

Webcode - Übungsdateien

5515-1E5A-E2D4

TBZ Technische Berufsschule Zürich



In Kooperation mit dem HERDT-Verlag stellen wir Ihnen eine PDF inkl. Zusatzmedien für Ihre persönliche Weiterbildung zur Verfügung. In Verbindung mit dem Programm HERDT-Campus ALL YOU CAN READ stehen diese PDFs nur Lehrkräften und Schüler\*Innen der oben genannten Lehranstalt zur Verfügung. Eine Nutzung oder Weitergabe für andere Zwecke ist ausdrücklich verboten und unterliegt dem Urheberrecht. Jeglicher Verstoß kann zivil- und strafrechtliche Konsequenzen nach sich ziehen.

---

Marc Haunschild

4. Ausgabe, 2. Aktualisierung, November 2017

ISBN 978-3-86249-701-0

## HTML5

Grundlagen der Erstellung  
von Webseiten

HTML5



<b>Bevor Sie beginnen ...</b>	<b>4</b>	<b>7 Hyperlinks erstellen</b>	<b>76</b>
		7.1 Wie werden Hyperlinks eingesetzt?	76
		7.2 Hyperlinks zu Seiten innerhalb eines Webauftritts erstellen	76
		7.3 Hyperlinks zu Seiten und Dokumenten im Web erstellen	81
		7.4 Hyperlinks innerhalb einer Seite	83
		7.5 Hyperlinkziele definieren	86
		7.6 Grafiken als Hyperlinks	86
		7.7 Blockelemente als Link benutzen	87
		7.8 Übung	88
<b>1 Vor der Technik</b>	<b>5</b>	<b>8 Listen erstellen</b>	<b>90</b>
1.1 Am Anfang steht die Idee	5	8.1 Mit Listen arbeiten	90
1.2 Das Konzept	6	8.2 Unsortierte Listen nutzen	90
1.3 Fazit	9	8.3 Sortierte (nummerierte) Listen	92
		8.4 Beschreibungslisten	93
		8.5 Listen verschachteln und kombinieren	95
		8.6 Übung	97
<b>2 HTML5 kennenlernen</b>	<b>10</b>	<b>9 Tabellen erstellen</b>	<b>99</b>
2.1 Basiswissen HTML und CSS	10	9.1 Grundlagen zu Tabellen	99
		9.2 Spalten- und Tabellenüberschriften	100
		9.3 Bereiche einer Tabelle organisieren	102
		9.4 Spalten und Zeilen verbinden	103
		9.5 Tabellen in responsiven Webseiten verwenden	108
		9.6 Übung	111
<b>3 Die erste Webseite mit HTML erstellen</b>	<b>15</b>	<b>10 Formulare erstellen</b>	<b>112</b>
3.1 Webseiten erstellen, speichern und testen	15	10.1 Formulare erstellen	112
3.2 Erste Schritte mit HTML	19	10.2 Eingabefelder erstellen	117
3.3 Der Quellcode eines HTML-Dokuments	21	10.3 Auswahllisten	123
3.4 Webseiten strukturieren	23	10.4 Kontrollkästchen und Optionsfelder	126
3.5 Elemente verschachteln	26	10.5 Schaltflächen mit Grafiken und JavaScript	128
3.6 Übung	27	10.6 Mehr Attribute für Formular-Elemente verwenden	130
		10.7 Übung	133
		<b>11 Webstandards und Codeoptimierung</b>	<b>134</b>
<b>4 Erste optische Anpassungen mit CSS</b>	<b>28</b>	11.1 HTML-Versionen	134
4.1 Erste Schritte mit CSS	28	11.2 Standardisiertes HTML schreiben	135
4.2 Eingebettete Stylesheets erstellen	32	11.3 HTML 4 und XHTML verstehen	138
4.3 Mit Vererbung richtig arbeiten	33	11.4 Code validieren	141
4.4 Externe Stylesheets verwenden	35	11.5 Webseiten in verschiedenen Browsern testen	143
4.5 Farbwerte definieren	37	11.6 Effizient arbeiten, Vorhandenes nutzen	145
4.6 Maßeinheiten in CSS festlegen	39	11.7 Übungen	149
4.7 Übung	42		
<b>5 Organisation und Struktur</b>	<b>43</b>		
5.1 Webseiten in Ordnern organisieren	43		
5.2 Relative und absolute Pfade definieren	45		
5.3 Elemente mit unterschiedlicher Darstellung	48		
5.4 Elemente richtig verschachteln	53		
5.5 Semantische Textauszeichnung verwenden	54		
5.6 Zeichencodierung und Kommentare	55		
5.7 Übung	57		
<b>6 Grafiken und Multimedia einbinden</b>	<b>58</b>		
6.1 Mit Grafiken arbeiten	58		
6.2 Grafiken im Web	61		
6.3 Grafiken für unterschiedliche Ausgabegeräte bereitstellen	65		
6.4 Videos einbetten	65		
6.5 Audio-Dateien einbinden	67		
6.6 Externe Inhalte nutzen	67		
6.7 Übungen	74		

<b>12 Webauftritt veröffentlichen</b>	<b>150</b>	14.3 Vor- und Nachteile der beiden Layout-Methoden	178
12.1 Suchmaschinenoptimierung (Search Engine Optimization – SEO)	150	14.4 Eine responsive Navigation mit Flexbox umsetzen	179
12.2 Einen Domainnamen registrieren	157	14.5 Benutzerfreundliche Menüs erstellen	181
12.3 Speicherplatz auf einem Webserver reservieren	159	14.6 Übungen	186
12.4 Einen Webauftritt veröffentlichen	161		
12.5 Übung	163	<b>A Anhang: HTML-Element-Referenz</b>	<b>187</b>
<b>13 Text mit CSS stylen</b>	<b>164</b>	<b>B Anhang: Editoren und Entwicklungs- umgebungen</b>	<b>192</b>
13.1 Das Aussehen von Text mit CSS gestalten	164	B.1 WYSIWYG-Editoren	192
13.2 Schriftfamilien definieren	166	B.2 Integrierte Entwicklungsumgebungen (IDE)	193
13.3 Schlüsselwörter verwenden	168	B.3 Reine Texteditoren	193
13.4 Bestimmte Textstellen mit CSS stylen	169		
13.5 Übung	171	<b>Stichwortverzeichnis</b>	<b>195</b>
<b>14 Layout in CSS</b>	<b>172</b>		
14.1 Zweispalter im Kopfbereich	172		
14.2 Flexbox – flexible, responsive Layouts ohne @media-Regel	176		

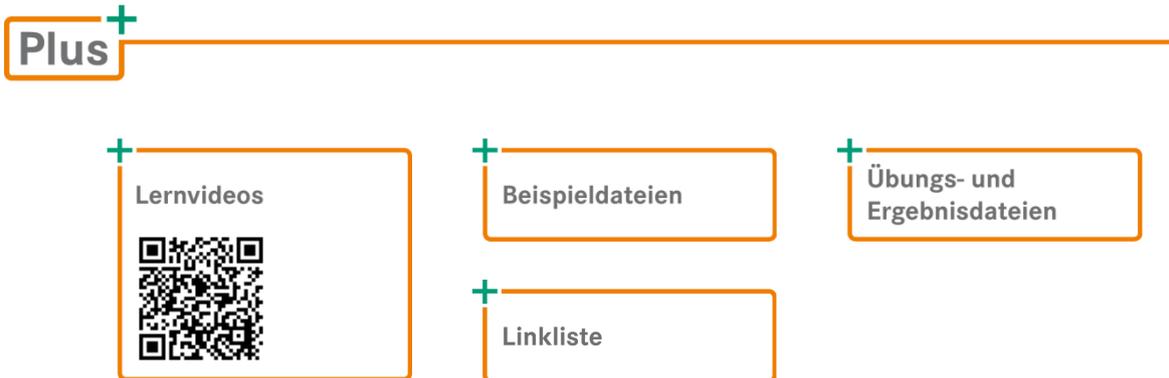
# Bevor Sie beginnen ...

## HERDT BuchPlus - unser Konzept:

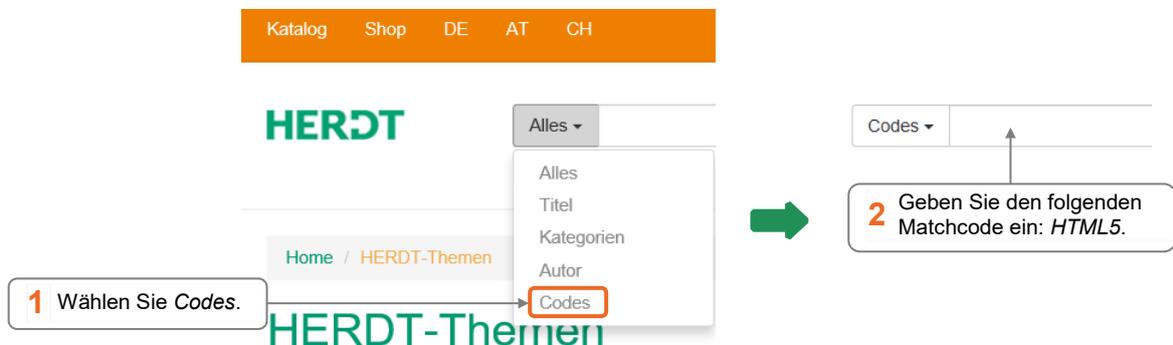
Problemlos einsteigen - Effizient lernen - Zielgerichtet nachschlagen

(weitere Infos unter [www.herdt.com/BuchPlus](http://www.herdt.com/BuchPlus))

Nutzen Sie dabei unsere maßgeschneiderten, im Internet frei verfügbaren Medien:



- ▶ Rufen Sie im Browser die Internetadresse [www.herdt.com](http://www.herdt.com) auf.



## 1

## Vor der Technik

## 1.1 Am Anfang steht die Idee

„Ich mache eine Website über xyz!“. Gut so! Jeder hat etwas, das zu teilen lohnt. Es gibt beispielsweise eine Webseite, auf der Menschen einander ihre Spleens und Ticks mitteilen. So erfährt man, dass praktisch jede (gute oder schlechte) Angewohnheit von vielen Menschen geteilt wird. Selbst wenn man im Bekanntenkreis dafür verlacht wird, weil kein anderer diesen Spleen teilt (oder es nicht zugeben mag) – es gibt die Menschen mit denselben Stärken und Schwächen!

Mit Webseiten erreichen Sie also neue, noch unbekannte Menschen, mit denen Sie etwas gemeinsam haben.

In dem Satz „Ich mache eine Website über xyz!“ **steht das xyz keineswegs für ein beliebiges Thema**, sondern es ist ein Platzhalter für **Ihr Thema**.

Um Menschen mit denselben Interessen erreichen zu können, müssen Sie zunächst einmal festlegen, was dieses „Interesse“ ist. Das kann ganz allgemein sein, z.B. „Reisen“ oder ein Land oder ein bestimmter Ort. Das Thema können aber auch Sie selber sein.

Bereits jetzt sollten Sie sich eine Frage stellen: **„Warum mache ich diese Webseite?“**

Wenn Sie Ihr Geschäft vorstellen möchten, wollen Sie sicher neue Kunden gewinnen, also Unbekannte erreichen. Wenn Sie über sich selbst oder Ihre Familie schreiben, möchten Sie vermutlich Ihrer Familie und Ihren Freunden – also Bekannten – mitteilen, was Sie öffentlich zu teilen bereit sind. Ein Passwortschutz schränkt diese Öffentlichkeit auf einen genau auswählbaren Benutzerkreis ein.

An diesen wenigen Beispielen sehen Sie bereits, wie unterschiedlich die Intention für die Erstellung einer Webseite sein kann.

Sich über das Ziel der Webseite klar zu werden ist essentiell bei der Erstellung eines Konzeptes und dem späteren Aufbau Ihrer Homepage.



Auf [Spleen24.tumblr.com](https://spleen24.tumblr.com) erfahren Sie, wie viele Menschen so ticken wie Sie

## 1.2 Das Konzept

### Thema und Zielgruppe

Wenn Sie nicht nur Ihren Bekanntenkreis über sich informieren möchten, müssen zuerst einmal Sie selber wissen, was Ihr Thema ist, sonst werden das auch Ihre Besucher nicht nachvollziehen können. Besucher verlassen Ihre Webseite in der Regel, wenn Sie nicht wissen, worum es dort geht. Es sei denn, genau das ist die Idee der Webseite: eine Art Ratespiel, eine Reise mit offenem, möglichst überraschendem Ende – aber selbst dann muss bereits zu Beginn erkennbar sein, dass diese Webseite erforscht werden will und dass den Besucher eine spannende Reise erwartet.

**In jedem Fall braucht Ihre Webseite einen roten Faden**, an dem sich Besucher entlang hangeln können.

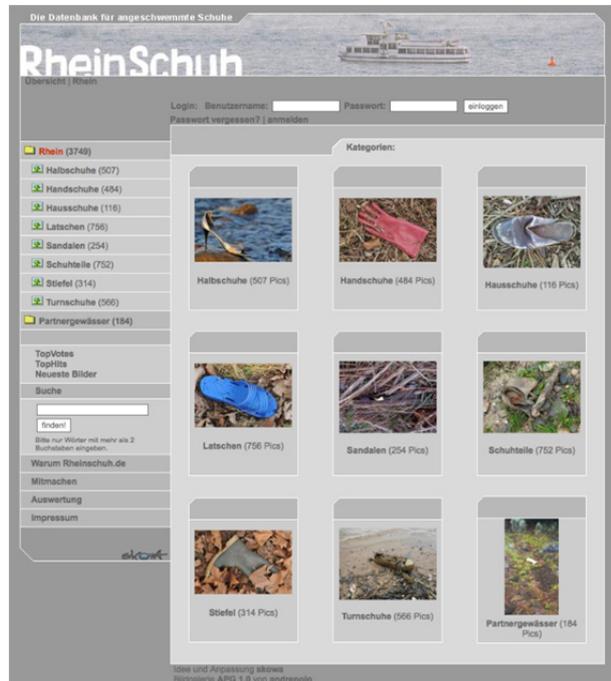
Aus dem Thema ergibt sich fast zwangsläufig eine Zielgruppe – und umgekehrt. Ein Beispiel: Wenn Sie eine Webseite über Reisen erstellen, erreichen Sie damit potentiell alle Menschen auf der Welt,

- ✓ die gerne reisen,
- ✓ die reisen müssen,
- ✓ die gerne reisen würden,
- ✓ die jemanden kennen, der gerne reist
- ✓ und sogar diejenigen, die nur mal wissen wollen, warum andere reisen, ohne es selber zu wollen, zu müssen oder jemanden zu kennen, der reist.

Damit dürften Sie also potentiell fast alle Menschen erreichen. Vermutlich aber auch wenige fesseln. Denn wenn Sie nicht die Ressourcen haben, ein großes Team an Schreibern, Fotografen, Designern usw. zu beschäftigen, werden Sie als einzelne Person recht wenig zu dem im Internet bereits allgegenwärtigen Thema „Reisen“ beitragen können.

Sie könnten aber auch eine Webseite erstellen, in der es um einen ganz speziellen Ort auf der Welt geht. Vielleicht nur um ein einzelnes Haus. Eines mit einer besonderen Geschichte, von der nur Sie selber etwas wissen.

Damit erreichen Sie potentiell nicht alle Menschen auf der Welt, aber diese einzigartigen Inhalte machen eine erfolgreiche Webseite aus. Ihre Zielgruppe ist automatisch sehr viel kleiner, in der Regel aber auch sehr viel interessierter. Sie können ein Forum anbieten, in dem sich Menschen über das von Ihnen bereitgestellte Thema austauschen und eine „Gemeinde“ aufbauen, die sich mit Ihrem Thema beschäftigt.



*Originell und einzigartig: Auf rheinschuh.de werden Schuhe gezeigt, die Vater Rhein getragen hat*

Der Unterschied ist folgender: eine Webseite zu betreiben mit einem Thema, das (fast) jeden interessiert und sich so einer riesengroßen Konkurrenz zu stellen, in der Hoffnung, dass für Sie genug abfällt von dem allgemeinen Interesse (was durchaus funktionieren kann) – oder eine Nische finden, die sehr speziell ist, für die es aber auch Interessierte gibt und die bisher noch von niemandem sonst angeboten wird.

Neben dem Thema und der Zielgruppe, die sich gegenseitig bedingen, gilt es bei einem Konzept noch anderes zu bedenken.

## Erfolg mit der eigenen Webseite

Ihre Webseite ist erfolgreich, wenn Sie damit die selbst gesteckten Ziele erreichen. Die Ziele sind ebenso unterschiedlich wie die möglichen Themen.

### Mögliche Ziele

- ✓ Ihre Bekannten über sich auf dem Laufenden halten
- ✓ Bekannter werden
  - sich einen Namen in einer bestimmten Gruppe machen
  - sich Kunden, Firmen oder Gleichgesinnten vorstellen
- ✓ Geld verdienen über
  - Werbenetzwerke
  - selbst verkaufte Werbung
  - den Verkauf von Waren
  - den Verkauf von Dienstleistungen
- ✓ Einen guten Zweck unterstützen, indem Sie
  - Bewusstsein für ein Problem schaffen
  - Hilfsbedürftigen ein Forum geben
  - Spenden sammeln

Diese wenigen Beispiele machen deutlich, dass Sie an Ihr Projekt je nach Formulierung Ihrer Ziele anders herangehen müssen.

Um nur auf zwei Beispiele einzugehen: wenn Sie sich als Experte für ein Thema vorstellen wollen, verfolgen Sie materielle Zwecke bestenfalls in zweiter Linie. Zunächst gilt es, Ihre Kompetenz zu belegen. Das können Sie mit Veröffentlichungen, Referenzen, Zeugnissen und anderen Belegen für Ihre Fähigkeiten.

Wenn die Inhalte stimmen, werden über das Thema interessierte Menschen auf Ihre Webseite aufmerksam werden; vor allem, wenn Sie sich mit diesen auch sonst vernetzen (Soziale Netzwerke, Veröffentlichungen in Offline-Medien, Vorträge im echten Leben usw.).

Hierzu ist Online-Werbung nur bedingt nützlich. Auch wenn Ihre Webseite damit bekannter gemacht werden kann, muss man sich überlegen, ob es sich lohnt, da dieses Geld bestenfalls über Umwege zurückkommen wird.

Ganz anders liegt der Fall, wenn Sie Waren oder Dienstleistungen verkaufen möchten, weil Sie ein Geschäft betreiben. Der Volksmund sagt von jeher: wer nicht wirbt, der stirbt. Das gilt auch online, denn Menschen müssen von Ihrem tollen Angebot erfahren, um es nutzen zu können. Bei Online-Werbung lässt sich der Erfolg einer Kampagne übrigens über diverse Statistiken besonders gut nachvollziehen.

## Conversion Rate – Klasse statt Masse

In wenigen Fällen lässt sich Erfolg daran festmachen, wie viele Menschen auf Ihre Webseite kommen. Ein solcher Fall ist gegeben, wenn Sie Werbung von anderen schalten wollen, die von möglichst vielen Menschen gesehen werden soll, weil Sie dann mehr Geld für die gezeigte Werbung erhalten.

Sonst liegen die Dinge ein wenig komplizierter. Wenn Sie einen Blumenladen haben und vor allem an Stammkunden aus Ihrem Stadtviertel verkaufen, nützt es Ihnen wenig, wenn Millionen von Menschen weltweit Ihre Seite wegen der hübschen Blumenfotos besuchen. Das ist schön, bringt aber keine Käufer, weil jemand aus Brasilien nicht bei Ihnen einkaufen wird und sie vermutlich auch nicht dorthin liefern.

Wenn Sie möchten, dass die Webseite Sie beim Verkauf Ihrer Blumen unterstützt, müssen Sie also die Seite so gestalten und bewerben, dass die Menschen in Ihrer Nachbarschaft darauf aufmerksam werden und vielleicht sogar online bestellen – denn zwei Straßen weiter gegen einen kleinen Obulus zu liefern können Sie vermutlich leisten, um so neue Kunden zu erreichen, die beispielsweise schlecht zu Fuß oder einfach bequem sind.

Ein anderes Beispiel ist eine Webseite für einen Facharzt. Fachärzte sind ausgebucht und benötigen keine Werbung. Eigentlich. Uneigentlich möchten sie vielleicht Dienstleistungen anbieten, die besonders lukrativ sind und die vielleicht nicht einmal der Praxisinhaber selbst durchführen muss. So dürfen angestellte Arzthelferinnen oder Krankenschwestern bestimmte Behandlungen nach entsprechender Ausbildung durchführen. Der Praxisinhaber kann eventuell eine zusätzliche Helferin für besonders lukrative Behandlungen einstellen. Diese Behandlungen sollten dann auch prominent auf der Praxis-Homepage präsentiert werden.

**Die Webseite sollte also keinen Selbstzweck erfüllen** (möglichst oft angeklickt werden oder massenhaft Patienten bringen, für die der Arzt gar keine Zeit hätte), **sondern Unternehmensziele unterstützen**. In diesem Fall müsste also vor dem Konzept für die Webseite ggfs. geklärt werden, wohin sich eine Firma entwickeln soll, bevor Sie sich über die Webseite Gedanken machen.

Eine gute Webseite sollte Sie in Ihren persönlichen Zielen unterstützen – ganz gleich ob diese privater oder beruflicher Natur sind.



*Auch wenn Ärzte in ihrer eigentlichen Arbeit ausgelastet sind, kann sich eine Webseite lohnen, die über weitere Leistungen der Praxis informiert*

*Foto: Víctor Santa María from Buenos Aires, Argentina (Suterh Solidario - Víctor Santa María), via Wikimedia Commons*

Dazu müssen Sie sich bewusst machen, was Sie von Ihrer Webseite erwarten und was Besucher dafür auf Ihrer Webseite tun sollten. Auch hierzu einige Beispiele:

Besucher sollen

- ✓ möglichst viele Informationen erhalten
- ✓ eine Dienstleistung oder Waren bestellen
- ✓ einen Termin ausmachen
- ✓ ihre persönlichen Daten (z.B. für einen privaten oder geschäftlichen Austausch) bereitstellen
- ✓ sich an der Website beteiligen (als Kommentator, als Autor, als Partner)
- ✓ die Seite bekannter machen (z.B. in sozialen Medien teilen)
- ✓ einen Newsletter bestellen.

Legen Sie solche Ziele für Ihre Seite fest und gelingt es, Besucher zu den gewünschten Aktionen zu bringen, misst man das mit der **Conversion Rate**: Wie hoch ist die Anzahl der Besucher, die einen Bestellvorgang erfolgreich abgeschlossen haben oder die vorbeigekommen sind, um einen Strauß Blumen bei Ihnen zu kaufen.

Dennoch sollen natürlich Leute auf Ihre Webseite kommen. Also muss Ihre Seite in Suchmaschinen (in Deutschland vor allem Google wegen seiner Monopol-Stellung) gefunden werden. Wie Sie das erreichen, ist ausführlicher im Kapitel „Webauftritt veröffentlichen“ beschrieben.

Dabei ist es wichtig, die Suchbegriffe, unter denen Sie gefunden werden sollen, passend zum Konzept und zu den Zielen Ihrer Webseite festzulegen.

## Ordnung ist das halbe Leben!

Falls Sie bereits Artikel erstellt haben, gibt es zur Strukturierung der Inhalte eine einfache und wirksame Methode: Schreiben Sie den Titel von jedem Text auf einen Zettel, und verteilen Sie die Zettel dann in Kisten oder auf große Blätter, die mit einem Thema beschriftet sind. So sehen Sie schnell, ob Sie an alles gedacht haben – insbesondere können Sie auf diese Weise leicht umsordern. Das funktioniert sogar mit Texten, die Sie noch gar nicht erstellt haben!

Was sonst noch beim Aufbau einer Webseite zu beachten ist, erfahren Sie später in einem eigenen Kapitel.

## 1.3 Fazit

Werden Sie sich über das genaue Thema klar, das Sie anbieten wollen.

Legen Sie Ziele für Ihre Homepage fest – selbst dann, wenn Sie diese (anfangs) nicht mit entsprechenden Statistiken messen.

Erstellen Sie zielgruppengerechte, thematisch passende Inhalte.

Erstellen Sie eine Struktur für Ihre Inhalte.

Lassen Sie sich nicht davon abhalten, dass Ihnen das für die wenigen Inhalte, die Sie anfangs haben, überdimensioniert scheint. Ihre Seite wird mit der Zeit wachsen (das tun sie alle). Dann werden Sie froh sein, ein Konzept zu haben, das ein geordnetes Wachstum zulässt und Ihre Webseite wird Sie in Ihrem beruflichen oder privaten Leben weiterbringen.

# 2

## HTML5 kennenlernen

### 2.1 Basiswissen HTML und CSS

#### Was versteht man unter HTML?

HTML (**H**ypertext **M**arkup **L**anguage) ist eine Auszeichnungssprache (Markup Language), mit der Textdokumente erstellt und strukturiert werden. Das Wort „**Hypertext**“ im Namen weist auf die **Hyperlinks** (kurz Links) hin, mit denen HTML-Dokumente mit beliebigen Dateien verknüpft werden können.

Webseiten werden nicht programmiert, sondern in HTML geschrieben. Textpassagen werden mithilfe von HTML-Tags ausgezeichnet. **HTML-Tags** (kurz Tags genannt) kennzeichnen die logische Bedeutung (bzw. Semantik) von Textpassagen innerhalb der Webseite. Sie werden in spitzen Klammern geschrieben, z. B.

<code>&lt;p&gt;</code>	Dieses Tag signalisiert, dass ein Absatz (engl.: paragraph) beginnt ( <b>öffnendes Tag</b> ). Es besteht aus dem <b>Namen des HTML-Elements in spitzen Klammern</b> .
<code>&lt;/p&gt;</code>	Dieses Tag signalisiert, dass der Absatz endet. Das <b>schließende Tag</b> besteht aus einem <b>Schrägstrich und dem Namen des HTML-Elements in spitzen Klammern</b> .

Ein **HTML-Element** besteht in der Regel aus einem öffnenden Tag, einem schließenden Tag und dem Text zwischen beiden. Ausnahmen sind leere Elemente, vgl. Kapitel 3.3.

Da HTML-Dokumente aus reinem Text bestehen, benötigen Sie keine spezielle Software, um sie zu erstellen. Ein beliebiger Texteditor genügt.

Hinweis: Mehr zu den einzelnen Versionen und der Geschichte von HTML erfahren Sie im Kapitel „Webstandards und Codeoptimierung“.

#### Webseiten anzeigen und analysieren

Es gibt zahlreiche Software, um Webseiten darzustellen oder zu analysieren. Dazu gehören die bekannten **Browser** (engl. to browse: stöbern) wie Firefox, Chrome, InternetExplorer und Safari. Außerdem gibt es nicht-grafische Textbrowser und Vorleseprogramme für Blinde (Screenreader).

Suchmaschinen können Ihre HTML-Dokumente ebenfalls einlesen und die enthaltenen Informationen verarbeiten.

Browser stehen nicht nur für Computer zur Verfügung, sondern z. B. auch für Smartphones, Tablets, Fernseher, Spielekonsolen und Computermonitore.

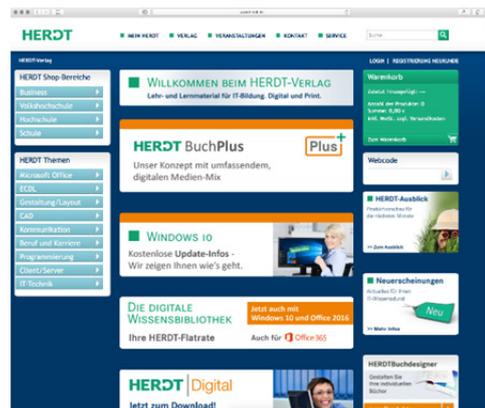
## Wie Browser arbeiten

Ein üblicher (grafischer) Browser liest den HTML-Code und ist für die Wiedergabe des Inhalts auf dem Computerbildschirm zuständig.

Er berücksichtigt neben der Struktur des Dokumentes (Überschriften, Absätze, Listen usw.) auch Formatierungsangaben, die Sie dem Dokument mitgeben (ähnlich wie Formatvorlagen für ein Textdokument, das Sie mit einer Textverarbeitungssoftware erstellen).

Den Vorgang aus dem reinen Text eine Webseite mit Layout und Bildern zu erstellen, nennt man **Rendering**.

Der Kern eines jeden Browsers ist daher die **Rendering Engine**. Diese ist für die Darstellung der Webseiten zuständig. Hier gibt es geräteabhängige, herstellerabhängige und versionsbedingte Unterschiede.



*Browser stellen Webseiten grafisch aufbereitet dar (hier die Webseite des HERDT-Verlags für Bildungsmedien)*

**Geräteabhängige Unterschiede** entstehen dadurch, dass eine Webseite auf einem Smartphone anders aussieht, als auf einem Computer oder extrem großen Fernseher.

**Herstellerabhängige Unterschiede** sind dadurch bedingt, dass die verschiedenen Hersteller von Browsern unterschiedliche Konzepte bei der Aktualisierung ihrer Browser verfolgen. So wird der Safari von seinem Hersteller Apple nur zusammen mit Betriebssystem-Updates auf den aktuellen Stand der Webtechnik gebracht und verliert bis zum nächsten Update (große Updates finden nur einmal im Jahr statt) mehr und mehr an Aktualität, während die Mozilla Foundation (Firefox) und Google (Chrome) ihre Browser ständig verbessern und in deutlich kürzeren Abständen automatisch Updates ausliefern, von denen der Anwender nichts merkt. So werden ständig neue Funktionen nachgeliefert. Der Benutzer kommt so kaum noch auf Webseiten, auf denen irgendetwas nicht mehr funktioniert.

Dadurch bleibt er idealerweise von **versionsbedingten Unterschieden** verschont. Darunter versteht man das Problem, dass alte Browser neue HTML- und CSS-Möglichkeiten nicht implementiert haben, die zum Zeitpunkt der Browserveröffentlichung noch gar nicht existierten.

Weitere **Unterschiede bedingt das Betriebssystem, auf dem ein Browser läuft**. So sind auf einem Linux-Desktop andere Schriften installiert, als auf einem Windows-System. Auch (Standard-) Schaltflächen, Pop-Ups und andere Dinge sehen in ein und demselben Browser (z. B. Firefox) anders aus, wenn er auf unterschiedlichen Betriebssystemen läuft.

Ihre Webseite kann deshalb in den verschiedenen Browsern und in den verschiedenen Versionen eines Browsers unterschiedlich aussehen. Das muss kein Problem sein, so lange die Webseite in allen relevanten Browsern funktioniert. Das heißt:

- ✓ alle Texte sind lesbar,
- ✓ Bedienelemente (vor allem die Navigation) funktionieren erwartungsgemäß,
- ✓ Formulare lassen sich ausfüllen und abschicken,
- ✓ Programmierte Elemente (JavaScript-Funktionen) können bedient werden und führen zu den festgelegten Ergebnissen.

Die Browserhersteller sind maßgeblich an der Weiterentwicklung von HTML beteiligt. Zum Wesen von HTML5 gehört es, dass neue Möglichkeiten zunächst von einem einzelnen Hersteller in den eigenen Browser integriert werden, bevor andere Hersteller nachziehen.

Sie können Möglichkeiten, die nur in einem Browser funktionieren, ohne Probleme verwenden, sofern diese in anderen Browsern auch dargestellt werden und benutzbar sind und lediglich anders aussehen.

Das Prinzip, alle Browser zu unterstützen und trotzdem die Fähigkeiten moderner Browser zu nutzen, nennt man **progressive enhancement**.



HTML-Code und die Darstellung im Browser (Beispieldatei „code.html“)

## Entwicklung und Formatierung von Webseiten

Die ersten Webseiten sahen recht simpel aus: schwarzer Text auf weißem Grund, hier und da ein Bild. Überschriften waren größer und fetter als normale Texte. Wichtige Texte konnten fett oder kursiv dargestellt werden und alle Links waren blau und unterstrichen.

Das genügte dem ursprünglichen Zweck, denn Webseiten sollten in erster Linie Informationen bereitstellen. Der Austausch unter Wissenschaftlern spielte bei der Entwicklung eine entscheidende Rolle. Fehlende Funktionen, wie die Möglichkeit mathematische Formeln darzustellen, wurden durch den Einsatz von Dokumenten in anderen Formaten (LaTeX oder Grafiken) umgangen.

Im Zuge der Kommerzialisierung des Webs kam das Bedürfnis nach immer umfangreicherer Gestaltung auf. HTML erhielt zunehmend mehr Möglichkeiten zur optischen Gestaltung.

### Beispiel:

<code>&lt;u&gt;Wichtig&lt;/u&gt;</code>	Mit dem u-Element war es möglich, dem Browser mitzuteilen, dass ein bestimmtes Wort unterstrichen dargestellt werden sollte.
---	--

Das war ziemlich mühselig, da eine Formatierung an jedem einzelnen Element vorgenommen werden musste. Wenn Webentwickler auf einer Website hunderte Überschriften rot darstellen wollten, mussten sie das an jede einzelne Überschrift schreiben. Ebenso umständlich war die Einarbeitung späterer Änderungen: Ein Webauftritt, auch Website genannt, besteht oft aus Hunderten einzelner Seiten. So mussten Webentwickler Änderungen in hunderten von Dateien vornehmen. Als Lösung wurde die Beschreibungssprache **CSS (Cascading Style-Sheets)** entwickelt.

## Das Traumpaar HTML und CSS

CSS bietet umfangreiche Formatierungsmöglichkeiten sowie den Vorteil einer Trennung der **Struktur** einer Webseite von der **Darstellung** der Webseite. Die Struktur von Webseiten wird in HTML ausgezeichnet und das Aussehen bzw. die Formatierung in CSS festgelegt. Die Regeln zur Darstellung der Webseite können zentral in einer CSS-Datei gespeichert werden. Somit können Änderungen am Layout **einmal** in dieser Datei vorgenommen werden und wirken sich automatisch auf **alle** HTML-Dokumente aus, die mit der CSS-Datei verknüpft sind.

Konsequenterweise wurden alle Formatierungsmöglichkeiten aus der Sprache HTML entfernt (in HTML 4.01 wurden sie zunächst als unerwünscht gekennzeichnet, in HTML5 sind sie nicht mehr erlaubt). **HTML5 dient ausschließlich der logischen Auszeichnung** von Texten und interaktiven Elementen wie Formularen.

**css Zen Garden**

**The Beauty of CSS Design**

A demonstration of what can be accomplished visually through CSS-based design. Select any style sheet from the list to load it into this page.

Download the sample [html file](#) and [css file](#)

**The Road to Enlightenment**

Littering a dark and dreary road lay the past relics of browser-specific tags, incompatible DOMs, and broken CSS support.

Today, we must clear the mind of past practices. Web enlightenment has been achieved thanks to the tireless efforts of folk like the W3C, WaSP and the major browser creators.

The css Zen Garden invites you to relax and meditate on the important lessons of the masters. Begin to see with clarity. Learn to use the (yet to be) time-honored techniques in new and invigorating fashion. Become one with the web.

**So What is This About?**

There is clearly a need for CSS to be taken seriously by graphic artists. The Zen Garden aims to excite, inspire, and encourage participation. To begin, view some of the existing designs in the list. Clicking on any one will load the style sheet into this very page. The code remains the same, the only thing that has changed is the external .css file. Yes, really.

➔



The screenshot shows the CSS Zen Garden website. On the left, there is a list of design styles under the heading 'Select a Design'. The styles listed include 'Under the Seat', 'Make 'em Proud', 'Orchid Beauty', 'Oscillations', 'CSS Co., Ltd.', 'Sakura', 'Kyoto Forest', and 'A Walk in the Garden'. On the right, a preview of the 'The Road to Enlightenment' design is shown, featuring a dark background with white text and a photograph of two people working at a computer.

Die Anzeige eines HTML-Dokuments vor und nach der Verbindung mit einer CSS-Datei (<http://www.csszengarden.com/tr/deutsch/>: „CSS Co, Ltd.“ von Benjamin Klemm)

Vorteil	Beschreibung
<b>Webseiten werden schneller geladen.</b>	Die HTML-Dateien sind kleiner, da sie nur den Code für den Inhalt enthalten. Formatierungshinweise müssen nicht an jedem Element vorgenommen werden.
<b>Der Formatierungscode kann einfacher gefunden bzw. übersprungen werden.</b>	Da der Formatierungscode in einer zentralen Datei bereitgestellt werden kann, ist die nachträgliche Bearbeitung der Webseiten einfacher. Änderungen wirken sich auf alle Seiten aus, welche die Datei mit den Formatierungen verwenden. Die Datei wird vom Browser einmalig geladen und im Cache zwischengespeichert. Weil die Formatierungen so nur einmal geladen werden müssen, sind die eigentlichen HTML-Dokumente wesentlich kleiner und werden insbesondere bei langsamen Internet-Verbindungen schneller geladen.

## Informationen zu HTML und CSS im Internet

Hilfreiche Nachschlagemöglichkeiten im Internet finden Sie hier:

<b>Online-Referenz HTML</b>	<a href="https://dev.w3.org/html5/html-author/">https://dev.w3.org/html5/html-author/</a> (offizieller engl. Standard) <a href="https://wiki.selfhtml.org/">https://wiki.selfhtml.org/</a> (deutsche Anleitung) <a href="http://www.w3schools.com">http://www.w3schools.com</a> (umfangreiche englischsprachige Anleitung mit Beispielen und Testmöglichkeiten)
<b>Online-Referenz CSS</b>	<a href="http://www.css-wiki.com">http://www.css-wiki.com</a>
<b>Weitere Nachschlagewerke</b>	<a href="http://www.canituse.com">http://www.canituse.com</a> (Übersicht über Browserfähigkeiten) <a href="http://html5please.com">http://html5please.com</a> (Neue Features browserübergreifend sinnvoll einsetzen – auch CSS und JavaScript)

Nach der Einführung von CSS wurden HTML-Elemente zur optischen Gestaltung der Webseite, wie zum Beispiel das `<u>`-Tag, als veraltet (engl.: deprecated) eingestuft. Im HTML5-Standard kommen diese Elemente nicht mehr vor.

## Was brauche ich, um Webseiten mit HTML zu erstellen?

Um Webseiten mit HTML zu erstellen, benötigen Sie einen **Texteditor** (kurz Editor). Wichtig ist es auch, mindestens einen Browser installiert zu haben, um die erstellte Webseite betrachten zu können.

Ein Texteditor ist ein Programm, das speziell für das Verfassen von Code entwickelt wurde. Solche Programme unterstützen Sie optimal, wenn Sie programmieren oder Webseiten erstellen möchten. Eine Auswahl an interessanten Editoren finden Sie im Anhang B (Werkzeuge).

# 3

## Die erste Webseite mit HTML erstellen

### 3.1 Webseiten erstellen, speichern und testen

#### Ordner für Ihre Webseiten und Dateien erstellen

Erstellen Sie einen Ordner auf Ihrem System, in dem Sie alle Dateien (z. B. HTML-, CSS- und Bild-dateien) für Ihren Webauftritt ablegen werden. Dieser Ordner wird Stammordner oder Wurzelverzeichnis (engl.: root directory) genannt. Für die Arbeit mit diesem Buch nennen Sie Ihren Ordner *hotel*.

#### Mit Texteditoren arbeiten

Wenn Sie sich noch nicht für einen Texteditor entschieden haben, können Sie auch einen auf Ihrem System bereits installierten Editor verwenden. Jedes Betriebssystem bringt einen einfachen Editor mit.

Schauen Sie sich immer wieder im Web nach neuen Editoren um – es kommen ständig neue auf den Markt. Eine Auswahl an Editoren finden Sie im Anhang B.

Windows-Anwendern steht der im Lieferumfang enthaltene Windows Editor (*notepad.exe*) zur Verfügung.

- ▶ Klicken Sie auf die -Schaltfläche von Windows und dann nacheinander auf die Einträge *Alle Programme - Zubehör - Editor*.

Ein leeres Editorfenster wird geöffnet, in dem Sie Ihren HTML-Code eingeben können.

Wenn Sie mit einem Macintosh arbeiten, können Sie die Macintosh-Anwendung TextEdit verwenden.

#### Das Grundgerüst einer Webseite

Jedes HTML-Dokument beginnt mit der Angabe der HTML-Version (der Dokumenttyp-Definition bzw. DOCTYPE). Mit `<!DOCTYPE html>` teilen Sie dem Browser mit, dass Sie den Standard HTML5 verwenden.

Das HTML-Dokument selbst wird von dem Tagpaar `<html>` und `</html>` umschlossen. Weil dieses Element der Ausgang für alle weiteren ist, nennt man es auch Wurzel- oder Root-Element. Jedes HTML-Dokument wird in zwei Bereiche unterteilt:

<b>Dokumentkopf</b>	<p>Der Bereich zwischen den Tags <code>&lt;head&gt;</code>- und <code>&lt;/head&gt;</code> (engl.: Kopf) enthält Informationen über das HTML-Dokument. Seine Inhalte werden nicht vom Browser dargestellt.</p> <p>Eine Ausnahme bildet der <b>Titel</b> des Dokumentes, der zwar nicht auf der eigentlichen Webseite erscheint, aber an vielen weiteren Stellen: Titelleiste des Browsers und/oder Tab, Windows-Taskbar, Google-Suchergebnis usw.</p> <p>Außerdem dient der Titel den meisten Browsern als Vorschlag für die Benennung von Lesezeichen (Favoriten).</p>
<b>Dokumentrumpf</b>	<p>Zwischen den <code>&lt;body&gt;</code>- und <code>&lt;/body&gt;</code>-Tags (engl.: Körper, Rumpf) befindet sich der eigentlich darzustellende Inhalt.</p>

## Webseite erstellen

- ▶ Geben Sie im Texteditor den folgenden HTML-Code ein:

```

① <!DOCTYPE html> ← HTML-Version
② <html> ← HTML-Starttag
③ <head>
④   <meta charset="utf-8">
⑤   <title>Hotel Vallora, Nordseeküste</title>
⑥ </head>
⑦ <body>
⑧   Hotel Vallora
⑨ </body>
⑩ </html> ← HTML-Endtag

```

Diagramm zur Struktur des HTML-Codes:

- Ein orangefarbener Rahmen umschließt die Zeilen 3 bis 6, beschriftet mit "Dokumentkopf".
- Ein orangefarbener Rahmen umschließt die Zeilen 7 bis 9, beschriftet mit "Dokumentrumpf".
- Die Zeilen 1 bis 2 sind als "HTML-Starttag" markiert.
- Die Zeile 10 ist als "HTML-Endtag" markiert.

- ① DOCTYPE-Definition teilt mit, dass HTML5 verwendet wird
- ② Zeigt den Beginn des HTML-Dokuments an
- ③ Bestimmt den Beginn des Dokumentkopfs
- ④ Angabe des verwendeten Zeichensatzes (sofern nicht vom Server per http-header gesendet)
- ⑤ Titel des Dokumentes  
Während andere Angaben im Dokumentenkopf optional sind, ist das `title`-Element für jede Webseite vorgeschrieben.
- ⑥ Zeigt das Ende des Dokumentenkopfes an
- ⑦ Bestimmt den Beginn des Dokumentenrumpfs
- ⑧ Inhalte, die Sie innerhalb des `body`-Elementes notieren, erscheinen auf Ihrer Webseite
- ⑨ Ende des Dokumentenrumpfs
- ⑩ Ende des HTML-Dokuments

Jede Webseite müssen Sie als Datei abspeichern. Verwenden Sie aussagekräftige Namen, um sich in Ihrem Webauftritt zurechtzufinden, ohne alle Seiten öffnen zu müssen.

## Die Bedeutung der Startseite

Wenn Sie Ihre Webseiten im Web veröffentlichen, wird Ihr Webauftritt über eine Adresse wie *www.example.com* abrufbar sein.

Wenn jemand diese Adresse in seinen Browser eingibt, geschieht folgendes: Der Browser (auch **Webclient** genannt) fordert ein **Webserver** genanntes Programm auf, diese Datei zu liefern. Sie möchten sicher eine bestimmte Webseite Ihres Webauftritts als Startseite festlegen, die ausgeliefert wird, wenn nicht eine ganz spezielle Seite wie *www.example.com/kontakt.htm* angefordert wird.

Die **Startseite** bzw. Einstiegsseite ist dasjenige HTML-Dokument, das standardmäßig ausgeliefert wird, wenn ein Besucher Ihren Webauftritt ohne Angabe einer bestimmten Seite aufruft. **Wie** Sie die Startseite benennen müssen, sodass sie vom Webserver als solche erkannt wird, schreibt Ihr Webpace-Provider vor. Meistens ist der Name *index.html*, *index.htm*, *default.html* oder *default.htm*. Bei manchen Providern können Sie einen beliebigen Namen definieren.

## Code übersichtlich schreiben

Um die Lesbarkeit Ihres Codes zu erhöhen, sollten Sie Zeilenumbrüche mit der -Taste und Einzüge mit der -Taste erzeugen oder Leerzeichen einfügen (dabei konsequent bleiben, nicht zum Teil mit Leerzeichen, zum Teil mit Tabs einrücken). So entstehen Leerräume (sogenannter Whitespace). Whitespace wird von Browsern „ignoriert“ bzw. zu einem einzigen Leerzeichen zusammengefasst, inklusive Leerräume, die durch mehrmaliges Betätigen der Leertaste entstehen.

```

<!DOCTYPE html> ← -Taste betätigen
<html> ← -Taste betätigen
  <head> ← -Taste betätigen

```

In der folgenden Abbildung sehen Sie den Code aus dem vorigen Beispiel ohne Leerräume. Der Code wird vom Browser identisch dargestellt; er ist jedoch für Menschen nicht gut zu lesen.

```

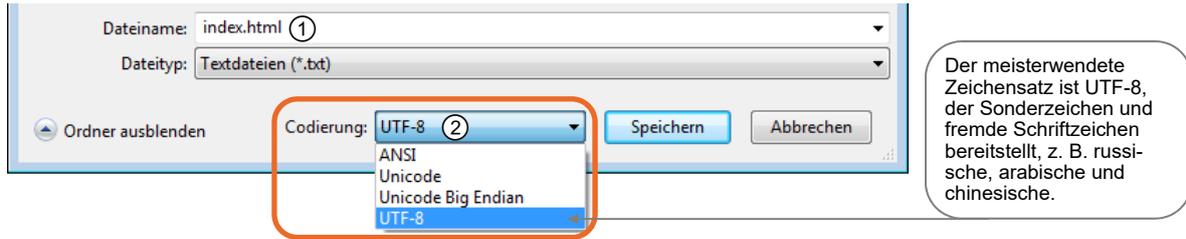
<!DOCTYPE html>
<html><head><meta charset="utf-8"><title>Hotel Vallora,
Nordseeküste </title></head><body>Hotel Vallora</body></html>

```

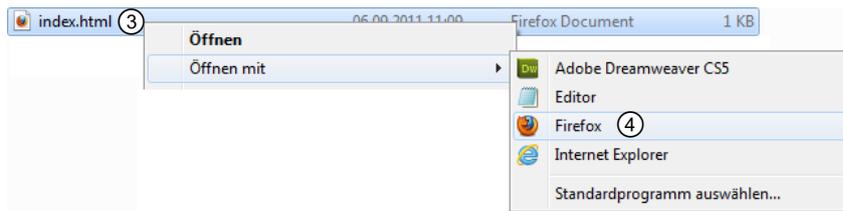
## Webseite speichern

- Speichern Sie die Datei als HTML-Dokument unter dem Namen *index.html* ① und legen Sie beim Speichern die Codierung UTF-8 ② fest.

Alle Webseiten müssen mit der Dateinamenserweiterung *.htm* oder *.html* versehen werden, sodass Browser diese als HTML-Dateien erkennen. Welche der beiden Schreibweisen Sie benutzen, bleibt Ihnen überlassen.



- ▶ Öffnen Sie die Webseite im Browser, indem Sie in Ihrem Dateimanager (z. B. Windows-Explorer) mit der rechten Maustaste auf die gespeicherte Datei (3) klicken und den Kontextmenüpunkt *Öffnen mit - Browsername* (z. B. Firefox) (4) wählen.



Der Quelltext wird vom Browser interpretiert und angezeigt. Das Element `title` wird als Titel Ihres Dokumentes (5) erkannt und als Tab-Titel dargestellt. Der Text, der sich zwischen den `<body>`-Tags befindet, stellt den Dokumentinhalt dar (6). Der Text wird in der Standardgröße und -schriftart angezeigt. Sie lernen im nächsten Kapitel, wie Sie den Text mithilfe von CSS formatieren.



Webseite im Browser anzeigen (Mozilla Firefox 6)

## Webseiten im Browser überprüfen

Wenn Sie die in diesem Buch verwendeten Beispiele betrachten wollen, genügt ein beliebiger Browser. Die Beispiele wurden bereits dahingehend getestet, ob sie in allen aktuellen Browsern funktionieren.

Sobald Sie an eigenen Seiten arbeiten, sollten Sie diese aber in mehreren Browsern betrachten, um Unterschiede zu finden und zu kennen. Ob diese Unterschiede so gravierend sind, dass sie entfernt werden müssen, liegt dann in Ihrer Entscheidung als Webseitenautor. Grundsätzlich sollte die Seite in jedem Browser funktionieren. Optisch identisch müssen Webseiten nicht sein – sofern nicht explizit gewünscht.

- ▶ Lassen Sie das Browserfenster mit der geladenen Webseite geöffnet.
- ▶ Nehmen Sie die gewünschten Änderungen im Editor vor und speichern Sie das Dokument.
- ▶ Wechseln Sie zum geöffneten Browserfenster und aktualisieren Sie die Ansicht, indem Sie je nach Browser auf das Symbol  klicken.

Mit der Funktionstaste **F5** bzw. der Tastenkombination **Strg R** können Sie Webseiten in Ihrem Browser schnell aktualisieren.

## 3.2 Erste Schritte mit HTML

### HTML-Code für eine einfache Webseite im Editor eingeben

Sie geben den Code in einen Editor ein und speichern die Datei als HTML-Dokument. **HTML-Dokumente bestehen aus HTML-Elementen.** Mittels **Tags** teilen Sie Browsern mit, wo ein Element beginnt und wo es endet.

Im folgenden Beispiel sehen Sie den HTML-Code für eine einfache Webseite, auf der ein fiktives Hotel vorgestellt wird. Die Tags werden in der Abbildung fett hervorgehoben. Die meisten Tags treten in Paaren auf, mit einem Start- und einem Endtag (öffnendes und schließendes Tag), die einen Text einschließen.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <title>Hotel Vallora, Nordseeküste</title>
  </head>
  <body>
    <h1>Hotel Vallora</h1>
    <p>Bei Ihrem nächsten Aufenthalt in unserer Gegend würden
      Sie gern persönlich begrüßen und mit unserem
      freundlichen und kompetenten Service verwöhnen.</p>
    <h2>Zimmerreservierungen:</h2>
    <p>Unsere Rezeption ist 24 Stunden besetzt. Sie können
      auch gern unser Online-Buchungsformular ausfüllen.</p>
  </body>
</html>
```

HTML-Code einer Webseite (Beispieldatei: „03\_Die erste Webseite mit HTML erstellen/code.html“)

### HTML-Dokument im Browser anzeigen

Der Browser erkennt anhand der Tags, welche Bedeutung die eingeschlossenen Texte haben. Sie können Text zum Beispiel als Überschrift, als Absatz, als Liste oder Tabelle auszeichnen.

Abhängig von der Bedeutung des Textes wählt der Browser eine passende Darstellung.



Darstellung Ihrer ersten Webseite im Browser Firefox

Die Abbildung zeigt, was der Browser aus Ihrem Quellcode macht. Er erkennt aufgrund Ihrer Angaben, welche Texte normale Absätze sind und welche Texte als Überschrift dargestellt werden sollen. **Überschriften sind hierarchisch und geben die Struktur des Dokumentes wieder.** Setzen Sie die Überschriften sinnvoll zur Strukturierung Ihres Dokumentes ein.

Sie können sich im Browser den HTML-Code von jeder Webseite anzeigen lassen, zum Beispiel um von anderen zu lernen. Den HTML-Code einer im Browser geladenen Webseite zeigen Sie beispielsweise in Mozilla Firefox über die Tastenkombination (Strg) (U), im Internet Explorer über den Menüpunkt *Ansicht - Quellcode* an. Noch mehr Analysemöglichkeiten bieten die in die Browser integrierten Entwicklertools, die sich in den meisten Browsern mit der Taste (F12) öffnen lassen. Diese geben weiterführende Informationen über den HTML-Quelltext hinaus, wie Formatierungsangaben in externen Dateien, die Sie für ein Element wie beispielsweise Überschriften gemacht haben. Für die ersten Schritte in HTML sind diese mächtigen Werkzeuge nicht nötig.

Bereits die HTML-Ansicht im Browser bietet Ihnen einigen Komfort: die Syntax des HTML-Dokumentes wird durch unterschiedliche Farben und normale oder fette Schriften verdeutlicht.

Verknüpfte Dateien lassen sich wie Links anklicken und dadurch ebenfalls öffnen. Um das zeigen zu können, wurde die obige Datei um die Zeile 11 ergänzt (siehe Abbildung). Hier sehen Sie, wie die Formatvorlage `vallorastyles.css` eingebunden wird. Auf das Einbinden von Formatvorlagen wird im nächsten Kapitel genauer eingegangen. Hier soll lediglich die Darstellung verknüpfter Dateien in der Quelltext-Ansicht gezeigt werden.

Nicht zuletzt erleichtert die Zeilennummerierung die Fehlersuche, da zahlreiche Werkzeuge, die Sie bei der Fehlersuche unterstützen, die Zeilen benennen, in denen Fehler gefunden wurden.

```

1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8">
5     <title>Hotel Vallora, Nordseeküste</title>
6     <link rel="stylesheet" type="text/css" href="vallorastyles.css">
7   </head>
8
9   <body>
10    <h1>Hotel Vallora</h1>
11    <p><a href="wegweiser.html">Wegbeschreibung</a></p>
12    
13    <p>
14      Bei Ihrem nächsten Aufenthalt in unserer Gegend würden wir Sie
15      gern persönlich begrüßen und mit unserem freundlichen und kompetenten
16      Service verwöhnen.
17    </p>
18
19    <h2>Zimmerreservierungen:</h2>
20    <p>
21      Unsere Rezeption ist 24 Stunden besetzt. Sie können
22      auch gern unser Online-Buchungsformular ausfüllen.
23    </p>
24  </body>
25 </html>

```

Quelltext-Darstellung Ihrer ersten Webseite im Browser Firefox

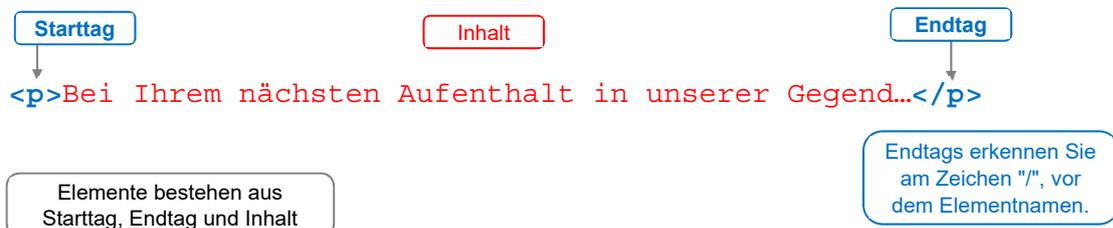
### 3.3 Der Quellcode eines HTML-Dokuments

#### Elemente

Die meisten **Elemente** bestehen aus einem **Starttag**, einem **Endtag** und **Inhalt** (in der Regel Text). Das Starttag wiederum besteht aus dem **Namen des Elements** in spitzen Klammern und das Endtag zusätzlich aus einem **vorangestellten Schrägstrich "/"**.

Für Absätze beispielsweise nutzen Sie das p-Element (für engl. paragraph):

#### Beispiel für ein HTML-Element



Genauso können Sie Überschriften, Listen, Zitate und alle anderen Texte eines HTML-Dokumentes auszeichnen. Daher hat die Hypertext-Auszeichnungssprache HTML ihren Namen. Browser können nun anhand Ihrer Auszeichnungen die Struktur einer Webseite analysieren und die Bedeutung der Texte erkennen.

## Leere Elemente

Einige Elemente haben keine Inhalte. Diese nennt man **leere Elemente**. Um ein Zeilenende anzuzeigen, nutzen Sie beispielsweise das leere Element `br` (für engl. break).

Notieren Sie die folgende Zeichenfolge für einen Zeilenumbruch: `<br>`.

Es gibt keinen sinnvollen Inhalt für einen Zeilenumbruch. Er zeigt nur das Ende einer Zeile an. Somit ist dieses Beispiel ein typischer Vertreter der leeren Elemente.

Leere Elemente können Sie innerhalb des Starttags mit einem abschließenden Schrägstrich `"/` schreiben. Notieren Sie die folgende Zeichenfolge für einen Zeilenumbruch: `<br />`.

Das ist nicht nötig, wird aber von manchen Entwicklern als guter Stil angesehen.

## Attribute

Öffnende Tags und Tags von leeren Elementen können zusätzlich zum Elementnamen noch ein oder mehrere Attribute enthalten. Attribute legen zusätzliche Informationen fest und gelten in der Regel nur für das Element, dem Sie ein Attribut mitgeben. So können Sie beispielsweise einem Element die verwendete Sprache mitgeben, was bei fremdsprachigen Zitaten sinnvoll ist.

Die Regel, dass Attribute nur für das Element gelten, in dessen öffnendem Tag sie stehen, hat ein paar Ausnahmen. Im Kopf-Element einer HTML-Datei (der Bereich zwischen `<head>` und `</head>`) befinden sich Meta-Angaben. Wie bei anderen leeren Elementen auch, befinden sich die wesentlichen Informationen in den Attributen. Die in den `meta`-Tags angegebenen Informationen in den Attributen gelten jedoch für das gesamte HTML-Dokument.

Ein Beispiel finden Sie bereits in der ersten Webseite, die Sie erstellt haben:

```
<meta ① charset="②utf-8" />
```

Das `<meta>`-Tag mit dem `charset`-Attribut

Attribute setzen sich aus einem **Attributnamen** ①, gefolgt von einem Gleichheitszeichen und einem **Attributwert** ② zusammen. Der Attributwert wird in Anführungszeichen eingeschlossen.

Im Beispiel wird dem Browser mitgeteilt, welche **Zeichenkodierung** Sie bei der Erstellung des Dokumentes verwendet haben. Diese Angabe muss mit der Zeichenkodierung übereinstimmen, die Sie im Editor eingestellt haben.

Die Angabe der verwendeten Kodierung gilt nicht nur für das `meta`-Element, sondern für das gesamte HTML-Dokument.

## Zeichensatz korrekt angeben

Geben Sie die falsche Kodierung an, werden Sonderzeichen nicht korrekt dargestellt.

UTF-8 hat sich durchgesetzt und ist somit empfehlenswert für Ihre Webseiten. In den meisten Editoren entspricht das dem voreingestellten Standard. Ob und wo das eingestellt werden kann, hängt vom verwendeten Editor ab – lesen Sie dazu die Anleitung Ihres Editors.

### Tag mit mehreren Attributen

Manchmal wird ein Tag mit mehreren Attributen ausgestattet. Die Attribute werden durch Leerzeichen voneinander getrennt. Die Reihenfolge der Attribute ist beliebig. Beim `<img>`-Tag sind zwei Attribute sogar Pflicht:

1. Das `src`-Attribut gibt an, welches Bild verwendet werden soll.
2. Das `alt`-Attribut liefert eine Beschreibung des Bildes (Alternativtext) für alle Fälle, in denen die Anzeige des Bildes unterbunden wird. Das kann von dem Nutzer selber getan werden (z. B. um Übertragungsvolumen zu sparen). Es kann aber auch sein, dass einem Nutzer kein grafischer Browser zur Verfügung steht, sondern z. B. ein Vorleseprogramm für Blinde.

```

```

Leerzeichen

`<img>`-Tag mit seinen zwei erforderlichen Attributen `src` und `alt`

### Groß- und Kleinschreibung in HTML

In HTML müssen Sie bei Element- und Attributnamen nicht auf Groß- und Kleinschreibung achten. Ob Sie `<title>`, `<TITLE>` oder `<Title>` schreiben, stellt für den Browser kein Problem dar. Ihr Code wird jedoch übersichtlicher, wenn Sie eine einheitliche Schreibweise verwenden. Die Kleinschreibung ist in der früher meist verwendeten Sprache XHTML zwingend, hat sich unter Webentwicklern damals durchgesetzt und gilt heute noch vielen als „guter Stil“.

## 3.4 Webseiten strukturieren

### Absätze erstellen

Auf Webseiten werden Absätze voneinander durch Abstände getrennt. Absätze erstellen Sie mit den Tags `<p>` und `</p>`.

```
<p>Text</p>
```

Syntax für Absätze

- ▶ Geben Sie im Dokumentrumpf das Starttag des `p`-Elements ein (`<p>`) ①, vgl. folgende Abbildung.

```
<body>
  ①<p>Bei Ihrem nächsten Aufenthalt in unserer Gegend würden wir
    Sie gern persönlich begrüßen und mit unserem freundlichen
    und kompetenten Service verwöhnen.</p> ②
  <p>Unsere Rezeption ist 24 Stunden besetzt. Sie können auch
    gern unser Online-Buchungsformular ausfüllen.</p>
</body>
```

- ▶ Geben Sie den Text für den Absatz ein.
- ▶ Schließen Sie den Absatz mit dem Endtag ② ab.
- ▶ Wiederholen Sie die vorherigen Schritte für alle Absätze der Webseite.

Bei Ihrem nächsten Aufenthalt in unserer Gegend würden wir Sie gern persönlich begrüßen und mit unserem freundlichen und kompetenten Service verwöhnen.

Unsere Rezeption ist 24 Stunden besetzt. Sie können auch gern unser Online-Buchungsformular ausfüllen.

*Absätze im Browser dargestellt*

## Zeilenumbrüche erstellen

Einen Zeilenumbruch innerhalb eines Absatzes können Sie mit dem leeren `br`-Element (von engl. break) erzwingen. Zeilenumbrüche sollten Sie dann verwenden, wenn Sie Informationen, die logisch zusammengehören (und deshalb in einem gemeinsamen Absatz stehen sollten) in mehreren Zeilen darstellen wollen.

`Text<br>`

*Beispiel für einen Zeilenumbruch*

Das Element `br` sollte nicht zum Erzeugen von Abständen missbraucht werden. Das geht über Formatierungsanweisungen in einer Formatvorlage viel besser, genauer und komfortabler.

Beispiele für die sinnvolle Verwendung von Zeilenumbrüchen sind die einzelnen Verse einer Liedstrophe oder eines Gedichtes.

Hier ein Beispiel für die Hotel-Vallora-Website:

```
<p>Unsere Rezeption ist 24 Stunden besetzt. Sie können auch  
gern unser Online-Buchungsformular ausfüllen.</p>  
① <p>Tel. 000 000 000<br> ②  
Fax 000 000 000<br>  
E-Mail: info@hotelvallora.com</p>
```

- ▶ Geben Sie das Starttag des `p`-Elements ① ein (`<p>`).
- ▶ Tragen Sie die erste Zeile Ihres Textes ein.
- ▶ Geben Sie `<br>` ② ein.
- ▶ Wiederholen Sie gegebenenfalls die letzten zwei Schritte, um andere Textzeilen und Zeilenumbrüche festzulegen.
- ▶ Schließen Sie den Absatz mit `</p>`.

Unsere Rezeption ist 24 Stunden besetzt. Sie können auch gern unser Online-Buchungsformular ausfüllen.

Tel. 000 000 000

Fax 000 000 000

E-Mail: info@hotelvallora.com

*Zeilenumbrüche im Browser*

## Überschriften erstellen

Die meisten Webnutzer wollen so schnell wie möglich an Informationen gelangen und neigen dazu, Webseiten zu überfliegen. Mithilfe von Überschriften können Sie Inhalte strukturieren und Besucher gezielt durch Ihr Webangebot führen.

```
<h1>Überschriftentext</h1>
```

Beispiel für eine Überschrift 1. Ordnung

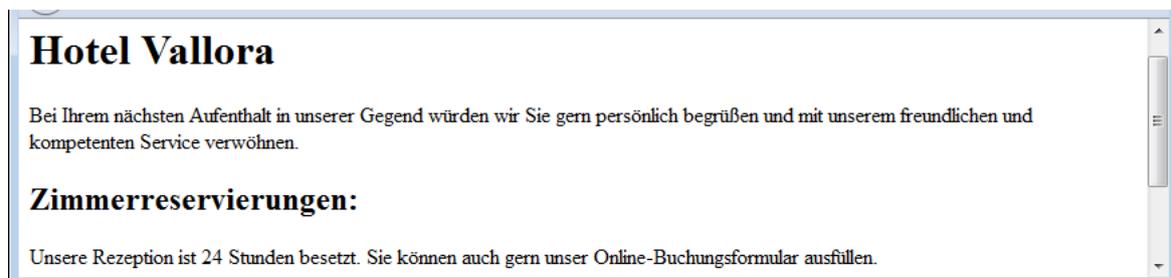
Es gibt 6 verschiedene Überschriften-Ebenen: h1, h2, h3, h4, h5 und h6. Jedes hx-Element (von engl. heading) hat eine Standardschriftgröße. Die Darstellung der Überschriften unterscheidet sich je nach Browser. Bei allen Browsern ist das h1-Element am größten und h6 am kleinsten.

```
① <h1>Hotel Vallora</h1> ②
  <p>Bei Ihrem nächsten Aufenthalt in unserer Gegend würden wir
    Sie gern persönlich begrüßen und mit unserem freundlichen
    und kompetenten Service verwöhnen.</p>
③ <h2>Zimmerreservierungen:</h2> ④
  <p>Unsere Rezeption ist 24 Stunden besetzt. Sie können auch
    gern unser Online-Buchungsformular ausfüllen.</p>
```

- ▶ Geben Sie zu Beginn des body-Elementes das Starttag ① und das Endtag ② des h1-Elements (<h1></h1>) ein.
- ▶ Tragen Sie den Text der Überschrift ein, hier *Hotel Vallora*.
- ▶ Geben Sie das Starttag ③ und das Endtag ④ des h2-Elements (<h2></h2>.) ein.
- ▶ Tragen Sie den Text für diese Überschrift ein, hier *Zimmerreservierungen*.

Um Fehler zu vermeiden, sollten Sie sich das oben beschriebene Vorgehen angewöhnen, d. h., schreiben Sie zunächst den öffnenden und den schließenden Tag – z. B. <p></p> – und geben Sie danach den Inhalt ein. So vermeiden Sie, den schließenden Tag zu vergessen. Dieses Vorgehen gilt auch für das Öffnen und Schließen von Klammern und Anführungszeichen. Viele Editoren unterstützen Sie dabei übrigens dadurch, dass sie solche Paare automatisch schließen.

Eine andere Herangehensweise ist es, lange fertige Texte in den Quellcode einzufügen und die Tags nachträglich hinzuzufügen. Auch hierbei sollten Sie die Tags immer paarweise notieren, also jedes geöffnete Tag gleich wieder schließen.



Überschriften mit Standardformatierung des Browsers

- ✓ Mitunter wirken die Standardschriftgrößen bzw. die Abstände von sinnvoll verwendeten Überschriften unpassend. Setzen Sie Überschriften dennoch so ein, dass eine nachvollziehbare Hierarchie entsteht. Sie können die Formatierung mit CSS genau festlegen.
- ✓ Überschriften wie `h1` und `h2` werden von Suchmaschinen stark gewichtet und haben deshalb einen großen Einfluss darauf, wie weit oben Ihre Webseiten in der Ergebnisliste von Suchmaschinen auftauchen (Ranking).
- ✓ Durch die Verwendung von Überschriften können Bildschirmlesegeräte für Sehbehinderte den Aufbau der Webseite genauer ermitteln und besser zusammenfassen.

## 3.5 Elemente verschachteln

Ihre Webseite besteht aus Elementen, die einander folgen oder ineinander verschachtelt sind.

Auf Ihrer ersten Beispiel-Webseite befindet sich ein Element `html`, darin `head` und `body` (nacheinander) und im `body` wiederum Überschriften und Absätze.

Absätze befinden sich nicht innerhalb von Überschriften – aber innerhalb der Elemente `body` und `html`.

Beachten Sie, dass es nur innere und äußere Elemente gibt (auch **Eltern-Element** bzw. **Kind-Element** genannt).

Stellen Sie sich HTML-Elemente wie Boxen bzw. Kartons vor: viele kleine Boxen können sich in einer großen Box befinden, sie können aber nicht „herausragen“.

In Ihrem Beispiel sehen Sie, dass alle Elemente (Überschriften wie Absätze) innerhalb des Dokumentrumpfes nicht nur geöffnet, sondern auch geschlossen werden.

Wenn Sie nicht korrekt öffnen **und** schließen, erzeugen Sie einen sogenannten **Verschachtelungsfehler** (engl.: nesting error).

**Verschachtelungsfehler führen oft zu Fehldarstellungen im Browser.**

## 3.6 Übung

### Eine weitere Webseite für den Webauftritt erstellen

Level		Zeit	ca. 15 min
Übungsinhalte	<ul style="list-style-type: none"> <li>✓ HTML-Dokument erstellen und Grundgerüst festlegen</li> <li>✓ Seitentitel definieren</li> <li>✓ Text mit Absätzen und Überschriften strukturieren</li> <li>✓ Bild einbinden</li> <li>✓ Hyperlink erstellen</li> </ul>		
Übungsdatei	<i>hotel03</i>		
Ergebnisdatei	<i>hotel03-e</i>		

1. Erstellen Sie eine neue Seite für das Hotel Vallora mit dem Seitentitel *Hotel Vallora - Ausflüge*.
2. Geben Sie eine beliebige Beschreibung von Ausflugszielen ein und strukturieren Sie den Text mit Absätzen und Überschriften.  
Tipp: Geben Sie in einer Suchmaschine die Worte „Blindtext-Generator“ ein, um Textbeispiele zu erhalten.
3. Speichern Sie die Webseite mit dem Namen *ausfluege.html* und testen Sie diese im Browser.

# 4

## Erste optische Anpassungen mit CSS

### 4.1 Erste Schritte mit CSS

#### Webseiten ohne und mit CSS

Prinzipiell funktionieren Webseiten ohne Anpassungen am CSS. Ohne CSS erhalten Sie standardmäßig Texte in schwarzer Schrift, Links sind unterstrichen und blau formatiert, der Hintergrund ist weiß.

Damit Sie wenigstens ein wenig Farbe und erste Anpassungen vornehmen können, finden Sie im vorliegenden Kapitel einige Informationen (z. B. zu Farben und Schriften), mit denen Sie Ihren Webauftritt deutlich sichtbar verändern können.

Wie Sie Ihre Webseite bis ins Detail durchgestalten können, erklärt Ihnen das HERDT-Buch *CSS – Cascading Style Sheets Level 3 (CSS3)*.

#### CSS-Regeln notieren

Die Darstellung Ihrer Webseiten legen Sie mithilfe von Formatierungsanweisungen in der Sprache **Cascading Style Sheets (CSS)** fest. Die sogenannten CSS-Regeln teilen dem Browser mit, wie bestimmte HTML-Elemente formatiert werden sollen (z. B. Größe, Farbe oder Abstand zu anderen Elementen).

Angaben zum Aussehen von Elementen können an drei verschiedenen Stellen notiert werden:

1. Mithilfe des Attributes `style` direkt am Element, das Sie formatieren möchten (gilt wie alle Attribute nur für das entsprechende Element). Da Sie die Anweisungen direkt im Fließtext angeben, werden sie im englischen **inline Styles** genannt. Diese Bezeichnung hat sich als Fachbegriff auch im Deutschen etabliert.
2. In einem eigenen Element im Kopfbereich eines HTML-Dokumentes (gilt nur für das jeweilige Dokument). So werden Style-Angaben in das HTML-Dokument eingebettet und daher nennt man sie oft **eingebettete Styles**.

3. In einer externen Datei (gilt für jedes mit dieser Datei verknüpfte HTML-Dokument, in der Regel alle Seiten eines Webauftritts). Üblich ist die Verwendung einer Datei, in der sich alle Formatierungsangaben für die gesamte Website befinden. Da an einer zentralen Stelle der komplette Webauftritt gepflegt werden kann, spricht man auch von einem **zentralen Stylesheet**.

Für Anfänger ist es häufig verführerisch, Formate direkt am Element zu notieren, also inline Styles zu verwenden. Dabei ist der Bezug zwischen der Formatierung und der Auswirkung besonders eng und gut nachvollziehbar.

Diese Methode ist auf Dauer die umständlichste, denn Formate müssen an jedes Element geschrieben werden. So müssen Sie jeder einzelnen Überschrift mitteilen, dass Sie rot sein soll, wenn Sie auf der gesamten Website rote Überschriften einsetzen möchten.

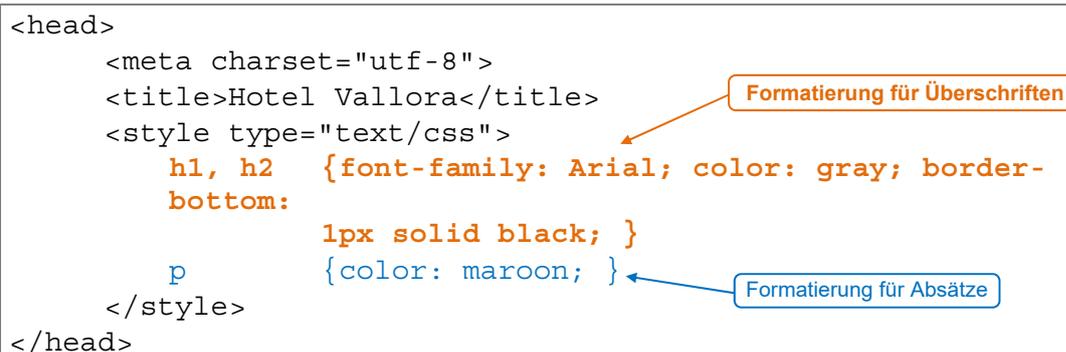
Eine wahre Sisyphos-Arbeit erwartet Sie, wenn Sie später einen anderen Farbton verwenden möchten, denn dazu müssen Sie sämtliche Überschriften finden und mit den neuen Angaben überschreiben.

Nur wenig besser ist die Verwendung eines eingebetteten Stylesheets, das im `<head>`-Bereich der Webseite eingefügt wird. Solche Angaben gelten ausschließlich für das HTML-Dokument, in dem sie notiert sind.

Sie können so immerhin für eine Seite alle Überschriften rot formatieren und müssen bei späteren Änderungen nur noch einmal pro Seite Anpassungen vornehmen. Bei großen Webseiten mit dutzenden von Einzelseiten ist das ebenfalls sehr umständlich.

Im folgenden Beispiel wurden CSS-Regeln für die Formatierung von Absätzen, Überschriften und Hyperlinks auf der Startseite des Hotels Vallora definiert. Die CSS-Regeln können Sie mit der Darstellung der Webseite im Browser auf der nächsten Seite vergleichen.

```
<head>
  <meta charset="utf-8">
  <title>Hotel Vallora</title>
  <style type="text/css">
    h1, h2 {font-family: Arial; color: gray; border-
    bottom:
      1px solid black; }
    p      {color: maroon; }
  </style>
</head>
```



Stylesheet im `head`-Bereich der Startseite (Beispieldatei: „04\_Erste optische Anpassungen mit CSS/hotel/stylesheet.html“)

Später werden Sie die eigentlichen Vorteile von CSS kennenlernen, indem Sie diese Regeln in einem separaten CSS-Dokument speichern. Sogenannte **externe oder zentrale Stylesheets** können Sie mit beliebig vielen Webseiten verknüpfen. Der Vorteil davon ist, dass Sie Ihre Formatierungsregeln zentral verwalten können.

## Wirkung von CSS im Browser betrachten

Die CSS-Regeln bewirken, dass Absatztext in einer dunkelroten Farbe und Überschriften in einer anderen Schriftart – in grauer Farbe und unterstrichen – dargestellt werden.

# Hotel Vallora

← Formatierte Überschrift

[Wegbeschreibung](#)



Bei Ihrem nächsten Aufenthalt in unserer Gegend würden wir Sie gern persönlich begrüßen und mit unserem freundlichen und kompetenten Service verwöhnen.

Zimmerreservierungen: ← Formatierte Überschrift

← Absätze mit der Schriftfarbe Maroon (Dunkelrot)

Unsere Rezeption ist 24 Stunden besetzt. Sie können auch gern unser [Online-Buchungsformular](#) ausfüllen.

Tel. 000 000 000  
 Fax 000 000 000  
 E-Mail: [info@hotelvallora.com](mailto:info@hotelvallora.com)

Darstellung der Webseite im Browser

## Veraltete Vorgehensweisen erkennen

Früher haben Webseitenentwickler die Formatierung ihrer Webseiten direkt in den Tags festgelegt. Die Hintergrundfarbe einer Webseite wurde beispielsweise im `<body>`-Tag definiert.

```
<body bgcolor="blue">
```

Das veraltete Attribut `bgcolor`

Solche Formatierungsattribute finden sich noch in Webseiten, die seit Jahren nicht verändert wurden. Sie sind veraltet und in HTML5-Dokumenten nicht erlaubt.

CSS wird von allen gängigen Browsern unterstützt. Allerdings gibt es Unterschiede in der Wiedergabe von bestimmten CSS-Eigenschaften. Das Testen Ihrer Webseiten in unterschiedlichen Browsern ist wichtig. Es ist nicht unbedingt nötig, dass eine Seite in allen Browsern gleich aussieht, aber Sie sollten dafür sorgen, dass Ihre Seite in jedem Browser ordentlich aussieht und bedienbar ist.

## CSS-Regeln erstellen

CSS ist eine eigenständige Sprache, in der Sie **Regeln** erstellen, aufgrund derer Elemente formatiert werden. So eine Regel kann lauten: Absätze blau darstellen. Sie müssen also nicht für jeden einzelnen Absatz Formatierungsanweisungen notieren. Welche Elemente Sie optisch anpassen, legen Sie mit **Selektoren** fest. Der einfachste Selektor ist der Element-Selektor. Dazu notieren Sie lediglich den Namen eines Elementes, bei Absätzen also ein `p`.

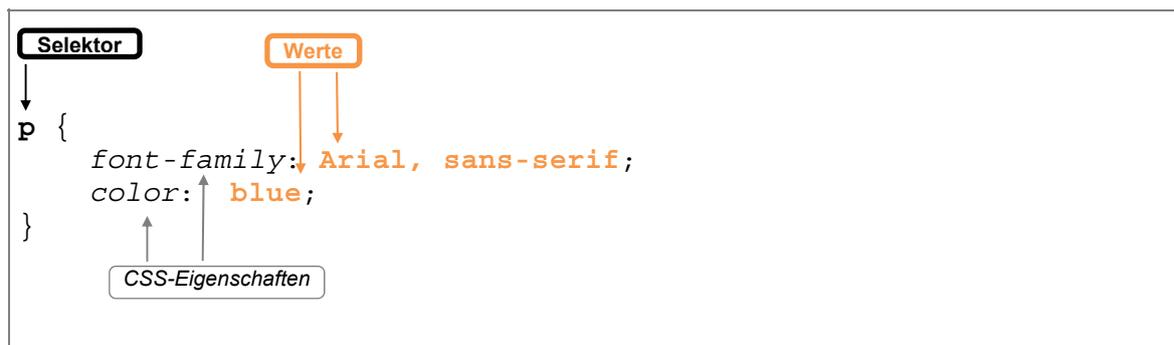
Die gewünschte Formatierung bestimmen Sie in einer **Deklaration**. Eine Deklaration besteht aus dem Namen einer Eigenschaft (z. B. Farbe) und einem Wert (z. B. blau). Solche Gebilde heißen **Eigenschaft-Werte-Paar**. Die Eigenschaft wird von ihrem Wert durch einen Doppelpunkt getrennt, mehrere Eigenschaft-Werte-Paare werden voneinander mit einem Semikolon getrennt.

### Aufbau einer CSS-Regel

```
Selektor1[, Selektor2, ...] {CSS-Eigenschaft1: Wert1; CSS-
Eigenschaft2: Wert2; [...] }
```

Syntax für CSS-Regeln

### Beispiel für eine CSS-Regel



CSS-Selektor (für Absätze) und zwei Deklarationen. Deklarationen werden mittels Semikolon voneinander getrennt.

Das Semikolon und der Leerschritt nach dem letzten Eigenschafts-/Wertpaar sind nicht notwendig. Es handelt sich hier um gute Codepraxis. Wenn Sie auch am Ende immer ein Semikolon und eingeben, können Sie später neue Regeln hinzufügen und gehen kein Risiko ein, dass Semikolon zu vergessen.

Wie bei HTML sind Leerschritte und Zeilenumbrüche nur im Code ersichtlich. Auch in CSS ist das Kleinschreiben der Deklarationen lediglich guter Stil. **Bei Selektoren, denen Sie selber Namen vergeben (z. B. Klassen oder IDs), wird dagegen zwischen Groß- und Kleinschreibung unterschieden.**

Auch Zeilenumbrüche und -einrückungen im Code sind optional. Wie im vorigen Beispiel ersichtlich, verbessern diese die Lesbarkeit des Codes, der Browser ignoriert sie. Daher können Sie Ihren Code mittels Einrückungen und Umbrüchen nach eigenen Wünschen gestalten. Wird mit mehreren Personen an einer Webseite gearbeitet, empfiehlt es sich Richtlinien festzulegen, damit sich jeder gleichermaßen im Code zurechtfindet.

Solche Richtlinien nennt man **Coding Guidelines**. Mit einer Suchmaschine können Sie Beispiele für Coding Guidelines von bekannten Projekten und Organisationen finden und für eigene Projekte als Anregung nutzen.

## 4.2 Eingebettete Stylesheets erstellen

### Einen Bereich für den CSS-Code definieren

Eingebettete Stylesheets werden im `head` der HTML-Dokumente im `style`-Element zwischen den Tags `<style>..</style>` notiert.

- ▶ Öffnen Sie ein HTML-Dokument in Ihrem Texteditor, z. B. die Datei *index.html*.
- ▶ Geben Sie im `<head>`-Bereich direkt unter dem Seitentitel das öffnende Tag und schließende Tag für das Element `style` ein.

```
<head>
  <meta charset="utf-8">
  <title>Hotel Vallora, Nordseeküste</title>
  <style>

  </style>
</head>
```

*Einen Stylesheet-Bereich festlegen*

### CSS-Regeln notieren

In diesem Beispiel notieren Sie Formatierungsdefinitionen für die HTML-Elemente `p`, `h1` und `h2`. Der Text in Absätzen sollte mit der Farbe Dunkelrot versehen werden. Hier wird die Eigenschaft `color` auf den Wert `maroon` (deutsch: dunkelrot) gesetzt. Das ist nur eine Möglichkeit, um Farbwerte festzulegen. Später in diesem Kapitel erfahren Sie mehr über Farbwerte.

Für die Überschriften `h1` und `h2` auf der Webseite legen Sie die Schriftart *Arial* und die Schriftfarbe *gray* (Grau) fest. Zusätzlich sollen alle Überschriften mit einer Linie unterstrichen werden, die sich über die ganze Breite der Webseite erstreckt.

- ▶ Platzieren Sie den Cursor direkt unter dem `<style>`-Tag.
- ▶ Geben Sie den Selektor `p` ① ein.

```
<style>

① p      {color: maroon; } ③
           ②
④ h1, h2 {font-family: Arial, sans-serif; ⑤
           color: gray;
           border-bottom: 1px solid black; } ⑥

</style>
```

*CSS-Regeln für Absätze und Überschriften*

- ▶ Notieren Sie die geschweiften Klammern. Unter Windows drücken Sie die folgenden Tasten `[AltGr] 7` (öffnende Klammer) und `[AltGr] 0` (schließende Klammer), auf einem Mac `[Alt] 8` zum Öffnen und `[Alt] 9` zum Schließen der Klammern.

Um Fehler zu vermeiden, gewöhnen Sie sich an, erst die Klammern und anschließend den Inhalt zu schreiben. So vermeiden Sie es, die schließende Klammer ③ zu vergessen.

- ▶ Ergänzen Sie die Eigenschaft `color` und den Doppelpunkt. Setzen Sie den Wert auf `maroon`, gefolgt von einem Semikolon ②.
- ▶ Geben Sie die Selektoren `h1` und `h2` durch ein Komma getrennt ④ ein.
- ▶ Tragen Sie die geschweiften Klammern, sowie die in der Abbildung ersichtlichen Eigenschaften und Werte ⑤ ein.
- ▶ Überprüfen Sie, ob Sie die schließende geschweifte Klammer ⑥ notiert haben.

### CSS-Regeln gruppieren

Ihr Code bleibt übersichtlicher, wenn Sie CSS-Regeln gruppieren, die mehrere Elemente betreffen. Im vorigen Abschnitt wurden die CSS-Regeln für die Elemente `h1` und `h2` gruppiert: Die zwei Selektoren `h1` und `h2` wurden durch ein Komma getrennt und die gemeinsamen Regeln nur einmal geschrieben.

```
h1, h2 {font-family: Arial; color: gray; }
```

*Gruppierung*

Möchten Sie beispielsweise, dass alle Überschriften mit derselben Schriftart und -farbe versehen werden und dass die Unterstreichung nur auf das Element `h1` angewendet wird, können Sie die CSS-Regeln wie in der folgenden Abbildung ersichtlich definieren. Auf diese Weise werden die gemeinsamen CSS-Regeln gruppiert und nur die sich unterscheidenden CSS-Regeln getrennt aufgelistet.

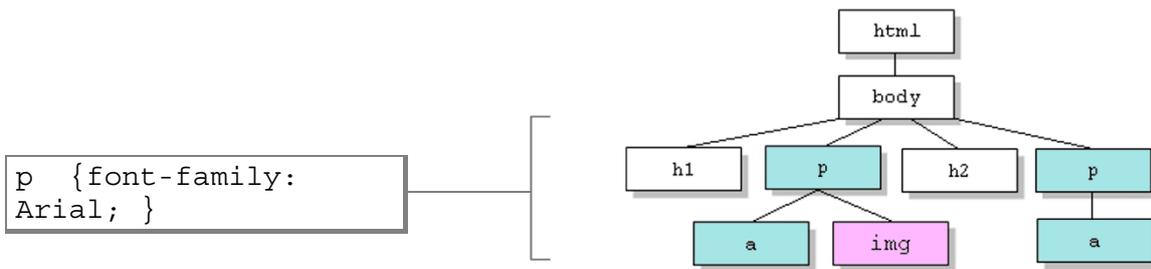
```
h1, h2 {font-family: Arial; color: gray; }
h1 {border-bottom: 1px solid black; }
```

*Gruppierung von Selektoren und einzelner Selektor*

## 4.3 Mit Vererbung richtig arbeiten

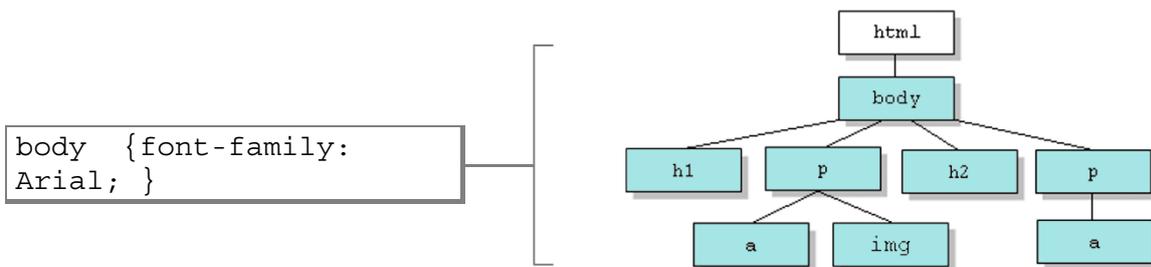
### Wie werden CSS-Regeln vererbt?

Wenn Sie eine CSS-Regel für ein Element festlegen, gilt diese Regel für alle Instanzen dieses Elements. Wenn Sie beispielsweise die Schriftart `Arial` für das `p`-Element definieren, gilt diese Eigenschaft für alle `p`-Elemente auf der Webseite. Fast alle CSS-Regeln werden vererbt. Vererbung (engl.: *inheritance*) ist das Übertragen der Eigenschaften von übergeordneten Elementen auf untergeordnete. Der Vorteil von Vererbung ist, dass Sie eine CSS-Regel für das übergeordnete Element erstellen können, ohne dass Sie diese Regel für jedes einzelne untergeordnete Element definieren müssen. Im folgenden Beispiel gilt die Eigenschaft `font-family` für alle `p`-Elemente und ihre untergeordneten Elemente.



Das `img`-Element steht für ein Bild auf Ihrer Webseite. Obwohl ein Bild nicht anders aussieht, wenn Sie eine andere Schriftart benutzen, betrifft die für `p` festgelegte Schriftart das Bild trotzdem: Für jedes Bild müssen Sie einen Alternativ-Text bereitstellen, falls das Bild einmal nicht geladen werden kann. Weil das `img`-Element dem `p`-Element untergeordnet ist, wird dieser Text dann in derselben Schriftart dargestellt, wie der restliche Absatz.

Im folgenden Beispiel wird die Schriftart `Arial` für das `body`-Element festgelegt. Die Schriftart gilt für alle untergeordneten Elemente des `body`-Elements. Da alle Elemente auf der Webseite dem `body`-Element untergeordnet sind, gilt die Schriftart als Standardschriftart für die Webseite.



## Vererbung

Um den Umgang mit Vererbung zu erlernen, definieren Sie die Standardschriftart im eingebetteten Stylesheet der Startseite.

```
<style>                                ①
  body {font-family: Arial, sans-serif; }
  p    {color: maroon; }
  h1, h2 {color: gray; border-bottom: 1px solid black; }
</style>
```

Standardschriftart festlegen (Beispieldatei 04: „hotel/vererbung.html“)

Da die Schriftart `Arial` im `body`-Element festgelegt wird ①, gilt die Schriftart für alle Elemente der Webseite.

Im Kapitel 13 werden Sie erfahren, weshalb Sie beim Tag `font-family` immer eine Liste von Schriftarten angeben sollten.

## Vererbung überschreiben

CSS-Regeln, die Sie für ein übergeordnetes Element festlegen, können Sie überschreiben. So können Sie Ausnahmen der ursprünglichen Regel festlegen. Auch diese Ausnahme ist wiederum eine Regel, nur für einen kleineren Bereich Ihrer Dokumente.

Sie haben im vorigen Abschnitt eine Standardschriftart für die Webseite (über das `body`-Element) festgelegt. Sie möchten jedoch eine andere Schriftart für Überschriften verwenden.

Sie legen wie zuvor die gewünschte Schriftart fest ①, benutzen aber einen anderen Selektor.

Auf diese Weise unterbrechen Sie die Vererbung. Von diesen Elementen an wird die neu zugewiesene Schriftart an alle Nachfahren weitergegeben.

```
<style>
  body      {font-family: Arial, sans-serif; }
  p         {color: maroon; }
  h1, h2    {font-family: Courier; color: gray; }
</style>
```

Eine Regel überschreiben (Beispieldatei 04: „hotel/vererbung\_ueberschreiben.html“)

## 4.4 Externe Stylesheets verwenden

### CSS effizient einsetzen

Oft besteht ein Webauftritt aus vielen Einzelseiten. Sie könnten Ihre CSS-Regeln kopieren und in alle Seiten einfügen. Diese Vorgehensweise ist sehr mühsam, vor allem, wenn Sie die CSS-Regeln weiterentwickeln wollen oder nachträgliche Änderungen einarbeiten müssen. Möchten Sie beispielsweise die Schriftfarbe Ihrer Webseiten ändern, müssten Sie jede einzelne Webseite öffnen und diese Änderung vornehmen.

Viel effizienter ist es, die CSS-Regeln in einem separaten Stylesheet-Dokument zu speichern und alle Webseiten, für die diese CSS-Regeln gelten sollen, mit dem Stylesheet zu verknüpfen. Änderungen, die Sie später im externen Stylesheet vornehmen, wirken sich auf alle damit verknüpften Webseiten aus. Durch den Einsatz externer Stylesheets können Sie ein konsistentes Erscheinungsbild gewährleisten und Sie sparen Zeit bei der Erstellung und Aktualisierung Ihrer Webseiten. Je nach Bedarf können Sie beliebig viele Stylesheets für Ihre Webseiten erstellen.

### Externe Stylesheets erstellen

Erstellen Sie in Ihrem Texteditor eine neue Datei und speichern Sie sie mit der Dateinamenserweiterung `.css` ab.

- ▶ Erstellen Sie in Ihrem Texteditor ein neues Dokument.
- ▶ Geben Sie die folgenden CSS-Regeln ein:

```
p          {color: maroon; }
h1, h2    {font-family: Arial; color: gray; }
h1        {border-bottom: 1px solid black; }
```

Externe Stylesheet-Datei

- ▶ Speichern Sie das Stylesheet mit der Dateinamenserweiterung `.css`, beispielsweise mit dem Namen `hotelstyles.css`.

Haben Sie ein eingebettetes Stylesheet erstellt, können Sie es in ein externes Stylesheet umwandeln, indem Sie die CSS-Regel kopieren, in ein neues Textdokument einfügen und dieses mit der Dateinamenserweiterung `.css` abspeichern.

**Achtung!** Die Tags `<style>` und `</style>` selber dürfen Sie dabei **nicht mitkopieren**. Diese sind nur der HTML-Container für das eigentliche CSS!

## Externe Stylesheets verknüpfen

Sie verknüpfen ein HTML-Dokument mit einem externen Stylesheet mithilfe eines `link`-Elements. Dieses leere Element tragen Sie im `head`-Bereich ein. Um ein externes Stylesheet mit einem HTML-Dokument zu verknüpfen, benötigt das `<link>`-Tag folgendes Attribute:

- ✓ `rel` legt das Verhältnis zwischen der Webseite und dem Stylesheet fest und erhält den Wert `stylesheet`, falls das Stylesheet ständig mit der Webseite verknüpft ist.
- ✓ `href` teilt dem Browser mit, wo das Stylesheet zu finden ist.

Wenn mehrere Attribute für ein Tag definiert werden, müssen diese durch Leerzeichen voneinander getrennt werden.

```
<link rel="stylesheet" href="Verzeichnis/Dateiname.css">
```

Syntax der Verknüpfung zum externen Stylesheet (Beispieldatei 04: „hotel/link.html“)

Als Beispiel geben Sie die Verknüpfung zum Stylesheet `hotelstyles.css` in die Startseite ein.

- ▶ Öffnen Sie die Webseite, die Sie mit einem Stylesheet verbinden möchten.
- ▶ Platzieren Sie den Cursor im `head`-Bereich und betätigen Sie `↵`.
- ▶ Geben Sie das öffnende Tag des Elements `link` ① ein.

```
<head>
  <meta charset="utf-8">
  <title>Hotel Vallora, Nordseeküste</title>
  <link rel="stylesheet" href="hotelstyles.css">
</head> ①      ②      ③
```

Link zu einem externen Stylesheet

- ▶ Legen Sie das `rel`-Attribut ② fest.
- ▶ Geben Sie das `href`-Attribut mit dem Pfad zum gewünschten Stylesheet ③ ein.  
In diesem Beispiel reicht der Dateiname aus, da sich die CSS-Datei im gleichen Ordner wie die Webseite befindet.

## 4.5 Farbwerte definieren

### Wie werden Farben in CSS festgelegt?

Farben auf Computermonitoren entstehen durch die Mischung von rotem, grünem und blauem Licht. Aus diesem Grund wird das Farbsystem RGB genannt (von den englischen Bezeichnungen *Red*, *Green* und *Blue*). Durch die additive Mischung dieser drei Grundfarben können über 16 Millionen Farbvarianten dargestellt werden. Es gibt mehrere Möglichkeiten, um Farbwerte in CSS festzulegen. Drei davon vergleicht die folgende Tabelle.

Farbname (dt. Bezeichnung)	RGB-Werte	Hexadezimalwerte
aqua (Hellblau)	0,255,255	#00FFFF
black (Schwarz)	0,0,0	#000000
blue (Blau)	0,0,255	#0000FF
fuchsia (Magenta)	255,0,255	#FF00FF
green (Grün)	0,128,0	#008000
gray (Grau)	128,128,128	#808080
lime (Hellgrün)	0,255,0	#00FF00
maroon (Dunkelrot)	128,0,0	#800000
navy (Dunkelblau)	0,0,128	#000080
olive (Olivgrün)	128,128,0	#808000
purple (Violett)	128,0,128	#800080
red (Rot)	255,0,0	#FF0000
silver (Silber)	192,192,192	#C0C0C0
teal (Blaugrün)	0,128,128	#008080
white (Weiß)	255,255,255	#FFFFFF
yellow (Gelb)	255,255,0	#FFFF00

### Farbnamen verwenden

Einige Farben können sie mit ihrem englischen Namen im CSS-Code benennen. Im nebenstehenden Beispiel wird die Schriftfarbe von Absätzen mithilfe einer Farbbezeichnung auf Blau gesetzt.

```
p {color: blue; }
```

*Farbname als Wert verwenden*

Nur wenige der 16,8 Millionen möglichen Farben sind benannt. Daher spielen Farbnamen in der Praxis nur eine kleine Rolle. Nur ein paar leicht zu merkende Namen wie *white* oder *black* werden von einigen Entwicklern hin und wieder eingesetzt.

Wenn Sie Farbnamen verwenden möchten, finden Sie unter der folgenden Adresse eine Liste erlaubter Werte: <http://www.w3.org/TR/css3-color/#colorunits>

## RGB-Werte einsetzen

Sie können Farben auch über ihre RGB-(Rot Grün Blau-) Werte festlegen. RGB-Werte werden oft von Farbwahlssystemen in Bildbearbeitungs- und anderen Programmen verwendet, und Sie können diese aus den entsprechenden Anwendungen in Ihren CSS-Code kopieren.

RGB-Werte bestehen aus drei durch Kommas getrennte Angaben, die wiederum definieren, wie hell die jeweilige Grundfarbe leuchten soll. Daher bedeutet der Wert 0 keine Farbe und der Wert 255 die höchstmögliche Farbintensität. Mit dieser Methode können Sie 16,8 Millionen verschiedene Farben definieren. Im nebenstehenden Beispiel wird die Schriftfarbe von Absätzen mithilfe von RGB-Werten auf Blau gesetzt.

```
p {color: rgb(0,0,255); }
```

*RGB-Wert verwenden*

## RGBa-Werte notieren

RGBa-Werte sind Farbangaben, die genau wie RGB-Werte notiert werden. Zusätzlich gibt es aber einen vierten Wert, der für die Deckkraft der Farbe steht. Ist dieser gleich Null, ist das eingefärbte Element komplett durchsichtig und daher unsichtbar. Bei einem Wert von 1 deckt die Farbe zu 100% und darunter liegende Ebenen scheinen nicht durch. Erlaubt sind Werte zwischen 0 und 1, also z. B. 0.5 für 50% Deckung.

```
p {  
  color: rgba(0,0,255,0.5);  
}
```

*RGB-Wert verwenden*

Obwohl die Angabe auch für Schriften erlaubt ist, macht das nur bei sehr großen und fetten Buchstaben Sinn. Deutlich sichtbar wird die Transparenz bei Hintergrund-Farben vor einem Hintergrundbild.

## Beispiel

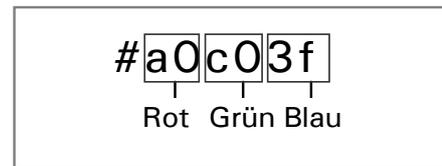
```
div {  
  background: rgba{255,0,0,0.5};  
  width: 10em;  
  height: 10em;  
}
```



*Auszug aus der Datei 03/rgba.html*

## Hexadezimalwerte verwenden

Die im Web am häufigsten verwendete Methode, um Farbwerte zu definieren, ist die Definition mithilfe von Hexadezimalwerten, auch Hex-Codes genannt. Die Angabe eines Hexadezimalwertes beginnt mit einer Raute (#). Dahinter werden die Rot-, Grün- und Blauwerte einer Farbe jeweils durch zweistellige Ziffer-/ Buchstabenkombinationen angegeben. Es handelt sich hier um eine andere Darstellungsweise der RGB-Werte. Sie können genauso viele Farbtöne ansprechen wie mit RGB-Werten.



Im nebenstehenden Beispiel wird die Schriftfarbe von Absätzen mithilfe eines Hexadezimalwerts auf Blau gesetzt.

```
p {color: #0000ff; }
```

*Hexadezimalwert verwenden*



Beachten Sie bei den **Angaben 0**, dass es sich hier um die **Ziffer 0 (Null)** handelt und **nicht um den Buchstaben O**.

## Hexadezimal- und RGB-Werte ermitteln

Mit sogenannten Browser-Erweiterungen (je nach Browser auch AddOn, PlugIn oder Extension genannt) können Sie Farben aus beliebigen Webseiten extrahieren. Suchen Sie nach solchen Erweiterungen mit den Stichworten „colorpicker“ oder „Farbwähler“.

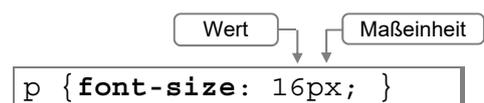
### Was sind „websichere“ Farben?

Früher konnten die meisten Monitore nur 256 Farben darstellen. Daraufhin wurde eine Liste von 216 Farben definiert, die auf allen Monitoren konsistent dargestellt werden konnten. Tatsächlich wurden lediglich 22 der 216 als „websicher“ aufgelisteten Farben auf allen Monitoren einheitlich dargestellt. Heutzutage ist dieses Thema nicht mehr relevant, da Monitore seit Jahren viel mehr Farben darstellen können.

## 4.6 Maßeinheiten in CSS festlegen

### Maßeinheiten für Größen in CSS festlegen

Zur Angabe von Höhen, Breiten oder Abständen müssen Sie einen Wert und eine Einheit angeben. Es ist ein Unterschied, ob ein Gegenstand einen Meter oder einen Kilometer lang ist. Sie können beispielsweise eine Schriftgröße in Pixeln (px), Millimetern (mm), Punkt (pt) oder als Prozentwert (%) angeben. Im nebenstehenden Beispiel wird die Schriftgröße von Absätzen auf 16 px gesetzt.



*Wert mit der Maßeinheit Pixel*

Achten Sie bei der Eingabe von Maßeinheiten auf Folgendes:

- ✓ Geben Sie **kein** Leerzeichen zwischen dem Wert und der Maßeinheit ein.
- ✓ Bei der Angabe des Werts sind Dezimalstellen erlaubt. Dabei müssen Sie einen **Punkt (.) als Dezimaltrennzeichen** verwenden.
 

p { font-size: 10.5px; }

  
*Die Ausgabe des Programms*
- ✓ In CSS gibt es keine Standardmaßeinheit. Sollten Sie einen Wert ohne Maßeinheit festlegen, kann die Angabe von Browser zu Browser unterschiedlich interpretiert oder ignoriert werden.

## Absolute Werte verwenden

Absolute Werte sind Werte, die nicht in Bezug zu anderen Werten auf Ihrer Seite stehen. Sie sind also unabhängig von allen anderen Angaben. Die folgenden Maßeinheiten können Sie in CSS verwenden, um absolute Werte zu definieren:

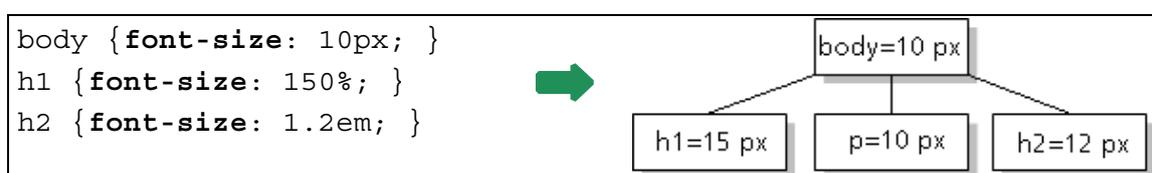
Maßeinheit	Beschreibung
<b>Pixel</b> (px)	Die einzelnen nebeneinanderstehenden Quadrate auf dem Computerbildschirm, die Werte zu Helligkeit bzw. Farben enthalten, werden Pixel genannt. Längenangaben werden oft mit dieser Maßeinheit festgelegt. Die genaue Größe eines Pixels kann von Bildschirm zu Bildschirm unterschiedlich sein und ändert sich je nach Bildschirmgröße und -auflösung. Ein Pixel in CSS entspricht oft nicht den tatsächlichen, physikalischen Pixeln moderner Geräte. Informationen hierzu finden Sie unter: <i><a href="http://alistapart.com/article/a-pixel-identity-crisis">http://alistapart.com/article/a-pixel-identity-crisis</a></i>
<b>Punkt</b> (pt)	Punkt ist ein typografisches Maß, das gleich 1/72 in (bzw. 0,353 mm) ist. Webentwickler, die schon Erfahrung mit Typografie bzw. Grafikgestaltung haben, kennen und verwenden die Maßeinheiten Punkt und Pica. Die Maßeinheit Punkt wird auch in der Textverarbeitung für Schriftgrößen verwendet. Die Angabe von Größen in Punkt ist nur für den Ausdruck sinnvoll.
<b>Inch</b> (in)	Ein englisches Längenmaß, das gleich 25,4 mm ist. Diese Maßeinheit ist im Verhältnis zu den vorherigen Maßeinheiten sehr groß. Da eine zuverlässige Umrechnung in Pixel nicht möglich ist, ist auch diese Maßeinheit lediglich für den Ausdruck interessant.
<b>Millimeter</b> (mm) <b>und</b> <b>Zentimeter</b> (cm)	Millimeter und Zentimeter, obwohl kleiner als Inches, sind problematisch, weil Bildschirme diese Maßeinheiten ebenso wie Inches nicht immer genau wiedergeben. Für den Druck können Angaben in mm gemacht werden.

## Relative Werte verwenden

Relative Werte sind skalierbar und ändern sich im festgelegten Verhältnis zu anderen Werten, von denen sie abhängen, zu denen sie also in Bezug stehen. Sie können die folgenden relativen Maßeinheiten für Ihre Werte verwenden:

Maßeinheit	Beschreibung
<b>Ems</b> (em)	Ein em ist so groß wie der Buchstabe "m". Da Schriftgrößen vererbt werden, ist ein em so groß, wie ein "m" im Elternelement. In der Regel entspricht ein em 16 Pixeln. Geben Sie für <code>body</code> eine Schriftgröße von 20 Pixeln an, ist ein em 20px groß.
<b>Exes</b> (ex)	Diese Maßeinheit funktioniert wie em, allerdings ist der Bezug die Größe des Buchstabens "x".
<b>Prozent</b> (%)	Breiten und Höhen von Elementen können Sie in Prozent angeben. Die Angabe bezieht sich auf die Größe des Elternelementes. Wenn Sie einem Bild in einem 600px breiten Artikel eine Breite von 25% geben, so wird das Bild 150px breit dargestellt. Sie können auch Schriftgrößen in Prozent angeben. Dann bezieht sich die Prozentangabe auf die Schriftgröße des Elternelementes. Wenn die Schrift des Elternelementes 16 Pixel groß ist, führt die Anweisung, eine 75% große Schrift zu verwenden dazu, dass die Schrift 12 Pixel groß dargestellt wird.
<b>Rems</b> (rem)	Wenn Sie die Größe der Schrift in rem notieren, wird diese in Abhängigkeit von der Größe eines em im Wurzel-Element html angegeben. So haben Sie keine Abhängigkeit vom Elternelement. Wenn Sie für html eine Schriftgröße von 16px festgelegt haben, entspricht 1rem immer 16px, egal wie groß die Schrift im Elternelement ist.
<b>Viewport-Breite</b> (vw)	Als Breite des Viewports (engl. <b>viewport width</b> ) wird der Abstand zwischen dem linken und dem rechten Rand der Webseite bezeichnet, also die Breite der Fläche, die der Browser für die Darstellung der Seiten nutzt. 1vw entspricht dabei einem Hundertstel der Viewport-Breite.
<b>Viewport-Höhe</b> (vh)	Mit Höhe des Viewports (engl. <b>viewport height</b> ) wird die Höhe der Fläche bezeichnet, die dem Browser für die Darstellung der Seiten zur Verfügung steht. 1vh entspricht einem Hundertstel der Viewport-Höhe.
<b>Kürzere/längere Kantenlänge</b> (vmin und vmax)	Diese Werte stehen für den jeweils kürzeren bzw. längeren Rand. Bei einem Viewport, der ähnlich einem Portrait höher als breit ist, würde der Browser bei Verwendung der Angabe 1vmax ein Hundertstel der Höhe verwenden und bei Verwendung von 1vmin ein Hundertstel der Breite (weil die horizontale Kante kürzer ist als die vertikale).

Relative Werte können Sie wie im folgenden Beispiel einsetzen:



Die Schriftgrößen der Elemente `h1` und `h2` sind Prozentwerte der Schriftgröße des `body`-Elements

## 4.7 Übung

### Ein externes Stylesheet erstellen und verknüpfen

Level		Zeit	ca. 15 min
Übungsinhalte	<ul style="list-style-type: none"> <li>✓ Externes Stylesheet erstellen</li> <li>✓ Verknüpfung zu externem Stylesheet herstellen</li> </ul>		
Übungsdatei	<i>hotel04</i>		
Ergebnisdatei	<i>hotel04-e</i>		

1. Erstellen Sie ein neues HTML-Dokument.
2. Legen Sie die folgende Formatierung in CSS fest.

Selektor	Deklaration
body	font-family: Arial;
p	color: #003366; font-size: 16px;
h1, h2	color: #339966;
h1	font-size: 24px; border-bottom: 1px solid #339966;
h2	font-size: 20px;

3. Speichern Sie die Datei unter dem Namen *vallorastyles.css*.
4. Vergleichen Sie Ihre CSS-Datei mit der Datei *vallorastyles.css* im Ergebnisordner *hotel03-e*.  
Haben Sie die Farbformatierung der zwei Überschriften gruppiert?
5. Öffnen Sie nacheinander die Webseiten *index*, *wegweiser* und *ausfluege* und verknüpfen Sie diese Dateien mithilfe des `<link>`-Tags mit dem neu erstellten Stylesheet.
6. Testen Sie die Webseiten im Browser.
7. Wechseln Sie zur Stylesheet-Datei und ändern Sie den Wert für die Überschriftenfarben sowie den Wert für die Unterstreichung des Elements `h1` in `#339966` um.
8. Speichern Sie das Stylesheet.
9. Laden Sie die damit verknüpften Webseiten erneut im Browser und betrachten Sie die Änderung.

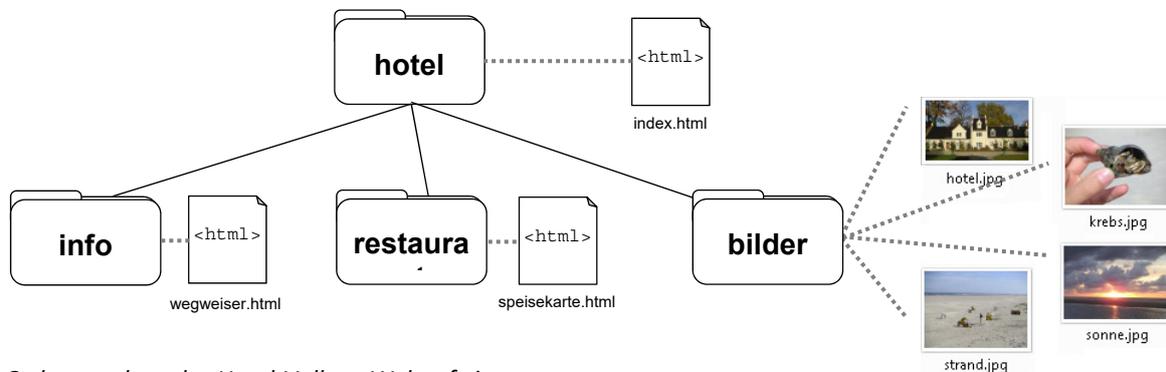
# 5

## Organisation und Struktur

### 5.1 Webseiten in Ordnern organisieren

#### Aufbau der Ordnerstruktur

Sie haben nun gelernt, wie Sie HTML-Dateien und CSS-Dateien erstellen können. Ein fertiger Webauftritt besteht aus vielen Webseiten und **Ressourcen** (z. B. Bild-, Sound- oder Videodateien). Um den Überblick über Ihre Dateien zu behalten, sollten Sie Ihre Webseiten in Ordnern organisieren. Der oberste Ordner Ihrer Ordnerstruktur, der alle Dateien und Ordner Ihres Webauftritts enthält, heißt **Wurzelverzeichnis** oder **Stammordner** (engl.: root directory).



Ordnerstruktur des Hotel-Vallora-Webauftritts

Die Startseite, also die Seite, die zuerst angezeigt wird, wenn Besucher Ihren Webauftritt im Browser laden, z. B. *index.html*, muss sich direkt im Wurzelverzeichnis befinden und darf nicht in einem Unterordner platziert werden. Sie können beliebig viele Unterordner erstellen, um die Übersicht über Ihre Dateien zu behalten. Sie sparen sich viel Aufwand, wenn Sie am Anfang eine sinnvolle Ordnerstruktur für Ihren Webauftritt planen. Die Ordnerstruktur des Webauftritts des Hotels Vallora könnte beispielsweise wie in der Abbildung oberhalb aussehen.

Außerdem ist es sinnvoll, eine Seite *index.html* in jeden Unterordner zu legen. Gibt ein Besucher im Browser *www.example.com/unterverzeichnis* ein, wird automatisch die Datei *www.example.com/unterverzeichnis/index.html* angezeigt.

### Dateinamenserweiterungen aller Dateien in Windows anzeigen

Falls Sie mit Windows arbeiten, ist es empfehlenswert, alle Dateinamenserweiterungen anzuzeigen. Standardmäßig blendet Windows die Dateinamenserweiterungen bekannter Dateitypen aus.

- ▶ Klicken Sie mit der rechten Maustaste auf die Startschaltfläche von Windows und wählen Sie den Kontextmenüpunkt *Explorer* bzw. *Windows-Explorer öffnen*.
- ▶ Wenn Sie mit Windows Vista oder neuer arbeiten, betätigen Sie im Windows-Explorer die Schaltfläche *Organisieren* und wählen Sie den Eintrag *Ordner- und Suchoptionen*.  
*oder* Wenn Sie mit Windows XP arbeiten, rufen Sie den Menüpunkt *Extras - Ordneroptionen* auf.
- ▶ Wechseln Sie im Dialogfenster *Ordneroptionen* zum Register *Ansicht*.
- ▶ Deaktivieren Sie im Feld *Erweiterte Einstellungen* das Kontrollfeld *Erweiterungen bei bekannten Dateitypen ausblenden*.
- ▶ Betätigen Sie die Schaltfläche *Für Ordner übernehmen*.
- ▶ Bestätigen Sie nacheinander mit *Ja* und *OK*.

### Ordner und Dateien benennen

Eine einheitliche Schreibweise bei der Benennung Ihrer Ordner und Dateien ist wichtig, da die Funktionalität der Webseiten von der Schreibweise abhängen kann.

Die folgenden Konventionen sollten Sie bei der Benennung Ihrer Ordner und Dateien beachten:

Konvention	Beschreibung
<b>Kleinschreibung verwenden</b>	Viele Webserver werden mit dem Betriebssystem UNIX betrieben, das zwischen Groß- und Kleinschreibung unterscheidet. Die Dateinamen <i>index.html</i> , <i>Index.html</i> und <i>INDEX.HTML</i> werden beispielsweise als drei verschiedene Namen behandelt.
<b>Keine Umlaute</b>	Da nicht alle Webserver mit Umlauten richtig umgehen können, sollten Sie ae, oe und ue verwenden, z. B. <i>uebersicht</i> .
<b>Keine Sonderzeichen</b>	Verwenden Sie keine Sonderzeichen wie zum Beispiel ß, =, @ oder %.
<b>Keine Leerzeichen</b>	Verwenden Sie statt Leerzeichen ein Minus <input type="checkbox"/> , um längere Dateinamen optisch zu unterteilen, z. B. <i>wir-ueber-uns</i> . Weit verbreitet ist auch die Kamelhöcker-Schreibweise, bei der die Unterteilung mittels Großbuchstaben vorgenommen wird, z. B. <i>wirUeberUns.html</i>

## Ordner erstellen

Erstellen Sie Ordner für den Beispiel-Webauftritt des Hotels Vallora und legen Sie Ihre Dateien in den entsprechenden Ordnern ab.

- ▶ Erstellen Sie im Stammordner (Wurzelverzeichnis) *hotel* drei Unterordner mit den Namen *info*, *restaurant* und *bilder*.
- ▶ Legen Sie die Webseite *wegweiser.html* im Ordner *info* und das Bild *hotel.jpg* im Ordner *bilder* ab.
- ▶ Laden Sie die Webseite *index.html* in das Browserfenster.

## 5.2 Relative und absolute Pfade definieren

### Neue Pfade angeben

Wenn Sie eine Datei in einen anderen Ordner verschieben, müssen Sie alle Verknüpfungen zu dieser Datei aktualisieren, indem Sie den neuen Pfad angeben.

- ▶ Öffnen Sie die Webseite *index.html* in Ihrem Texteditor.
- ▶ Fügen Sie dem Hyperlink den Pfad zum Ordner *info* ① hinzu.

```
<h1>Hotel Vallora</h1>
<p><a href="①info/wegweiser.html">Wegbeschreibung</a></p>
```

*Pfade aktualisieren (Beispieldatei 05: „hotel/neue\_ordner\_und\_aktualisierte\_pfade.html“)*

- ▶ Speichern Sie die Webseite und testen Sie sie im Browser.  
Der Hyperlink zur Webseite *wegweiser.html* funktioniert. Die Webseite *wegweiser.html* wird jedoch mit der Standardtextformatierung angezeigt. Das liegt daran, dass in dieser Datei der Link zum Stylesheet noch nicht aktualisiert ist. Diese Aktualisierung nehmen Sie in der Übung am Ende des Kapitels vor.

### Relative Pfade benutzen

Im vorigen Beispiel haben Sie **relative Pfade** festgelegt. Relative Pfade benötigen einen Bezugspunkt. So, als ob Sie jemandem den Weg von Ihrem aktuellen Standpunkt zu einer bestimmten Adresse erklären, teilen Sie dem Webserver mit einem relativen Pfad mit, wie er sich durch die Verzeichnisstruktur bewegen muss, um eine bestimmte Datei zu finden. Ausgang dieser Wegbeschreibung (Pfadangabe) ist immer der aktuelle „Standort“, also die aktuell angezeigte Seite.

### Beispiel

Indem Sie die Pfadangabe *info/wegweiser.html* in dem Dokument *index.html* notieren, weisen Sie den Webserver an, im Verzeichnis, in dem *index.html* liegt, nach einem Ordner *info* zu suchen, diesen zu öffnen und darin die Datei *wegweiser.html* zu suchen und an den Browser zu senden.

Relative Pfade werden verwendet, um auf andere Seiten oder Dateien im selben Webauftritt zu verweisen. Wenn Sie Ihre Website später im Internet veröffentlichen, kopieren Sie diese auf einen anderen Computer (den Server). Weil Sie dabei die komplette Struktur kopieren, wird es auch auf diesem Computer neben der Datei *index.html* den Ordner *info* geben. Relative Pfade werden also auch nach der Veröffentlichung funktionieren, ohne dass Sie Anpassungen vornehmen müssen.

### Nach unten wechseln

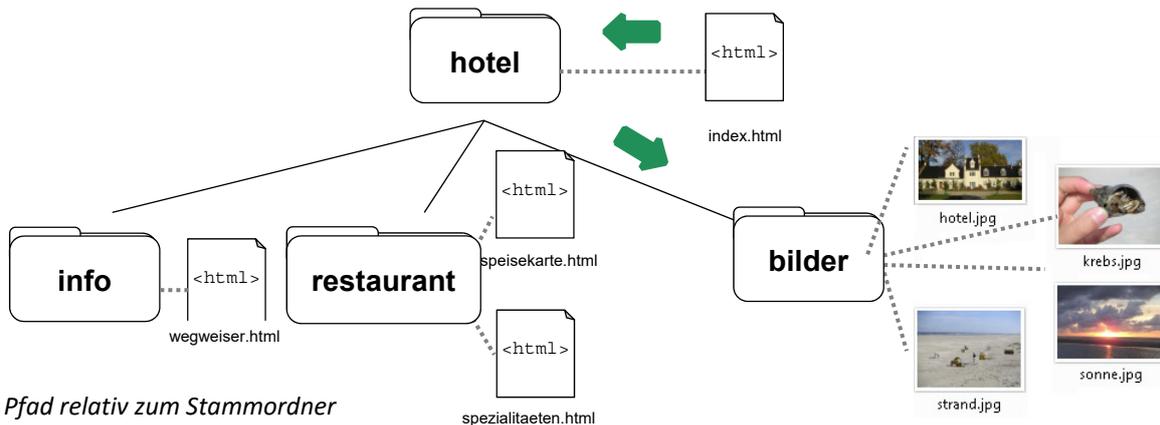
Relative Pfade weisen den Weg relativ zum **aktuellen** Standort aus. Zum Beispiel wird der Weg von der Beispielwebseite *index.html* im Stammordner *hotel* nach **unten** zur Bilddatei *hotel.jpg* im Unterordner *bilder* wie folgt definiert.

```

```

Unterordner

*Pfad zu einer Datei in einem Unterordner des Stammordners*



*Pfad relativ zum Stammordner*

Möchten Sie eine Verknüpfung zu einer Datei im gleichen Ordner erstellen, reicht es den Dateinamen anzugeben.

Einen Hyperlink von der Webseite *speisekarte.html* zur Webseite *spezialitaeten.html* im Ordner *restaurant* legen Sie beispielsweise wie folgt fest:

```
<a href="spezialitaeten.html">Spezialitäten aus unserer Region</a>
```

*Pfad zu einer Datei im selben Ordner*

### Nach oben wechseln

Sie können auch relative Pfade zu Dateien festlegen, die sich weiter **oben** in der Ordnerstruktur befinden. Mithilfe von zwei Punkten, gefolgt von einem Schrägstrich, teilen Sie dem Webserver mit, dass er die Zieldatei **eine** Ebene höher in der Ordnerstruktur findet. Diese Schreibweise ist eindeutig, da jeder untergeordnete Ordner genau einen übergeordneten Ordner hat.

Beispielsweise erstellen Sie einen Hyperlink auf der Webseite *wegweiser.html* im Ordner *info*, der zurück zur Startseite *index.html* im Stammordner *hotel* führt, wie folgt:

```
<a href=" ../index.html ">Zurück zur Startseite</a>
```

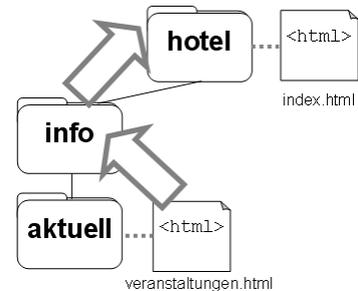
Übergeordneter Ordner

*Pfad zu einer Datei im übergeordneten Ordner*

Wenn Sie mehr als eine Ebene nach oben wechseln möchten, geben Sie die entsprechende Anzahl von Punktpaaren gefolgt von Schrägstrichen an. Möchten Sie beispielsweise auf einer Webseite in einem Unterordner des Ordners *info* einen Link zur Startseite erstellen, legen Sie den folgenden Pfad fest:

```
<a href=" ../../index.html ">Zur Startseite</a>
```

*Pfad zu einer Datei zwei Stufen höher in der Ordnerstruktur*



## Absolute Pfade benutzen

Sie können auch **absolute Pfade** definieren, indem Sie die komplette Adresse der zu findenden Datei angeben. Absolute Pfade werden beispielsweise für Hyperlinks zu anderen Webseiten im Internet verwendet. Hier geben Sie die URL (kurz für Unified Resource Locator) ein. Die URL einer Webseite fängt üblicherweise mit *http://* oder *https://* (verschlüsselt) an.

Beispielsweise möchten Sie auf einer Webseite im Webauftritt des Hotels Vallora einen Hyperlink zur Startseite der Touristenzentrale erstellen. Sie legen hier einen absoluten Pfad fest, indem Sie die URL der Startseite angeben, z. B.:

```
<a href="http://www.touristikzentrale.com">Touristikzentrale</a>
```

*Pfad zu einer (fiktiven) Webseite*

Möchten Sie beispielsweise einen Hyperlink zur Webseite *hotels.html* der Touristikzentrale erstellen, die sich im Ordner *info* befindet, legen Sie den folgenden absoluten Pfad fest:

```
<a href="http://www.touristikzentrale.com/info/hotels.html">Hotels</a>
```

*Pfad zu einer (fiktiven) Webseite im Unterordner eines Webauftritts*

## Die Angabe des Protokolls ist für absolute Pfade erforderlich

Um Daten verlässlich zu transportieren, sind Protokolle notwendig. In diesen ist geregelt, wie Daten abgerufen und übertragen werden, aber auch, was bei Fehlern geschehen soll. Für die Übertragung von Webseiten verwenden Sie das Hyper Text Transfer Protocol (*http*) oder das Hypertext Transfer Protocol Secure (*https*). Bei letzterem werden die Daten verschlüsselt übertragen, was das Mitlesen (auf legalem Weg) verhindert.

Da es viele verschiedene Protokolle gibt, müssen Sie dem Browser mitteilen, wie er eine Datei von einem anderen Computer anfordern soll. Daher müssen absolute Pfade mit der Angabe des Protokolls (für Webseiten: *http://www.touristikzentrale.com*) beginnen. Allerdings ist es nicht notwendig, das zu verwendende Protokoll explizit zu benennen. Folgende Angabe funktioniert ebenfalls, vorausgesetzt die Seite, die Sie betrachten, wurde von einem Webserver ausgeliefert und nicht aus dem Datei-System Ihres Rechners geladen: *//www.touristikzentrale.com*

Dies veranlasst den Browser implizit das Protokoll weiter zu benutzen, mit dem die aktuell geöffnete Seite übertragen wurde. Das heißt: lautet die Adresse der aktuellen Seite *http://www.hotel-vallora.de*, wird versucht, *http://www.touristikzentrale.com* zu öffnen. Betrachten Sie aber *https://www.hotel-vallora.de*, wird versucht *https://www.touristikzentrale.com* zu öffnen. Das setzt allerdings voraus, dass der Server unter *www.touristikzentrale.com* eine Übertragung per *https* anbietet.

Die implizite Verwendung des Protokolls macht vor allem Sinn, wenn Sie selber *https* verwenden. Wenn Sie Inhalte von anderen Seiten per *http* einbinden (z. B. Bilder oder CSS-Dateien), erhält der Besucher Ihrer Webseite einen Warnhinweis vom Browser, dass nicht sichere Inhalte nachgeladen werden sollen und er verliert womöglich das Vertrauen in Ihre Website – wenn Sie einen Online-Shop betreiben, kann das beispielsweise dazu führen, dass ein Kauf abgebrochen wird.

### Lokale absolute Pfade

Auch auf Ihrem Computer können Sie mit absoluten Pfaden arbeiten. Diese sehen unter Windows so aus *Laufwerk:\Verzeichnis\Datei.endung* – diese sind aber für die Verwendung in Webseiten ungeeignet, da dieser Pfad bei den Besuchern Ihrer Webseite wohl nicht vorhanden sein dürfte.

## 5.3 Elemente mit unterschiedlicher Darstellung

Jedes HTML5-Element besitzt eine logische Bedeutung (Semantik).

HTML-Dokumente sollten Sie streng nach logischen Gesichtspunkten auszeichnen. Verwenden Sie beispielsweise keine Überschrift, wenn ein Text groß dargestellt werden soll. Nehmen Sie alle optischen Anpassungen über CSS-Formatierungen vor.

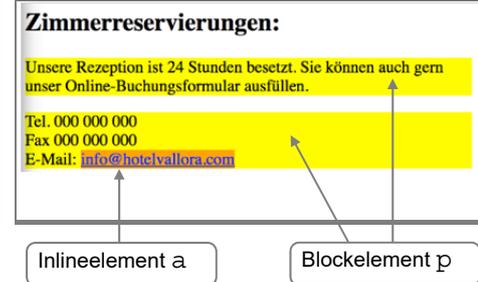
An den bisherigen Beispielen haben Sie gesehen, dass sich die Elemente unterschiedlich verhalten. Absätze beispielsweise bilden einen abgegrenzten Block. Sie gehören zu den Blockelementen.

**Blockelemente** sind Elemente, die einen eigenständigen Absatz im Textfluss erzeugen. Dazu gehören zum Beispiel alle Überschriften, Listen (aber keine Listenpunkte) und Absätze.

Für Links wäre dieses Verhalten sehr lästig, denn dann würde jeder Link für sich allein in einer Zeile stehen. Links beanspruchen keine eigenständigen Blöcke, sie fließen im Text in den vorhandenen Zeilen mit. Elemente, die in den Zeilen bleiben, nennt man **Inline-Elemente** (von engl. line = Zeile).

Blockelemente wie Absätze (gelb) werden als Block dargestellt, der die gesamte Seitenbreite einnimmt. Nachfolgende Elemente werden darunter dargestellt. Inlinenelemente wie Links (orange) beanspruchen nur so viel Platz, wie ihr Inhalt benötigt. Sie fließen wie Worte in der Zeile mit.

Darstellung der Datei 04/block-inline.html



**Daneben gibt es zahlreiche weitere Darstellungsmöglichkeiten:** Tabellen, Tabellenzellen, geordnete und ungeordnete Listen, Listeneinträge, Bilder – all diese Elemente verhalten sich unterschiedlich.

Eine Übersicht über die zahlreichen Darstellungsmöglichkeiten und wie sich diese mittels CSS sinnvoll verändern lassen, finden Sie im Herdt-Buch *CSS – Cascading Stylesheets Level 3 (CSS3)*.

## HTML-Dokument in logische Bereiche unterteilen

Im Folgenden werden Elemente vorgestellt, mit denen Sie Ihr HTML-Dokument in Bereiche unterteilen können, die eine logische Bedeutung innehaben. Diese Bedeutung wird beispielsweise von Screenreadern, Suchmaschinen und anderen Programmen verstanden.

Üblicherweise lässt sich eine Webseite mindestens in folgende Bereiche aufteilen: Kopfbereich, Navigation, Hauptbereich und Fußbereich. Sehr oft gibt es außerdem noch eine Marginal-Spalte mit weiterführenden Informationen zum Hauptbereich.

Für jeden dieser Bereiche stellt HTML ein Element bereit: `header`, `nav`, `main`, `footer` und ggf. `aside`. Wichtig ist der bestimmungsgemäße Gebrauch. Diese Elemente sind nicht darstellungsabhängig zu verwenden. Ein `footer` ist beispielsweise kein Kasten am unteren Ende der Webseite. Er ist im HTML-Standard über seinen Inhalt definiert und darf sich durchaus an anderer Stelle befinden (siehe Tabelle unten).

Eine Liste mit Erläuterungen, den Standard-Rollen und allen erlaubten ARIA-Rollen sowie Code-Beispielen finden Sie unter <https://www.w3.org/TR/html5/sections.html>.

```
<h1>Hotel Vallora</h1>
<nav>
  <h2>Hauptnavigation</h2>
  <ul>
    <li><a href="index.html">Startseite</a></li>
    [...]
  </ul>
</nav>
<article>
  <h2>Wir über uns</h2>
  <p>Lorem ipsum...</p>
  <aside>
    <h3>Verwandte Artikel</h3>
```

```

    <p>Lorem ipsum...</p>
  </aside>
</article>
<article>
  <h2>Kontakt</h2>
  <p>Lorem ipsum...</p>
</article>

```

Diesen Code finden Sie in der Beispieldatei 05: „html5-outlining.html“

Jeder Bereich kann übrigens ein `header`-Element (Kopf) und/oder ein `footer`-Element für Anmerkungen beinhalten. Hier eine Übersicht über HTML5-Struktur-Elemente und deren logische Bedeutung.

Elementname	Bedeutung	Standard-Aria-Rolle
<code>article</code>	In sich abgeschlossener Inhalt, der in der Regel für sich allein genommen verständlich ist	<code>article</code>
<code>section</code>	Ein (thematisch zusammengehörender) Bereich einer Seite, der z.B. mehrere Artikel zu einem Thema zusammenfasst. Ein solcher Bereich kann aber auch einen Artikel in Kapitel unterteilen. Bereiche beginnen typischerweise mit einer Überschrift.	<code>region</code>
<code>nnav</code>	Navigationsbereich, in der Regel nur die Hauptnavigation Eine längere Liste von Links im Fußbereich zu Seiten im Angebot kann ebenfalls mit <code>nav</code> ausgezeichnet werden, muss aber nicht. Das <code>footer</code> -Element mit seiner Bedeutung gibt schon genug Anhaltspunkte über die Bedeutung der Links. In jedem Fall sollte für eine Liste von Links das Listenelement <code>ul</code> benutzt werden.	<code>navigation</code>
<code>aside</code>	Weiterführende Informationen, die beispielsweise einem <code>article</code> mitgegeben werden Oft sind das Links zu ähnlichen Inhalten, es kann aber auch kontextbezogene Werbung sein o.ä. Dargestellt werden diese Inhalte auf Desktop-Ansichten typischerweise in einer Randspalte.	<code>complementary</code>
<code>h1</code> , <code>h2</code> , <code>h3</code> , <code>h4</code> , <code>h5</code> , <code>h6</code>	Gliedernde Überschriften, welche ein Element hierarchisch strukturieren	<code>heading</code>
<code>header</code>	Thematische Einführung für das direkt folgende Element oder den Bereich, in dem es sich befindet (z.B. einen <code>article</code> )	<code>banner</code>
<code>footer</code>	Enthält Informationen über den Bereich, in dem er sich befindet Bei einem Artikel z. B. Informationen zum Autor (sollten mittels <code>address</code> ausgezeichnet werden, wenn sie Kontaktinformationen enthalten), Veröffentlichungsdatum, Urheberrechtshinweise usw.	<code>contentinfo</code>

## Blockelemente verwenden

Das Verhalten von verschiedenen Blockelementen können Sie anhand des folgenden Beispiels erkennen. Hier werden Sie die Blockelemente auf der Startseite des Hotels Vallora betrachten und die Webseite mit dem Element `blockquote` ergänzen, um ein Zitat hinzuzufügen. Die vorhandenen `<img>`- und `<a>`-Tags sollten Sie zunächst nicht beachten.

- ▶ Öffnen Sie im Texteditor die in den vorherigen Kapiteln erstellte Seite *index.html*.
- ▶ Platzieren Sie den Cursor unter dem ersten Absatz und geben Sie die abgebildeten `<blockquote></blockquote>`-Tags, Texte und `<cite></cite>`-Tags ① ein. Mit dem `cite`-Element geben Sie die Quelle des Zitats an.

```
<p>Bei Ihrem nächsten Aufenthalt in unserer Gegend würden wir Sie
gern persönlich begrüßen und mit unserem freundlichen und
kompetenten Service verwöhnen.</p>
<blockquote>
Hier bin ich Mensch, hier darf ich sein.
<cite>Johan Wolfgang von Goethe</cite>
</blockquote>
```

Die Blockelemente `p` und `blockquote`

- ▶ Speichern Sie die Webseite und betrachten Sie sie im Browser. Das Element `blockquote` wird durch Abstände oben und unten von anderen Blockelementen getrennt. Es wird in der Standardtextformatierung angezeigt, da Sie in Ihrem externen Stylesheet noch keinen Stil für dieses Element definiert haben.

## Was sind Inline-Elemente?

Inline-Elemente...

- ✓ erzeugen keinen neuen Absatz im Textfluss
- ✓ verhalten sich wie Worte in einem Fließtext

HTML bietet Ihnen die Möglichkeit, kurze Zitate zur Kennzeichnung von wörtlicher Rede zu verwenden, ohne für diese einen eigenen Block bereitstellen zu müssen. Dazu dient das Element `q` (engl.: quotation).

```
<h1> ... </h1>
<p>
<q> ... </q>
</p>
```

Inline-Element

Mit `cite`, das ebenfalls zu den Inline-Elementen gehört, können Sie den Urheber eines Zitates benennen.

Inline-Element	Beschreibung
<code>q</code>	Dieses Inline-Element wird für kurze Zitate innerhalb eines Absatzes verwendet. Text innerhalb dieses Elements wird standardmäßig in Anführungszeichen dargestellt.
<code>cite</code>	Dieses Inline-Element wird verwendet, um die Quelle eines Zitats auszuzeichnen. Text innerhalb dieses Elements wird standardmäßig mit Kursivschrift dargestellt.

## Inline-Elemente einsetzen

Das Verhalten von Inline-Elementen können Sie anhand des folgenden Beispiels beobachten:

- ▶ Wechseln Sie zur Webseite *index.html*.
- ▶ Geben Sie direkt vor dem `</p>`-Tag <sup>①</sup> im letzten Absatz den abgebildeten Code ein:

```
<p>Bei Ihrem nächsten Aufenthalt in unserer Gegend würden wir Sie gern persönlich begrüßen und mit unserem freundlichen und kompetenten Service verwöhnen.
<q>Die größte Offenbarung ist die Stille</q><cite> (Laotse)</cite>. ① </p>
```

- ▶ Speichern Sie die Webseite und betrachten Sie diese im Browser.

Das Zitat und seine Quelle sind Bestandteile des Absatzes und erben die für das Element `p` definierten Formatierungen, vgl. folgende Abbildung. Beide Elemente werden zusätzlich mit ihren Standardformatierungen versehen – das `q`-Element erscheint in Anführungszeichen und das `cite`-Element wird in Kursivschrift angezeigt.

Bei Ihrem nächsten Aufenthalt in unserer Gegend würden wir Sie gern persönlich begrüßen und mit unserem freundlichen und kompetenten Service verwöhnen. "Die größte Offenbarung ist die Stille" (Laotse).

*Inline-Elemente `q` und `cite` im Browser*

- ▶ Speichern Sie die Webseite und aktualisieren Sie die Ansicht im Browser.

Der Unterschied zu Blockelementen ist offensichtlich. Inline-Elemente werden nicht von dem restlichen Text abgesetzt. Sie bilden keinen eigenen Block, sondern laufen im Text weiter mit.

Dennoch können Sie die Elemente an ihren Anführungszeichen (`q`) und an der kursiven Schrift (`cite`) erkennen.

Hier das Ergebnis, wenn Sie einen Zitat-Block und ein Inline-Zitat mit Autorenangabe verwenden.

Bei Ihrem nächsten Aufenthalt in unserer Gegend würden wir Sie gern persönlich begrüßen und mit unserem freundlichen und kompetenten Service verwöhnen. "Die größte Offenbarung ist die Stille" (Laotse).

Gönnen Sie sich  
Erholung und Entspannung  
in unserer Oase des Wohlfühlens  
im landschaftlichen Idyll.

*Die Elemente `q` und `blockquote` im Browser (Beispieldatei 05: „hotel/block\_inline.html“)*

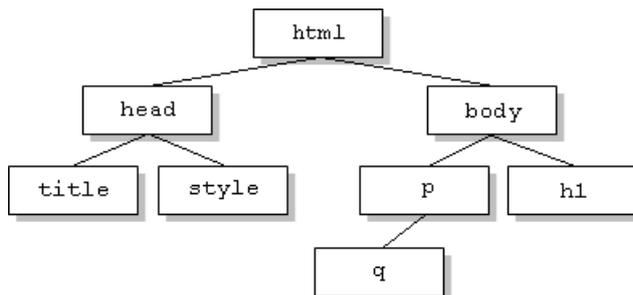
Auch Links zählen zu den Inline-Elementen. So können Sie in einem Fließtext einzelne oder mehrere zusammengehörende Worte eines Satzes verlinken, ohne dass diese aus dem Sinnzusammenhang in einen eigenen Block gestellt werden.

Neben Inline-Elementen und Block-Elementen gibt es zahlreiche weitere Darstellungsarten, die in den nächsten Kapiteln benannt werden.

## 5.4 Elemente richtig verschachteln

### Wie funktioniert die Verschachtelung in HTML?

HTML-Elemente werden ineinander verschachtelt. Sie können sich die Struktur eines HTML-Dokuments wie einen Stammbaum vorstellen. Das Element `html` steht ganz oben und alle anderen Elemente im Dokument sind Nachfahren dieses Elements.



„Stammbaum“ eines HTML-Dokuments

Die Beziehung zwischen Elementen wird anschaulich mit Eltern-Kind-Beziehungen beschrieben. Es gibt **Geschwister-Elemente**, **Eltern-Elemente**, **Kind-Elemente**, **Vorfahren-Elemente** und **Nachfahren-Elemente**. Eltern-Elemente sind gleichzeitig auch Vorfahren-, Kind-Elemente auch Nachfahren-Elemente.

Betrachten Sie den oben abgebildeten beispielhaften „Stammbaum“. Die direkten Nachfahren von `html` sind die Elemente `head` und `body`. Direkte Nachfahren werden Kind oder Kind-Element genannt. Zwischen den weiteren Nachfahren wird nicht unterschieden, Enkel oder Urenkel gibt es daher nicht. Die Elemente `title` und `style` sind dem Element `head` direkt untergeordnet und somit Kinder **und** Nachfahren von `head`. Vom Element `html` sind sie ebenfalls Nachfahren, ohne weitere Unterscheidung. Die Elemente `p` und `h1` sind wiederum dem Element `body` direkt untergeordnet, also Kinder und Nachfahren von `body`, aber ebenfalls nur Nachfahren von `html`.

Weiterhin kann das Element `p` eigene Kinder haben, wie zum Beispiel das Element `q`. Das Element `q` ist ebenfalls ein Nachfahre von `body` und `html`. Elemente auf der gleichen Ebene im Stammbaum, wie zum Beispiel `p` und `h1` oder `title` und `style`, sind Geschwister-Elemente.

### Auf die korrekte Reihenfolge von Tags achten

Nur wenn Sie Elemente richtig ineinander verschachteln, können Sie gewährleisten, dass Ihre Webseiten in allen Browsern richtig dargestellt werden.

Wenn Sie zum Beispiel ein kurzes Zitat in einem Absatz mit dem Element `q` auszeichnen, sieht der HTML-Code wie folgt aus:

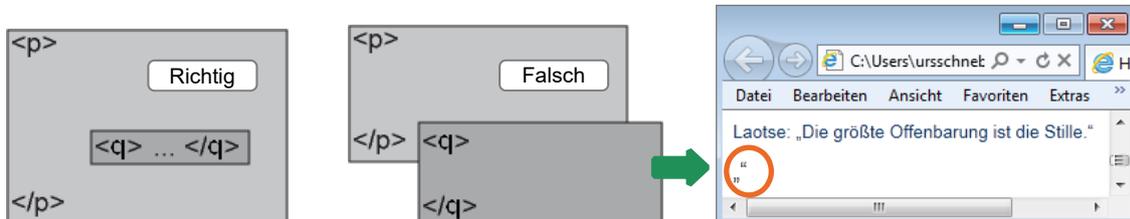
```
<p>Laotse: <q>Die größte Offenbarung ist die Stille.</q></p>
```

Richtige Reihenfolge

Eine Grundregel für das Schreiben von HTML-Code lautet: Jedes Element wird geschlossen, bevor das Elternelement geschlossen wird. Das Element `q` dürfen Sie beispielsweise nicht erst nach dem Element `p` schließen.

```
<p>Laotse: <q>Die größte Offenbarung ist die Stille.</p></q>
```

*Falsche Reihenfolge*



*Das Element `q` richtig und falsch verschachtelt und das Aussehen der Fehlverschachtelung im Internet Explorer (Beispieldatei 05: „hotel/verschachtelung.html“)*

## 5.5 Semantische Textauszeichnung verwenden

HTML dient dazu, Texte entsprechend ihrer Bedeutung auszuzeichnen. In HTML5 gibt es daher keine Möglichkeit, den Browser anzuweisen, einer Schrift ein bestimmtes Aussehen mitzugeben.

Zwar stellt der Browser Überschriften automatisch groß und fett dar, Autorengaben kursiv und Inline-Zitate in Anführungszeichen, Sie sollten dieses Verhalten aber nie nutzen, um das Aussehen Ihrer Webseite zu beeinflussen.

Jede Software, die Ihre Webseite verarbeitet, verlässt sich auf die korrekte Auszeichnung der Inhalte.

Webseiten werden nicht nur mit Webbrowsern angezeigt. Blinde Menschen nutzen beispielsweise Screenreader. Blinde Besucher sind darauf angewiesen, dass ein Text, der als Überschrift ausgezeichnet ist, auch einen neuen Abschnitt der Dokumentshierarchie beginnt.

Auch Suchmaschinen besuchen Ihre Seiten im Internet. Unlogische Auszeichnung kann bei Suchmaschinen zu fehlerhaften Auswertungen führen und damit die Platzierung verschlechtern.

Es ist schlechter Stil und macht unnötige Arbeit bei der Pflege der Seiten, wenn Sie logisch unpassende Elemente für optische Manipulationen verwenden.

Hier eine Übersicht über einige HTML-Elemente und ihre Standard-Darstellung. Eine vollständige alphabetische Übersicht finden Sie im Anhang A.

Tags	Verwendung	Standarddarstellung
<code>&lt;a&gt; ... &lt;/a&gt;</code>	Anker-Element für Links und Linkziele	Inline (in Kombination mit dem <code>href</code> -Attribut blau und unterstrichen)
<code>&lt;abbr&gt; ... &lt;/abbr&gt;</code>	Abkürzungen	Inline (in Kombination mit dem <code>title</code> -Attribut gepunktet unterstrichen)
<code>&lt;q&gt; ... &lt;/q&gt;</code>	Kurze Zitate, wörtliche Rede	Inline (in Anführungszeichen)
<code>&lt;strong&gt; ... &lt;/strong&gt;</code>	Wichtiger Text	Inline (fett)
<code>&lt;figure&gt; ...&lt;/figure&gt;</code>	Container für Illustrationen, wie Bilder, Tabellen, Animationen o. ä., optional eine Beschreibung in <code>&lt;figcaption&gt;...&lt;/figcaption&gt;</code>	Block (auch <code>figcaption</code> )
<code>&lt;img&gt;</code>	Bild oder Grafik	Inline-Block
<code>&lt;details&gt; ...&lt;/details&gt;</code>	Verborgener Inhalt, enthält <code>&lt;summary&gt; ...&lt;/summary&gt;</code> - Ein Klick auf <code>summary</code> öffnet die Details	Darstellung wie Block (nur <code>summary</code> ) aber mit einem Pfeil, der anzeigt, ob die Details aufgeklappt sind
<code>table&gt; ...&lt;/table&gt;</code>	Für die Darstellung tabellarischer Daten (Daten, die sich in Spalten mit Überschriften darstellen lassen). Enthält zwingend weitere Elemente, wie <code>tr</code> , <code>td</code>	Tabelle
<code>&lt;div&gt; ...&lt;/div&gt;</code>	Gruppiert Blockelemente, ohne eine weitere Bedeutung hinzuzufügen	Block
<code>an&gt; ...&lt;/span&gt;</code>	Gruppiert Worte oder Inline-Elemente, ohne eine weitere Bedeutung hinzuzufügen	Inline

## 5.6 Zeichencodierung und Kommentare

### Zeichencodierung festlegen

Am Anfang des `<head>`-Bereichs wird die Zeichencodierung festgelegt. Die (falsche) Bezeichnung Zeichensatz (engl. character set) hat historische Gründe.

UTF-8 ist eine Kodierung, Unicode ist ein Zeichensatz.

```
<head>
  <meta charset="utf-8">
</head>
```

Die Zeichencodierung im `<head>`-Bereich festlegen

Mit der Zeichencodierung UTF-8 stellen Sonderzeichen, landestypische Zeichen und kaufmännische Zeichen im Text des HTML-Dokuments, wie zum Beispiel Umlaute, ©, è, €, £ oder ¥ kein Problem dar. Aber auch komplette fremdländische Texte mit chinesischen oder kyrillischen Zeichen sind möglich.

So können Sie ohne Probleme in ein ansonsten deutsches Dokument ein russisches Zitat einfügen. Vergessen Sie dabei nicht die Sprachangabe mittels lang-Attribut!

```
<p>Gorki hat einmal gesagt:</p>
<blockquote lang="ru">
  Слово — одежда всех фактов, всех мыслей.
</blockquote>
<p>Auf deutsch würde man sagen: <q>Das Wort - Bekleidung aller
Fakten, aller Gedanken.</q></p>
```

Mit UTF-8-Kodierung können auch kyrillische Buchstaben verwendet werden.

## HTML-Code mit Kommentaren versehen

Kommentare sind Texte, die dem HTML-Code hinzugefügt werden, um nicht selbsterklärende Code-Bereiche zu erläutern. Kommentare werden im Browser nicht angezeigt. So können Sie beispielsweise kennzeichnen, wo bestimmte Bereiche des Codes anfangen bzw. aufhören oder an welchen Stellen künftig neuer Code hinzugefügt wird.

Ein HTML-Kommentar beginnt mit `<!--` und wird mit der Zeichenfolge `-->` abgeschlossen. Text innerhalb dieser Tags wird nicht im Browser angezeigt. Ein Kommentar wird beispielsweise auf Webseiten eingegeben, um den Autor und die Aktualität des Dokumentes anzugeben:

```
<!--
Autor:   Marie Mustermann (Abteilung 123)
Stand:   xx/xx/xxxx
Projekt: Selbstdarstellung der Muster AG
-->
```

*Kommentar eingeben*

Kommentare werden auch häufig eingesetzt, um Code-Bereiche vorübergehend zu deaktivieren. Hierdurch können Sie an einem bestimmten Code-Bereich weiterarbeiten und ihn erst „freischalten“, nachdem der Code fertig ist.

Darüber hinaus ist das Auskommentieren von Teilen der Seite hilfreich bei der Fehlersuche. So können Sie große Bereiche oder den gesamten Inhalt einer Seite auskommentieren und dann mehr und mehr Teile „reaktivieren“, bis Sie die problematische Stelle identifiziert haben.



Obwohl Ihre Kommentare im Browser nicht angezeigt werden, haben Besucher die Möglichkeit, den Quellcode der Webseite anzuzeigen. Somit haben sie auch Zugang zu den Kommentaren im Code. Achten Sie deshalb darauf, dass Ihre Kommentare keine vertraulichen Informationen beinhalten.

## Stylesheets mit Kommentaren versehen

Ihre CSS-Dateien können Sie auch mit Kommentaren versehen. Kommentare sind besonders nützlich, um bestimmte Formatierungen schnell wiederzufinden. In CSS fängt ein Kommentar mit der Zeichenfolge `/*` an und wird mit der Zeichenfolge `*/` abgeschlossen.

```
/*Formatierung für den body
Hintergrundfarbe, Schriftfarbe und Schriftfamilie*/

body {background-color: #F0E68C; color: #0000FF; font-family:
Verdana, Arial, Helvetica, sans-serif;}
```

Sowohl HTML-Kommentare, als auch CSS-Kommentare können nicht geschachtelt werden!

## 5.7 Übung

### Dateien in Ordner verschieben und anpassen

Level		Zeit	ca. 15 min
Übungsinhalte	<ul style="list-style-type: none"> <li>✓ Ordner erstellen</li> <li>✓ Dateien verschieben</li> <li>✓ Kommentare eingeben</li> </ul>		
Übungsdatei	<i>hotel05</i>		
Ergebnisdatei	<i>hotel05-e</i>		

1. Erstellen Sie im Stammordner *hotel* einen neuen Ordner mit dem Namen *stylesheets*.
2. Verschieben Sie die Stylesheets *hotelstyles* und *vallorastyles* in den Ordner *stylesheets*.
3. Passen Sie die Verknüpfungen zum Stylesheet dem neuen Speicherort an.
4. Geben Sie einen Hyperlink in die Webseite *wegweiser* (im Ordner *info*) ein, der zurück zur Startseite führt.
5. Verschieben Sie die Webseite *ausfluege.html* in den Ordner *info* und passen Sie hier die Links zum Stylesheet, zur Startseite sowie die URL des `img`-Elementes entsprechend an.
6. Öffnen Sie die Webseite *index* und passen Sie den Hyperlink zur Webseite *wegweiser* an, sodass dieser zum neuen Speicherort dieser Datei führt.
7. Geben Sie einen Kommentar oben in der Webseite *wegweiser* ein.
8. Testen Sie die Webseiten und ihre Links im Browser.
9. Zeigen Sie den Quellcode der Webseite *wegweiser* in Ihrem Browser an.

# 6

## Grafiken und Multimedia einbinden

### 6.1 Mit Grafiken arbeiten

#### Grafiken in Webseiten integrieren

Der Einsatz von Fotos, Bildern und Diagrammen sowie Multimedia-Elementen (Audio und Video) bietet Ihnen die Möglichkeit, Ihre Webseite anschaulich und interessant zu gestalten.

! Beachten Sie die Lizenzbestimmungen von allen Werken, die Sie einsetzen. Häufig ist die Verwendung nicht kostenfrei bzw. überhaupt nicht erlaubt. Gehen Sie deshalb sehr sorgsam bei der Auswahl von fremden Inhalten vor. Wenn möglich, erstellen Sie die benötigten Inhalte selbst oder geben Sie deren Erstellung in Auftrag. Dabei muss vertraglich ausdrücklich festgelegt sein, dass die Verwendung in Webseiten erlaubt ist.

Sogenannte Foto-Stocks sind eine gute Anlaufstelle für Bilder (eine Suche nach „photo stock“ in der Suchmaschine Ihrer Wahl wird Ihnen zahlreiche Treffer liefern). Auch hier sollten Sie die Lizenzbestimmungen lesen, in denen geregelt wird, wofür Sie die erworbenen Inhalte verwenden dürfen.

Viele kostenlose Medien-Inhalte bietet <http://commons.wikimedia.org/wiki/Hauptseite>

#### Grafiken einfügen und mit einem Alternativtext versehen

Mit dem Tag `<img>` (von engl. image = Grafik) binden Sie eine Grafik in ein HTML-Dokument ein. Über das Attribut `src` (von engl. source = Quelle) geben Sie die URL zur Grafik an, z. B. einen Dateinamen.

Die Syntax des `<img>`-Tags lautet:

```

```

Ein Bild sagt mehr als tausend Worte. Bilder sind also Inhalte. Deswegen ist es wichtig, Bilder auch jenen zugänglich zu machen, die die Bilder nicht sehen können.

Für jede Grafik, die Sie in Ihr Dokument einbinden, ist laut HTML-Standard eine aussagekräftige Kurzbeschreibung (Alternativtext) Pflicht. Den Alternativtext legen Sie über das `alt`-Attribut fest. Dieser Text wird angezeigt, wenn die Grafik nicht dargestellt werden kann oder soll. Dies ist beispielsweise der Fall, wenn der Besucher Ihrer Webseite die Darstellung von Grafiken ausgeschaltet hat. Auch Benutzern, die mit Bildschirmlesegeräten im Web unterwegs sind (z. B. Blinde), wird der Alternativtext vorgelesen.

Für Sie als Webseitenbetreiber ist der Alternativtext zudem interessant, weil er Schlüsselwörter (engl.: keywords) enthalten kann und damit Suchmaschinen wichtige Hinweise zur Grafik gibt. Das wiederum wirkt sich positiv auf die Auffindbarkeit Ihrer Site aus.

Wenn Sie eine Grafik mit einem Link versehen, sollte der Alternativ-Text außerdem Aufschluss über das Linkziel geben.

Nur rein dekorative Schmuckgrafiken erhalten ein leeres `alt`-Attribut.

```

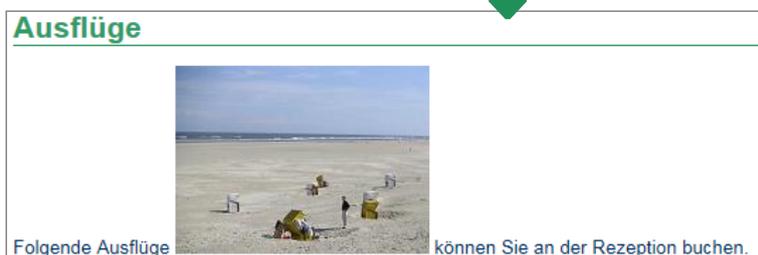
```

So wird eine Grafik von Screenreadern stillschweigend übergangen.

### Beispiel: Ein Bild in eine Webseite integrieren

Im Beispiel platzieren Sie die vorhandene Grafik in der Beispieldatei `06_Grafiken einbinden/hotel/info/ausfluege.html` mitten im Text, um die Auswirkung zu beobachten. Zusätzlich versehen Sie die Grafik mit einem Alternativtext.

```
<h1>Ausflüge</h1>
<p>Folgende Ausflüge  können Sie an der Rezeption buchen.</p>
```



Eingebundene Grafik wird angezeigt (Beispieldatei 06: „hotel/info/ausfluege.html“)

Die Abbildung zeigt, dass sich Grafiken einerseits wie Buchstaben verhalten – sie stehen im Text, laufen mit diesem mit und werden auch am Zeilenende umgebrochen. Grafiken haben also einige Darstellungseigenschaften mit Inline-Elementen gemeinsam.

Andererseits erzwingen Sie eine große Zeilenhöhe, da Grafiken nicht automatisch von Text umflossen werden. Dass Elemente eine gewisse Höhe für sich beanspruchen, kennen Sie bereits von Block-Elementen. Grafiken werden als **Inline-Block-Elemente** dargestellt, die standardmäßig auf der Grundlinie einer Zeile stehen – darunter ist noch Platz für die Unterlängen von Kleinbuchstaben wie „p“ oder „g“.

Um den Alternativtext im Browser anzeigen zu lassen, können Sie die Anzeige von Grafiken im Browser ausschalten, um beispielsweise teure Kosten bei Datentarifen im Ausland zu vermeiden. In den folgenden Schritten wird die Vorgehensweise im Internet Explorer beschrieben.

- ▶ Klicken Sie zunächst die Titelleiste mit der rechten Maustaste an und wählen Sie den Eintrag *Menüleiste*.
- ▶ Rufen Sie in der Menüleiste die Menüpunkte *Extras - Internetoptionen* auf.
- ▶ Wechseln Sie im Dialogfenster *Internetoptionen* zum Register *Erweitert*.
- ▶ Scrollen Sie nach unten zur Rubrik *Multimedia* und deaktivieren Sie das Kontrollfeld *Bilder anzeigen*.
- ▶ Laden Sie das Dokument *ausfluege.html* im Internet Explorer gegebenenfalls erneut. Anstatt der Grafik wird der Alternativtext angezeigt.
- ▶ Wiederholen Sie die vorigen Schritte, sodass Grafiken im Internet Explorer wieder angezeigt werden.



Alternativtext ansehen

### Text als Tooltip erscheinen lassen

Mithilfe des Attributs `title` können Sie einen Text erscheinen lassen, wenn der Besucher den Mauszeiger über das Bild bewegt. Legen Sie einen sogenannten Tooltip wie folgt fest:

```

```



- ! Die meisten Menschen nutzen das Web inzwischen auf touchfähigen Mobilgeräten ohne Maus. Diesen werden `title`-Texte nicht angezeigt, weil es keinen Mauszeiger gibt, der auf einem Element ruhen kann.

## Grafiken mit Beschriftungen versehen

Das Element `figure` wird für Illustrationen jeglicher Art genutzt. Eine Illustration im Sinne von HTML5 kann auch eine Tabelle oder eine Animation sein.

Benutzen Sie die Elemente `figure` und `figcaption`, um Grafiken eine Beschriftung zuzuordnen:

```
<figure>
  
  <figcaption>Sandkrebs mit Haus</figcaption>
</figure>
```



Code und Darstellung im Browser (Beispieldatei „06\_Grafiken einbinden/beschriftung.html“)

## 6.2 Grafiken im Web

### Ein Grafikformat wählen

Es gibt vier digitale Grafikformate, die für das Web besonders geeignet sind und sich in der Webentwicklung etabliert haben: SVG, GIF, JPG und PNG. Die vier Formate unterscheiden sich in den folgenden Punkten:

<b>Farbtiefe</b>	Die Farbtiefe bestimmt, wie viele Bits zum Speichern der Farbe eines Pixels verwendet werden. Je größer die Farbtiefe, umso klarer und detailgetreuer sind die Farben in der Grafik. Die Farbtiefe ist besonders für Fotos wichtig, da diese viele Farben aufweisen. Eine hohe Farbtiefe bedingt andererseits große Dateien.
<b>Kompression</b>	Die Dateigröße einer Grafik können Sie mithilfe eines Bildbearbeitungsprogramms verringern. Die Formate PNG, GIF und JPG nutzen sogenannte verlustbehaftete Komprimierungsverfahren, bei denen Informationen verloren gehen. Sie müssen die Kompressionsrate und den damit verbundenen Verlust gegen den Gewinn (schnellere Übertragung durch geringe Dateigröße) abwägen.
<b>Animation</b>	Bei vielen einfachen Animationen im Web handelt es sich nicht um Videos, sondern um einzelne Grafiken, die nacheinander angezeigt werden. Diese lassen sich im GIF-Format speichern.
<b>Transparenz</b>	Bei GIF-Grafiken kann eine Farbe im Bild komplett durchsichtig gemacht werden. Bei PNG- und SVG-Grafiken hingegen können Bereiche mit mehreren Farben entweder komplett oder auch teilweise durchsichtig gemacht werden. Auch teil-transparente Verläufe sind möglich.
<b>Skalierbarkeit</b>	Im Gegensatz zu allen anderen Formaten lassen sich SVG-Grafiken endlos verlustfrei vergrößern.

## Übersicht über die Grafikformate

	GIF	JPG	PNG	SVG
Farbtiefe	Bis 256 versch. Farben	Bis 16,7 Mio. Farben	Bis 16,7 Mio. Farben	Bis 16,7 Mio. Farben
Kompression	Verlustbehaftet	Verlustbehaftet	Verlustfrei	Verlustfrei
Transparenz	Nur Volltransparenz	Nein	Voll- und Teiltransparenz	Voll- und Teiltransparenz
Animation	Animated GIF	Nein	Über das MNG-Format	Ja
Eignung	Für Grafiken	Für Fotos	Für Grafiken und Fotos	Für Grafiken
Unterstützung	Alle Browser	Alle Browser	Alle Browser*	Alle Browser**

\* Animated PNG unterstützen nur Firefox und Safari ab Version 8 (auch unter iOS 8)

\*\* Android erst ab Version 4.4. SVG stellt viele Möglichkeiten zur Verfügung, die nicht alle von jedem Browser vollständig unterstützt werden, ggfs. schließen Plugins solche Lücken (siehe [http://de.wikipedia.org/wiki/Scalable\\_Vector\\_Graphics](http://de.wikipedia.org/wiki/Scalable_Vector_Graphics))

## Die Dateigröße von Grafiken verringern

Digitale Bilder (außer SVG) sind aus einzelnen Bildpunkten (sogenannten Pixeln) aufgebaut. Die Anzahl der Pixel in einem Bild wird **absolute Auflösung** genannt. Die absolute Auflösung wird immer als Breite und Höhe angegeben, und zwar in genau dieser Reihenfolge. Ein Bild mit einer absoluten Auflösung von 800 x 600 ist 800 Pixel breit und 600 Pixel hoch.

Je höher die Bildschirmauflösung, desto kleiner wird das Bild.



Die Dateigröße sollten Sie so klein wie möglich halten, sodass das Bild schnell lädt. Die Dateigröße verringern Sie, indem Sie die Anzahl der Pixel (absolute Auflösung) oder die Farbtiefe verringern. JPG-Grafiken können stärker komprimiert werden. Alle diese Methoden führen zu einer Verringerung der Qualität.

## Bildgrößen angeben

Mit den Attributen `height` und `width` können Sie die Höhe und Breite der Grafik angeben, z. B. ``

In diesem Fall wird die Grafik automatisch auf die festgelegte Größe skaliert. Wenn eine große Grafik eingebunden und mit den Attributen `height` und `width` verkleinert wird, beansprucht diese mehr Speicherplatz als ein Bild, das schon vorab mit einem Bildbearbeitungsprogramm verkleinert wurde. Wird hingegen ein kleines Bild eingebunden und nachträglich vergrößert, verliert das Bild an Schärfe und kann grobkörnig wirken. Wird das Bild unproportional vergrößert bzw. verkleinert, wirkt das Bild gestaucht oder gezerrt.

Die Verwendung der Attribute `height` und `width` ist sinnvoll, damit der Browser schon vor dem vollständigen Laden des Bildes den benötigten Platz bei der Berechnung der Webseite berücksichtigen kann. Wenn Sie ein Bild mehrmals und in verschiedenen Größen in einem einzigen HTML-Dokument verwenden möchten, vermeiden Sie durch das Laden nur einer einzigen Datei unnötige Serveranfragen und Datenverkehr. Beispielsweise möchten Sie ein größeres Bild an einer Stelle anzeigen und woanders auf derselben Seite als Miniaturbild einfügen.

Auf hochauflösenden Displays gilt es ein weiteres Phänomen zu berücksichtigen. Hochauflösende Geräte setzen Angaben für Höhe, Breite und Schriftgröße nicht wie angegeben um. Damit wird vermieden, dass auf normalen Bildschirmen ausreichend große Schriften auf Handys unleserlich klein werden.

Dasselbe gilt für Bilder. Wenn Sie als Größe 100px x 200px angeben, werden dem Bild 200px x 400 px tatsächlicher Bildschirmauflösung bereitgestellt. Wenn Sie hochauflösende Bildschirme unterstützen wollen, sollten Sie die Bilder also doppelt so breit und doppelt so hoch wie gewünscht abspeichern. Dann erscheinen sie auch auf Displays mit 300dpi und mehr gestochen scharf.

## Grafiken verkleinern

Wenn Sie Grafiken in der Größe verändern, ist es empfehlenswert, die veränderte Datei unter einem neuen Namen zu speichern. Auf diese Weise überschreiben Sie die Ausgangsdatei nicht und können später immer wieder auf sie zurückgreifen.

Auch mit einfachen Grafikprogrammen wie dem Programm Microsoft Paint aus dem Lieferumfang von Windows bzw. dem Programm Vorschau, das mit OS X installiert ist, können Sie Ihre Grafiken verkleinern und webtauglich machen. Je nach Programm gehen Sie wie in den folgenden zwei Abschnitten beschrieben vor:



Originalbild  
(„smiley.png“)



Vergrößert



Verzerrtes Bild

### Microsoft Paint verwenden

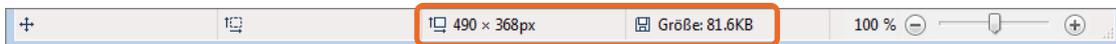
- ▶ Klicken Sie mit der rechten Maustaste auf die Grafikdatei, z. B. die Beispieldatei *06\_Grafiken einbinden/obst.jpg*, und wählen Sie den Kontextmenüpunkt *Öffnen mit - Paint*. Die Höhe und Breite der Grafik sowie die Dateigröße werden unten im Fenster angezeigt.



- ▶ Klicken Sie in Paint im Register *Start* in der Gruppe *Bild* auf die Schaltfläche *Größe ändern*.

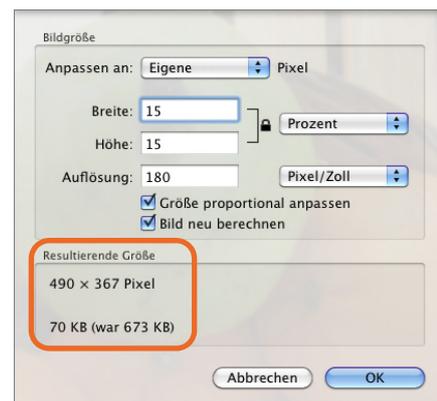


- ▶ Geben Sie im Dialogfenster *Größe ändern/Zerren* im Feld *Horizontal* einen Prozentwert ein, z. B. 15. Die Breite wird automatisch proportional angepasst.
- ▶ Speichern Sie die Datei über das Register , Menüpunkt *Speichern unter*, unter einem neuen Namen, z. B. *obst\_klein.jpg*.
- ▶ Die neuen Dimensionen werden sofort nach der Größenänderung angezeigt, die neue Dateigröße jedoch erst nach dem Speichern. Die Dimensionen und die Dateigröße variieren von Programm zu Programm.



### Mac Vorschau verwenden

- ▶ Klicken Sie doppelt auf die Grafikdatei, z. B. die Datei *obst.jpg* im Ordner *Beispieldateien 06*. Das Programm Vorschau wird gestartet und die Grafik geöffnet.
- ▶ Rufen Sie den Menüpunkt *Werkzeuge - Größenkorrektur* auf.
- ▶ Geben Sie im geöffneten Dialogfenster im Feld *Breite* einen Prozentwert ein, z. B. 15. Die Höhe wird automatisch proportional angepasst. Die neuen Dimensionen sowie die aktuelle und ursprüngliche Dateigröße werden unten im Dialogfenster angezeigt.
- ▶ Speichern Sie die Bilddatei unter einem neuen Namen, z. B. *obst\_klein.jpg*.



Sehr viele Bildmanipulationen lassen sich auch mit dem für Windows kostenlos erhältlichen Programm IrfanView durchführen, das auch ein sehr schneller Bildanzeiger ist.

## 6.3 Grafiken für unterschiedliche Ausgabegeräte bereitstellen

Das Element `picture` bietet Ihnen die Möglichkeit, mehrere Bilder bereit zu stellen und den Browser anzuweisen, wann welches Bild angezeigt werden soll.

### Anwendung des `picture`-Elementes

```
① <picture>
②   <source srcset="bild-HD.jpg"
      media="(min-width: 64em)" >
③   <source srcset="bild-normal.jpg"
      media="(min-width: 40em)" >
④   <source srcset="bild-klein.jpg"
      media="(min-width: 20em)" >
⑤   
      <p>Beschriftung</p>
⑥ </picture>
```

*Beispiel für die Verwendung des Elementes `picture`*

- ① Zunächst öffnen Sie das `picture`-Element mit dem öffnenden Tag.
- ② Geben Sie nun eine Quelle für ein Bild an und legen Sie eine Bedingung fest, die zutreffen soll, damit der Browser das Bild anzeigt (im Beispiel soll das Fenster mindestens 64em breit sein).
- ③ Geben Sie eine weitere Quelle und Bedingung an, hier für mittelgroße Bildschirme.
- ④ Kleine Bildschirme, wie sie in Smartphones verbaut werden, werden häufig unterwegs benutzt, wo die Internet-Verbindung schlecht und teuer sein kann – hier sollten Sie ein Bild mit möglichst wenig Kilobyte bereit stellen, damit das Bild schnell geladen wird und wenig Kosten verursacht.
- ⑤ HTML-Elemente, die ein Browser nicht kennt, werden ignoriert. Geben Sie zuletzt deshalb ein Bild im `img`-Element an, das angezeigt wird, wenn ein Browser das `picture`-Element nicht kennt oder keine der angegebenen Bedingungen zutrifft (im Beispiel wären das Bildschirme, die weniger als 20em Breit sind)
- ⑥ Schließen Sie das `picture`-Element.

## 6.4 Videos einbetten

Videos werden mittels `video`-Element eingebunden. Die Verwendung des Elementes ist sehr einfach. Probleme gibt es dagegen mit den unterschiedlichen Video-Formaten. Nicht jeder Browser unterstützt alle Formate.

Um Videos zu verwenden, gehen Sie wie folgt vor:

- ▶ Legen Sie ein `video`-Element an, indem Sie das öffnende und schließende Video-Tag notieren: `<video></video>`
- ▶ Geben Sie mindestens eine Quelle für ein Video an.
- ▶ Geben Sie einen Text an, der ausgegeben werden soll, falls das Video nicht abgespielt werden kann. Verlinken Sie dort auf die Datei, die wiedergegeben werden soll, damit sie heruntergeladen und in einem anderen Player abgespielt werden kann.

```
<video>
  <source src="video.ogv" type="video/ogv">
  Ihr Browser unterstützt keine HTML5-Videos. Sie können <a
href="video.ogv">die Datei video.ogv hier herunterladen</a>.
</video>
```

In der Praxis sollten Sie das MPEG-4/H.264 Video-Format verwenden, da dies als einziges von allen aktuellen Browsern unterstützt wird (außer Opera Mini, der aber auch das Video-Element nicht unterstützt):

```
<video>
  <source src="video.mp4" type="video/mp4">
  Ihr Browser unterstützt keine HTML5-Videos. Sie können <a
href=" video.mp4">die Datei video.mp4hier herunterladen</a>.
</video>
```

Außerdem ist es eine gute Idee, die Höhe und Breite des Videos anzugeben, so dass Sie folgenden endgültigen HTML-Code erhalten:

```
<video width="400" height="600">
  <source src="video.mp4" type="video/mp4">
  Ihr Browser unterstützt keine HTML5-Videos. Sie können <a
href=" video.mp4">die Datei video.mp4hier herunterladen</a>.
</video>
```

Überlegen Sie sich gut, ob Sie Videos überhaupt selber bereitstellen wollen. Eine gute Alternative ist die Nutzung eines Video-Dienstes wie YouTube oder My Video. Dann muss nicht Ihr eigener Server die Videos ausliefern. Die Server der großen Anbieter sind in der Regel schneller und besser für solche Aufgaben geeignet. Sie stehen außerdem weltweit bereit.

Nicht zuletzt sind diese Video-Seiten als Marketing-Plattformen für Ihre Webseite geeignet. Das Video befindet sich dann nicht nur auf Ihrer eigenen Webseite, sondern auch bei dem eigentlichen Dienst und kann dort ebenfalls gefunden werden. Links zu Ihrer eigenen Seite und weitere Informationen über sich und das Video können Sie dort zusätzlich angeben und so auf sich und Ihre Website aufmerksam machen.

## 6.5 Audiodateien einbinden

Audiodateien binden Sie wie Videodateien ein. Nur das mp3-Format wird uneingeschränkt von allen Browsern unterstützt. Das AAC-Format wird von allen Browsern unterstützt; vom Firefox aber nur in einem mp4-Container und nur dann, wenn die entsprechenden Codecs systemweit verfügbar sind. Daher ist das mp3-Format die sicherste Lösung, wenn es darum geht, dass jeder Besucher die Inhalte hören soll.

```
<audio>
  <source src="audio.mp3" type="audio/mpeg" />
  Ihr Browser unterstützt kein HTML5-Audio
</audio>
```

## 6.6 Externe Inhalte nutzen

Viele Webseiten bieten Ihnen die Möglichkeit, die angebotenen Dienste in Ihre eigene Webseite zu integrieren. Dazu gehören Video-Dienste wie Vimeo oder Youtube, Karten-Dienste wie Google Maps oder Bing-Maps oder Multimedia-Anbieter wie Media Wiki.

Grundsätzlich sind solche Dienste mit Vorsicht zu nutzen. Da es sich um externe Anbieter handelt, machen Sie sich nicht nur von der Erreichbarkeit dieser Dienste abhängig. Sie müssen auch die Nutzungsbedingungen beachten. Allzu sorgloser Umgang kann hier zu Rechtsstreitigkeiten führen.

Wenn man sich an die Bedingungen hält, stellen die Dienstanbieter ihren Service in der Regel gerne zur Verfügung.

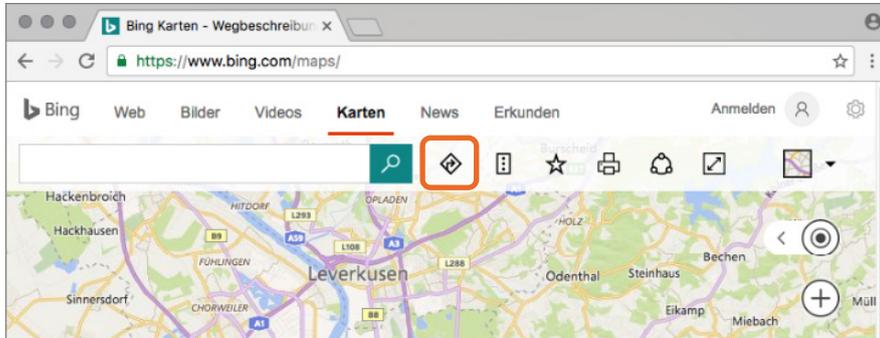
Die Nutzung solcher Dienste hat viele Vorteile. Der wichtigste: Sie können professionelle Dienste wie Routenplaner oder Kalender in Ihre Webseite integrieren.

Ein weiteres Vorteil ist, dass die Server der großen Diensteanbieter auf der ganzen Welt verteilt stehen und sehr gut an das Internet angebunden sind. So werden insbesondere große und zeitkritische Inhalte wie hochauflösende Videos auch in anderen Ländern der Welt meist ruckelfrei dargestellt.

Normalerweise müssen Sie auf den Anbieter des jeweiligen Diensts hinweisen und machen so für diesen Werbung. Im Gegenzug macht er es Ihnen leicht, Karten, Kalender, Videos, Umfragen oder Bilder den Besuchern Ihrer Seite anzubieten.

## Beispiel 1: Karten von Microsoft

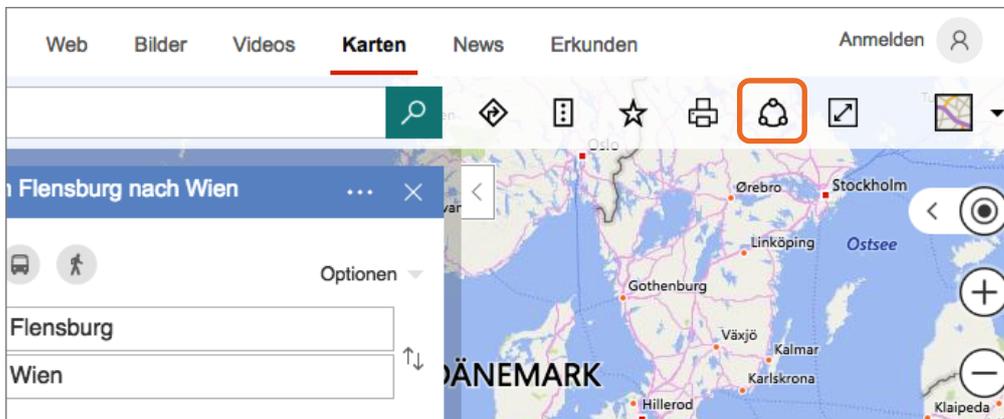
- ✓ Besuchen Sie die Seite <https://www.bing.com/maps/> und klicken Sie auf das Routen-Symbol (siehe oranger Rahmen).



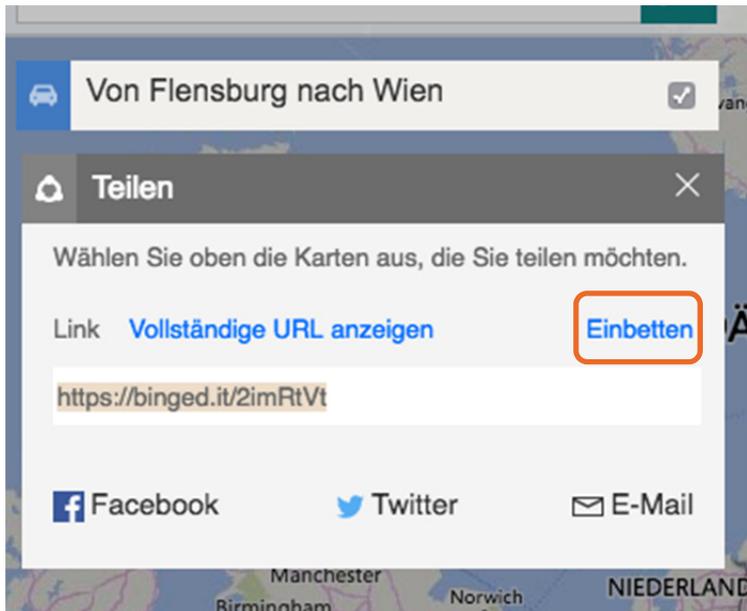
- ✓ Geben Sie einen Startort und ein Ziel ein, zum Beispiel Flensburg und Wien.



- ✓ Klicken Sie auf den „Teilen“-Button.



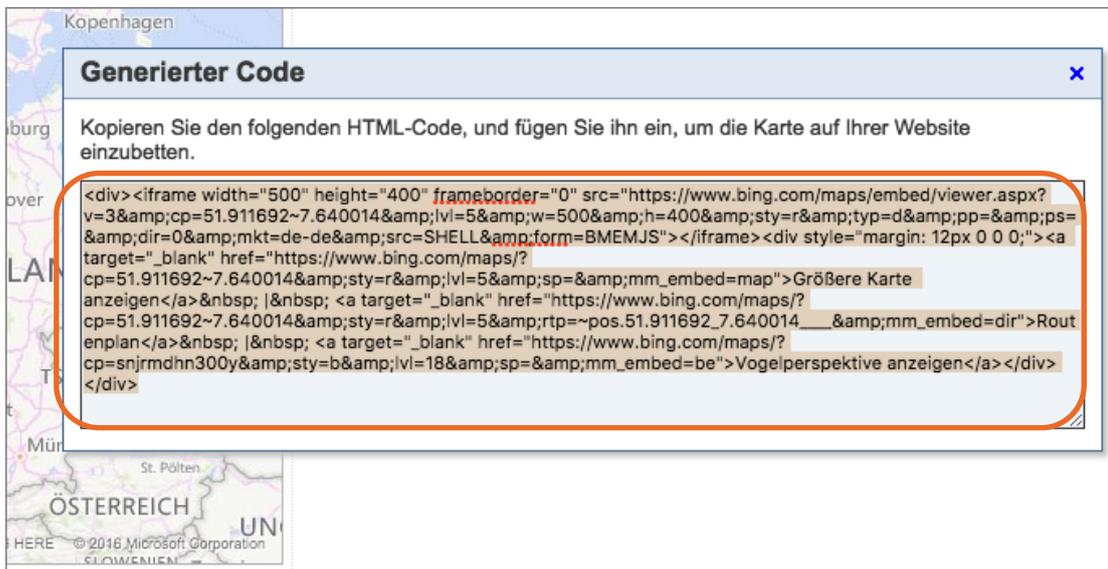
- ✓ Klicken Sie auf „einbetten“.



- ✓ In der folgenden Ansicht können Sie ein paar Optionen nach Ihren Bedürfnissen anpassen. Klicken Sie auf „Code generieren“.



- ✓ Im nächsten Schritt gibt Ihnen der Bing-Karten-Dienst HTML-Quellcode aus, den Sie markieren, kopieren und in Ihre Webseite einfügen können.



Dieser Code bindet in Ihre Seite ein `iframe`-Element ein. Das ist ein HTML-Element, das komplette Webseiten beinhalten kann. Die eingebundene Webseite wird im `src`-Attribut angegeben.

## Beispiel 2: Multimedia-Inhalte von Wiki Media Commons

Wiki Media Commons ist ein Schwesterprojekt der Wikipedia, in der alle multimedialen Inhalte hinterlegt sind. Wie die Wikipedia selber, sind auch die Inhalte der Wikimedia in der Regel fre nutzbar. Zu jedem Inhalt stehen genaue Angaben zu den Bedingungen bereit, die bei der Verwendung in der eigenen Webseite beachtet werden müssen.

Auch die Wikimedia Commons stellen Code bereit, den Sie nur in Ihre Webseite kopieren müssen. Die Seite stellt vor allem Bilder, aber auch Videos und Audiodateien bereit. Dazu müssen Sie folgende Schritte durchführen:

- ✓ Öffnen Sie die Seite <https://commons.wikimedia.org>.

Hauptseite
Diskussion
Lesen
Quelltext anzeigen
Versionsgeschichte

Q

## Wikimedia Commons

eine Sammlung von 35.588.836 (unter Bedingungen) frei verwendbaren Mediendateien, zu der jeder beitragen kann.

Bilder
Töne
Filme
Hochladen

**Bild des Tages**



Das Reichstagsgebäude in Berlin bei Nacht. Davor ein Weihnachtsbaum. Links neben dem Reichstag sind das internationale Handelszentrum (Friedrichstraße) und der Fernsehturm (Alexanderplatz) zu sehen. Über dem Fernsehturm leuchtet der Mond. Die Kuppel des Reichstagsgebäudes wurde von Norman Foster entworfen.

+/- (de), +/- (en)

[Tweet](#) · [RSS feed](#) · [Subscribe](#) · [Frühere Bilder des Tages](#)

**Fotowettbewerb**

Mache Fotos und lade sie hoch, um an unserem **monatlichen Wettbewerb** teilzunehmen und wertvolle Anregungen zu erhalten!

Lerne die Wettbewerbe dieses Monats kennen

**Höhepunkte**

Beim ersten Stöbern durch Commons sollte man bei den **Exzellenten Bildern**, den **Qualitätsbildern** oder den **Wertvollen Bildern** beginnen. Unter **Unsere Fotografen stellen sich vor** und **Unsere Illustratoren stellen sich vor** findet man einige Beiträge unserer besten Freiwilligen.



Bild des Jahres 2015

**Inhalte**

**Nach Themen**

**Natur**  
Tiere · Fossilien · Landschaft · Meeresleben · Pflanzen · Wetter

**Gesellschaft · Kultur**  
Kunst · Architektur · Glaube · Wappen · Unterhaltung · Ereignisse · Flaggen · Essen · Geschichte · Sprache · Literatur · Musik · Objekte · Menschen · Orte · Politik · Sport

**Naturwissenschaft**  
Astronomie · Biologie · Chemie · Mathematik · Medizin · Physik · Technologie

**Ingenieurwissenschaften**  
Chemieingenieurwesen · Hoch- und Tiefbau · Elektrotechnik · Umwelttechnik · Ingenieurgeologie

- ▶ Media types
- ▶ Categories
- ▶ Commons
- ▶ Copyright statuses
- ▶ Featured items
- ▶ Gallery pages
- ▶ Media by source
- ▶ Topics

**Multimediateil des Tages**



- ✓ Die Startseite bietet Ihnen zahlreiche Einstiegspunkte, um zu interessanten Inhalten zu gelangen. Besonders gelungene Bilder und Videos finden Sie unter „Bild des Tages“ oder „Multimediateil des Tages“. Oben rechts können Sie Inhalte nach Format auswählen (Bilder, Töne, Filme). Darunter finden Sie Aktionen (Fotowettbewerb), Höhepunkte (besonders gelungene Inhalte) und thematische Einstiegspunkte (Inhalte).
- ✓ Um bestimmte Inhalte zu erreichen, kann die Suche hilfreich sein. Geben Sie in das Suchfeld oben rechts (orangener Rahmen im Bild oben) „Big Buck Bunny“ ein, um den gleichnamigen Open Source Film zu finden.
- ✓ Klicken Sie auf „Multimedia“ (orangener Rahmen), um zu den Filmen zu gelangen.

The screenshot shows the Wikimedia Commons page for 'Big Buck Bunny'. The page title is 'Big Buck Bunny' and it is in German. The page content includes a description in German and English, a table of contents, and an 'Official image gallery' section. The 'Multimedia' section is highlighted with a red box, and an orange arrow points from it to the 'Multimedia' section in the second screenshot.

Wikimedia Commons

English Not logged in Talk Contributions Create account Log in

Gallery Discussion View Edit History Search Wikimedia Commons

## Big Buck Bunny

Wikimedia Commons auf Deutsch

From Wikimedia Commons, the free media repository

**Deutsch:** Big Buck Bunny ist ein computer-generierter Kurzfilm, der unter einer CC-BY-Lizenz veröffentlicht wurde. Du kannst Screenshots und Clips mit dem Lizenzschild {{Big Buck Bunny}} hochladen.

**English:** Big Buck Bunny is a computer generated movie that has been released under a CC-BY license. You can upload screenshots and clips with {{Big Buck Bunny}}.

**Contents** [hide]

- Official image gallery
  - Movie snapshots
    - Production photos
      - Working
  - DVD Cover
  - Multimedia**

**Official image gallery** [edit]

**Movie snapshots** [edit]

The screenshot shows the 'Multimedia' section of the Big Buck Bunny page. It features a video player with a play button and a caption 'full movie, low bitrate'. Below the video player is a large image of the white rabbit character from the movie.

## Multimedia

[Bearbeiten]

full movie, low bitrate

- ✓ Klicken Sie auf ein Bild oder den blauen Link darunter, kommen Sie auf die Seite mit diesem Multimedia-Inhalt. Auf der Seite für den Inhalt erfahren Sie alles Wichtige zu Lizenz- und Nutzungsbedingungen. Auch den Knopf zum Einbinden des Films in die eigene Webseite finden Sie hier.

**File:Big Buck Bunny medium.ogv**

Aus Wikimedia Commons, dem freien Medienarchiv

Herunterladen Einbinden Einbinden Link per Mail Information



Größe der JPG-Vorschau dieser OGG-Datei: 800 x 450 Pixel. Weitere Auflösungen: 320 x 180 Pixel | 640 x 360 Pixel | 896 x 504 Pixel.  
 Originaldatei (Ogg-Audio-/Video-Datei, Theora/Vorbis, Länge: 9 min 56 s, 896x504 Pixel, 1,32 Mbps insgesamt)

<b>Beschreibung</b>	<b>English:</b> Big Buck Bunny short film by the Blender Foundation. Downsampled to bitrate of 1317 kbps from original 1280x720 OGG movie using these ffmpeg2theora-0.25 command options: --width 896 --videobitrate 1240 --audiobitrate 96
<b>Datum</b>	März 2008
<b>Quelle</b>	<a href="http://mirror.bigbuckbunny.de/peach/bigbuckbunny_movies/big_buck_bunny_720p_stereo.ogg">http://mirror.bigbuckbunny.de/peach/bigbuckbunny_movies/big_buck_bunny_720p_stereo.ogg</a> linked from <a href="http://www.bigbuckbunny.org/index.php/download/">http://www.bigbuckbunny.org/index.php/download/</a>
<b>Urheber</b>	(c) copyright Blender Foundation/www.bigbuckbunny.org
<b>Genehmigung</b> (Weiternutzung dieser Datei)	<p>This file is from the movie <i>Big Buck Bunny</i> or the associated website, which has been licensed under <a href="#">Creative Commons Attribution 3.0</a> License. Their website states:</p> <p>The attribution is either one of the following cases</p> <ol style="list-style-type: none"> <li>If you redistribute or screen or broadcast the movie itself: include the entire credits scroll.</li> <li>In all other cases, attribute it as: (c) copyright Blender Foundation   www.bigbuckbunny.org</li> </ol> <p>This is the first time such a large collection of files get launched under 'the commons', and therefore still a bit complex to grasp what to credit when exactly. The guidelines above are just meant to ensure that attribution happens in a reasonable way. If in doubt, use the second choice above.</p> <p>Excluded from the Creative Commons is:</p> <ul style="list-style-type: none"> <li>all logos on this website (including Blender logo, Peach logo, Creative Commons logo) and associated trademarks</li> </ul> <p>Diese Datei ist unter der <a href="#">Creative-Commons-Lizenz „Namensnennung 3.0 nicht portiert“</a> lizenziert.</p> <p>Dieses Werk darf von dir</p> <ul style="list-style-type: none"> <li><b>verbreitet werden</b> – vervielfältigt, verbreitet und öffentlich zugänglich gemacht werden</li> <li><b>neu zusammengestellt werden</b> – abgewandelt und bearbeitet werden</li> </ul> <p>Zu den folgenden Bedingungen:</p> <ul style="list-style-type: none"> <li><b>Namensnennung</b> – Du musst den Namen des Autors oder Rechteinhabers in der von ihm festgelegten Weise nennen (aber nicht so, dass es so aussieht, als würde er dich oder deine Verwendung des Werks unterstützen).</li> </ul>

Nun öffnet sich ein Kasten mit Code, den Sie in Ihre Webseite kopieren können.



Ein weiteres Video-Beispiel finden Sie in den BuchPlus-Inhalten:

**Plus** Lernvideo: *youtubevideo-verwenden.mp4*

## 6.7 Übungen

### Übung 1: Mit Grafiken arbeiten

Level		Zeit	ca. 20 min
Übungsinhalte	<ul style="list-style-type: none"> <li>✓ Grafiken einfügen und mit Alternativtext versehen</li> <li>✓ Die Grafikhöhe und -breite festlegen</li> </ul>		
Übungsdatei	<i>hotel06</i>		
Ergebnisdatei	<i>hotel06-e</i>		

1. Öffnen Sie im Ordner *info* die Datei *ausfluege.html* und fügen Sie aus dem Ordner *bilder* die Grafiken *ballon.jpg*, *dom.jpg*, *boote.jpg* sowie *tour.jpg* ein.
2. Vergeben Sie dabei aussagekräftige Alternativtexte.
3. Überprüfen Sie die Webseite im Browser.
4. Schalten Sie die Anzeige von Grafiken in Ihrem Browser aus und laden Sie die Webseite erneut, um die Alternativtexte zu betrachten. Schalten Sie anschließend die Grafikanzeige wieder ein.

5. Fügen Sie Miniaturbilder aller Grafiken oben im Dokument ein, indem Sie eine einheitliche Höhe von 75 Pixeln und eine Breite von 100 Pixeln für jedes Bild festlegen.
6. Zeigen Sie die Webseite im Browser an.
7. Öffnen Sie nacheinander alle anderen HTML-Dokumente des Webauftritts und versehen Sie die Bilder mit `alt`-Attributen.

### Ausflüge

Folgende Ausflüge können Sie an der Rezeption buchen.







← Miniaturbilder

1. Ballonglühen:



Dieses Bild erscheint verzerrt, weil es unproportional verkleinert wurde. Alle anderen Bilddateien sind 150 Pixel hoch und 200 Pixel breit und wurden deshalb proportional (um die Hälfte) verkleinert.

Beim Ballonglühen werden mehrere Heißluftballons am Boden verankert und nach Einbruch so besonders gut zur Geltung. Ein atemberaubendes Schauspiel für Jung und Alt.

2. Historische Sehenswürdigkeiten der Umgebung



Bilder in normaler Größe

Ergebnisdatei „[hotel06/info/ausfluege.html](http://hotel06/info/ausfluege.html)“

## Übung 2: Karten einbinden

<b>Level</b>		<b>Zeit</b>	ca. 10 min
<b>Übungsinhalte</b>	✓ Karte einbinden		
<b>Übungsdatei</b>	<i>hotel06/wegweiser.html</i>		
<b>Ergebnisdatei</b>	<i>hotel06-e/wegweiser.html</i>		

1. Binden Sie eine Karte in die Seite *wegweiser.html* ein, indem Sie die Schritte zum Einbinden von Karten nachvollziehen.
2. Geben Sie keine Route an, sondern nur eine beliebige Adresse für das Hotel (z. B. in Cartagena, Spanien). Besucher können dann einen eigenen Startort festlegen, um sich den Weg zum Hotel anzeigen zu lassen.

# 7

## Hyperlinks erstellen

### 7.1 Wie werden Hyperlinks eingesetzt?

**Links** (auch **Verweise** oder **Hyperlinks**) dienen dazu, Webseiten miteinander zu verbinden. Folgende Arten von Links gibt es:

- ✓ Hyperlinks zu anderen Seiten innerhalb eines Webauftritts,
- ✓ Hyperlinks zu Seiten im Internet,
- ✓ Hyperlinks auf bestimmte Ziele innerhalb einer Webseite oder in einer anderen Seite,
- ✓ Hyperlinks auf Email-Adressen

### 7.2 Hyperlinks zu Seiten innerhalb eines Webauftritts erstellen

#### Nutzer führen

Mit Hyperlinks geben Sie Besuchern Ihrer Webseite eine Möglichkeit, sich Ihren Webauftritt zu erschließen. Da durch die Art, wie Sie Links anordnen und präsentieren, Nutzer verleitet werden, bestimmten Links häufiger zu folgen (oder auch nicht), haben Sie einen großen Einfluss darauf, wie sich Nutzer durch Ihre Seiten bewegen. Links sind neben Texten und Design das wichtigste Mittel zur **Nutzerführung**.

Von zentraler Bedeutung zur Erschließung der gesamten Inhalte ist die **Hauptnavigation**. Hierüber sollten Sie alle weiteren Seiten erreichbar machen.

Die Hauptnavigation sollte die Ordnerstruktur widerspiegeln, die Sie, wie in Kapitel 5 beschrieben, angelegt haben.

## Relative Links erstellen

Die meisten Links, die Sie anlegen, verweisen auf Seiten innerhalb Ihres eigenen Webauftritts. Für diese Hyperlinks können Sie relative Pfade verwenden.

```
<a href="Hyperlink">Hyperlinktext</a>
```

### Beispiel: Hyperlinks erstellen

Im Beispiel haben Sie im Stammordner *hotel* zwei Seiten erstellt: die Startseite des Hotels Vallora und eine Seite mit Wegbeschreibung.

Zu jedem weiteren Thema (Zimmerausstattung, Restaurant, Dienstleistungen, Freizeitangebot usw.) sollten Sie jeweils eine eigene Seite anlegen. Von der Startseite (*index.html*) verweisen Sie dann auf die jeweilige Unterseite und von den Unterseiten wieder zurück zur Startseite.

**Hotel Vallora**

[Wegbeschreibung](#) [Über uns](#)



Bei Ihrem nächsten Aufenthalt in unserer Gegend würden wir Sie gern persönlich begrüßen und mit unserem freundlichen und kompetenten Service verwöhnen.

**Zimmerreservierungen:**

Unsere Rezeption ist 24 Stunden besetzt. Sie können auch gern unser Online-Buchungsformular ausfüllen.

Tel. 000 000 000  
Fax 000 000 000  
E-Mail: [info@hotelvallora.com](mailto:info@hotelvallora.com)

[Nehmen Sie mit uns Kontakt auf.](#)

**Wegbeschreibung**

Fahren Sie in Richtung Küste. Kurz vor dem Ortseingang von Küstendorf biegen Sie links in die Küstenallee ein. Nach der Kurve finden Sie unser Hotel. Parkmöglichkeiten befinden sich direkt vor dem Hotel.



**Über uns**

Wir wünschen Ihnen einen angenehmen und erholsamen Aufenthalt im Familienhotel Vallora. Unser erfahrenes und kinderfreundliches Betreuungsteam steht Montag bis Samstag ab 9 Uhr für Sie bereit. Wir bieten jeden Tag ein abwechslungsreiches Programm für Kinder zwischen 8 und 14 Jahren an. Sie finden eine Beschreibung der Aktivitäten an der Rezeption sowie in unserem Betreuungsbereich. Bei Ausflügen müssen die Kinder am Nachmittag vorher angemeldet werden.

Bei gutem Wetter finden unsere Abenteuer- und Spielprogramme im Freien statt. Die Kinder sollten ihre Badebekleidung, Handtücher und Badesandalen mitbringen. An regnerischen Tagen bieten wir Bewegungsspiele und Bastelprojekte in unserer Freizeithalle an. In der Freizeithalle befinden sich eine Kletterwand, drei Trampoline und zahlreiche Spielgeräte (wie zum Beispiel Einräder und Stelzen), die unter Aufsicht benutzt werden dürfen. Tischtennisschläger und -bälle sowie Bälle für Kicker können in der Freizeithalle gegen Pfand ausgeliehen werden.

Startseite mit Links zu weiteren Unterseiten

Erstellen Sie den Hyperlink zu der Webseite *ueberuns.html* im Ordner *hotel/info* in der Webseite *index.html* wie folgt:

- ▶ Öffnen Sie die Seite, in der Sie den Hyperlink einfügen möchten (z. B. *index.html*).
- ▶ Fügen Sie den Hyperlink im `body`-Bereich an einer sinnvollen Stelle ein, im Beispiel unter dem vorhandenen Hyperlink zur *Wegbeschreibung* ①.

```
<body>
<h1>Hotel Vallora!</h1>
<p><a href="info/wegweiser.html">Wegbeschreibung</a>
  <a href="info/ueberuns.html">Über uns</a></p>
```

Code für den Hyperlink eingeben

Damit der Besucher Ihres Webauftritts wieder auf die Startseite Ihres Webauftritts findet, erstellen Sie auf der Seite *ueberuns.html* einen Hyperlink auf die Startseite.

[Zurück zur Startseite](#)

### Tipps für die Bezeichnung von Hyperlinks

Der Besucher Ihres Webauftritts soll bereits beim Betrachten der Hyperlinks wissen, was ihn auf den damit verbundenen Seiten erwartet, z. B.

Nicht aussagekräftiger Hyperlinktext	Aussagekräftiger Hyperlinktext
<u>zurück</u>	<a href="#">Zurück zur Startseite</a>
Weiter geht's <u>hier</u> .	<a href="#">Nähere Informationen zu den Öffnungszeiten</a>

Der Text in Hyperlinks sollte aus mehreren Gründen aussagekräftig sein:

- ▶ Von Sehenden wird der Text schnell gefunden. Längere Texte lassen sich so nach Links „scannen“, um schnell zu erfahren, wo es weiterführende Informationen gibt. Linktexte wie „Hier Klicken“ oder „Weiter“ wären dabei keine Hilfe.
- ▶ Blinde Menschen können von Link zu Link „springen“, um sich wie Sehende über weiterführende Informationen zu informieren.
- ▶ Suchmaschinen und andere Software kann die verlinkten Texte besonders bewerten.

### ! Umgang mit Fehlermeldungen

Wenn Sie versuchen, eine Seite aufzurufen, die nicht existiert, erscheint im Browser die Meldung, dass die gewünschte Seite nicht geladen werden kann. Gehen Sie wie folgt vor:

- ✓ Prüfen Sie, ob sich beide HTML-Dokumente im selben Verzeichnis befinden. Kopieren Sie gegebenenfalls die Dateien in dasselbe Verzeichnis.
- ✓ Vergewissern Sie sich, dass Sie keinen Tippfehler im Pfad haben, beachten Sie auch die Groß- und Kleinschreibung des Dateinamens! Ändern Sie ggf. die Schreibweise des Links.

### Hyperlinks auf eine E-Mail-Adresse erstellen

Möchten Sie, dass Ihnen die Besucher Ihrer Webseite über einen Link eine E-Mail schicken können, benutzen Sie das Schlüsselwort `mailto:` vor dem Linkziel (in diesem Fall ist das Linkziel Ihre Email-Adresse):

```
<a href="mailto:mail@example.com">...</a>
```

- ✓ Bei einem Klick auf den Hyperlink wird das voreingestellte E-Mail-Programm gestartet und das Feld *Empfänger* mit der angegebenen E-Mail-Adresse ausgefüllt.

### Beispiel: E-Mail-Link für Hotel Vallora

Wenn der Besucher Ihrer Webseite auf den Link *Nehmen Sie mit uns Kontakt auf* klickt, wird das Standard-E-Mail-Programm geöffnet und als Empfänger der E-Mail die Adresse *info@hotelvallora.com* eingetragen.

```
<a href="mailto:info@hotelvallora.com">Nehmen Sie mit uns Kontakt auf.</a>
```

### Weitere Angaben für die E-Mail erstellen

Außer dem Aufruf des E-Mail-Programms mit der entsprechenden Empfängeradresse können noch zusätzliche Angaben vordefiniert werden. Dazu wird nach der Angabe der E-Mail-Adresse ein Fragezeichen (?) eingefügt.

```
<a href="mailto:Adresse?subject=Betreff der E-Mail">Text</a>
```

Der Parameter `subject` füllt das Feld *Betreff* im E-Mail-Programm mit dem Text nach dem Gleichheitszeichen aus. Der Text kann auch Leerzeichen enthalten.

```
<a href="mailto:Adresse?body=vorgegebener Text">Text</a>
```

Der Parameter `body` fügt einen vorgegebenen Text in das Feld für den E-Mail-Text ein. Der Text kann auch Leerzeichen enthalten.

```
<a href="mailto:Adresse?cc=Adresse">Text</a>
```

Mit der Angabe `cc` können Sie eine Kopie der E-Mail an eine andere E-Mail-Adresse verschicken. Alle E-Mail-Adressen der Kopie-Empfänger sind bei jedem Empfänger der Nachricht sichtbar.

```
<a href="mailto:Adresse?bcc=Adresse">Text</a>
```

Durch die Angabe von `bcc` verschicken Sie ebenfalls Kopien der E-Mail. Im Gegensatz zu `cc` sind die E-Mail-Adressen der anderen Kopie-Empfänger beim Empfänger der Nachricht nicht sichtbar.

Da E-Mail-Adressen personenbezogene Daten sind, dürfen Sie diese nicht ohne Einwilligung der Besitzer weitergeben. Daher sollten Sie zum Versand an mehrere Empfänger immer die Methode mit `bcc` verwenden, wenn Sie nicht das Einverständnis aller Mail-Empfänger zur Weitergabe der Adressdaten schriftlich vorliegen haben.

Sie können auch die einzelnen Vorgaben miteinander kombinieren. Diese werden mit dem Zeichen (&) voneinander getrennt angegeben.

```
<a href="mailto:email@example.com?subject=Kontaktaufnahme&body=Ich bitte um folgende Informationen&cc=kopie@email.de">Nehmen Sie mit mir Kontakt auf.</a>
```

## Hyperlink auf andere Dokumenttypen erstellen

Mit einem Hyperlink können Sie auf jeden beliebigen Dokumenttyp verweisen, z. B. auf Textdateien, Grafiken, Videos, Musikdateien oder Programme.

Was beim Nutzer nach Aktivierung des Links passiert, hängt vom verwendeten Browser, Betriebssystem und den installierten Programmen ab. Viele Dateien können moderne Browser eigenständig wiedergeben. Dazu gehören Audio- und Video-Dateien. Wenn der Browser etwas nicht selbstständig wiedergeben kann, bietet er dem Nutzer in der Regel an, ein PlugIn herunterzuladen, welches die benötigte Fähigkeit nachrüstet.

Es kann aber auch sein, dass eine Betriebssystem-Funktion oder ein weiteres Programm (z. B. Textverarbeitung oder Video-Player) zur Darstellung oder Wiedergabe genutzt werden.

Immer hat der Besucher der Webseite die Möglichkeit, sich die angebotene Datei abzuspeichern. Das betrifft auch Dateien, für die kein geeignetes Programm zum Öffnen der verlinkten Datei vorhanden ist.

Dokumenttyp	Häufig ausgeführte Aktion
Textdatei (*.txt) <a href="text.txt">	Eine Textdatei wird im Browser angezeigt.
Grafik (*.jpg, *.gif, *.png) <a href="grafik.jpg">	Die Grafik wird im Browser angezeigt.
Video (*.mpg, *.avi) <a href="video.avi">	Haben Sie ein entsprechendes Programm zum Abspielen von Videodaten installiert, wird dieses geöffnet und das geladene Video abgespielt.
Musik (*.mp3, *.wav) <a href="musik.mp3">	Diese Dateien werden wie Videodaten behandelt: Existiert ein entsprechendes Programm, werden die Musikdaten abgespielt.
Programm, Datendatei (*.zip, *.gz, *.exe) <a href="programm.zip">	Im Internet wird eine Vielzahl von Programmen angeboten, die Sie auf Ihren Rechner überspielen können (Download). Häufig werden Dateien auch gepackt (zip, gz) abgelegt, um die Menge der zu übertragenden Daten zu verringern.
Andere Formate (*.pdf, *.doc bzw. *.docx, *.xls bzw. *.xlsx) <a href="dokument.pdf">	Befindet sich ein Programm auf Ihrem Rechner, das diese Dateiformate anzeigen kann, wird die Datei automatisch in das jeweilige Programm geladen bzw. direkt im Browserfenster geöffnet.

Die Programme zum Anzeigen der hier dargestellten Beispiele können von Computer zu Computer unterschiedlich sein. Abhängig ist dies von der jeweils installierten Software.

## 7.3 Hyperlinks zu Seiten und Dokumenten im Web erstellen

### Absolute Links erstellen

Hyperlinks zu anderen Webseiten im Internet werden auch als externe Links bezeichnet. Bei diesen Hyperlinks arbeiten Sie mit absoluten Hyperlinks. Sie müssen die vollständige Internetadresse (URL) angeben.

```
<a href="http://www.wetter.com">Wetterinfo</a>
```

Sie können, wenn Sie in ein bestimmtes Verzeichnis des anderen Webauftritts oder auf eine bestimmte Webseite verweisen möchten, dafür den absoluten Pfad angeben:

```
<a href="http://www.wetter.com/wetter_aktuell/">Wettervorhersage</a>
<a href="http://www.geo.de/GEO/reisen/europa/">Geo Reisen
Europa</a>
```

### URI, URL und URN

URI	Universal Resource Identifier	<ul style="list-style-type: none"> <li>✓ Zeichenkette, über die eine Ressource im Netzwerk gefunden oder identifiziert werden kann (entweder eine URL oder eine URN),</li> <li>✓ könnte zukünftig URL und URN ersetzen,</li> <li>✓ URIs werden in URL und URN unterteilt</li> </ul>
URL	Uniform Resource Locator	<ul style="list-style-type: none"> <li>✓ Gibt einen Ort an, an dem sich eine Ressource befindet. Da Dateien kopiert werden, kann sich eine Datei an mehreren Orten befinden, es können also auch mehrere URLs z. B. auf ein Buch verweisen, das auf mehreren Webseiten angeboten wird.</li> </ul>
URN	Uniform Resource Name	<ul style="list-style-type: none"> <li>✓ Damit eine Ressource eindeutig identifiziert werden kann, benötigt sie einen eindeutigen Namen. Dieser wird als URN bezeichnet. Darüber kann man eine Ressource jedoch nicht erreichen; das geht nur über einen Ort, der wie oben beschrieben mittels URL mitgeteilt wird.</li> </ul>

Eine URL, wie sie beispielsweise in Hyperlinks genutzt wird, hat den folgenden Aufbau:

```
http://www.example.com:portnummer/ordner/webseite.html?ParameterListe
```

Bestandteil	Beschreibung
<code>http://</code>	Kennzeichnet das verwendete Protokoll, hier Hypertext-Transfer-Protokoll (http). Es wird durch die Zeichenfolge <code>://</code> vom nächsten Teil der URL getrennt.
<code>www.domain-name.de</code>	Es folgt der Domainname mit der Domainendung. Alternativ kann auch die IP-Adresse des Servers, z. B. 217.80.10.2, angegeben werden, wenn diesem kein Domainname zugeordnet ist.
<code>:portnummer</code>	Dem Domainnamen kann optional durch einen Doppelpunkt getrennt eine Portnummer angefügt werden. Standardmäßig wird der Port 80 für das HTTP-Protokoll verwendet und muss in diesem Fall nicht mit angegeben werden.
<code>ordner</code>	Durch das Zeichen <code>/</code> (Slash) getrennt wird ein Verzeichnispfad auf dem Server angefügt.
<code>webseite.html</code>	Es wird eine konkrete Datei angegeben. Es ist allerdings nicht immer notwendig, eine konkrete Webseite in einer URL zu benennen. Geben Sie beispielsweise nur <code>http://www.XYZ.de/</code> ein, verwendet der Webserver automatisch eine vordefinierte Startseite wie z. B. <code>http://www.XYZ.de/index.html</code> .
ParameterListe	Als letzter Bestandteil kann optional nach einem Fragezeichen eine Liste von Parameterwerten angehängt werden, die von der angegebenen Datei ausgewertet werden.

In der Regel verwenden URLs nicht alle diese Bestandteile, sodass der Aufbau wesentlich einfacher ist. Geben Sie die Namen der URLs wie `http://www.domainname.de/` oder `http://www.domainname.de/ordner/` mit abschließendem Slash `/` ein. Dies beschleunigt den Aufruf der Datei, da der Server, auf dem das Dokument liegt, sofort erkennt, dass ein Verzeichnis aufgerufen wird:

`http://www.herd.com/`  
`http://www.w3.org/`

Bei der Angabe von Hyperlinks müssen Sie darauf achten, dass die Namen der aufzurufenden Dateien genau übereinstimmen. Das bedeutet, dass Sie auch auf die Groß- und Kleinschreibung der Dateinamen achten müssen. Fast alle Webserver laufen auf UNIX/Linux-Betriebssystemen, die zwischen Groß- und Kleinschreibung unterscheiden. Rufen Sie Webseiten, auf die Sie verlinken möchten, am besten selber im Browser auf und kopieren Sie dann die Adresse aus dem Adressfeld des Browsers in Ihren Editor.

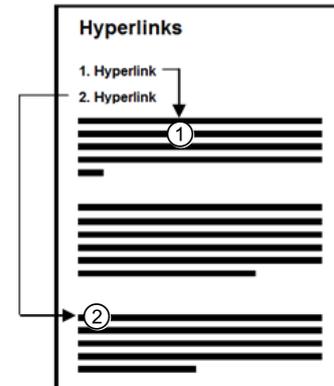
## 7.4 Hyperlinks innerhalb einer Seite

### Innerhalb eines HTML-Dokumentes zu einer bestimmten Stelle springen

Auch wenn Sie Ihren Webauftritt durch Aufteilung der Informationen auf mehrere Seiten strukturieren, kann es vorkommen, dass einzelne Seiten über so viel Inhalt verfügen, dass viel gescrollt werden muss. Dieses Problem verschärft sich, wenn für die Betrachtung einer Seite ein Smartphone oder anderes Gerät mit kleinem Display verwendet wird.

Mit seiteninternen Hyperlinks geben Sie dem User die Möglichkeit, schnell auf bestimmte Stellen innerhalb einer Seite zuzugreifen.

In der Abbildung verweist der erste Hyperlink auf die Position ① in der Webseite, der zweite auf Position ②.



Hyperlink auf eine andere Position in der Webseite

Folgende Arbeitsschritte sind zum Erstellen eines Hyperlinks notwendig:

- ✓ ein Ziel (Anker) an der Position definieren, auf die Sie verlinken möchten,
- ✓ einen Hyperlink zu dem definierten Ziel setzen.

### Hyperlinkziel definieren (Anker)

Um einen Anker für mögliche Hyperlinkziele zu definieren, geben Sie dem Element, das Ziel eines Links werden soll, eine eindeutige Bezeichnung mittels `id`-Attribut. Den Namen des Linkziels (Wert des `id`-Attributes) können Sie frei wählen. Achten Sie darauf, innerhalb eines HTML-Dokumentes **nicht mehrmals denselben Wert für IDs zu vergeben**, sonst kann der Browser nicht wissen, zu welcher dieser IDs er springen soll. Daher sind mehrere identische Werte für IDs laut HTML5-Standard nicht zulässig.

<code>&lt;... id="ziel" ...</code>	Diese Angabe eines Ankers kann eine Identifikation innerhalb eines beliebigen Tags setzen.
------------------------------------	--



Das Attribut `id` muss mindestens ein Zeichen, darf aber keine Leerzeichen enthalten. Die Namen der Anker müssen im HTML-Dokument eindeutig sein, d. h., sie dürfen nur einmal je HTML-Dokument verwendet werden.

```
<h1 id="ziel">Zielelement</h1>
```

Sinnvollerweise sollten Sie Links an die Anfänge wichtiger Abschnitte auf Ihrer Seite setzen (beispielsweise Beginn von Kapiteln und Unterkapiteln, Navigationsbereich, Weiterführende Informationen).

Hyperlinkziele können z. B. sein:

- ✓ Bereiche (Artikel `article`, Hauptnavigation `nav`, weiterführende Informationen `aside` oder Sektionen `section`)
- ✓ eine Überschrift
- ✓ Kopfbereiche `header` oder Fußbereiche `footer`
- ✓ Abbildungen und Illustrationen `figure`

### Beispiel: Anker setzen

Im Folgenden werden Sie im Ordner `hotel/info` in der Datei `ueberuns.html` die Hyperlinkziele `ueber` und `ablauf` definieren, die im nächsten Schritt verlinkt werden.

- ▶ Öffnen Sie ein HTML-Dokument, in dem Sie interne Hyperlinks setzen möchten, z. B. die neu erstellte Seite `ueberuns.html`.
- ▶ Fügen Sie für die Überschriften `Über uns` ① bzw. `Tagesablauf` ② einen Anker ein.

```
<h1 id="ueber">Über uns</h1> ①
<p>Wir wünschen Ihnen einen angenehmen
kinderfreundliches Betreuungsteam
steht
abwechslungsreiches Programm für
Kinder
```

```
<p>Eltern können sich in unserem
Wellness
unseren Fitness-Club. Sie können an
geplan
<h2 id="ablauf">Tagesablauf</h2> ②
```

### Hyperlink setzen

Um auf einen Anker zu verweisen, verwenden Sie, wie bei anderen Hyperlinks auch, das `a`-Element mit dem Attribut `href`. Der Anker besteht aus der Definition des Ziels und einem beschreibenden Text.

```
<a href="#ziel">Ziel</a>
```

Das Attribut `href` enthält das Verweisziel. Durch die Angabe des Zeichens `#` (Raute, Lattenzaun, Gatter) vor dem Namen des Ankers wird dem Browser mitgeteilt, dass es sich bei dem Ziel um ein Element mit einer ID handelt.

Der durch die Tags `<a>` und `</a>` eingeschlossene Text stellt den auf der Webseite sichtbaren Hyperlink dar. Wenn Sie auf diesen Text klicken, wird an die entsprechende Position innerhalb der Webseite gesprungen. Das Ziel, auf das Sie verweisen, muss über das Attribut `id` definiert sein.

### Beispiel: Hyperlink auf ein Element mit einer ID setzen

Im folgenden Beispiel wird auf die zuvor in der Datei `ueberuns.html` gesetzten Anker `ueber` und `ablauf` verwiesen.

- ▶ Gehen Sie an die Position, von der aus Sie auf eine andere Stelle innerhalb der Seite verweisen wollen.

- ▶ Setzen Sie den Hyperlink wie folgt:

```
<a href="#ueber">Über uns</a>
<a href="#ablauf">Tagesablauf</a>
```



Über uns Tagesablauf

**Über uns**

Wir wünschen Ihnen einen angenehmen und erholsamen Aufenthalt im Familienhotel Vallora. Unser und kinderfreundliches Betreuungsteam steht Montag bis Samstag ab 9 Uhr für Sie bereit. Wir bieten

Eltern können sich in unserem Wellness-Bereich verwöhnen lassen. Besuchen Sie unsere Sauna-Landschaft oder unseren Fitness-Club. Sie können an geplanten Ausflügen teilnehmen oder Ihr eigenes Tagesprogramm gestalten.

**Tagesablauf**

09:00 bis 13:00 Uhr Kinderprogramm  
 13:00 bis 14:30 Uhr Mittagessen  
 14:30 bis 17:30 Uhr Kinderprogramm  
 18:00 bis 19:30 Uhr Abendessen  
 19:30 bis 21:30 Uhr Familienprogramm

Einen definierten Anker über einen Hyperlink erreichen

Wird auf einen Hyperlink geklickt, sucht der Browser die gewünschte Position auf der Webseite. Findet er sie, springt er automatisch an die betreffende Stelle und zeigt das Dokument ab der gesetzten Markierung an. Gestattet es die Länge der Webseite, wird die Stelle am oberen Rand des Browsers angezeigt. Ist die Webseite zu kurz, blendet der Browser das Ende der Webseite ein.

Der (sichtbare) Sprung zu einem Anker funktioniert nur dann, wenn die Webseite auch tatsächlich mehr Inhalt als eine Bildschirmseite hat. Zum Testen können Sie das Browserfenster verkleinern.

Der Textcursor wird allerdings in jedem Fall an das Ziel gesetzt. Das können Sie überprüfen, wenn Sie bei gedrückter Umschalt-Taste  die Pfeiltaste zum Markieren von Text benutzen. Der Text neben dem Sprungziel sollte nun farblich hinterlegt werden.

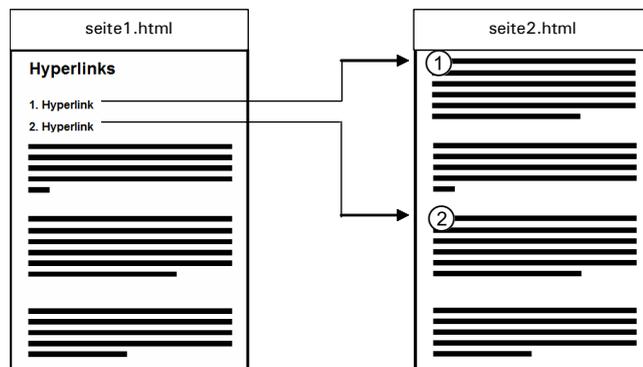
## Hyperlink auf eine bestimmte Position in einer anderen Webseite

Sie können auch auf ein Element mit einer ID in einer anderen Seite verweisen:

```
<a href="datei.html#ziel"
>Hyperlinktext</a>
```

Lade die Webseite *datei.html*, suche darin die Position *ziel* und springe dorthin.

```
<a
href="../datei.html#ziel"
>Hyperlinktext</a>
```



Verweis auf eine Position in einer anderen Webseite

Lade die Webseite *datei.html* aus dem übergeordneten Ordner, suche darin die Position (*id*) *ziel* und springe dorthin.

```
<a
href="http://www.example.com/datei.html#ziel"
>Hyperlinktext</a>
```

Lade die Webseite *datei.html* von der Internetadresse *http://www.domainname.de/*, suche darin die Position *ziel* und springe dorthin. (Der entsprechende Anker muss auf der externen Webseite vorhanden sein.)

## 7.5 Hyperlinkziele definieren

### Wo werden Hyperlinks geöffnet?

Wenn Sie den Besucher Ihres Webauftritts mit einem absoluten Link auf einen anderen Webauftritt schicken, wird der verlinkte Webauftritt automatisch in dem Browserfenster geöffnet, in dem er aufgerufen wurde. Es ist möglich, mithilfe des Attributs `target` (Ziel) und des Parameters `"_blank"` einen Link in einem neuen Browserfenster zu öffnen:

```
<a href="http://www.wetter.com" target="_blank">Wetterinfo</a>
```

Dieses Attribut ist umstritten, da Besucher selbst entscheiden sollten, ob sie Webseiten in demselben Fenster (durch Anklicken mit der linken Maustaste) oder in einem neuen Tab bzw. Fenster (durch Betätigen der rechten Maustaste und Auswahl des gewünschten Kontextmenüpunkts ①) öffnen.

KompoZer-Projekt mit einer Spende unterstützen



Link in neuem Tab öffnen



Link in neuem Fenster öffnen

Hyperlink mit der rechten Maustaste anklicken

! Wenn Sie `_blank` als Wert von `target` angeben, wird jedes Mal ein neues **unbenanntes Fenster** geöffnet. Andere Werte (z. B. nur `blank` ohne Unterstrich) führen dazu, dass ein neues Fenster mit dem Namen „blank“ geöffnet wird – jeder weitere Link mit demselben `target`-Wert wird in diesem Fenster geöffnet. Es wird dann also **nicht** für jeden Link ein weiteres Fenster geöffnet: Da sich die benannten Fenster meist im Hintergrund befinden, bekommt der Nutzer davon nichts mit und vermutet einen defekten Link.

## 7.6 Grafiken als Hyperlinks

### Grafiken mit Hyperlinks versehen

Auch Grafiken können als Linkinhalt dienen. Dann ist statt eines Textes das Bild anklickbar. Genau wie bei Texten sollte die Grafik dann das Linkziel erkennen lassen.

Häufig verlinkte Grafiken sind:

- ✓ das Logo einer Seite, das in der Regel zur Startseite führt.
- ✓ eine kleine Version eines Bildes, die zu einer Großansicht (Zoom-Darstellung) führt.
- ✓ ein Bild in einem Anrisstext, das mit einem Artikel verlinkt ist.

```
<a href="vollbild.jpg"></a>
```

- ✓ Das Tag `<a>` leitet den Hyperlink ein.
- ✓ Das Attribut `href` verweist auf die Bilddatei (schöner wäre eine Seite mit einer Großansicht des Bildes und weiteren Navigationsmöglichkeiten).

- ✓ Das Element `img` wird als „Beschriftung“ des Hyperlinks eingesetzt. Dazu wird eine Grafik im Attribut `src` angegeben und ein Alternativ-Text im Attribut `alt`.  
**Geben Sie im `alt`-Text nicht nur an, was auf dem Bild zu sehen ist, sondern auch was passiert, wenn man den Link anklickt!**
- ✓ Das schließende Tag `</a>` beendet die Definition des Hyperlinks.

```
<a href="datei.html"></a>
```

Verlinkte Grafiken sind außer in den o. g. Fällen aufgrund zahlreicher Probleme aus der Mode gekommen. Grafiken sind nicht verlustfrei skalierbar, die Einbindung ist aufwändig und fehleranfällig und die Anpassung an ein neues Design mit hohem Aufwand verbunden.

## 7.7 Blockelemente als Link benutzen

### Blockelemente mit Hyperlinks versehen

Manchmal kann es sinnvoll sein, größere, gestaltete und frei mit Text und Bildern versehene Flächen zu verlinken.

Im folgenden Beispiel kann ein `figure`-element komplett mit enthaltener Grafik und Bildbeschreibung `figcaption` angeklickt werden.

Legen Sie zunächst das Element an, welches verlinkt werden soll:

```
<figure>
  
  <figcaption>Unser Hotel ist hübsch</figcaption>
</figure>
```

Um dieses HTML-Konstrukt legen Sie nun einen Link mit den Tags `<a>` und `</a>`.

```
<a href="hotel_groß.jpg" title="Hier klicken für eine größere
Ansicht">
  <figure>
    
    <figcaption>Unser Hotel ist hübsch</figcaption>
  </figure>
</a>
```

- ! Achten Sie darauf, dass solch ein Konstrukt keine weiteren interaktiven Elemente enthält (weitere Links, Formularfelder, Buttons o. ä.). Der Browser kann sonst nicht interpretieren, was geschehen soll, wenn ein solches interaktives Element angeklickt wird – soll dann der alles umgebende Link aufgerufen oder das enthaltene Element ausgeführt werden?

## 7.8 Übung

### Hyperlinks erstellen

Level		Zeit	ca. 15 min
Übungsinhalte	<ul style="list-style-type: none"> <li>✓ Externe Hyperlinks einfügen</li> <li>✓ Hyperlink zu einer E-Mail-Adresse erstellen</li> <li>✓ Interne Hyperlinks einfügen</li> </ul>		
Übungsdatei	<i>hotel06</i>		
Ergebnisdatei	<i>hotel06-e</i>		

1. Öffnen Sie die Webseite *index.html* im Übungsordner *hotel06* und fügen Sie zu den schon vorhandenen Hyperlinks *Wegbeschreibung* und *Über uns* den Hyperlink zu der neuen Seite *speisekarte.html* ein.
2. Fügen Sie unter dem Adressblock am Ende der Seite *index.html* die Grafik *brief.gif* ein und versehen Sie diese mit einem E-Mail-Hyperlink (*info@hotelvallora.com*).
3. Öffnen Sie die Webseite *restaurant/speisekarte.html* und fügen Sie am Ende der Seite mithilfe der Grafik *oben.jpg* einen internen Hyperlink auf den Anfang der Webseite ein.
4. Speichern Sie die Seite *speisekarte.html* und öffnen Sie sie im Browser. Verkleinern Sie das Browserfenster und scrollen Sie nach unten, um den Anker-Hyperlink zu testen.
5. Ergänzen Sie auch die Webseiten *info/wegweiser.html*, *info/ueberuns.html* sowie *speisekarte.html* um die fehlenden Hyperlinks zu den anderen Seiten (*wegweiser.html*, *speisekarte.html* und *ueberuns.html*).
6. Öffnen Sie die Webseite *wegweiser.html* und erstellen Sie einen Hyperlink zu einem externen Webauftritt, beispielsweise zu einem Routenplaner.  
Der externe Webauftritt soll dabei in einem neuen Fenster geöffnet werden.

[Wegbeschreibung](#) [Über uns](#)

## Speisekarte



Lachsfilet  
mit Reis und Gemüse € 11,90

Putensteak  
mit Pfannengemüse € 8,50

Spaghetti  
Bolognese mit Salat € 6,20

Salatkreation  
Salat der Saison mit Dressing € 9,20

Gemüsegratin  
Blumenkohl, Möhren und Brokkoli in Sahnesauce mit Käse überbacken € 8,90

Hüftsteak  
mit Folienkartoffel und Salat € 12,30



## Unser Buffet

An unserem Mittags- und Abendbuffet bieten wir jeden Tag eine abwechslungsreiche Auswahl an: Vorspeisen, Suppen, Salaten, Hauptgerichten und Desserts

Erwachsene € 8,50 Kinder bis 12 Jahre € 6,00

[nach oben](#)

[Zurück zur Startseite](#)

Ergebnisdatei „speisekarte.html“



[Wegbeschreibung Speisekarte](#)

[Über uns](#) [Tagesablauf](#)

## Über uns

Wir wünschen Ihnen einen angenehmen und erholsamen Aufenthalt im Familienhotel Vallora. Unser erfahrenes und kinderfreundliches Betreuungsteam steht Montag bis Samstag ab 9 Uhr für Sie bereit. Wir bieten jeden Tag ein abwechslungsreiches Programm für Kinder zwischen 8 und 14 Jahren an. Sie finden eine Beschreibung der Aktivitäten an der Rezeption sowie in unserem Betreuungsbereich. Bei Ausflügen müssen die Kinder am Nachmittag vorher angemeldet werden.

Bei gutem Wetter finden unsere Abenteuer- und Spielprogramme im Freien statt. Die Kinder sollten ihre Badebekleidung, Handtücher und Badesandalen mitbringen. An regnerischen Tagen bieten wir Bewegungsspiele und Bastelprojekte in unserer Freizeithalle an. In der Freizeithalle befinden sich eine Klettenwand, drei Trampoline und zahlreiche Spielgeräte (wie zum Beispiel Einräder und Stelzen), die unter Aufsicht benutzt werden dürfen. Tischtennisschläger und -bälle sowie Bälle für Kicker können in der Freizeithalle gegen Pfand ausgeliehen werden.

Eltern können sich in unserem Wellness-Bereich verwöhnen lassen. Besuchen Sie unsere Sauna-Landschaft oder unseren Fitness-Club. Sie können an geplanten Ausflügen teilnehmen oder Ihr eigenes Tagesprogramm gestalten.

## Tagesablauf

09:00 bis 13:00 Uhr Kinderprogramm  
13:00 bis 14:30 Uhr Mittagessen  
14:30 bis 17:30 Uhr Kinderprogramm  
18:00 bis 19:30 Uhr Abendessen  
19:30 bis 21:30 Uhr Familienprogramm

Unsere Rezeption ist 24 Stunden besetzt.  
Sollten Sie Fragen haben, wenden Sie sich an unser Personal.

[nach oben](#)

[Zurück zur Startseite](#)

Ergebnisdatei „ueberuns.html“

# 8

## Listen erstellen

### 8.1 Mit Listen arbeiten

#### Anwendungsmöglichkeiten für Listen

Listen oder Aufzählungen sollten Sie für alle Aufzählungen verwenden, um z. B. die Reihenfolge eines Arbeitsablaufs zu kennzeichnen.

Auch Navigationsmenüs sind typische Vertreter von Listen (Aufzählung von Links).

Listeneinträge können automatisch beispielsweise numerisch oder alphabetisch fortlaufend gekennzeichnet werden.

Es gibt vier Listentypen in HTML5:

- ✓ Unsortierte Listen (`ul`): Die Reihenfolge der Objekte ist beliebig.
- ✓ Sortierte Listen (`ol`): Die Objekte in der Liste sind nachvollziehbar sortiert.
- ✓ Beschreibungslisten (`dl`): Listen mit Begriffen und deren Beschreibungen.  
Mittels `dfn` kann diese Liste zu einer Definitionsliste, z. B. in Glossars, gemacht werden.

### 8.2 Unsortierte Listen nutzen

#### Unsortierte Listen erstellen

Für alle Listen (sortierte wie unsortierte) gilt: Umfasst ein Absatz mehrere Zeilen, werden die Folgezeilen standardmäßig eingezogen dargestellt.

- ✓ Standardmäßig stellen die meisten Browser die Listenelemente durch eingerückte Absätze mit einem vorangestellten dicken schwarzen Punkt oder Sternchen oder anderen Symbolen dar.
- ✓ Unsortierte Listen werden in CSS formatiert.

<code>&lt;ul&gt;</code>	Das Tag <code>&lt;ul&gt;</code> (von engl. <i>unordered list</i> = unsortierte Liste) leitet eine Aufzählung ein, die bis zum abschließenden Tag <code>&lt;/ul&gt;</code> fortgeführt wird.
<code>&lt;li&gt;Text&lt;/li&gt;</code> <code>&lt;/ul&gt;</code>	In die Tags <code>&lt;li&gt;</code> bzw. <code>&lt;/li&gt;</code> (von engl. <i>list item</i> = Listenelement) wird jeweils ein Listenpunkt eingeschlossen.

### Beispiel: Unsortierte Liste

Erstellen Sie eine unsortierte Liste, z. B. innerhalb der Beispieldatei `08_Listen erstellen/hotel/info/ueberuns.html`, wie folgt:

①	<code>&lt;p&gt;In der Freizeithalle befinden sich:&lt;/p&gt;</code>
②	<code>&lt;ul&gt;</code>
③	<code>&lt;li&gt;eine Kletterwand,&lt;/li&gt;</code>
③	<code>&lt;li&gt;drei Trampoline,&lt;/li&gt;</code>
③	<code>&lt;li&gt;zahlreiche Spielgeräte (wie zum Beispiel Einräder, Stelzen, Federballspiele sowie Fahrräder),&lt;/li&gt;</code>
④	<code>&lt;/ul&gt;</code>

- ① Der Satz *In der Freizeithalle befinden sich:* wird mithilfe der Tags `<p>` und `</p>` als normaler Fließtext (Absatz) ausgezeichnet.
- ②, ④ Fügen Sie zur Kennzeichnung einer unsortierten Liste das Tag `<ul>` vor dem ersten Listeneintrag und das Tag `</ul>` hinter dem letzten Listeneintrag ein.
- ③ Die einzelnen Einträge der Liste werden in den öffnenden und schließenden `li`-Tags eingeschlossen.

<p>In der Freizeithalle befinden sich:</p> <ul style="list-style-type: none"> <li>• eine Kletterwand,</li> <li>• drei Trampoline,</li> <li>• zahlreiche Spielgeräte (wie zum Beispiel Einräder, Stelzen, Federballspiele sowie Fahrräder),</li> </ul> <p>die unter Aufsicht benutzt werden dürfen.</p>
--

Eine unsortierte Liste im Browser

- ✓ Wenn Sie die Fenstergröße des Browsers verändern, erfolgt automatisch ein Umbruch.
- ✓ Um die neu erstellte Liste dem Format der restlichen Webseite anzupassen, empfiehlt es sich, in der Stylesheet-Datei das Format für die unsortierte Liste zu ergänzen. Ergänzen Sie die Stylesheet-Datei um den Eintrag ①.

```
ul {color: #003366;
  ① font-size: 1em;}
```

Die Ausgabe des Programms

## 8.3 Sortierte (nummerierte) Listen

### Nummerierte Listen erstellen

Objekte, die in einer bestimmten Reihenfolge präsentiert werden sollen, wie z. B. eine Wegbeschreibung, sollten Sie mithilfe einer sortierten Liste darstellen. Die Vorgehensweise beim Anlegen von sortierten Listen ist bis auf das Element `ol` die gleiche wie bei unsortierten Listen.

#### Wegbeschreibung

1. Fahren Sie in Richtung Küste.
2. Kurz vor dem Ortseingang von Küstendorf biegen Sie links in die Küstenallee ein.
3. Nach der Kurve finden Sie unser Hotel.
4. Parkmöglichkeiten befinden sich direkt vor dem Hotel.

*Eine sortierte Liste im Browser*

- ✓ Browser stellen Listenelemente eingerückt mit vorangestellten fortlaufenden Nummern dar.
- ✓ Das Format der Aufzählung (mittels römischer Ordnungszahlen, alphabetisch u. a. m.) kann mittels CSS festgelegt werden.

<pre>&lt;ol&gt;   &lt;li&gt;Text&lt;/li&gt; &lt;/ol&gt;</pre>	Das Tag <code>&lt;ol&gt;</code> (von engl. <i>ordered list</i> = sortierte Liste) leitet eine Aufzählung ein, die bis zum abschließenden Tag <code>&lt;/ol&gt;</code> fortgeführt wird.
---	---

Die Wegbeschreibung innerhalb der Datei *wegweiser.html* können Sie zu einer sortierten Liste verändern:

<pre>&lt;h1&gt;Wegbeschreibung&lt;/h1&gt; ① &lt;ol&gt; ②   &lt;li&gt;Fahren Sie in Richtung Küste.&lt;/li&gt;      &lt;li&gt;Kurz vor dem Ortseingang von Küstendorf biegen Sie        links in die Küstenallee ein.&lt;/li&gt;      &lt;li&gt;Nach der Kurve finden Sie unser Hotel.&lt;/li&gt;      &lt;li&gt;Parkmöglichkeiten befinden sich direkt vor dem        Hotel.&lt;/li&gt; ③ &lt;/ol&gt;</pre>	
---	--

- ①, ③ Fügen Sie zur Kennzeichnung einer sortierten Liste das Tag `<ol>` vor dem ersten Listeneintrag und das schließende `</ol>`-Tag hinter dem letzten Listeneintrag ein.
- ② Die einzelnen Einträge der Liste werden in den öffnenden und schließenden `<li>`-Tags eingeschlossen.

### Startwert festlegen

Die Nummerierung einer Liste beginnt standardmäßig mit dem Wert 1. Mithilfe des Attributs `start` wird festgelegt, dass die Nummerierung mit einem anderen Wert beginnen soll.

<code>&lt;ol start="Wert"&gt;</code>	<code>&lt;ol start="3"&gt;</code>
--------------------------------------	-----------------------------------

Alternativ dazu kann auch direkt im ersten Listeneintrag über das Attribut `value` der Startwert festgelegt werden.

Mithilfe des Attributs `reversed` (= umgekehrt) können Sie ab HTML5 die Listenummerierung von aufsteigend in absteigend ändern. Bislang besitzt dieses Attribut jedoch noch keine Browserunterstützung.

## 8.4 Beschreibungslisten

### Beschreibungslisten erstellen

Paare von zusammengehörenden Elementen, können Sie mit Beschreibungslisten erstellen.

Für die Erstellung solcher Listen verwenden Sie in HTML ...

- ✓ die Tags `<dl>` und `</dl>` zum Umschließen der Liste,
- ✓ `<dt></dt>` des Fachbegriffs und
- ✓ `<dd></dd>` der Beschreibung.

<pre>&lt;dl&gt;   &lt;dt&gt;Ausdruck&lt;/dt&gt;   &lt;dd&gt;Beschreibung&lt;/dd&gt;   &lt;!-- optional: --&gt;   &lt;dd&gt;weitere dd&lt;/dd&gt;    &lt;dt&gt;2. Ausdruck&lt;/dt&gt;   &lt;dd&gt;Beschreibung&lt;/dd&gt; &lt;/dl&gt;</pre>	<p>Das Tag <code>&lt;dl&gt;</code> (engl.: description list) leitet die Beschreibungsliste ein, die bis zum abschließenden Tag <code>&lt;/dl&gt;</code> fortgeführt wird. In der Liste werden Paare der beiden folgenden Tags eingefügt.</p> <p><code>dt</code> (engl.: description term) gibt den zu beschreibenden Ausdruck an.</p> <p><code>dd</code> (engl.: description definition) enthält den erläuternden Text.</p> <p>Jedem <code>dt</code> können <b>ein oder mehrere</b> <code>dd</code> zugeordnet werden.</p>
--	--

- ✓ Standardmäßig stellen die Browser die Begriffe in eigenen Blöcken dar und ihre Beschreibungen in eingerückten Blöcken.
- ✓ Innerhalb der Listeneinträge von Beschreibungslisten (`dt` und `dd`) können Sie weitere Elemente verwenden.

Solche Paare finden sich häufig:

- ✓ **Beschriftung und Wert**  
*Autor: Max Mustermann*  
 oder  
*Datum: 1.1.1970*
- ✓ **Sprecher und Text**  
 DER HERR: Kennst du den Faust?  
 Mephistopheles: Den Doktor?  
 DER HERR: Meinen Knecht!
- ✓ **Glossar**  
 Köln: Stadt am Rhein, Erzbistum, ...  
 Berlin: Hauptstadt an der Spree, Regierungssitz, ...

### Beispiel: Beschreibungsliste erstellen

Von der Datei `hotel/info/ueberuns.html` (`ueberuns1.html` bis `ueberuns3.html`) finden Sie im Beispielfolder mehrere Varianten mit den verschiedenen Listenarten.

**Unser Wellness-Angebot:**

Dampfbad - Hier können Sie ausgiebig schwitzen!  
 Das Dampfbad ist eine Variante der Sauna mit niedrigerer Temperatur und höherer Luftfeuchtigkeit.  
 Wir bieten Ihnen ein sogenanntes Hamam (türkisches Dampfbad) an.

Sauna  
 In unserer Sauna-Landschaft finden Sie neben einer finnischen Sauna im original finnischen Blockhaus  
 zwei weitere Saunen mit unterschiedlichen Temperaturen,  
 einen kleinen Saunahof sowie einen gemütlichen Ruheraum.

Eine Beschreibungsliste im Browser

```

<h2> Unser Wellness-Angebot:</h2>
① <dl>
② <dt>Dampfbad</dt>
③ <dd>Das Dampfbad ist eine Variante der Sauna mit niedrigerer
    Temperatur und höherer Luftfeuchtigkeit.
    </dd>
    <dd>
    Wir bieten Ihnen ein sogenanntes Hamam (türkisches Dampfbad)
    an.</dd>
② <dt>Sauna</dt>
③ <dd>In unserer Sauna-Landschaft finden Sie neben einer finnischen
    Sauna im original finnischen Blockhaus
    </dd>
    <dd>
    zwei weitere Saunen mit unterschiedlichen Temperaturen,
    </dd>
    <dd>
    einen kleinen Saunahof sowie einen gemütlichen Ruheraum.
    </dd>
    <dd>
③ </dt>
④ </dl>
  
```

- ①, ④ Die Beschreibungsliste wird als Beschreibungslistenelement mit dem öffnenden `<dl>`- und dem schließenden `</dl>`-Tag gekennzeichnet.
- ② Die Begriffe innerhalb der Beschreibungsliste werden eingeschlossen in die Tags `<dt>` und `</dt>`.
- ③ Die Definitionen der Begriffe werden eingeschlossen in die Tags `<dd>` und `</dd>`.

## 8.5 Listen verschachteln und kombinieren

### Listen verschachteln

HTML bietet Ihnen die Möglichkeit, mehrere Aufzählungen und Nummerierungen ineinander zu verschachteln. Dazu fügen Sie eine weitere Liste in eine bereits bestehende Liste ein.

Verschachtelte Listen werden automatisch etwas mehr eingerückt. In der Regel verwenden die Browser bei unsortierten Listen unterschiedliche Aufzählungszeichen für die Verschachtelungsebenen.

Achten Sie darauf, die **innere Liste in die Listeneinträge** `li` **der äußeren Liste** einzutragen, da eine Liste als Kind-Element nur Listeneinträge beinhalten darf. In Listeneinträgen (`li`-Element) dürfen Sie neben Listen alle anderen Elemente verwenden.

Neben Listen dürfen Listeneinträge fast alle anderen Elemente enthalten. So können Sie beispielsweise eine ArtikelListe umsetzen.

```

① <h2>Neueste Nachrichten</h2>
② <ul>
③   <li>
      <article>
        <h3>Kletterwand eröffnet</h3>
        <p>Mit der Kletterwand wurde das Hotel Vallora
          um eine weitere Attraktion bereichert.
        <p><a href="kletterwand.html">Kletterwand</a></p>
      </article>
    </li>
③   <li>[weitere Nachrichten-Teaser...]</li>
④ </ul>

```

- ① Der Satz *Neueste Nachrichten* wird mithilfe der Tags `<h2>` und `</h2>` als Überschrift ausgezeichnet.
- ②, ④ Fügen Sie zur Kennzeichnung einer unsortierten Liste das Tag `<ul>` vor dem ersten Listeneintrag und das Tag `</ul>` hinter dem letzten Listeneintrag ein.
- ③ Die einzelnen Einträge der Liste werden in den öffnenden und schließenden `li`-Tags eingeschlossen und enthalten in diesem Beispiel komplexe Konstrukte (Artikel mit Überschriften und Teaser in einem Absatz).

```

<h2>Tagesablauf</h2>
<ul>
  <li>07:00 bis 09:00 Uhr Frühstück</li>
  <li>
    09:00 bis 13:00 Uhr Kinderprogramm
    <ol>
      <li>03 - 05 Jahre: Mäuse-Gruppe, Raum 101</li>
      <li>06 - 10 Jahre: Hasen-Gruppe, Raum 102</li>
      <li>11 - 14 Jahre: Bären-Gruppe, Raum 103</li>
    </ol>
  </li>
  <li>13:00 bis 14:30 Uhr Mittagessen</li>
  <li>14:30 bis 17:30 Uhr Kinderprogramm</li>
  <li>18:00 bis 19:30 Uhr Abendessen</li>
  <li>19:30 bis 21:30 Uhr Familienprogramm</li>
</ul>

```

**Tagesablauf**

- 07:00 bis 09:00 Uhr Frühstück
- 09:00 bis 13:00 Uhr Kinderprogramm
  1. 03 - 05 Jahre: Mäuse-Gruppe, Raum 101
  2. 06 - 10 Jahre: Hasen-Gruppe, Raum 102
  3. 11 - 14 Jahre: Bären-Gruppe, Raum 103
- 13:00 bis 14:30 Uhr Mittagessen
- 14:30 bis 17:30 Uhr Kinderprogramm
- 18:00 bis 19:30 Uhr Abendessen
- 19:30 bis 21:30 Uhr Familienprogramm

Verschachtelte Liste in der Browseransicht

## Listen kombinieren

Sie haben auch die Möglichkeit, Beschreibungslisten mit Aufzählungen und Nummerierungen zu kombinieren.



Kombinierte Liste in der Browseransicht (Mozilla Firefox)

### Beispiel: Listen kombinieren

In der Datei *ueberuns.html* wird innerhalb der zweiten Erläuterung, um Übersichtlichkeit zu schaffen, die Definitionsbeschreibung mit einer Aufzählung (unsortierte Liste) kombiniert.

```

<h2>Unser Wellness-Angebot:</h2>
① <dl>
  ...
② <dt>Sauna</dt>
③ <dd>In unserer Sauna-Landschaft finden Sie
④ <ul>
  <li>eine finnische Sauna im original Blockhaus</li>
⑤ <li>zwei weitere Saunen mit unterschiedlichen
  Temperaturen,</li>
  <li>einen kleinen Saunahof,</li>
  <li>einen gemütlichen Ruheraum.</li>
④ </ul>
③ Hier können Sie ausgiebig schwitzen!
  </dd>
⑥ </dl>

```

- ① Die Beschreibungsliste fängt mit dem öffnenden `<dl>`-Tag an.
- ② Die Begriffe innerhalb der Beschreibungsliste werden eingeschlossen in die Tags `<dt>` und `</dt>`.
- ③ Die Definitionen der Begriffe werden eingeschlossen in die Tags `<dd>` und `</dd>`.
- ④ Fügen Sie zur Kennzeichnung einer unsortierten Liste das Tag `<ul>` vor dem ersten Listeneintrag und das schließende `</ul>`-Tag hinter dem letzten Listeneintrag ein.
- ⑤ Die einzelnen Einträge der Liste werden in den öffnenden und schließenden `<li>`-Tags eingeschlossen.
- ③ Mit dem `</dl>`-Tag wird die Beschreibungsliste abgeschlossen.

! Listen werden in den verschiedenen Browsern unterschiedlich dargestellt.

## 8.6 Übung

### Mit Listen arbeiten

Level		Zeit	ca. 15 min
Übungsinhalte	<ul style="list-style-type: none"> <li>✓ Sortierte und unsortierte Listen erzeugen</li> <li>✓ Sprunglinks einfügen (Wiederholung)</li> </ul>		
Übungsdatei	hotel08		
Ergebnisdatei	hotel08-e		

1. Öffnen Sie die Webseite *info/ausfluege.html* im Übungsordner *hotel08* und verändern Sie die Ausflugsbeschreibung, indem Sie sie durch Listen strukturieren, vgl. Abbildung.
  - ▶ Benutzen Sie eine unsortierte Liste, um die Vorschau-Bilder zu gruppieren
  - ▶ Verwenden Sie eine sortierte Liste für die Ausflugsziele – jedes Ziel soll einen eigenen Listenpunkt erhalten
  - ▶ Die historischen Sehenswürdigkeiten zählen Sie in einer unsortierten Liste auf.
  - ▶ Fügen Sie je eine ID zu jedem Ausflug hinzu und verlinken Sie die Vorschaubilder mit den neu erstellten IDs – so dienen die Bilder als seiteninterne Navigation.

Das Aussehen der Webseite hat unter der neuen Struktur gelitten. Insbesondere die Darstellung der Vorschaubilder am Seitenanfang nimmt viel Platz in der Vertikalen ein, während in der Horizontalen viel unnötiger Weißraum entsteht.

Dennoch sollten Sie HTML dazu nutzen, Ihre Dokumente sinnvoll zu strukturieren. Mehrere Bilder – insbesondere, wenn diese verlinkt sind – stellen eine Liste von Links dar und sollten daher auch als Liste ausgezeichnet werden.

Mit ein wenig CSS setzen Sie die Bilder wieder nebeneinander:

```
<ul style="list-style-type:none;display:inline;">
```

In den folgenden beiden Kapiteln erfahren Sie mehr über CSS. Eine ausführliche Beschreibung der Formatierungssprache finden Sie im Herdt-Buch *CSS – Cascading Style Sheets*.

## Ausflüge

Folgende Ausflüge können Sie an der Rezeption buchen.



•



•



•



•

### 1. Ballonglücken:



Beim Ballonglücken werden mehrere Heißluftballons am Boden verankert und nach Einbruch der Dunkelheit durch das Befeuern der balloneigenen Gasbrenner zum Glühen gebracht. Die bunten Farben kommen so besonders gut zur Geltung. Ein atemberaubendes Schauspiel für Jung und Alt.

### 2. Historische Sehenswürdigkeiten der Umgebung



Bei diesem Ausflug fahren wir Sie zu folgenden interessanten Sehenswürdigkeiten der Region:

- Römersiedlung aus dem 2. Jh. n. Chr.
- Ruine der Kirche Sankt Peter aus dem 13. Jahrhundert
- Kloster Maria Hilf mit beeindruckendem gotischen Kreuzgang und gemütlicher Klosterschenke

Die Eintrittspreise der Sehenswürdigkeiten sind im Preis des Ausfluges inbegriffen.

### 3. Bootsfahrt



Begleiten Sie uns auf eine sachkundig geführte Bootsfahrt durch das Naturschutzgebiet Krötenwiese.

Ergebnisdatei „ausfluege.html“ in der Browseransicht (Mozilla Firefox). Die unschöne Darstellung der Bilder untereinander und mit vorangestellten Listenpunkten lässt sich per CSS umgestalten.

# 9

## Tabellen erstellen

### 9.1 Grundlagen zu Tabellen

#### Einsatzgebiete von Tabellen

HTML-Tabellen dienen dazu, tabellarische Daten wie z. B. Zeitpläne, Messwerte etc. darzustellen. Die Gestaltung von Tabellen wird mit CSS durchgeführt (vgl. Kapitel 14.5).

#### Aufbau einer Tabelle

HTML-Tabellen beinhalten Zeilen, die Zeilen wiederum enthalten tabellarische Daten, die als Zellen dargestellt werden.

<pre>&lt;table&gt; ... &lt;/table&gt;</pre>	Das Tag <code>&lt;table&gt;</code> zeigt den Beginn einer Tabelle an, das schließende Tag <code>&lt;/table&gt;</code> beendet die Tabelle.
---	--

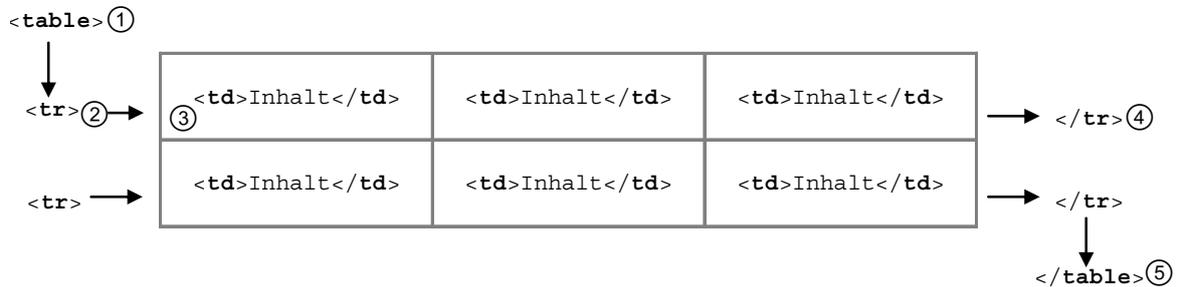
Die Angabe des `table`-Elementes alleine reicht nicht aus, um eine Tabelle darzustellen. Zusätzlich sind mindestens eine Zeile und eine Zelle der Tabelle anzugeben. Dies erfolgt mit den Elementen `tr` und `td`.

<pre>&lt;tr&gt; &lt;td&gt;Tabella - risches Datum &lt;/td&gt; &lt;/tr&gt;</pre>	<p>Das Tag <code>&lt;tr&gt;</code> (von engl. table row = Tabellenzeile) beginnt eine Zeile.</p> <p>Das Tag <code>&lt;td&gt;</code> (von engl. table data = Daten der Tabelle) öffnet eine Zelle, in der Sie die eigentlichen tabellarischen Daten eingeben können.</p> <p>Mit den Tags <code>&lt;/td&gt;</code> und <code>&lt;/tr&gt;</code> beenden Sie je eine Zelle und eine Zeile einer Tabelle.</p> <p><b>Die Anzahl der <code>td</code>-Elemente muss in allen Zeilen gleich sein.</b></p>
---	---

Browser passen die Größe der Zellen automatisch an den darin befindlichen Text an. Ein Umbruch des Textes erfolgt erst, wenn die Gesamtbreite der Tabelle die Breite des Browserfensters überschreitet. Eine einfache Tabelle kann maximal die Breite des Fensters annehmen. Dies können Sie überprüfen, indem Sie die Breite Ihres Browserfensters verkleinern. Die Tabelle wird neu formatiert und ihr Inhalt wird nach Möglichkeit umgebrochen.

### Die wichtigsten Elemente und deren Bedeutung innerhalb einer Tabelle

Im folgenden Beispiel wird eine Tabelle mit 2 Zeilen und 3 Spalten betrachtet:



- ① Das Tag `<table>` beginnt eine Tabelle.
- ② Das Tag `<tr>` beginnt eine neue Zeile,
- ③ in der sich wiederum einzelne Zellen befinden (`<td></td>`).
- ④ Eine Zeile wird mit dem Tag `</tr>` geschlossen.  
In einer einfachen Tabelle müssen alle Zeilen die gleiche Anzahl an Zellen besitzen.  
So verfahren Sie für alle benötigten Zeilen.
- ⑤ Die Tabelle wird mit dem Tag `</table>` beendet.

### Leere Zellen

Tabellenzellen dürfen auch leer sein. Wenn Sie in einer Zelle keine Daten eingeben möchten, lassen Sie den Inhalt der Tags `<td>` und `</td>` leer. Allerdings stellen einige Browser leere Zellen nicht dar. Füllen Sie deshalb die leere Zelle mit einem Zeichen, z. B. einem Strich (Minus). So wird sichtbar angezeigt, dass kein entsprechender Wert vorliegt.

```
<td>-</td>
```

Leere Zelle mit einem Minus-Zeichen füllen

## 9.2 Spalten- und Tabellenüberschriften

### Überschriften mit `<th>` auszeichnen

Um Tabellen verständlich zu machen, sollten Sie die enthaltenen Daten mit Überschriften beschreiben. Verwenden Sie für Überschriften das Element `th`. Diese werden standardmäßig fett hervorgehoben und zentriert dargestellt.

<code>&lt;th&gt;&lt;/th&gt;</code>	Mit dem Element <code>th</code> (von engl. table header = Tabellenkopf) zeichnen Sie Überschriften in Tabellen aus.
------------------------------------	---

### Beispiel: Tabelle erstellen

In der Datei *ueberuns.html* werden die Daten des Tagesablaufs in eine Tabelle umgewandelt:

```

① <table>
②   <tr>
③     <th>Uhrzeit</th>
     <th>Programm</th>
②   </tr>
④   <tr>
⑤     <td>09:00 bis 13:00 Uhr</td>
     <td>Kinderprogramm</td>
④   </tr>
     ...
⑥ </table>

```

#### Tagesablauf

Uhrzeit	Programm
09:00 bis 13:00 Uhr	Kinderprogramm
13:00 bis 14:30 Uhr	Mittagessen
14:30 bis 17:30 Uhr	Kinderprogramm
18:00 bis 19:30 Uhr	Abendessen
19:30 bis 21:30 Uhr	Familienprogramm
ab 22.00 Uhr	Nachtruhe

- ① Die Tabelle wird mit `<table>` geöffnet.
- ② Die erste Zeile wird eingeschlossen in die Tags `<tr>` und `</tr>`.
- ③ Mit den Tags `<th>` und `</th>` kennzeichnen Sie Überschriftzellen.
- ④ Eine weitere Zeile der Tabelle wird mit dem `<tr>`-Tag erstellt.
- ⑤ Die Zellen der neuen Zeile kennzeichnen Sie mit den Tags `<td>` und `</td>`.
- ⑥ Die Tabelle wird mit dem Tag `</table>` geschlossen.

Um die neu erstellte Tabelle sichtbar zu machen bzw. dem Format der restlichen Webseite anzupassen, empfiehlt es sich, in der CSS-Datei einige weitere Angaben zu notieren. Erweitern Sie die CSS-Datei um den Eintrag ①.

```

table, th, td {
  color: #003366;
  font-size: 1em;    ①
  border: 1px solid #003366;
  border-collapse: collapse;
}

```

*CSS-Formatierung der Tabelle*

### Tabellenüberschrift mit dem Element `caption`

Mit dem Element `caption` haben Sie die Möglichkeit, der Tabelle eine Überschrift bzw. eine Beschriftung zu geben.

<code>&lt;caption&gt;Überschriftentext&lt;/caption&gt;</code>	Über das Element <code>caption</code> (Überschrift) beginnen Sie eine Tabellenüberschrift. Dieses Tag muss direkt nach dem öffnenden Tag <code>table</code> angegeben werden.
---	---

```

<table>
<caption>Tagesablauf</caption>
  <tr>
    <th>Uhrzeit</th>
    <th>Programm</th>
  </tr>
  <tr>
    <td>09:00 bis 13:00 Uhr</td>
    <td>Kinderprogramm</td>
  </tr>
  ...
</table>

```

### Tagesablauf

Uhrzeit	Programm
09:00 bis 13:00 Uhr	Kinderprogramm
13:00 bis 14:30 Uhr	Mittagessen

Überschrift der Tabelle mit *caption*

Das Element `caption` ist ein Block-Element und kann neben Text fast alle Elemente (aber keine weiteren Tabellen) enthalten.

## 9.3 Bereiche einer Tabelle organisieren

### Mit Kopf-, Rumpf- und Fußbereichen arbeiten

HTML-Tabellen sollten logisch in Kopf-, Rumpf- und Fußbereich organisiert werden.

Die Aufteilung der Tabellenelemente in Kopf, Fuß und Rumpf hat mehrere Vorteile, die sich nicht auf den ersten Blick erschließen. Kopf- und Fußbereiche werden beim Druck auf allen Seiten ausgegeben, auf denen sich Teile der Tabelle befinden. Suchmaschinen, Vorleseprogramme und andere Software können Ihre Tabellen besser verstehen, was sich im Falle von Suchmaschinen positiv auf das Ranking von Seiten auswirken kann. Blinden wiederum werden die Navigation in der Tabelle und das Verständnis erleichtert.

<pre> &lt;thead&gt;...&lt;/thead&gt; &lt;tfoot&gt;...&lt;/tfoot&gt; &lt;tbody&gt;...&lt;/tbody&gt; </pre>	<p>Mit diesen Tags organisieren Sie eine Tabelle in drei verschiedene Bereiche:</p> <ul style="list-style-type: none"> <li>✓ <code>thead</code> Kopfteil der Tabelle; speziell für Überschriften gedacht</li> <li>✓ <code>tbody</code> Rumpf der Tabelle; enthält die eigentlichen Daten (darf mehrfach vorkommen)</li> <li>✓ <code>tfoot</code> Fußteil der Tabelle; für Erläuterungen zu den Angaben innerhalb der Tabelle</li> </ul>
---	---



- ✓ Jedes Element muss mindestens eine Tabellenzeile `tr` enthalten.
- ✓ Die Elemente `thead` und `tfoot` dürfen in einer Tabellendefinition nur einmal auftreten.
- ✓ Der HTML-Standard gibt die Reihenfolge vor, in der Sie die Elemente notieren müssen: Tabellenkopf, Tabellenrumpf, Tabellenfooter. **Auch wenn Sie sich nicht daran halten, werden die Bereiche immer in dieser Reihenfolge ausgegeben.**

### Beispiel: Einteilen einer Tabelle in Bereiche

Die Tabelle in der Datei *hotel/info/ueberuns.html* wurde ergänzt um Kopf, Rumpf und Fuß (am Aussehen ändert sich dadurch nichts):

```

<table><caption>Tagesablauf</caption>
① <thead>
  <tr>
    <th>Uhrzeit</th>
    <th>Programm</th>
  </tr>
</thead>

③ <tbody>
  <tr>
    <td>09:00 bis 13:00 Uhr</td>
    <td>Kinderprogramm</td>
  </tr>
  <tr>
    <td>13:00 bis 14:30 Uhr</td>
    <td>Mittagessen</td>
  </tr>
</tbody>

② <tfoot>
  <tr>
    <td>© Hotel Vallora</td>
    <td>Stand: 01.01.1970</td>
  </tr>
</tfoot>
</table>
    
```

Tagesablauf

Uhrzeit	Programm
09:00 bis 13:00 Uhr	Kinderprogramm
13:00 bis 14:30 Uhr	Mittagessen
© Hotel Vallora	Stand: 01.01.1970

*Tabelle mit Beschreibung (caption), Überschriften, Tabellen-Rumpf und Tabellen-Fuß*

- ① Der Kopfbereich der Tabelle wird mit dem Tag `<thead>` geöffnet.
- ③ Im Tabellenrumpf zwischen `<tbody>` und `</tbody>` befinden sich die eigentlichen Daten.
- ② Der Fußbereich gehört an das Ende der Tabelle und enthält Informationen über die Tabelle.

## 9.4 Spalten und Zeilen verbinden

Über das Verbinden von Zellen, Spalten oder Zeilen können Sie unterschiedliche Zellenhöhen und -breiten erhalten.

<b>Verbundene Zeilen und Spalten</b>					

*Beispiel für verbundene Zeilen und Spalten einer Tabelle*

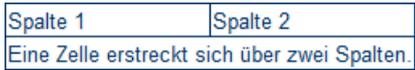
## Spalten verbinden

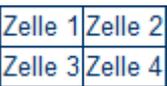
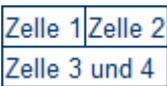
Sie können mehrere Zellen innerhalb einer Zeile verbinden, sodass sich eine Zelle in dieser Zeile über mehrere Spalten erstreckt. Das kann sinnvoll sein, wenn Sie im Fußbereich einer Tabelle nur eine einzige Angabe zum Beispiel zum Copyright oder Stand der Daten bereitstellen möchten.

<code>colspan="Anzahl"</code>	Das Attribut <code>colspan</code> (engl. für Spalten überbrücken) gibt die Anzahl der Zellen an, über die sich die jeweilige Spalte erstrecken soll. Dieses Attribut kann in den Tags <code>&lt;th&gt;</code> und <code>&lt;td&gt;</code> angegeben werden.
-------------------------------	--

### Beispiel: Spalten verbinden

Die erste Zeile der Tabelle enthält zwei einzelne Spalten. Die zweite Zeile enthält nur eine Zelle, die sich jedoch über 2 Spalten erstrecken soll.

HTML-Code	Tabelle
<pre>&lt;tr&gt;   &lt;td&gt;Spalte 1&lt;/td&gt;   &lt;td&gt;Spalte 2&lt;/td&gt; &lt;/tr&gt; &lt;tr&gt;   &lt;td colspan="2"&gt;     Eine Zelle erstreckt sich     über zwei Spalten.   &lt;/td&gt; &lt;/tr&gt;</pre>	

Grundtabelle	Modifizierte Tabelle
	
<pre>&lt;table&gt;   &lt;tr&gt;     &lt;td&gt;Zelle 1&lt;/td&gt;     &lt;td&gt;Zelle 2&lt;/td&gt;   &lt;/tr&gt;   &lt;tr&gt;     &lt;td&gt;Zelle 3&lt;/td&gt;     &lt;td&gt;Zelle 4&lt;/td&gt;   &lt;/tr&gt; &lt;/table&gt;</pre>	<pre>&lt;table&gt;   &lt;tr&gt;     &lt;td&gt;Zelle 1&lt;/td&gt;     &lt;td&gt;Zelle 2&lt;/td&gt;   &lt;/tr&gt;   &lt;tr&gt;     &lt;td colspan="2"&gt;Zelle 3 und 4&lt;/td&gt;   &lt;/tr&gt; &lt;/table&gt;</pre>

Es folgt eine Auflistung von möglichen Spaltenverbindungen innerhalb einer Tabelle, vgl. *colspans.html*.

- Die Anzahl der Spalten einer Tabelle muss in jeder Zeile gleich sein. Wenn sich eine Tabelle beispielsweise über vier Spalten erstreckt, kann eine einzelne Zelle im Höchstfall vier Spalten umfassen, d. h., es kann dem Attribut `colspan` maximal der Wert 4 zugewiesen werden.

Spalte 1	Spalte 2	Spalte 3	Spalte 4
<code>&lt;td colspan=4&gt;</code>			

Spalte 1	Spalte 2	Spalte 3	Spalte 4
<code>&lt;td colspan=2&gt;</code>		<code>&lt;td colspan=2&gt;</code>	

Spalte 1	Spalte 2	Spalte 3	Spalte 4
Zeile 2	<code>&lt;td colspan=3&gt;</code>		

Spalte 1	<code>&lt;td colspan=2&gt;</code>		Spalte 4
<code>&lt;td colspan=2&gt;</code>		<code>&lt;td colspan=2&gt;</code>	

Das Erstellen einer Tabelle mit verbundenen Zellen ist sehr fehleranfällig. Tritt ein Darstellungsfehler auf, ist es meistens sehr schwer, die Ursache zu lokalisieren. Je umfangreicher die Tabelle dabei ist, umso schwieriger wird die Fehlersuche. Deshalb sollten Sie bei der Erstellung der Tabelle so vorgehen, dass Sie zuerst eine Tabelle mit allen Zellen erstellen und testen. Erst danach werden die Spalten miteinander verbunden.

### Zeilen verbinden

<code>rowspan="Anzahl "</code>	Das Attribut <code>rowspan</code> (engl. für Zeilen überbrücken) gibt die Anzahl der zu verbindenden Zeilen an. Die Höhe bezieht sich hierbei auf die Anzahl der Zellen, über die sich eine Zelle erstrecken soll. Dieses Attribut kann mit den Tags <code>&lt;th&gt;</code> und <code>&lt;td&gt;</code> verwendet werden.
--------------------------------	---

### Beispiel: Zeilen verbinden

Die Summe der Zeilen muss in jeder Spalte der Tabelle identisch sein. Da sich die erste Zelle über 2 Zeilen erstreckt, müssen Sie in der zweiten Zeile die erste Spalte nicht mehr angeben.

HTML-Code	Tabelle				
<pre>&lt;tr&gt;   &lt;td rowspan="2"&gt;Zelle 1&lt;/td&gt;   &lt;td&gt;Zelle 2&lt;/td&gt; &lt;/tr&gt; &lt;tr&gt;   &lt;td&gt;Zelle 3&lt;/td&gt; &lt;/tr&gt;</pre>	<table border="1"> <tr> <td>Zelle 1</td> <td>Zelle 2</td> </tr> <tr> <td></td> <td>Zelle 3</td> </tr> </table>	Zelle 1	Zelle 2		Zelle 3
Zelle 1	Zelle 2				
	Zelle 3				

Es folgt eine Auflistung von möglichen Zeilenverbindungen innerhalb einer Tabelle, vgl. *rowspans.html*.

Zeile 1	<td rowspan=4>
Zeile 2	
Zeile 3	
Zeile 4	

Zeile 1	<td rowspan=2>
Zeile 2	
Zeile 3	<td rowspan=2>
Zeile 4	

Zeile 1	Zeile 1
Zeile 2	<td rowspan=3>
Zeile 3	
Zeile 4	

Zeile 1	<td rowspan=2>
<td rowspan=2>	
Zeile 4	<td rowspan=2>

Grundtabelle	Modifizierte Tabelle							
<table border="1"> <tr><td>Zelle 1</td><td>Zelle 2</td></tr> <tr><td>Zelle 3</td><td>Zelle 4</td></tr> </table>	Zelle 1	Zelle 2	Zelle 3	Zelle 4	<table border="1"> <tr><td>Zelle 1</td><td rowspan="2">Zelle 2 und 4</td></tr> <tr><td>Zelle 3</td></tr> </table>	Zelle 1	Zelle 2 und 4	Zelle 3
Zelle 1	Zelle 2							
Zelle 3	Zelle 4							
Zelle 1	Zelle 2 und 4							
Zelle 3								
<pre> &lt;tr&gt;   &lt;td&gt;Zelle 1&lt;/td&gt;   &lt;td&gt;Zelle 2&lt;/td&gt; &lt;/tr&gt; &lt;tr&gt;   &lt;td&gt;Zelle 3&lt;/td&gt;   &lt;td&gt;Zelle 4&lt;/td&gt; &lt;/tr&gt; </pre>	<pre> &lt;tr&gt;   &lt;td&gt;Zelle 1&lt;/td&gt;   &lt;td rowspan="2"&gt;Zelle 2 und 4&lt;/td&gt; &lt;/tr&gt; &lt;tr&gt;   &lt;td&gt;Zelle 3&lt;/td&gt;   &lt;!-- nicht mehr vorhandene Zelle --&gt; &lt;/tr&gt; </pre>							

- !
  - ✓ Das Erstellen einer Tabelle mit verbundenen Zeilen ist schwieriger als die Erstellung von Spaltenverbindungen. Während Sie das Zusammenfassen von Spalten nur in der betreffenden Zeile durchführen, sind beim Zusammenfassen von Zeilen auch alle anderen Zeilen betroffen. Sie müssen in jeder betroffenen Zeile eine oder mehrere Zellen entfernen.
  - ✓ Das Verbinden von Zellen über mehrere Spalten und Zeilen hinweg erfordert nicht nur bei der Erstellung von Tabellen viel Aufmerksamkeit, es macht Tabellen auch für den Nutzer in jedem Fall schwer verständlich. Besonders die mehrdimensionale Zuordnung von Überschriften zu zahlreichen Daten überfordert manchen Besucher Ihrer Seite. In den meisten Fällen sind solche komplexen Tabellenkonstrukte für Blinde mit einem Screenreader überhaupt nicht mehr begreifbar. Statt in einer einzelnen großen und komplexen Tabelle lassen sich Daten immer auch in mehreren leicht verständlichen Tabellen präsentieren.
  - ✓ Verwenden Sie Tabellen **nicht zur Gestaltung von Webseiten**. Tabellen sind ausschließlich für tabellarische Daten vorgesehen.

	HTML-Code	Tabelle
①	<code>&lt;tr&gt;</code>	
②	<code>&lt;td&gt;&amp;nbsp;&lt;/td&gt;</code> <code>&lt;td rowspan="4"&gt;&amp;nbsp;&lt;/td&gt;</code>	
③	<code>&lt;/tr&gt;</code> <code>&lt;tr&gt;</code> <code>&lt;td&gt;&amp;nbsp;&lt;/td&gt;</code>	
④	<code>&lt;/tr&gt;</code> <code>&lt;tr&gt;</code> <code>&lt;td&gt;&amp;nbsp;&lt;/td&gt;</code>	
⑤	<code>&lt;tr&gt;</code> <code>&lt;td&gt;&amp;nbsp;&lt;/td&gt;</code> <code>&lt;/tr&gt;</code>	

- ① In der ersten Tabellenzeile wird die erste Zelle definiert.
- ② Die rechte Zelle der ersten Tabellenzeile umfasst diese und die nächsten drei Zeilen (rowspan="4").
- ③ - ⑤ Die Zeilen 2 bis 4 enthalten jeweils nur noch eine Zelle, da die rechte Zelle bereits durch die Zelle der ersten Tabellenzeile ② festgelegt ist.

	HTML-Code	Tabelle
①	<code>&lt;tr&gt;</code> <code>&lt;td&gt;&amp;nbsp;&lt;/td&gt;</code>	
②	<code>&lt;td rowspan="2"&gt;&amp;nbsp;&lt;/td&gt;</code> <code>&lt;/tr&gt;</code>	
③	<code>&lt;tr&gt;</code> <code>&lt;td&gt;&amp;nbsp;&lt;/td&gt;</code> <code>&lt;/tr&gt;</code>	
④	<code>&lt;tr&gt;</code> <code>&lt;td&gt;&amp;nbsp;&lt;/td&gt;</code>	
⑤	<code>&lt;td rowspan="2"&gt;&amp;nbsp;&lt;/td&gt;</code> <code>&lt;/tr&gt;</code>	
⑥	<code>&lt;tr&gt;</code> <code>&lt;td&gt;&amp;nbsp;&lt;/td&gt;</code> <code>&lt;/tr&gt;</code>	

- ① In Tabellenzeile 1 wird die erste Zelle definiert.
- ② Die rechte Zelle der ersten Tabellenzeile umfasst zwei Zeilen (rowspan="2").
- ③ Die Zeile 2 enthält nur noch eine Zelle.
- ④ - ⑥ Die Zeilen 3 und 4 entsprechen den Zeilen 1 und 2 (① bis ③).

**!** Komplexe Tabellen mit zahlreichen verbundenen Zellen werden schnell unübersichtlich und sind nur noch schwer verständlich. Für responsive Webseiten sind sie völlig ungeeignet (siehe Beispieldatei *09/komplexe-tabelle.html*).

## Freizeitangebote

Sport									
Tennis			Wassersport				Fußball		
Kinder	Jugendliche	Erwachsene	Tauchen		Hochseeangeln	Schwimmen		Kinder	Jugendliche
Der Tennisplatz ist für Kinder nur in Begleitung von Erziehungsberechtigten zu folgenden Zeiten geöffnet...	Jugendliche dürfen den Tennisplatz nur unter Anleitung unserer Tennislehrer nutzen. Dazu stehen Ihnen folgende Zeiten zur Verfügung...	Erwachsene können den Platz täglich von 8:00 Uhr bis 20:00 Uhr nutzen. Sonntags wird der Platz bereits ab 15:00 Uhr geschlossen.	Mit dem Tauchanbieter "Fernandos Unterwassertouren" haben wir einen Vertrag abgeschlossen, der es unseren Gästen ermöglicht, alle Angebote mit einem Rabatt von 10% zu nutzen. Die Öffnungs- und Tourzeiten finden Sie direkt bei "Fernandos Unterwasserwelt".		Für Hochseeangeltouren chartern wir jeden Dienstag ein Boot für unsere Gäste. Die Tour findet nur statt, wenn sie mindestens 5 Teilnehmer finden. Die Anmeldung ist verbindlich. Für Gruppen ab 5 Personen können individuelle Termine vereinbart werden.	Freies Schwimmen ist in unserem Hotelpool jederzeit für alle Gäste möglich.	Schwimmkurse (10 Einheiten) unter fachkundiger Anleitung nur auf Anfrage	Vormittags steht der hoteleigene Fußballplatz den Kindern zur Verfügung	Die Jugendliche dürfen den Platz nachmittags nutzen.
kostenlos	10,- EUR je Stunde für den Trainer	Kostenlos	Preise laut Preistabelle von "Fernandos Unterwasserwelt"		Maximal 200,- EUR / je Person (bei mehr Teilnehmern wird es günstiger)	kostenlos	100,- EUR	kostenlos	kostenlos
Entspannung und Wellness									
Sauna (täglich von 08:00 bis 20:00 Uhr)			Kosmetik						
Damen	Herren	gemischt	Friseur			Maniküre		Massage	
täglich von 10:00 Uhr bis 12:00 Uhr	täglich von 14:00 Uhr bis 16:00 Uhr	in den übrigen Zeiten	Damen		Herren	Damen	Herren	Teilmassage	Ganzkörpermassage
kostenlos			50,- EUR		25,- EUR	Leistungen nach Preisliste im Studio		30,- EUR (eine halbe Stunde)	60,- EUR (eine ganze Stunde)

Solche Tabellen sind sehr unübersichtlich und kaum noch zu verstehen. Besser: Unterteilen Sie dieses komplexe Konstrukt in mehrere oder Texte mit Listen, die durch Zwischenüberschriften strukturiert sind (09/komplexe-tabelle.html)

## 9.5 Tabellen in responsiven Webseiten verwenden

Prinzipbedingt benötigen Tabellen umso mehr Bildschirmbreite, je mehr Spalten enthalten sind.

Das führt auf kleinen Bildschirmen schnell zu Problemen.



Tabellen mit vielen Spalten können selbst auf großen Bildschirmen dazu führen, dass der Platz nicht ausreicht, um die Tabelle in voller Breite komplett anzuzeigen. Auf Smartphones kann das so unübersichtlich werden, dass Tabellen unverständlich werden. Versuchen Sie daher Tabellen klein zu halten, beispielsweise indem Sie eine große Tabelle in mehrere kleinere Tabellen aufteilen. Auch ist es sinnvoll, wichtige Erkenntnisse, die aus den Daten einer Tabelle gewonnen werden können, in Textform zusammenzufassen.

Darüber hinaus gibt es mehr oder weniger komplexe technische Lösungen. Keine davon eignet sich für alle Tabellen. So kann beispielsweise per JavaScript bestimmt werden, welche Spalten angezeigt werden sollen und welche nicht. Bei sehr vielen Spalten kann aber bereits das Auswahlmü unübersichtlich werden und Nutzer, die die Möglichkeit übersehen, werden ausgeblendete Inhalte nicht zu Gesicht bekommen. Außerdem wäre wohl immer die Mehrzahl der Daten ausgeblendet.

Hier ein paar Online-Quellen für die weitere Lektüre:

Sie möchten ...	
Spalten ausblenden	<ul style="list-style-type: none"> <li>▶ <a href="http://filamentgroup.github.io/tablesaw/demo/toggle.html">http://filamentgroup.github.io/tablesaw/demo/toggle.html</a> *</li> <li>▶ <b>Vorteile</b> <ul style="list-style-type: none"> <li>✓ Benutzer behält die Kontrolle</li> <li>✓ Leicht verständlich</li> </ul> </li> <li>▶ <b>Nachteile</b> <ul style="list-style-type: none"> <li>✓ Bedienelemente können leicht übersehen werden.</li> <li>✓ Wenn Daten von Anfang an ausgeblendet werden, können sie übersehen werden.</li> <li>✓ Werden alle Daten zu Anfang eingeblendet, passt die Tabelle nicht auf den Bildschirm.</li> </ul> </li> </ul>
Tabellen neu anordnen	<ul style="list-style-type: none"> <li>▶ <a href="http://fooplugins.github.io/FooTable/docs/getting-started.html#examples">http://fooplugins.github.io/FooTable/docs/getting-started.html#examples</a></li> <li>▶ <b>Vorteile</b> <ul style="list-style-type: none"> <li>✓ Umfangreiche Manipulation, funktioniert bei jeder Bildschirmbreite</li> </ul> </li> <li>▶ <b>Nachteile</b> <ul style="list-style-type: none"> <li>✓ Der Bezug zwischen den Daten wird erschwert (Was gehört zusammen).</li> <li>✓ Der Nutzer kann das Verhalten nicht selber bestimmen.</li> <li>✓ Unübersichtlich, wenn einige Spalten längere Texte enthalten (bei einer Kunden-Datenbank z. B. eine Beschwerde, die der Kunde gemeldet hat).</li> </ul> </li> </ul>
Spalten aus dem sichtbaren Bereich schieben (und zurückholen lassen)	<ul style="list-style-type: none"> <li>▶ <a href="http://filamentgroup.github.io/tablesaw/demo/swipe.html">http://filamentgroup.github.io/tablesaw/demo/swipe.html</a> *</li> <li>▶ <b>Vorteile</b> <ul style="list-style-type: none"> <li>✓ Benutzer behält die Kontrolle</li> </ul> </li> <li>▶ <b>Nachteile</b> <ul style="list-style-type: none"> <li>✓ Die Bedienelemente werden leicht übersehen.</li> <li>✓ Der Nutzer muss sich mit der Bedienung auseinandersetzen.</li> <li>✓ Es sind nie alle Daten eines Datensatzes auf einen Blick sichtbar.</li> </ul> </li> </ul>

Sie möchten ...	
Tabellendarstellung durch Blockdarstellung ersetzen	<ul style="list-style-type: none"> <li>▶ <a href="http://codepen.io/maddesigns/full/pHqnt">http://codepen.io/maddesigns/full/pHqnt</a></li> <li>▶ <b>Vorteile</b> <ul style="list-style-type: none"> <li>✓ Alle Daten eines Datensatzes sind als Einheit erkennbar.</li> <li>✓ Beliebig viele Daten lassen sich darstellen.</li> <li>✓ Optisch ansprechende Aufbereitungen möglich (z. B. als Karteikarte).</li> <li>✓ Weitere Optimierungen denkbar, z. B. mittels flexbox könnten zwei Blöcke nebeneinander dargestellt werden, wenn der Platz dafür reicht, für die Tabelle aber noch zu klein ist.</li> <li>✓ Barrierefreie Umsetzung ist möglich.</li> </ul> </li> <li>▶ <b>Nachteile</b> <ul style="list-style-type: none"> <li>✓ Der Nutzer hat keinen Einfluss auf die Darstellung.</li> </ul> </li> </ul>

\* Die Möglichkeiten der filament group Tabellen lassen sich miteinander kombinieren:  
 (siehe <http://filamentgroup.github.io/tablesaw/demo/kitchensink.html>)  
 Das Script ist mächtig, die Einarbeitung benötigt Zeit, die Dokumentation ist nur auf Englisch verfügbar.

In den meisten Fällen dürfte die letzte Methode „Tabellendarstellung durch Blockdarstellung ersetzen“ die beste sein, sie setzt aber fortgeschrittene CSS-Kenntnisse voraus.

## 9.6 Übung

### Mit Tabellen arbeiten

Level		Zeit	ca. 30 min
Übungsinhalte	<ul style="list-style-type: none"> <li>✓ Tabellen erstellen</li> <li>✓ Tabellen- und Spaltenüberschriften erzeugen</li> <li>✓ Zellen verbinden</li> </ul>		
Übungsdatei	hotel09		
Ergebnisdatei	hotel09-e		

1. Öffnen Sie die Webseite *speisekarte.html* im Übungsordner *hotel09/restaurant* und verändern Sie die Speisekarte, indem Sie sie durch Tabellen strukturieren, vgl. Abbildung.

[Wegbeschreibung](#) [Über uns](#)

## Speisekarte



Lachsfilet mit Reis und Gemüse	€ 11,90
Putensteak mit Pfannengemüse	€ 8,50
Spaghetti Bolognese mit Salat	€ 6,20
Salatkreation Salat der Saison mit Dressing	€ 9,20
Gemüsegratin Blumenkohl, Möhren und Brokkoli in Sahnesauce mit Käse überbacken	€ 8,90
Hüftsteak mit Folienkartoffel und Salat	€ 12,30



## Unser Buffet

An unserem Mittags- und Abendbuffet bieten wir jeden Tag eine abwechslungsreiche Auswahl an: Vorspeisen, Suppen, Salaten, Hauptgerichten und Desserts

Erwachsene € 8,50 Kinder bis 12 Jahre € 6,00

[nach oben](#)

[Zurück zur Startseite](#)

Ergebnisdatei „speisekarte.html“

# 10

## Formulare erstellen

### 10.1 Formulare erstellen

In diesem Kapitel wird die Erstellung einfacher Formulare erläutert. Die Verarbeitung der abgesendeten Daten übernehmen **üblicherweise** Programme, die nicht in HTML erstellt werden können. Zu diversen Programmiersprachen (z. B. Java, PHP und JavaScript) gibt es entsprechende Herdt-Bücher.

#### Wozu werden Formulare benötigt?

Bisher können die Benutzer Ihrer Webseiten nur durch die Eingabe einer URL oder das Betätigen eines Links eine andere Seite aufrufen. Der Kontakt der Benutzer zu Ihnen ist nur über einen E-Mail-Link möglich. Obwohl das für die meisten Nutzer die bequemste und vertrauteste Möglichkeit der Kontaktaufnahme ist, möchten Sie darüber hinaus vielleicht ein Kontaktformular anbieten. Hier kommen HTML-Formulare ins Spiel. Damit können aber noch weit mehr Aufgaben erledigt werden, beispielsweise:

- ✓ über ein Kontaktformular Informationen, Kataloge usw. anfordern,
- ✓ eine Online-Bestellung anbieten,
- ✓ Nutzer melden sich in einem geschützten Bereich an,
- ✓ Suchfelder einbinden.

#### Wo werden Formulare definiert?

Ein Formular können Sie an jeder beliebigen Stelle innerhalb einer Webseite definieren. Auf einer Webseite können Sie auch mehrere Formulare anbieten. Da Formulare aus mehreren Elementen bestehen, müssen diese entweder als zusammengehöriges Formular definiert werden oder die frei in einer Seite notierten Formular-Elemente müssen einem Formular mit einer eindeutigen Bezeichnung (ID) zugeordnet werden. Das Element `FORM` (dt. Formular) benötigen Sie, um ein Formular zu erstellen. In ihm können Sie Schaltflächen, Eingabefelder und Auswahllisten notieren.

Folgendes ist beim Erstellen eines Formulars zu beachten:

- ✓ Elemente, die zu einem Formular gehören, müssen sich innerhalb der Tags `<form>` und `</form>` befinden oder mittels **form-Attribut** einem Formular mit einer bestimmten ID zugeordnet werden.
- ✓ Formulare können nicht ineinander verschachtelt werden. Möchten Sie mehrere Formulare auf einer Webseite integrieren, müssen diese nacheinander angelegt werden.

## Beispiele für Formulardefinitionen

### Formular-Elemente innerhalb eines Formulars

```
<form id="Formularname" action="mailto:mail@example.com"
method="POST|GET"
  enctype="text/plain" target="Ziel">
  Elemente des Formulars
</form>
```

Mit dem Attribut `id` legen Sie den eindeutigen Namen des Formulars fest, um es später z. B. in JavaScript ansprechen zu können. Mit `target` legen Sie das Zielfenster für die Rückmeldung des Skripts fest.

Zur Definition eines Formulars gehören noch weitere Attribute, die bestimmen, wie die Daten des Formulars verarbeitet werden sollen. Das Attribut `method` bestimmt die Methode, mit der die Formulardaten versendet werden. Mögliche Werte, um Anfragen an einen Adressaten zu versenden, sind `POST` und `GET`.

Der Empfänger eines Formulars kann ein lokales E-Mail-Programm, ein Skript in einer Webseite oder eine Anwendung auf einem Webserver sein. Das HTML-Formular übergibt die eingegebenen Daten lediglich. Die Programme, denen die Daten übergeben werden, stellen die eigentliche Funktion bereit; sie verarbeiten die übergebenen Daten.

Ein paar Beispiele, was mit den abgesendeten Daten passieren kann:

- ✓ Speichern der Daten (z. B. in einer Adressdatenbank, Kundendatenbank etc)
- ✓ Versenden von E-Mails an einen oder viele Empfänger
- ✓ Annahme einer Bestellung oder Buchung
- ✓ Weitergabe einiger oder aller Daten an weitere Programme, zum Beispiel eine Navigations-App, eine Fitness-App oder einen elektronischen Kalender
- ✓ Weiterverarbeitung der Daten innerhalb der Webseite (Anzeige der eingegebenen Daten, Manipulation vorhandener Daten, Abgleich mit bereits vorhandenen Daten, um zum Beispiel eine Benutzererkennung zu bestätigen und Zugang zu einem geschützten Bereich zu gewähren)

### Formular-Elemente außerhalb eines Formulars

Wenn Sie Formular-Elemente außerhalb eines Formulars einsetzen möchten, müssen Sie diese einem Formular mit einer ID zuordnen. Das geschieht durch das Attribut `form`. Diesem geben sie als Wert die ID des Formulars mit, zu dem es gehört.

```

<form id="Formularname" action="mailto:mail@example.comd"
method="POST|GET"
      enctype="text/plain" target="Ziel">
  [Elemente des Formulars]
</form>
[...]
<input form="Formularname" type="eingabetyp" >

```

Darüber hinaus unterscheidet sich die Verwendung von Formularelementen außerhalb von Formularen nicht vom Einsatz innerhalb von Formularen.

Folgenden Elementen können Sie im öffnenden Tag das `form`-Attribut mitgeben: `button`, `fieldset`, `input`, `optgroup`, `option`, `select`, `textarea`.

## Formulardaten versenden

Bei der `GET`-Methode werden die Daten des Formulars an die URL angehängt und zum Webserver übertragen. Es dürfen nicht zu viele Daten übertragen werden – die Maximallänge der URL variiert von Browser zu Browser. Der Vorteil von `GET` ist, dass Sie auf diese Daten per Hyperlink verweisen können, da alle Daten in der URL integriert sind. Zusätzlich können Sie diese Daten auch als Favorit in Ihrem Browser ablegen.

`POST` wird normalerweise dafür verwendet, dem Server eine größere Menge von Daten zur Verarbeitung zu übermitteln. Mit der Methode `POST` können Sie z. B. Binär-Daten auf einen Webserver übertragen (uploaden). Die Menge der übertragenen Daten ist im Gegensatz zu `GET` nicht begrenzt.

	POST	GET
Übermittlung größerer Datenmengen	ja	nein
Verwenden von Passwortabfragen	ja	nein
Hyperlink auf Formular-Parameter	nein	ja
Daten als Favoriten	nein	ja
Datenversand per E-Mail	ja	nein

## Formulardaten ausführen

Mit dem Attribut `action` geben Sie an, welches Programm auf dem Webserver nach dem Abschicken der Formulardaten ausgeführt werden soll. Ein solches Programm kann beispielsweise ein Skript sein, das in den Sprachen Perl, PHP oder ASP programmiert wurde. Ihre eingegebenen Daten werden an das Server-Programm übergeben.

Alles weitere bestimmt das Programm, das die Daten entgegen genommen hat – es kann diese zum Beispiel auf einer Webseite anzeigen oder in eine Datenbank schreiben.

Die Übermittlung von Formulardaten per E-Mail ist mithilfe eines `mailto`-Verweises ebenfalls möglich. Dann werden die Daten über das Mail-Programm des Nutzers versendet. **Wie bei E-Mail-Hyperlinks gibt es keine Garantie dafür, dass der E-Mail-Versand für jeden Besucher funktioniert. Darüber hinaus akzeptieren manche E-Mail- Programme aus Sicherheitsgründen keine Formulardaten.**

### Formulardaten codieren

Formulardaten werden nach einem bestimmten Schema codiert. Das verwendete Schema wird über das Formularattribut `enctype` angegeben. Möchten Sie die Daten per E-Mail versenden, weisen Sie dem Attribut `enctype` den Wert `text/plain` zu. Diese Codierung ermöglicht eine lesbare, korrekte Datenübertragung.

### Formularelemente und Plausibilitätsprüfungen

Häufig benutzte Elemente in Formularen sind Schaltflächen (auch Buttons genannt), Eingabefelder und Auswahllisten. Die Schaltflächen kann der Benutzer anklicken, in Eingabefelder kann er beliebigen Text eintragen und über Listen kann er eine Auswahl treffen.

Für die meisten dieser Eingabemöglichkeiten benötigen Sie das `input`-Element zusammen mit dem Attribut `type`, dem Sie folgende Werte zuweisen können:

Wert	Formularelement
<code>text</code>	Einzeiliges Text-Eingabefeld. <b>Vorgabe-Wert, kann daher weggelassen werden.</b>
<code>password</code>	Einzeiliges Text-Eingabefeld mit nicht sichtbaren Zeichen (Passwortfeld)
<code>checkbox</code>	Kontrollfeld, von denen mehrere innerhalb einer Gruppe markiert werden können (Browser stellt ein Quadrat dar, das mit einem Häkchen versehen werden kann)
<code>radio</code>	Optionsfelder, bei denen nur eine Option pro Gruppe markiert werden kann (Browser stellt einen Kreis dar, der mit einem Punkt versehen werden kann)
<code>image</code>	Schaltfläche, für die eine Grafik hinterlegt werden kann (Browser zeigt die Grafik an)
<code>number</code>	Angabe von Zahlen möglich (Browser stellt ein gewöhnliches Texteingabefeld dar, überprüft, ob tatsächlich nur Zahlen eingegeben wurden, <b>0 zu Beginn des Ausdrucks wird nicht übertragen</b> , d.h. 01234 wird zu 1234)

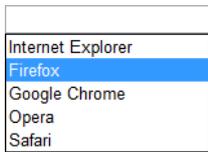
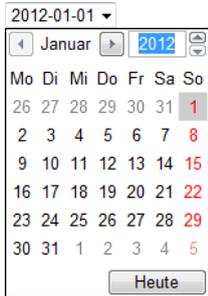
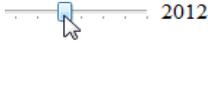
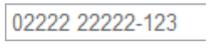
**Kennt ein (älterer) Browser einen Feldtypen nicht, stellt er ein gewöhnliches Texteingabe-Feld dar**, so dass in jedem Fall sichergestellt ist, dass benötigte Angaben gemacht und übergeben werden können. Es spricht also keine Inkompatibilität älterer Browser gegen die Verwendung der Typen. **Wenn die Grundfunktionalität sichergestellt ist, aber zusätzliche Fähigkeiten neuer Browser ausgenutzt werden, um Aussehen oder Nutzerfreundlichkeit weiter zu erhöhen, spricht man von *progressive enhancement*.**

Stellt der Browser fest, dass eine Eingabe nicht plausibel ist, verhindert er das Versenden des Formulars und zeigt eine Fehlermeldung am ersten nicht korrekt ausgefüllten Feld.

Neben der ggf. unterschiedlichen Darstellung und der Prüfung durch den Browser, hat die korrekte Verwendung der Typen noch einen wichtigen Vorteil. Immer mehr Menschen sind mit Smartphones im Netz unterwegs. Auf den kleinen Bildschirmen ist der Platz für Tastaturen begrenzt.

Smartphones bieten nur benötigte und erlaubte Zeichen auf ihren Tastaturen an, wenn ein Eingabefeld ausgewählt wurde. So erhalten Nutzer nach einem Klick auf ein Feld vom Typ `number` oder `tel` einen Ziffernblock mit großen Schaltflächen oder für ein Feld vom Typ `url` eine komfortable Möglichkeit Top-Level-Domains wie `.com` oder `.de` an eine Domain anzuhängen.

### Beispielhafte Darstellung einiger `input`-Typen

Feldtyp	HTML-Code	Darstellung
Auswahl- liste	<pre>&lt;input list="browsers"&gt; &lt;datalist id="browsers"&gt;   &lt;option value="Internet Explorer"&gt;   &lt;option value="Firefox"&gt;   &lt;option value="Google Chrome"&gt;   &lt;option value="Opera"&gt;   &lt;option value="Safari" &gt; &lt;/datalist&gt;</pre>	
Datum	<pre>&lt;input type="date" min="2012-01-01" max="2014-12-31" value="2012-01-01"&gt;</pre>	
E-Mail	<pre>&lt;input type="email" value="name@provider.de"&gt;</pre>	
Farb- wähler	<pre>&lt;input type="color"&gt;</pre>	
Regler	<pre>&lt;input type="range" min="1950" max="2050" value="2012" onInput="year.value=value"&gt; &lt;output id="year"&gt;&lt;/output&gt;</pre>	
Suchfeld	<pre>&lt;input type="search" placeholder="Suche"&gt;</pre>	
Telefon	<pre>&lt;input type="tel" placeholder="02222 22222-123"&gt;</pre> <p>Auf modernen Smartphones wird beim Klick in dieses Feld eine numerische Tastatur angezeigt.</p>	
Uhrzeit	<pre>&lt;input type="time"&gt;</pre>	

Feldtyp	HTML-Code	Darstellung
URL	<code>&lt;input type="url" placeholder="http:"&gt;</code>	<input type="text" value="http://"/>
Zahl	<code>&lt;input type="number" min="1" max="10" value="1"&gt;</code>	<input type="text" value="1"/>

## 10.2 Eingabefelder erstellen

### Einzeilige Eingabefelder erstellen

Alle einzeiligen Eingabefelder werden über die folgenden Attribute definiert (der senkrechte Strich symbolisiert eine ODER-Option):

```
<input
  type="text|password|email|search|tel|url|hidden"
  id="eindeutige Bezeichnung"
  name="Feldname"
  size="Größe (Breite) "
  maxlength="Maximallänge"
  value="Vorgabewert"
  placeholder="Platzhalter">
```

- ✓ Mit dem `<input>`-Tag legen Sie Eingabefelder an.
- ✓ Über das Attribut `type` legen Sie den Typ fest. Mit dem Typ `hidden` kennzeichnen Sie Felder, die nicht angezeigt, deren Werte aber an das verarbeitende Programm übergeben werden.
- ✓ Das Attribut `name` enthält den eindeutigen Namen des Eingabefeldes.
- ✓ Durch die Verwendung von `size` bestimmen Sie die sichtbare Anzahl der Zeichen.
- ✓ Über `maxlength` legen Sie die maximale Zeichenanzahl fest.
- ✓ Das Attribut `value` ermöglicht einen Vorgabetext im Feld.
- ✓ `readonly` wird eingesetzt, wenn ein Feld mit einem Wert belegt ist, der vom Benutzer nicht geändert werden darf.

### Eingabefelder beschriften

Damit Nutzer wissen, welche Angaben Sie in einem bestimmten Feld erwarten, ist es wichtig, Eingabefelder zu beschriften.

Dazu dient das Element `label`. Die Beschriftung muss einem Feld eindeutig zugeordnet sein. Eine Möglichkeit ist, das Eingabefeld in das `label` zu schreiben:

```
<label>Vorname: <input type="text" ></label>
```

Häufiger wird eine Beschriftung einem Eingabefeld allerdings über die Attribute `for` und `id` zugeordnet. Das gibt Ihnen mehr Freiraum bei der Gestaltung eines Formulars.

```
<label for="vorname">Vorname:</label>
<input type="text" id="vorname">
```

Diese logische Zuordnung können auch Programme verstehen, was den Besuchern Ihrer Seite die Nutzung des Formulars erheblich erleichtert.

Browser werten beispielsweise einen Klick auf eine solche Beschriftung wie einen Klick in ein Eingabefeld. Das erleichtert vor allem das Aktivieren kleiner Eingabefelder, wie ein Kästchen, das abgehakt werden soll. Besonders die Nutzer von Smartphones profitieren hiervon. Noch wichtiger ist eine korrekte Beschriftung mittels `label` für Blinde, die nur so den Zusammenhang zwischen einem Text und einem Eingabefeld wahrnehmen können.

### Beispiel: Formulare mit einzeiligen Eingabefeldern erstellen

Die Beispieldatei `10_Formulare_erstellen/formular1.html` zeigt Ihnen verschiedene Arten von Eingabefeldern und die jeweils möglichen Attribute, die Aussehen und Funktion der Felder beeinflussen.

```
<body>
  <h3>Eingabefelder</h3>
  ① <form action="mailto:mail@example.com" method="POST"
    enctype="text/plain">
  ②   <label>Vorname: <input type="text" name="Vorname"></label>
  ③   <label>Name: <input name="Name" size="23"></label>
    <label>
  ④   E-Mail: <input type="email" name="EMail"
      placeholder="mail@example.com">
    </label>
  ⑤   <label>Postleitzahl: <input type="text" name="PLZ"
      pattern="^(?: [Dd] \s* [---] ?\s*)? [0-9]{5}$"></label>
  ⑥   <label>Password: <input type="password"
      name="Passwort"></label>
  ⑦   <p><input type="submit" name="Abschicken"
      value="Absenden">Abschicken</p>
  ⑧ </form>
</body>
```

- ① Das HTML-Formular wird mit `<form>` geöffnet. Im Attribut `action` legen Sie fest, dass die Daten des Nutzers nach dem Absenden des Formulars per E-Mail an die Adresse `mail@example.com` versendet werden. Bei einer E-Mail-Übermittlung der Daten wird die Versandmethode `method="POST"` verwendet. Das Attribut `enctype="text/plain"` codiert die Daten im Textformat.
- ② Eingabefelder erzeugen Sie mit `<input>`. Das Feld soll normalen Text enthalten. Notieren Sie daher den Typ `text`. Der Browser stellt ein einzeliges Eingabefeld dar. Zur weiteren Verwendung der Formulardaten müssen Sie jedem Formularelement einen eindeutigen Namen zuweisen. Dies realisieren Sie über das Attribut `name`. In diesem Fall heißt das Feld `Vorname`.
- ③ Das zweite Eingabefeld besitzt den Namen `Name`. Das optionale Attribut `size` verändert die Größe des Eingabefeldes. Die Breite wird auf 23 sichtbare Zeichen eingestellt. Beachten Sie, das `type="text"` weggelassen wurde. Da `text` der Standardwert ist, muss dieser nicht ausdrücklich angegeben werden.
- ④ Zur Angabe der E-Mail-Adresse wird das Eingabefeld `EMail` angelegt. Das Feld soll bereits nach dem Laden der Webseite einen Vorschlag beinhalten. Dazu verwenden Sie das Attribut `placeholder` mit dem anzuzeigenden Text.
- ⑤ Die Postleitzahl besteht in Deutschland aus 5 Ziffern. Mit `pattern` und einem regulären Ausdruck teilen Sie dem Browser mit, dass genau 5 Zahlen erwartet werden. Außerdem werden der Buchstabe „D“ und/oder Bindestriche und/oder Leerzeichen erlaubt, da manche Menschen etwas wie „D-“ oder „D -“ einer Postleitzahl voranstellen. Mehr zu regulären Ausdrücken erfahren Sie im nächsten Abschnitt. Vor der Übernahme der Daten in eine Datenbank, sollten Sie gegen eine Liste existierender Postleitzahlen prüfen, um sicherzustellen, dass die angegebene Postleitzahl vorhanden ist.
- ⑥ Passwortfelder definieren Sie über den Elementtyp `password`. Damit werden alle Eingaben im Feld `Password` durch andere Zeichen ersetzt, z. B. ★ oder ●.
- ⑦ Damit die Formulardaten abgesendet werden können, fügen Sie eine Schaltfläche (Button) ein. Diese ist vom Typ `submit` (absenden). Die Beschriftung der Schaltfläche legen Sie mit dem Text im Element (also zwischen `<button>` und `</button>`) fest.
- ⑧ Zum Schluss schließen Sie das Formular mit dem Endtag `</form>`.

Unterschiedliche Eingabefelder („formular1.html“)

## Mit regulären Ausdrücken auf Plausibilität prüfen

Es scheint naheliegend zu sein, einem Feld mit einer Postleitzahl, den Typ `number` mitzugeben. Der ist aber nur für echte Nummern vorgesehen, nicht für Daten wie es Postleitzahlen sind. Der Unterschied lässt sich leicht verstehen, wenn man sich klar macht, dass eine Zahl nicht mit einer Null beginnt, was bei Postleitzahlen durchaus der Fall sein kann.

Somit ist eine Postleitzahl eine **normale Zeichenkette** (string), die nichts anderes ist, als der Standardfall – um das deutlich zu machen steht im Formular ausdrücklich `type="text"`, obwohl der Standardwert nicht ausdrücklich angegeben werden muss (und nicht sollte, um Fehler zu vermeiden).

Sie können mit dem Attribut `pattern` festlegen, was in das Textfeld eingegeben werden darf. Als Wert wird dem Attribut ein **regulärer Ausdruck** mitgegeben. Reguläre Ausdrücke sind Regeln, die bestimmte Zeichen oder Zeichenfolgen beschreiben. Auch die Erstellung von Mustern sind möglich. Sie können z. B. abfragen, ob `zeichenkette@zeichenkette.zeichenkette` eingegeben wurde, um zu überprüfen, ob eine gültige E-Mail-Adresse angegeben wurde.

Reguläre Ausdrücke sind ein umfangreiches Thema, das in diesem Buch nicht vollständig behandelt werden kann. Zwei weitere Beispiele seien aber genannt.

### Passwort-Vorgaben überprüfen

Ein Passwort soll mindestens acht Zeichen lang sein und mindestens einen Großbuchstaben, einen Kleinbuchstaben und eine Ziffer beinhalten:

```
<label> Passwort (mindestens 8 Zeichen, davon mindestens ein  
Großbuchstabe, ein Kleinbuchstabe und eine Ziffer):  
  <input type="password" name="pw"  
    pattern="(?=.*\d) (?=.*[a-z]) (?=.*[A-Z]) .{8,}">  
</label>
```

### Vorgabe für eine URL überprüfen

Wenn eine Webseite angegeben wird, soll diese mit `http://` oder `https://` beginnen.

```
<label> Webseite (Angabe mit http beginnen):  
  <input type="url" name="website"  
    pattern="https?://.+">  
</label>
```

Mehr über reguläre Ausdrücke finden Sie im deutschen Mozilla Developer Network unter [https://developer.mozilla.org/de/docs/Web/JavaScript/Guide/Regular\\_Expressions](https://developer.mozilla.org/de/docs/Web/JavaScript/Guide/Regular_Expressions).

Verwenden Sie solche Beschränkungen mit Augenmaß. Einerseits können diese eine Hilfe sein. In Feldern wo tatsächlich nur Ziffern Sinn machen (z. B. eine Altersangabe), sollten Sie den dafür gedachten Eingabe-Type `number` auch verwenden. Dann wird auf Geräten mit virtueller Tastatur ein Ziffernblock angezeigt, statt einer vollständigen Tastatur. Die wenigen dafür benötigten Tasten werden sehr groß angezeigt und sind daher leicht zu tippen. Es können auch nicht versehentlich Buchstaben eingetippt werden, weil diese gar nicht angeboten werden.

Sind die Vorgaben zu ungenau, lassen sich Felder praktisch beliebig befüllen und Nutzer von assistiven Techniken wie Blinde mit Vorleseprogrammen erfahren nicht, welche Daten Sie an dieser Stelle erwarten. Wenn Sie die übergebenen Daten dann serverseitig überprüfen und ablehnen, kann das frustrieren und Nutzer füllen das Formular oft nicht erneut aus.

Zu strenge Vorgaben können ebenfalls frustrieren und Nutzer davon abhalten, Formulare auszufüllen.

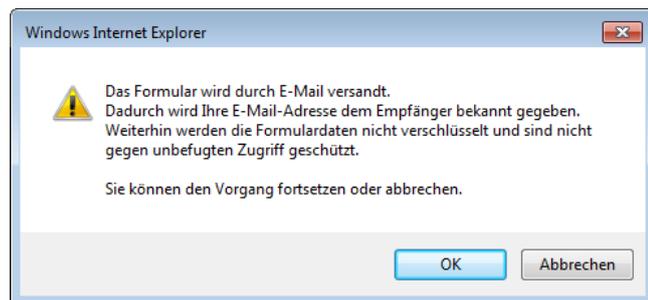
Grundsätzlich sollten Sie möglichst viel durchlassen, so lange darin die gewünschte Information erkennbar ist und es dadurch nicht zu Sicherheitsproblemen kommt.

Andererseits sollten Sie selber so penibel wie möglich auf konsistente Daten achten.

**Die Vorteile und Nachteile der Regeln und wie streng die Regeln sein müssen, lässt sich nur bei einem konkreten Formular im Einzelfall abwägen** und hängt z. B. mit der Zusammensetzung Ihrer Besucher zusammen. Werten Sie hierzu ggf. Ihre Statistiken aus.

Auch sollten Sie Daten beim Empfang auf dem Server überprüfen. Im Browser lässt sich die Prüfung (z. B. mit Erweiterungen) einfach abschalten, um Ihnen unerwünschte Daten zuzusenden. **Daher müssen Sie die Daten serverseitig unbedingt noch einmal überprüfen.** Die Überprüfung im Browser ist nur eine Usability-Maßnahme, die Ihren Besuchern das Ausfüllen des Formulars erleichtert. **Die Daten vom Browser überprüfen zu lassen ist keine geeignete Maßnahme, um für Sicherheit oder konsistente Daten zu sorgen.**

Nach der Eingabe der Daten und dem Absenden des Formulars werden die Daten beim Benutzer über das eingestellte E-Mail-Programm versendet. Der Internet Explorer blendet beispielsweise vor dem Versand eine entsprechende Meldung ein.



Der im Formular angegebene Empfänger erhält aufgrund des angegebenen Tags `enctype="text/plain"` eine E-Mail mit dem Betreff *Formular-Post von Browsername* und folgendem Inhalt:

```
Vorname=Sally
Name=Wagner
EMail=mail@example.com
PLZ=12345
Passwort=mein Passwort
Abschicken=Absenden
```

Die im Formular hinterlegten Bezeichnungen der Felder werden mit den vom Benutzer eingegebenen Daten formatiert aufgelistet.

Wenn Sie im Formular das Attribut `enctype` mit dem Wert `text/plain` nicht angeben, werden die Daten per E-Mail wie folgt versendet:

```
Vorname=Sally&Name=Wagner&EMail=mail@example.com&PLZ=12345&Passwort=
mein+Passwort&Abschicken=Absenden
```



Personenbezogene Daten dürfen nur verschlüsselt versendet werden. Mehr dazu unter <https://www.datenschutzbeauftragter-info.de/website-rechtssicher-erstellen-so-gehts/>

Ebenso wichtig ist es, Passwörter niemals unverschlüsselt zu übertragen. **Unverschlüsselter Netzwerkverkehr kann von jedem mitgelesen werden.**

## Mehrzeilige Eingabefelder erzeugen

Einzeilige Eingabefelder sind gut geeignet, um kurze Ausdrücke aus wenigen Wörtern aufzunehmen. Für längere Texteingaben sind sie ungeeignet. Dafür stehen Ihnen mehrzeilige Eingabefelder zur Verfügung.

```
<textarea name="Feldname" rows="Zeilenanzahl"
cols="Spaltenanzahl">
</textarea>
```

- ✓ Das Tag `<textarea>` leitet das Formularelement ein.
- ✓ Das Attribut `name` enthält den eindeutigen Namen des Eingabefeldes.
- ✓ Über das Attribut `rows` bestimmen Sie die sichtbare Zeilenanzahl. Genauere Angaben können Sie mittels CSS machen.
- ✓ Durch `cols` legen Sie die Breite des Feldes fest. Es wird die Anzahl der Zeichen angegeben, die innerhalb einer Zeile angezeigt werden sollen. Genauere Angaben können Sie mittels CSS machen.
- ✓ Im Gegensatz zu `input` benötigt dieses Element ein Endtag `</textarea>`.
- ✓ Der Text, den der Benutzer eingibt, darf unbegrenzt lang sein.

### Beispiel: Formulare mit mehrzeiligen Eingabefeldern erstellen

In der Beispieldatei `10_Formulare_erstellen/formular2.html` wird ein Formular angelegt, das ein Eingabefeld von 5 Zeilen und 30 Spalten (eine „Spalte“ entspricht einem Zeichen) enthält.

**Mehrzeiliges Eingabefeld**

Dieser Text wird automatisch im mehrzeiligen Eingabefeld dargestellt.



```
<html>
  <head>
    <title>Formular zur Dateneingabe</title>
  </head>
  <body>
    <h3>Mehrzeiliges Eingabefeld</h3>
    <form action="mailto:name@mein.web" method="POST"
    enctype="text/plain">
      <label>Ihre Nachricht: <textarea id="Mehrzeiler" rows="5"
      cols="30">Dieser Text wird automatisch im mehrzeiligen
      Eingabefeld dargestellt.</textarea></label>
      <p><button type="submit" name="Abschicken"
      value="Abschicken">Abschicken</button></p>
    </form>
  </body>
</html>
```

## 10.3 Auswahllisten

### Auswahllisten bzw. Listenfelder erstellen

Eine Möglichkeit, dem Benutzer eine Auswahl zwischen mehreren Punkten zu ermöglichen, sind Auswahllisten.

```
<select name="Listenname" size="Anzahl" multiple>
  <option value="Uebergabewert" selected>Anzeigewert</option>
</select>
```

- ✓ Das Tag `<select>` öffnet eine Auswahlliste.
- ✓ Mit `name` vergeben Sie einen im Formular eindeutigen Namen.
- ✓ Die Anzahl der gleichzeitig sichtbaren Listeneinträge legen Sie über das Attribut `size` fest. Standardmäßig wird ein Eintrag angezeigt und die Liste wird als Drop-down-Liste dargestellt (aufklappbares Listenfeld). Bei manchen Listen kann es von Interesse sein, dass auch mehrere Einträge ausgewählt werden könnten. Dann geben Sie das Attribut `multiple` an.
- ✓ Die einzelnen Einträge der Liste werden jeweils über das Tag `<option>` geöffnet. Zwischen Start- und Endtag geben Sie den Text an, der **angezeigt und versendet** werden soll.
- ✓ Soll ein anderer Wert als der angezeigte übermittelt werden, geben Sie diesen mit dem Attribut `value` an.
- ✓ Eine Vorauswahl der Listeneinträge realisieren Sie mit dem Attribut `selected`.

Mit HTML5 haben Sie zusätzlich die Möglichkeit, Textfelder mit Auswahllisten zu kombinieren. Legen Sie dazu ein Textfeld an und verknüpfen Sie damit eine Liste von Vorschlägen:

```
<input list="browsers" type="text" >
<datalist id="browsers">
  <option value="Internet Explorer">
  <option value="Firefox">
  <option value="Chrome">
  <option value="Opera">
  <option value="Safari">
</datalist>
```

Je nach Browser gibt es mehrere Möglichkeiten, dieses Feld zu befüllen. Google Chrome zeigt beispielsweise ein Texteingabefeld. Wenn dieses Feld mit der Maus überfahren wird (um den Cursor hineinzusetzen) zeigt Chrome ein Dreieck an der rechten Seite an. Klickt ein Nutzer auf das Dreieck, öffnet sich eine Auswahlliste, ähnlich dem Element `select`. Klickt der Nutzer in das Feld, kann er Text per Tastatur eingeben.

Beim Firefox wird die Liste der Einträge sichtbar und auswählbar, sobald der Nutzer zweimal in das Texteingabefeld klickt (sonst bietet der Firefox bei einem solchen zweimaligen Klick eine Liste von früheren Eingaben an).

Gibt man einen Text ein, bieten Browser, die `datalist` unterstützen, passende Begriffe als Autovervollständigungsoption an.

Safari unterstützt in der aktuellen Version `datalist` noch nicht.

Noch unterschiedlicher ist die Unterstützung der Browser für `datalist` in Verbindung mit anderen Feldtypen als Text. So unterstützte bereits der Internet Explorer `datalist` in Verbindung mit `range`, Chrome zeigt sogar vorgegebene Farben beim Eingabetyp `color` an. Aufgrund der ständigen Weiterentwicklung hilft hier nur eigenständiges Testen.

Beachten Sie: Dass einer, mehrere oder sogar alle Browser eine Funktion nicht unterstützen, heißt nicht, dass man diese nicht verwenden kann. Die Browserentwicklung schreitet rasant voran und Sie können heute bereits Komfortfeatures in Ihre Website einsetzen, die Besuchern erst in ein paar Wochen, Monaten oder Jahren zur Verfügung stehen.

**Das Prinzip für alte oder aktuelle Browser die reine Funktionalität sicherzustellen und für spätere Generationen mehr Komfort zu bieten, nennt sich progressive enhancement.**

### Beispiel: Formulare mit Auswahllisten erstellen

In der Beispieldatei `10_Formulare_erstellen/formular3.html` werden in einem Formular zwei Auswahllisten eingebunden.

```

<DOCTYPE html>
<html lang="de">
  <head>
    <title>Auswahllisten</title>
  </head>
  <body>
    <form action="mailto:wer@adresse.web" method="POST"
    enctype="text/plain">
      <label>Wählen Sie Ihr Lieblingshaustier:
      ① <select size="4" id="haustier" multiple>
      ②   <option selected> Hund</option>
      ③   <option> Katze</option>
        <option selected> Vogel</option>
        <option> Meerschweinchen</option>
      ④ </select></label>
      <label>Eine "Drop-Down"-Liste mit anderen
      Übergabewerten:
      ⑤ <select id="haustier2">
      ⑥   <option value="1"> Hund mit Übergabewert 1</option>
        <option value="2"> Katze mit Übergabewert 2</option>
        <option selected value="3"> Vogel mit Übergabewert
      3</option>
        <option value="4"> Meerschweinchen mit Übergabewert
      4</option>
      ⑦ </select></label>
      <label>Wählen Sie Ihr Lieblingshaustier mal auf die
      moderne Art:

```

```

⑧ <input list="haustier3" type="text" name="haustier3">
⑨ <datalist id="haustier3">
    <option value="Hund">
    <option value="Katze">
    <option value="Vogel" selected>
    <option value="Meerschweinchen">
  </datalist>
</label>
<p><button type="submit" id="Abschicken"
value="Abschicken">Abschicken</button></p>
</form>
</body>
</html>

```



- ① Innerhalb des Formulars wird die Auswahlliste mit dem Tag `select` eingeleitet. Zur Identifizierung der Daten erhält die Liste den Namen `haustier`. Vier Einträge sollen sichtbar und eine Mehrfachauswahl (`multiple`) möglich sein.
- ② Der erste Listeneintrag wird so erstellt, dass der Text `Hund` angezeigt wird. Dieser Text ist gleichzeitig der Wert, der beim Selektieren und Absenden des Formulars übermittelt wird. Nach dem Laden der Webseite ist automatisch der Eintrag ausgewählt (`selected`).
- ③ Der zweite Eintrag hat den Inhalt `Katze` und ist nicht vorausgewählt. Es folgen die weiteren Einträge der Liste.
- ④ Die erste Liste wird mit dem Endtag `</select>` beendet.
- ⑤ Die zweite Auswahlliste erhält die Bezeichnung `haustier2`. Da kein Attribut `size` angegeben ist, erstellt der Browser automatisch eine Drop-down-Liste. Dies ist eine Liste, in der nur ein Eintrag zu sehen ist und in der die restlichen Einträge erst durch Aufklappen sichtbar werden.
- ⑥ Der erste Eintrag erhält einen separaten Übergabewert (`value="1"`). Dies bedeutet, dass der Nutzer zwar den Eintrag `Hund` mit Übergabewert `1` sieht, der Empfänger des Formulars jedoch mit dem Wert `1` eine automatische Weiterverarbeitung durchführen kann.
- ⑦ Nach der Angabe der restlichen Einträge wird die Liste geschlossen.
- ⑧ Im öffnenden Tag eines `input`-Elementes notieren Sie das Attribut `list`, um ein Eingabefeld mit einer Liste vorgegebener Auswahlmöglichkeiten zu erstellen
- ⑨ Nutzen Sie das Element `datalist`, um eine Liste mit Begriffen zu erstellen, die Sie dem Nutzer der Seite zur Auswahl anbieten.

## Hotel Vallora

Wählen Sie Ihr Lieblingshaustier:

Hund  
 Katze  
 Vogel  
 Meerschweinchen

Eine "Drop-Down"-Liste mit anderen Übergabewerten:

Vogel mit Übergabewert 3

Wählen Sie Ihr Lieblingshaustier mal auf die moderne Art:

*Darstellen von Auswahlfeldern („formular3.html“)*

Wenn Sie nach dem Laden der Webseite das Formular unverändert absenden, erhält der Empfänger eine E-Mail mit folgendem Inhalt:

Da in der ersten Liste zwei Einträge selektiert waren, werden sie einzeln als Werte von `haustier1` aufgelistet. Der Inhalt der zweiten Liste `haustier2` ist der separat angegebene Wert 3 vom Eintrag `Vogel`. Die dritte Liste `haustier3` enthält den vorgegebenen Wert des Elementes `data1`.

```
haustier1=Hund
haustier1=Vogel
haustier2=2
haustier3=Vogel
Abschicken=Abschicken
```

## Optionen gruppieren

Mit dem Element `optgroup` lassen sich Einträge in einer Auswahlliste gruppieren, vgl. folgende Abbildung.

```
<select>
  <optgroup label="Musik">
    <option value="cd">CD</option>
    <option value="mp3">MP3</option>
  </optgroup>
  <optgroup label="Filme">
    <option value="dvd">DVD</option>
    <option value="bluray">Blu-ray</option>
  </optgroup>
</select>
```



Einträge im HTML-Code mit `optgroup` gruppieren und das Aussehen im Browser (Beispieldatei „10\_Formulare erstellen\formular4.html“)

## 10.4 Kontrollkästchen und Optionsfelder

### Kontrollkästchen erzeugen

Eingabefelder vom Typ **Checkbox** werden als Quadrate dargestellt, die mit einem Häkchen versehen werden können. Mehrere solcher Eingabefelder mit demselben Namen bilden eine Gruppe. **Aus einer Gruppe kann der Anwender entweder keine, eine oder mehrere Möglichkeiten selektieren.** Eine solche Auswahl erstellen Sie wie folgt:

```
<input type="checkbox" name="Gruppenname" value="Übergabewert" checked>
```

- ✓ Ein einzelnes Kontrollkästchen legen Sie mit `<input type="checkbox">` fest.
- ✓ Neben dem Namen der Gruppe können Sie den Übergabewert festlegen.
- ✓ Ob beim Laden der Webseite das Kontrollkästchen bereits gewählt sein soll, entscheiden Sie mit dem Attribut `checked`.

Üblicherweise werden Kontrollkästchen in Umfragen eingesetzt, bei denen zu jeweils einer bestimmten Frage eine Auswahl getroffen werden soll. Mehrere Antwortmöglichkeiten zu einer Frage werden in einer Gruppe zusammengefasst. Um mehrere Kontrollkästchen zu einer Gruppe zusammenzufassen, geben Sie diesen Kontrollkästchen denselben Wert für das Attribut `name`.

## Optionsfelder erstellen

Mit Optionsfeldern, auch **Radiobuttons** genannt, erstellen Sie Gruppen von **Auswahlmöglichkeiten**, aus denen der Anwender nur jeweils eine Option auswählen kann.

```
<input type="radio" name="Gruppenname" value="Übergabewert" checked>
```

Hierzu notieren Sie die Angabe `type="radio"`. Auch hier werden Elemente mit demselben Namen zu einer Gruppe zusammengefasst, indem den jeweiligen Elementen derselbe Name gegeben wird.

### Beispiel: Formulare mit Kontrollkästchen und Optionsfeldern erstellen

In der Beispieldatei `10_Formulare_erstellen/formular5.html` wird jeweils eine Gruppe von Kontrollkästchen und eine Gruppe von Optionsfeldern erstellt.

```
<!DOCTYPE html>
<html lang="de">
  <head>
    <title>Kontrollkästchen und Optionsfelder</title>
  </head>
  <body>
    <form action="mailto:mail@example.com" method="POST"
enctype="text/plain">
      <fieldset>
        <legend>
          Sie möchten Ihr Fahrrad mit folgender Ausstattung:
        </legend>
        ① <label><input type="checkbox" name="Ausstattung"
           checked value="27Gang">27-Gang-Schaltung</label>
        ② <label><input type="checkbox" name="Ausstattung"
           value="Gelsattel">Gelsattel</label>
          <label><input type="checkbox" name="Ausstattung"
           value="Staender">Ständer</label>
        </fieldset>
        <fieldset>
          <legend>
            Ihr Fahrrad soll folgende Farbe haben:
          </legend>
          ③ <label><input type="radio" name="Farbe" value="Rot"
             checked> Rot
            <label><input type="radio" name="Farbe" value="Blau">
              Blau
            <label><input type="radio" name="Farbe"
              value="Schwarz"> Schwarz</label>
          </fieldset>
          <p><button type="submit" name="Abschicken"
            value="Abschicken"></p>
        ④ </form>
      </body>
    </html>
```

- ① Nach der Definition des Formulars wird mit `input type="checkbox"` das erste Kontrollkästchen angelegt. Es bekommt den Namen `Ausstattung` und ist beim Laden der Webseite bereits ausgewählt (`checked`). Als Übergabewert wird `27Gang` festgelegt.
- ② Das zweite Kontrollkästchen mit dem Wert `Gelsattel` erhält ebenfalls den Namen `Ausstattung`. Somit gehören das erste und das zweite Element zu der Gruppe `Ausstattung`. Folglich gehört damit auch das dritte Kästchen `Staender` zur Gruppe.
- ③ Für die Auswahl der Farbe werden Optionsfelder `type="radio"` genutzt. Diese drei bilden aufgrund desselben Namens `Farbe` eine Gruppe, bei der nur eine der angegebenen Optionen gewählt werden kann.
- ④ Notieren Sie `<button type="reset">` wird im Formular eine zusätzliche Schaltfläche integriert. Diese ermöglicht dem Nutzer, die bisher getätigte Auswahl auf den Ausgangswert zurückzusetzen. Aus Gründen der Nutzerfreundlichkeit sollten Sie auf die Verwendung dieser Schaltfläche verzichten. Ein Formular nicht abzusenden hat denselben Effekt wie ein Klick auf den Reset-Button. **Allerdings löscht ein versehentlicher Klick auf diesen Button alle bisher getätigten Eingaben.**

Wünscht der Kunde ein rotes Fahrrad mit 27 Gängen und einem bequemen Gelsattel, erhält der Empfänger z. B. eine E-Mail mit folgendem Inhalt:

```
Ausstattung=27Gang
Ausstattung=Gelsattel
Farbe=Rot
Abschicken=Abschicken
```

## 10.5 Schaltflächen mit Grafiken und JavaScript

### Grafiken in Schaltflächen einbinden

Neben den beiden Schaltflächen `submit` und `reset` gibt es noch die Schaltfläche vom Typ `image`. Mit dieser können Sie eine grafische Schaltfläche erzeugen, mit der die Daten eines Formulars abgesendet werden, vgl. `10_Formulare_erstellen\formular5.html`.

```
<input type="image" name="Name"
      src="URL der Grafikdatei">
```

Wie immer ist der Einsatz von Grafiken mit den üblichen Nachteilen verbunden (Textalternativen müssen bereitgestellt werden, schlechte Skalierbarkeit, hoher Umsetzungsaufwand, schlechte Wartbarkeit usw.)

Mehrere zusammengehörige Radiobuttons oder Checkboxes sollten Sie mittels `fieldset` und `legend` gruppieren und beschriften.

Sie möchten Ihr Fahrrad mit folgender Ausstattung:

27-Gang-Schaltung  
 Gelsattel  
 Ständer

Ihr Fahrrad soll folgende Farbe haben:

Rot  
 Blau  
 Schwarz

 **Bestellung absenden**

Grafik zum Absenden des Formulars („formular6.html“)

<pre> &lt;fieldset&gt;   &lt;legend&gt;Sie möchten ihr Fahrrad mit folgender Ausstattung:&lt;/legend&gt;   &lt;input type="checkbox" name="Ausstattung" checked value="27Gang" id="27Gang"&gt;&lt;label for="27Gang"&gt;27-Gang- Schaltung&lt;/label&gt;   [...] &lt;/fieldset&gt; &lt;fieldset&gt;   &lt;legend&gt;Ihr Fahrrad soll folgende Farbe haben:&lt;/legend&gt;   &lt;input type="radio" name="Farbe" value="rot" id="rot" checked&gt;   &lt;label for="rot"&gt;rot&lt;/label&gt;   [...] &lt;/fieldset&gt; &lt;p&gt;   &lt;input type="image" name="Abschicken" src="img/grafbutt.gif"&gt; &lt;/p&gt; </pre>		<div style="border: 1px solid black; padding: 5px;"> <h3 style="margin: 0;">Hotel Vallora</h3> <p style="margin: 0;">Sie möchten ihr Fahrrad mit folgender Ausstattung:</p> <div style="border: 1px solid gray; padding: 5px;"> <input checked="" type="checkbox"/> 27-Gang-Schaltung  <input type="checkbox"/> Gelsattel  <input type="checkbox"/> Ständer </div> <p style="margin: 0;">Ihr Fahrrad soll folgende Farbe haben:</p> <div style="border: 1px solid gray; padding: 5px;"> <input checked="" type="radio"/> rot  <input type="radio"/> blau  <input type="radio"/> schwarz </div> <div style="text-align: right; margin-top: 10px;">  <input type="button" value="Bestellung absenden"/> </div> </div>
---	---	--

formular6-fieldset.html

## Schaltflächen mit JavaScript steuern

Eine weitere Schaltfläche zum Klicken erhalten Sie durch die Angabe von `type="button"`, mit der Sie JavaScript-gesteuerte Anweisungen ausführen können.

```

<button id="Bestellbutton"
type="button" value="Bestellung
absenden" onclick="alert('Danke für
Ihre Bestellung!');">Bestellung
</button>

```

JavaScript ist eine Skriptsprache, die für den Einsatz im Web entwickelt wurde. Mit JavaScript ist es möglich, Inhalte zu generieren und nachzuladen. Somit können Sie dynamische Inhalte und Interaktivität in Ihre Webseiten einbinden.

Zum Erlernen von JavaScript können Sie das HERDT-Buch *JavaScript 1.8 – Grundlagen* verwenden.

Sie möchten Ihr Fahrrad mit folgender Ausstattung:

27-G  
 Gels  
 Stän

Ihr Fahr

Rot  
 Blau  
 Schwarz

Danke für Ihre Bestellung!

Schaltfläche mit JavaScript-Anweisung („formular7.html“)

## 10.6 Mehr Attribute für Formular-Elemente verwenden

HTML5 erlaubt Ihnen, das Verhalten von Formularen zu beeinflussen und Komfort für den Nutzer bereitzustellen.

### Plausibilitätsprüfung verhindern

Wenn Sie dem Browser verbieten möchten, zu überprüfen, ob Pflichtfelder ausgefüllt und Felder vom Typ `number`, `date` usw. plausibel befüllt wurden, notieren Sie im `input`-Element zum Absenden der Daten `formnovalidate`:

```
<button type="submit" formnovalidate>Absenden</button>
```

### Formularfeld automatisch fokussieren

Um in ein Feld schreiben zu können, muss der Nutzer Ihrer Webseite in dieses Feld mit der Maus klicken. In alle weiteren Felder kann er mittels Druck auf die Tab-Taste gelangen. Verwenden Sie `autofocus`, um den Cursor sofort beim Laden der Seite ins erste Feld zu setzen. Die Besucher Ihrer Webseite können sofort mit der Eingabe beginnen.

```
<label>Vorname: <input name="Vorname" autofocus></label>
```

### Formularfeld mit Beispieltext versehen

Verwenden Sie das Attribut `placeholder`, um in Formularen Texte anzuzeigen, die verschwinden, sobald der Besucher Ihrer Webseite eigene Daten eingibt. So können Sie mit Beispielen verständlich machen, welche Daten Sie erwarten, vgl. Beispieldatei *10\_Formulare erstellen/placeholder.html*.

```
<label>Vorname: <input name="Vorname"  
placeholder="Max" ></label>
```

Vorname:

### Pflichtfelder festlegen

Geben Sie einem Feld das Attribut `required`, um ein Befüllen des Feldes durch den Nutzer zu erzwingen. Moderne Browser werden das Formular nicht abschicken, bevor nicht alle Pflichtfelder plausibel befüllt sind. Werden Pflichtfelder nicht oder nicht plausibel ausgefüllt, erhält der Nutzer eine erläuternde Fehlermeldung, was von ihm erwartet wird.

```
<label>Vorname: <input name="Vorname" required></label>
```

## Automatische Vervollständigung verbieten

Moderne Browser erleichtern das Ausfüllen von Formularen, indem Sie anbieten, Werte, die der Nutzer früher bereits in ein Formularfeld geschrieben hat, in weiteren Formularen automatisch einzufügen.

Wenn Sie beispielsweise einmal in einem Formular "Martina Mustermann" als Name angegeben haben, bieten Ihnen Browser an, in anderen Formularen ein Feld mit der Bezeichnung "Name" ebenfalls mit "Martina Mustermann" zu befüllen.

Unter anderem aus Sicherheitsgründen kann dieses Verhalten unerwünscht sein. Mit `autocomplete="off"` können Sie Browsern mitteilen, dass die Autovervollständigung abgeschaltet werden soll.

Der Standard-Wert für dieses Attribut lautet `on`.

```
<label>Vorname: <input name="Vorname" autocomplete="off"></label>
```

## Beschreiben von Formularfeldern verhindern

Mit Vergabe des Attributes `readonly`, verhindern Sie, dass der Besucher Ihrer Webseite ein Feld mit eigenen Angaben befüllt. Im Gegensatz zu Feldern, die mittels `type="hidden"` verborgen wurden und heimlich Daten versenden, können Sie so Werte mitgeben und Transparenz beim Nutzer schaffen.

```
<label>Vorname: <input type="text" name="Vorname" readonly></label>
```

## Formularfelder schützen

Notieren Sie `disabled` im öffnenden Tag, um ein Feld unclickbar und unbeschreibbar zu machen. Sinnvoll ist das beispielsweise, wenn erst ein Häkchen in einem anderen Feld gesetzt werden muss, um ein JavaScript zu veranlassen, das Attribut zu entfernen und somit das Element bedienbar zu machen.

```
<label>Vorname: <input name="Vorname" disabled></label>
```

## Rechtschreibkorrektur abschalten

Besonders ärgerlich ist es, wenn die Autokorrektur beim Ausfüllen von Formularfeldern über-eifrig ist. Bei Namen von Personen und Orten und manch anderem sind die Vorschläge wenig hilfreich. Mit `spellcheck="false"` (`autocorrect="off"` für iOS) lässt sich die Autokorrektur abschalten.

Für iOS können Sie mittels `autocapitalize="off"` außerdem die automatische Groß-schreibung in Formularfeldern unterbinden (kein HTML5-Standard).

```
<label>Vorname: <input name="Vorname" spellcheck="false"
autocorrect="off"></label>
```

## Werte-Bereiche definieren

Für Felder, in denen Zahlen erwartet werden, können Sie Minimal- und Maximal-Werte festlegen. Werte, die sich nicht im erlaubten Bereich befinden, bestehen die Plausibilitätsprüfung nicht und führen zu einer entsprechenden Fehlermeldung.

Die folgende Altersabfrage erlaubt nur die Angabe von Zahlen zwischen 18 und 88:

```
<label>Alter: <input type="number" min="18" max="88"></label>
```

## 10.7 Übung

### Ein Formular erstellen

Level		Zeit	ca. 40 min
Übungsinhalte	✓ Formularelemente in einem Formular erstellen		
Übungsdatei	hotel10		
Ergebnisdatei	hotel10-e		

- Öffnen Sie die Webseite *newsetter.html* im Übungsordner *hotel10/info* und erstellen Sie ein Formular zur Anmeldung zu einem Newsletter, vgl. Abbildung.

### Anmeldung zum Newsletter des Hotels Vallora

Vorname:

Nachname:

E-Mail-Adresse:

Wie sind Sie auf unser Hotel aufmerksam geworden?

Freunde  Zeitschrift  
 Internet  Werbung

Wann haben Sie Ihren nächsten Urlaub geplant?

in ca. 1 - 3 Monaten  
 in ca. 6 Monaten  
 in diesem Jahr nicht mehr

Wenn Sie uns etwas mitteilen möchten, so finden Sie hier Platz dafür:

⋮

[Zurück zur Startseite](#)

Ergebnisdatei „newsetter.html“

# 11

## Webstandards und Codeoptimierung

### 11.1 HTML-Versionen

#### HTML-Versionen kennenlernen

HTML existiert seit 1990. Das W3C (World Wide Web Consortium, <http://www.w3.org/>) entscheidet, welche Richtung die Entwicklung von HTML nimmt, sowie welche neuen Funktionalitäten in die HTML-Spezifikationen aufgenommen werden.

Die aktuelle Entwicklung können Sie auf folgender Seite verfolgen (engl.):

<http://www.w3.org/html/wg/drafts/html/master/Overview.html>

Viele interessante Anleitungen und Dokumentationen werden auf Englisch veröffentlicht. Auch die Standards sind alle auf Englisch verfasst, offizielle Übersetzungen (wie die der WCAG 2.0) sind selten. Allerdings sind Online-Übersetzungsdienste inzwischen so gut, dass sie Ihnen das Verständnis englischsprachiger Dokumente erleichtern. Nutzen Sie daher die in diesem Buch empfohlenen Quellen auch dann, wenn Sie mit der englischen Sprache nicht sehr vertraut sind.

Elemente aus früheren Spezifikationen werden nach wie vor von den Browsern dargestellt. Es existieren auch noch viele ältere Seiten, die in HTML 4.01 und XHTML 1.0 oder früher geschrieben wurden. Deshalb ist es wichtig, die Unterschiede zwischen diesen Versionen und HTML5 zu kennen.

<b>HTML 4.01</b>	Letzte HTML-Version im traditionellen Sinn. Bis hierher fand eine echte Versionierung statt. Das bedeutet, es wurden neue Features für die Sprache vorgeschlagen, öffentlich diskutiert und zuletzt verabschiedet. Dieser Prozess war sehr zeitaufwändig. Kurzfristige Entwicklungen, wie sie dem Wesen des World Wide Web entsprechen, konnten nicht aufgegriffen werden. Eine ganze Reihe von zusätzlichen Technologien wie Flash, Java Applets, Real-Audio, Silverlight und viele andere wurden entwickelt, um Webseiten immer reicher an Funktionen und Designmöglichkeiten werden zu lassen.
------------------	---

<b>XHTML</b>	Diese Version ist eine strenge Variante von HTML gemäß XML (eXtensible Markup Language). Das X in XHTML (Extensible HTML) und XML steht für „erweiterbar“. Erweiterbar bedeutet, dass Entwickler bei Bedarf eigene Elemente festlegen können. Dies ist besonders für Webdatenbankanwendungen nützlich, da Webseitenentwickler eigene Tags für Datenfelder definieren können.  In der Praxis wurde XHTML aber in aller Regel identisch zu HTML 4.01 eingesetzt und bot daher keine Vorteile. Eine echte Weiterentwicklung sollte XHTML 2.0 sein, das allerdings nie verabschiedet wurde.
<b>HTML5</b>	HTML5 ist keine weitere Version von HTML, sondern der Nachfolger von HTML 4.01. Das Ziel von HTML5 ist es, weitere Versionen unnötig zu machen. Daher fehlt zwischen „HTML“ und der „5“ auch ein Leerzeichen. Zukünftig soll HTML schrittweise weiterentwickelt werden. Browserhersteller können neue Funktionen vorschlagen und diese werden häufig auch von anderen Herstellern übernommen, wenn sich abzeichnet, dass die Funktion von den Entwicklern angenommen wird und Webseiten für Nutzer besser macht.  HTML5 integriert vieles, wofür zuvor eigene Sprachen nötig waren. Dazu gehören Multimedia-Elemente, Zeichenflächen für Animationen, offline-Speichermöglichkeiten u.v.a.m. Neue Funktionen kommen ständig hinzu.

## 11.2 Standardisiertes HTML schreiben

### Warum sind Webstandards wichtig?

Sie wenden viel Arbeit auf, damit Ihre Webseiten die gewünschten Inhalte anzeigen und optisch ansprechend aussehen. Sie wollen natürlich, dass Besucher Ihre Webseiten so angezeigt bekommen, wie Sie es vorgesehen haben. Hier spielen Webstandards eine große Rolle.

Die Idee hinter Webstandards ist einfach: Wenn alle Browser- und Webseitenentwickler die gleichen Standards einhalten, werden Webseiten von allen Browsern gleich dargestellt. Die im Web eingesetzten Standards werden vom W3C veröffentlicht. Die Einhaltung dieser Standards ist empfehlenswert, da Sie hierdurch ein Höchstmaß an Zuverlässigkeit erreichen.

Ein wichtiger Bestandteil dieser Standards ist die Einhaltung von Spezifikationen für die Barrierefreiheit. Hierdurch gewährleisten Sie, dass Ihre Webinhalte für alle Webbenutzer unabhängig von körperlichen oder technischen Möglichkeiten zugänglich sind. Einen guten Einstieg in barrierefreies Webdesign (auch Web Accessibility genannt) finden Sie auf der folgenden Webseite:

<http://www.barrierefreies-webdesign.de>.

## Welche Standards begegnen Ihnen im Internet?

HTML5 ist der aktuelle HTML-Standard und damit vergleichsweise neu. Den ersten Vorschlag für die Textauszeichnungssprache machte Tim Berners-Lee am 13. März 1989 am CERN (Europäische Organisation für Kernforschung) in Genf. Es sollte eine Möglichkeit geschaffen werden, um wissenschaftliche Ergebnisse nicht nur festzuhalten (das passierte ohnehin), sondern auch zu teilen. Das CERN hatte bereits Standorte in Frankreich und in der Schweiz und von beiden aus sollten die Dokumente zugänglich sein.

Die erste Version hieß HTML und kam noch ohne Versionsnummer aus. Wichtige Standards, die häufig über einen längeren Zeitraum benutzt wurden, waren die Version 3.2, Version 4.01 und XHTML 1.0.

Webseiten bleiben oft – im wesentlichen Kern – unverändert über mehrere Jahre im Netz. Daher werden Sie immer wieder Webseiten sehen, die noch in HTML4.01 oder XHTML 1.0 geschrieben wurden.

Auch nach der Veröffentlichung eines solchen Standards hat es immer noch eine Weile gedauert, bis sich alle Entwickler an die neuen Standards gewöhnt und diese eingesetzt haben.

Für neue Projekte sollten Sie die veralteten Standards nicht verwenden.

Ein wichtiger Unterschied zwischen HTML5 und früheren Versionen besteht darin, dass erst in HTML5 Inline-Elemente wie Links direkt im `body`-Element erlaubt sind. In früheren Versionen waren nur Blockelemente als Kinder von `body` erlaubt.

Die folgende Tabelle listet die **Gemeinsamkeiten der bisherigen Standards** auf, an die Sie sich also unabhängig von der verwendeten Version halten sollten.

Die Dokumenttyp-Deklaration muss oben im Dokument stehen.	Mit dieser Codezeile teilen Sie dem Browser mit, welche HTML-Version Sie verwenden. Bei HTML5 und HTML 4 muss die Dokumententyp-Deklaration direkt am Dokumentanfang stehen und bei XHTML muss diese als zweite Codezeile im Dokument angegeben werden.
Direkt nach der Dokumenttyp-Deklaration steht das <code>&lt;html&gt;</code> -Tag und am Dokumentende steht das <code>&lt;/html&gt;</code> -Tag.	Das Tag <code>&lt;html&gt;</code> kommt direkt unter dem <code>&lt;!DOCTYPE&gt;</code> . Das <code>&lt;/html&gt;</code> -Tag muss am Ende der Webseite eingetragen werden. Alle anderen Elemente befinden sich innerhalb dieser zwei Tags.
Das Root-Element ( <code>html</code> ) hat nur zwei Kinder: <code>head</code> und <code>body</code> .	Diese Tags werden als einzige Tags direkt innerhalb der <code>&lt;html&gt;</code> -Tags notiert. Alle anderen Elemente müssen innerhalb der <code>&lt;head&gt;</code> - bzw. <code>&lt;body&gt;</code> -Tags platziert werden.
Seitentitel eingeben	Sie müssen im <code>head</code> -Bereich einen Seitentitel mit dem <code>title</code> -Element festlegen.

In der Regel dürfen Blockelemente nicht in Inline-Elementen vorkommen.	Ein <code>&lt;p&gt;</code> -Tag darf z. B. nicht in einem <code>q</code> -Element vorkommen. Es gibt aber Ausnahmen. In HTML5 dürfen z. B. Links (Inline-Element) komplexe Strukturen inklusive Blockelementen beinhalten.
Keine Blockelemente in <code>&lt;p&gt;</code> -Tags schreiben	In Absätzen dürfen nur Text sowie Inline- und Inline-Block-Elemente vorkommen.
<code>li</code> -Elemente nur in <code>ul</code> und <code>ol</code> verwenden. Listen haben keine Kinder außer <code>li</code> .	Das <code>&lt;li&gt;</code> -Tag muss Kind von ungeordneten Listen <code>&lt;ul&gt;</code> oder geordneten Listen <code>&lt;ol&gt;</code> sein; kann selbst jedoch nahezu alle anderen Elemente enthalten.
Den Text eines <code>blockquote</code> korrekt auszeichnen	Auch Texte im <code>blockquote</code> -Element sollten Sie auszeichnen, normalen Fließtext also in einen Absatz ( <code>p</code> ) setzen, Überschriften in <code>h</code> x usw.

## Beispielwebauftritt standardgemäß anpassen

In der Beispieldatei `11_Webstandards und Codeoptimierung/hotel/index.html` müssen Sie noch einige Änderung vornehmen, um die Standards aus der obigen Tabelle einzuhalten.

- ▶ Die neue Art eine Zeichenkodierung anzugeben, gilt erst seit HTML5. Für ältere Versionen des HTML-Standards muss die Zeichenkodierung wie folgt angegeben werden:

```
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
```

- ▶ Schließen Sie den Text des Blockquotes in die `<p>` und `</p>` Tags ein.

```
<blockquote><p>Gönnen Sie sich  
Erholung und Entspannung  
in unserer Oase des Wohlfühlens  
im landschaftlichen Idyll.</p></blockquote>
```

- ▶ Setzen Sie die Links in ein `p`-Element, speichern Sie die Seite und testen Sie sie im Browser.

```
<p><a href="info/wegweiser.html">Wegbeschreibung</a></p>  
</body>
```

- ▶ Vergleichen Sie Ihre Datei mit den Beispieldateien `11_Webstandards und Codeoptimierung/hotel/html5_standardkonform.html` und `11_Webstandards und Codeoptimierung/hotel/html4_standardkonform.html`.

## 11.3 HTML 4 und XHTML verstehen

### Dokumenttyp-Deklarationen verstehen

**Verwenden Sie für neue Seiten immer die aktuellste Version des HTML-Standards!**

Wenn Sie Ihre Webseite nicht selber erstellen, sondern mit älteren Seiten arbeiten müssen, kann es sein, dass diese nicht dem aktuellen Standard HTML5 entsprechen. Hier erfahren Sie mehr über die Besonderheiten älterer Standards.

Die Dokumenttyp-Deklarationen (DTD) für HTML 4 und XHTML sind deutlich komplexer als für HTML5. Daher empfiehlt es sich, diese nicht immer wieder abzutippen, sondern diese aus bestehenden Dokumenten zu übernehmen.

Das selfhtml-Wiki bietet dafür geeignete Kopiervorlagen unter [http://wiki.selfhtml.org/wiki/HTML/Dokumentstruktur\\_und\\_Aufbau](http://wiki.selfhtml.org/wiki/HTML/Dokumentstruktur_und_Aufbau)

Die letzte Version von HTML 4 ist HTML 4.01 – XHTML 1.0 besitzt denselben Funktionsumfang wie HTML 4.01 und sollte zu XHTML 2.0 weiter entwickelt werden. Dazu kam es jedoch nie. Stattdessen wurde HTML5 im Jahr 2014 zur Empfehlung (Recommendation), wie es in der offiziellen Sprachregelung heißt.

Die drei verschiedenen Varianten Strict, Transitional und Frameset sind in der folgenden Tabelle beschrieben. Da die Bedeutungen für HTML 4.01 und XHTML identisch sind, werden die Beschreibungen zusammengefasst.

<b>Strict</b>	<p>HTML 4.01: <code>&lt;!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd"&gt;</code></p> <p>XHTML 1.0: <code>&lt;!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict //EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd"&gt;</code></p> <p>Die Variante Strict (engl. für <i>streng</i>) erlaubt weder Elemente noch Attribute, die vom W3C als <i>deprecated</i> (unerwünscht) eingestuft sind. Auch Framesets (Framesseiten) sind mit dieser Variante nicht möglich.</p>
<b>Transitional</b>	<p>HTML 4.01: <code>&lt;!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd"&gt;</code></p> <p>XHTML 1.0: <code>&lt;!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd"&gt;</code></p> <p>Verwenden Sie die Variante Transitional (engl. für Übergang), wenn Sie Elemente verwenden müssen, die in der Strict-Variante nicht erlaubt sind.</p>

<b>Frameset</b>	<p>HTML 4.01: <code>&lt;!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset/ /EN" "http://www.w3.org/TR/html4/frameset.dtd"&gt;</code></p> <p>XHTML 1.0: <code>&lt;!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Frameset/ /EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd"&gt;</code></p> <p>Die Transitional- bzw. Übergangsvariante wird um die Frame-Definition ergänzt. Diese Variante wird nur für Webauftritte mit Frames benötigt. Frames erlaubte es, eine Seite aus mehreren Dokumenten zusammzusetzen. Typischerweise wurden für unveränderliche Teile wie Seitenkopf oder Fußbereich HTML-Dokumente eingebunden, um sich so das Mitsenden dieser Daten für alle Einzelseiten zu sparen. Allerdings gibt es in modernen Seiten kaum noch unveränderliche Bereiche. Abhängig von Kontext und Zweck einer Einzelseite werden alle Bereiche angepasst.</p> <p>Frames haben zudem große Nachteile unter anderem für Menschen mit Behinderungen oder die Suchmaschinenoptimierung. Die Technik war nicht robust. Immer wieder wurde auf einzelne Seitenbereiche verlinkt, sodass die in Frames ausgelagerten Bereiche wie Kopfbereich oder sogar Navigation nicht zur Verfügung standen.</p> <p>Aufgrund der zahlreichen und schwerwiegenden Nachteile und dem Wegfall der wenigen Vorteile gibt es in HTML5 keine Frames mehr.</p>
-----------------	--

- ✓ Die Dokumenttyp-Deklaration können Sie in einer Zeile schreiben oder mit einem Zeilenumbruch eingeben. Achten Sie dabei darauf, dass Sie keinen Zeilenumbruch in einem Teil mit Anführungszeichen eingeben.
- ✓ Eine URL wird in der Dokumenttyp-Deklaration angegeben. Hierbei handelt es sich jeweils um das W3C-Dokument mit den Regeln zur entsprechenden HTML-4.01-Variante. Mithilfe dieser URL kann der Browser die damit verbundenen Regeln auslesen. Da die Browser diese Information bereits beinhalten, werden diese Dokumente jedoch nicht jedes Mal geladen. Die Angabe der URL kann deshalb entfallen.
- ✓ Alle Dokumenttyp-Deklarationen stehen Ihnen in der Beispieldatei *standards.html* zur Verfügung. Am schnellsten geht es, wenn Sie die benötigten Zeilen aus dieser Beispieldatei kopieren und in Ihre Webseiten einfügen.

## Zeichencodierung und Inhaltstyp angeben

Wie Sie bereits aus Kapitel 3 wissen, legen Sie in HTML5 mit einem `<meta>`-Tag fest, dass Sie Sonderzeichen wie Umlaute im HTML-Dokument verwenden. Diese Angabe müssen Sie als erstes Element im `<head>`-Bereich eingeben.

```
<meta charset="utf-8">
```

*Die Zeichencodierung für HTML5 festlegen*

Für HTML 4 und XHTML müssen Sie zusätzlich den Inhaltstyp angeben. Achten Sie bei der Eingabe darauf, dass Sie keinen Zeilenumbruch in einem Teil mit Anführungszeichen einfügen. Diese Zeile steht auch in der Beispieldatei *11\_Webstandards und Codeoptimierung/standards.html* zum Kopieren zur Verfügung.

```
<meta http-equiv="Content-Type" content="text/html; charset="utf-8">
```

Inhaltstyp und die Zeichencodierung für HTML 4 und XHTML

## Strengere Schreibregeln in XHTML

XHTML basiert auf XML. **Der wichtigste Unterschied zu HTML besteht darin, dass alle Elemente geschlossen werden müssen – auch leere!**

Normale Elemente schließen Sie wie gewohnt mit einem schließenden Tag.

```
<p>Text</p>
```

Leere Elemente schließen Sie entweder mit einem Schrägstrich im öffnenden Tag **oder** mit einem schließenden Tag. Beides ist erlaubt, die erste Schreibweise hat sich aber durchgesetzt, weil sie kürzer ist.

```
<br />
```

oder

```
<br></br>
```

## XHTML-Regeln kennenlernen

Die Regeln, die Sie sich für XHTML merken müssen, sind in der nachfolgenden Tabelle erläutert. Den HTML-Code finden Sie zusätzlich in der Beispieldatei *11\_Webstandards und Codeoptimierung/hotel/xhtml\_standardkonform.html*.

```
① <?xml version="1.0" encoding="utf-8"?>
② <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict
//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
③ <html xmlns="http://www.w3.org/1999/xhtml" lang="de"
xml:lang="de">
④ <head>
    <meta http-equiv="Content-Type" content="text/html;
charset=utf-8">
    <title>Hotel Vallora</title>
    <link rel="stylesheet" type="text/css" href="stylesheets/
vallorastyles.css" />
⑤ </head>
    <body>
        <h1>Hotel Vallora</h1>
```

```

⑥ <p></p>
    <p>Bei Ihrem nächsten Aufenthalt in unserer Gegend würden
    wir Sie gern persönlich begrüßen und mit unserem freundlichen
    und kompetenten Service verwöhnen.</p>
    <p>Tel. 000 000 000<br />
    Fax 000 000 000<br />
    E-Mail: info@hotelvallora.com</p>
</body>
</html>

```

- ① Der Bezug zu XML und die Zeichencodierung müssen am Dokumentanfang stehen.
- ② Als nächste Codezeile kommt die Dokumenttyp-Deklaration. Hier **muss** die Angabe `html` kleingeschrieben werden.
- ③ Das Starttag des `html`-Elements wird erweitert. Über die Attribute `lang` und `xml:lang` geben Sie an, dass das Dokument auf Deutsch geschrieben ist.
- ④ Elementnamen **müssen** kleingeschrieben werden. (In HTML 4 sowie HTML5 ist die Kleinschreibung optional.)
- ⑤ Leere Elemente **müssen** geschlossen werden (hier mit einem Schrägstrich `/` vor der schließenden spitzen Klammer). Es wird empfohlen, einen Leerschritt zwischen den Namen des Elements und den Schrägstrich einzufügen.
- ⑥ Attributwerte **müssen** in (doppelten oder einfachen) Anführungszeichen angegeben werden.

XHTML-Dateien werden ebenfalls mit der Dateinamenserweiterung `.htm` und `.html` gespeichert.

- ✓ Obwohl es möglich ist, XHTML-Dateien mit der Dateinamenserweiterung `.xhtm` bzw. `.xhtml` zu versehen, werden üblicherweise die Dateinamenserweiterungen `.htm` bzw. `.html` verwendet. An der Dokumenttyp-Deklaration im `<head>`-Bereich erkennt der Browser, dass es sich um eine XHTML-Datei handelt.
- ✓ Eine ausführliche Beschreibung der Unterschiede zwischen HTML und XHTML finden Sie auf der folgenden Webseite: <http://wiki.selfhtml.org/wiki/Doku:HTML/Unterschiede>.

## 11.4 Code validieren

### Den Code Ihrer Webseite vom W3C-Validator überprüfen lassen

Das W3C stellt einen Validator (Überprüfungstool) zur Verfügung, mit dem Sie den Code der Webseiten gemäß der angegebenen Dokumenttyp-Deklaration (DTD) überprüfen können. Falls der Validierer Fehler findet, werden diese aufgelistet. Die Webseite sowie die angezeigten Hinweise zur Fehlerbehebung sind in englischer Sprache.

- ▶ Öffnen Sie Ihren Browser und geben Sie die URL <https://validator.w3.org> ein.
- ▶ Klicken Sie auf das Register *Validate by File Upload* ①.  
Eine bereits im Web veröffentlichte Webseite können Sie über das Register ② (Validate by URI) prüfen. Über das Register ③ (Validate by direct input) können Sie den zu überprüfenden Code direkt im Browserfenster eingeben.

Webseite auf dem lokalen Rechner validieren

- ▶ Wechseln Sie über *Durchsuchen* ④ zum Speicherort der zu überprüfenden Datei, hier die Datei *11\_Webstandards und Codeoptimierung/hotel/index.html*.
- ▶ Betätigen Sie *Check*.  
Sie sollten die Meldung erhalten, dass der Validator Fehler auf der Webseite entdeckt hat.

Fehlermeldung

- ▶ Scrollen Sie im Browserfenster ggf. nach unten, um die Beschreibung möglicher weiterer Fehler zu lesen.

**In der Datei `index.html`, die Sie in diesem Kapitel standardkonform überarbeitet haben, haben Sie diesen Fehler jedoch behoben.**

- ▶ Testen Sie die angepasste Datei `index.html` bzw. die Datei `html5_standardkonform.html` (im Ordner *Beispieldateien*) mit dem Validator.

Jetzt sollte der Code die Prüfung bestehen.

**Document checking completed. No errors or warnings to show.**

*Meldung bei bestandener Prüfung. Das Dokument ist gültig (valide).*

Sie können genauso vorgehen, wenn Sie HTML 4 bzw. XHTML-Code überprüfen wollen. Anhand der Dokumenttyp-Definition erkennt der Validator automatisch, nach welchem Standard er prüfen soll. Zur Übungszwecken können Sie die Beispieldatei `11_Webstandards und Codevalidierung/hotel/index.html` den HTML 4- bzw. XHTML-Standards anpassen und validieren. Vergleichen Sie Ihre Ergebnisse mit den Beispieldateien `html4_strict.html` bzw. `xhtml_strict.html`.

### Arbeit erleichtern mit Browser-PlugIns

Viele typische Testaufgaben (auch das Validieren von Webseiten) erleichtern die Firefox-Erweiterungen „**Web Developer**“ und „**Firebug**“. Daneben bringen sowohl Firefox als auch alle anderen Browser sogenannte **Entwickler-Werkzeuge** mit, die Ihnen Testaufgaben und Fehlersuche erleichtern.



**Lernvideo:** *Entwicklerwerkzeuge.mp4*

**Beispieldatei:** *entwicklerwerkzeuge.html*

## 11.5 Webseiten in verschiedenen Browsern testen

### Übersicht über die gängigsten Browser

Da sich Browser unterscheiden, sollten Sie Ihre Webseiten in verschiedenen Browsern testen. Unterschiede treten besonders bei der Darstellung von CSS-Formatierungen und der Unterstützung neuer HTML5-Funktionen auf. Alle Browser können Sie kostenlos im Internet herunterladen. Die am häufigsten verwendeten Browser sind in der folgenden Tabelle aufgelistet.

Browser	Beschreibung
 <p data-bbox="245 504 430 533">Mozilla Firefox</p>	<p data-bbox="467 293 1367 506">Mozilla Firefox ist der zweithäufigste verwendete Browser in Deutschland und unter Webentwicklern sehr beliebt, weil er die Standards des W3C genau beachtet. Wenn Sie normgerechten Code schreiben, werden Ihre Webseiten in Firefox richtig dargestellt. Mozilla Firefox wurde als Open-Source-Projekt entwickelt und kann für Windows, Macintosh sowie Unix-Systeme kostenlos heruntergeladen werden.</p>
 <p data-bbox="240 710 434 739">Google Chrome</p>	<p data-bbox="467 557 1367 801">Google entwickelt seinen Browser Chrome mit hohem Tempo weiter und setzt aktuelle Neuerungen im HTML5-Standard oft früher um als andere Hersteller. Auch in anderer Hinsicht ist der Google-Browser technischer Vorreiter. So sind die eingebauten Entwickler-Werkzeuge die am weitesten entwickelten mit den meisten Möglichkeiten. Auch bei der Geschwindigkeit, mit der Webseiten dargestellt werden, setzt der Browser regelmäßig Maßstäbe.</p> <p data-bbox="467 815 1321 880">Die Live-Vorschau des Editors Brackets setzt einen installierten Chrome voraus.</p> <p data-bbox="467 896 1294 960">Chrome ist der weltweit meistverwendete Browser, wird aber häufig wegen der Datensammelei seines Herstellers kritisiert.</p>
 <p data-bbox="261 1182 413 1211">Apple Safari</p>	<p data-bbox="467 983 1362 1122">Apple Safari geht mit Webseiten ähnlich normgerecht um wie der Mozilla Firefox. Allerdings wird Safari als Bestandteil der Betriebssysteme MacOS und iOS in der Regel nur mit Betriebssystem-Updates aktualisiert. Die Update-Zyklen sind hier deutlich langsamer als bei den anderen Browsern.</p> <p data-bbox="467 1135 1174 1164">Dennoch ist die Unterstützung der Webstandards sehr gut.</p> <p data-bbox="467 1178 1326 1243">Aktuelle Features stellt das WebKit-Team zwar unter folgender Adresse bereit, das Angebot dürften aber die wenigsten Anwender nutzen:</p> <p data-bbox="467 1256 842 1285"><a href="https://webkit.org/downloads/">https://webkit.org/downloads/</a></p>
 <p data-bbox="233 1440 435 1469">Internet Explorer</p>	<p data-bbox="467 1301 1362 1440">Der Internet Explorer belegt den vierten Platz und wird mit dem Windows-Betriebssystem automatisch installiert. Die neueste Version ist Internet Explorer 11. Da viele Benutzer mit älteren Versionen dieses Browsers im Internet unterwegs sind, sollten Sie im Idealfall auch mit diesen testen.</p> <p data-bbox="467 1453 1326 1585">Microsoft bietet dafür virtuelle Maschinen zum Download an. Deutlich einfacher in der Handhabung, aber weniger zuverlässig sind die Emulationsmöglichkeiten des Internet Explorer 11, der versuchen kann, Webseiten so darzustellen, als würde er in einer älteren Version ausgeführt.</p> <p data-bbox="467 1599 1326 1664">Die Entwicklung des Internet Explorers wurde allerdings eingestellt. Der Nachfolger heißt Edge.</p>
 <p data-bbox="304 1843 368 1872">Edge</p>	<p data-bbox="467 1686 1367 1787">Der jüngste Browser ist Edge von Microsoft, der den Internet Explorer ersetzen soll. Er hat es aber bisher nicht geschafft, die Gunst der Anwender zu erlangen und fristet neben dem Opera ein Nischendasein.</p>
 <p data-bbox="300 2024 384 2054">Opera</p>	<p data-bbox="467 1895 1353 2033">Opera verwendet inzwischen dieselbe Rendering-Engine wie Chrome. Aufgrund dieser Tatsache und weil der Marktanteil von Opera in Deutschland sehr gering ist, müssen Tests in diesem Browser nicht den Schwerpunkt Ihrer Qualitätssicherung bilden.</p>

Eine Statistik der am häufigsten in Deutschland verwendeten Browser finden Sie auf der Webseite <http://www.browser-statistik.de>. Werten Sie unbedingt Ihre eigenen Statistiken aus. Die für Deutschland geltenden Mittelwerte können sich erheblich von denen für Ihre Webseite unterscheiden.

## 11.6 Effizient arbeiten, Vorhandenes nutzen

Webentwickler bilden sich im WWW fort, Sie verabreden sich zu Konferenzen und stellen Ihre eigenen Tools und Entwicklungen vor.

Viele Entwickler verdanken der Online-Community Hilfe in schwierigen Situationen und möchten dafür – oft kostenlos – etwas zurückgeben.

Zunächst werden Sie als Anfänger zu dem globalen Wissensschatz nicht viel beitragen können, **greifen Sie aber zu, wenn es um fertige Lösungen geht**. Diese nehmen Ihnen nicht nur viel Arbeit ab, Sie lernen auch, wenn Sie sich mit der Arbeit anderer auseinandersetzen.

### HTML5-Boilerplate verwenden

Unter <http://html5boilerplate.com> steht Ihnen ein komplettes Starterpaket für Ihre Webseite zur Verfügung. Alle unbedingt benötigten Dateien und Ordner sind bereits vorhanden, inklusive einer *index.html* mit HTML-Grundgerüst, Zeichensatz-Angabe und eingebundenen CSS-Dateien.

Die Idee hinter dieser Kochplatte ist: Wozu jedes Mal einen Herd bauen, wenn man sich ein Spiegelei braten möchte?

Viele nützliche Angaben werden gemacht, die Sie – sobald Sie diese verstehen – an Ihre eigenen Bedürfnisse anpassen können. Sie können die Boilerplate (Kochplatte) aber auch einsetzen, ohne etwas zu ändern.

HTML5-Boilerplate beinhaltet außerdem das Skript Modernizr, das ältere Browser für viele neue Elemente und CSS-Eigenschaften fit macht und Voraussetzung für weitere unterstützende Skripte ist. Die CSS-Datei *normalize.css* sorgt dafür, dass browser-spezifische Unterschiede entfernt („normalisiert“) werden.

**HTML5 ★ BOILERPLATE**  
SOURCE CODE DOCS OTHER PROJECTS

## The web's most popular front-end template

HTML5 Boilerplate helps you build **fast**, **robust**, and **adaptable** web apps or sites. Kick-start your project with the combined knowledge and effort of 100s of developers, all in one little package.

[Download v5.3.0](#) [Get a custom build](#)

[See the CHANGELOG](#)

### Save time. Create with confidence.

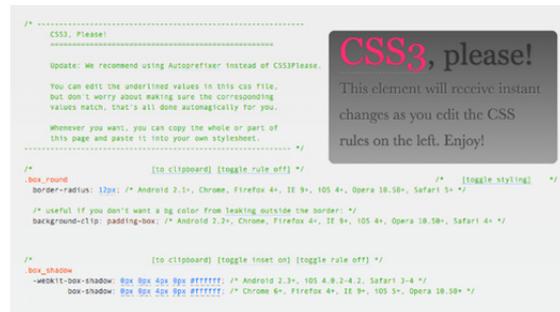
- ★ **Analytics, icons, and more**  
A lean, mobile-friendly HTML template; optimized Google Analytics snippet; placeholder touch-device icon; and docs covering dozens of extra tips and tricks.
- ★ **Normalize.css and helpers**  
Includes *Normalize.css* — a modern, HTML5-ready alternative to CSS resets — and further base styles, helpers, media queries, and print styles.
- ★ **JQuery and Modernizr**  
Get the latest minified versions of two best-of-breed libraries: *jQuery* (via Google's CDN, with local fallback) and the *Modernizr* feature detection library.
- ★ **High performance**  
Apache settings to help you deliver excellent site performance. We independently maintain *server configs*, and an *ant build script*.

## Online-Tools verwenden

### CSS3 bitte

Haben Sie noch Schwierigkeiten mit dem Schreiben von CSS? Gerade bei komplexen Regeln, in denen mehrere Browser unterschiedlich angesprochen werden wollen, hilft Ihnen <http://CSS3please.com>.

Viele komplexe Eigenschaften sind dort beispielhaft aufgeführt. Sie können Werte ändern und das Ergebnis per „copy und paste“ in die eigene CSS-Datei übernehmen. Vertipper sind auf diese Weise ebenso ausgeschlossen, wie fehlende Unterstützung älterer Browser.



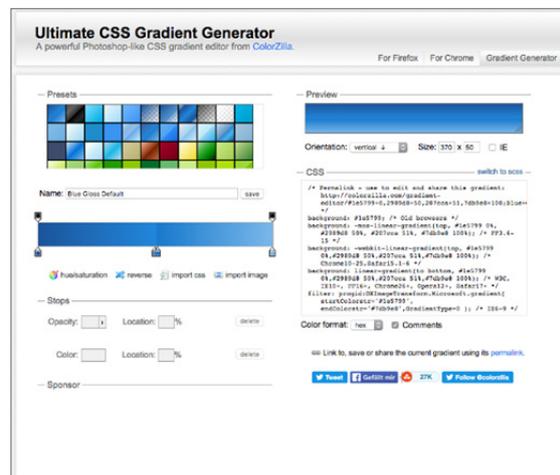
### Lorem ipsum

**Blindtext-Generatoren** helfen Ihnen dabei, Webseiten mit großen Textmengen zu befüllen, um so einen besseren Eindruck für das Layout zu bekommen. Suchen Sie einfach nach Blindtext-Generator oder „Lorem ipsum“.

### Aufwändige Verläufe

Sie möchten ein Element mit einem komplexen Verlauf versehen, sind aber (noch) nicht in der Lage, den Verlauf in CSS zu schreiben? Auch für Verläufe (engl.: gradient) gibt es Generatoren.

Einen empfehlenswerten Generator finden Sie unter <http://www.colorzilla.com/gradient-editor/> – hier können Sie auch Verläufe, die Sie in einem Grafikprogramm erstellt haben, als Bild hochladen. Das Online-Tool erstellt Ihnen das benötigte CSS für Ihre CSS-Datei automatisch.



### Google besser bedienen

Viele Informationen und Werkzeuge, um Ihre Webseite für Google aufzubereiten, finden Sie in den **Google Webmaster Tools** (<https://www.google.com/webmasters/tools/home?hl=de>). Dazu ist ein Google-Konto nötig.

Ähnliches bieten auch andere Suchmaschinenbetreiber an.

Videos für Webmaster finden Sie bei Youtube unter:

<https://www.youtube.com/user/GoogleWebmasterHelp>

### Tools in den Browser integrieren

Häufig benutzte Tools sollten Sie sich als **PlugIn** in Ihren Browser installieren. So sind wichtige Funktionen für Sie nur noch einen Mausklick weit entfernt.

Empfehlenswert ist der Werkzeugkasten des Mozilla Firefox (<https://addons.mozilla.org/DE/firefox/extensions/web-development/>). Aber auch für Google Chrome und andere Browser stehen viele PlugIns für die Webentwicklung zur Verfügung.

### In englischsprachigen Quellen suchen

Bei Problemen finden Sie häufig die Lösung im WWW. Oft kommen Sie mit englischen Suchbegriffen weiter, da viele Tools aus Gründen der Internationalität in englischer Sprache vorliegen. Die Bedienung erschließt sich meist auch mit geringen Englisch-Kenntnissen, passende Suchbegriffe in englischer Sprache liefern Ihnen **Online-Übersetzungsdienste**.

Wenn Sie das Gefundene nicht verstehen: Der Übersetzer von Google (<https://translate.google.de>) übersetzt Ihnen Webseiten ins Deutsche; im Google-eigenen Browser Chrome ist die Übersetzungsfunktion bereits eingebaut.

### Fertige Skripte nutzen

Neben HTML und CSS bestehen moderne Webseiten in der Regel auch aus einigen JavaScript-Dateien, mit denen beispielsweise Bildergalerien oder andere interaktive Elemente umgesetzt werden können. Auch horizontale Aufklappmenüs, wie Sie auf den meisten Seiten eingesetzt werden, erfordern JavaScript.

Ohne diese Sprache selber lernen zu müssen, können Sie fertige Skripte nutzen. In der Regel sind gute Skripte auch gut dokumentiert. Wenn Sie die Anleitungen der Skripte lesen, sind Sie mit den hier erworbenen HTML- und CSS-Kenntnissen in der Lage, fertige JavaScripte in Ihre Seiten einzubinden und entsprechend der Anleitungen zu konfigurieren oder mittels CSS eigene Farb- anpassungen durchzuführen.

Viele moderne Skripte sind sogenannte **jQuery-Plugins**, die voraussetzen, dass jQuery in Ihrer Seite eingebunden ist. Auch jQuery liefert Ihnen die HTML5-Boilerplate bereits mit.

Viele fertige Skripte gibt es auf **GitHub** (<https://github.com>) – Sie finden aber gewünschte Lösungen (z. B. JavaScript-Bildergalerie) auch leicht mit der Suchmaschine Ihrer Wahl.

## Informationenquellen im WWW

Da HTML5 ständig weiter entwickelt wird und sich diese Entwicklung auch bei den Browsern fortsetzt, können Sie auf der Suche nach aktuellsten Informationen bei Online-Informationenquellens fündig werden. In der nachfolgenden Tabelle finden Sie eine zum Zeitpunkt der Bucherstellung empfehlenswerte Auswahl:

Website	Beschreibung
<a href="https://selfhtml.org">https://selfhtml.org</a>	Deutschsprachiges Technik-Wiki mit umfangreichen Beispielen und Erläuterungen und einem angebundenes Forum für Fragen rund um alle Themen des Web
<a href="http://html5please.com">http://html5please.com</a>	Was lässt sich in welchen Browsern verwenden und wie rüstet man fehlende Funktionalitäten in Browsern nach?
<a href="http://www.w3schools.com">http://www.w3schools.com</a>	Englische Seite, auf der alle wichtigen Webtechniken erklärt werden. Alle HTML-Beispiele können selber ausprobiert werden, und Standard-Werte werden benannt.
<a href="http://html5doctor.com">http://html5doctor.com</a>	Englische Site, die alle Elemente sehr ausführlich erklärt, ähnliche Elemente voneinander abgrenzt und viele Tipps für die praktische Verwendung mitgibt.
<a href="http://www.w3.org/TR/html5">http://www.w3.org/TR/html5</a>	Die offizielle HTML5-Website
<a href="https://www.w3.org/Style/CSS/specs">https://www.w3.org/Style/CSS/specs</a>	Die offizielle CSS-Seite
<a href="http://webkrauts.de">http://webkrauts.de</a>	Deutsche Seite, die erklärt, wie man Webseiten besser macht.
<a href="http://www.barrierefreies-webdesign.de">http://www.barrierefreies-webdesign.de</a> <a href="http://www.einfach-fuer-alle.de">http://www.einfach-fuer-alle.de</a>	Auf den hier genannten Webseiten erfahren Sie, wie Sie Webseiten für alle zugänglich machen können. Man spricht auch von barrierefreiem Webdesign. Da sehr viele Menschen mehr oder weniger starke Beeinträchtigungen haben (Alterskurzsichtigkeit, Zittern, Rot-Grün-Blindheit), die die Nutzung von Webseiten erschweren können, erschließen Sie sich mit einer zugänglichen Webseite eine nicht unerhebliche Nutzergruppe.

## 11.7 Übungen

### Übung 1: Webseiten normgerecht anpassen und validieren

Level		Zeit	ca. 15 min
Übungsinhalte	<ul style="list-style-type: none"> <li>✓ Code-Standards einhalten</li> <li>✓ Code validieren</li> </ul>		
Übungsdatei	<i>hotel11</i>		
Ergebnisdatei	<i>hotel11-e</i>		

1. Öffnen Sie die Webseite *ueberuns-html4.html* im Übungsordner *hotel10/info* und passen Sie diese den Standards für HTML5 an.
2. Speichern Sie die Webseite und validieren Sie sie mit dem Validator (<http://validator.w3.org>).
3. Öffnen Sie die im Übungsordner *hotel11/restaurant* befindliche Webseite *speisekarte-fehler.html* und passen Sie diese gleichfalls an die Standards für HTML5 an.
4. Speichern Sie die Webseite und validieren Sie sie.

### Übung 2: HTML 4- und XHTML-Dokumente in HTML5-Dokumente umwandeln

Level		Zeit	ca. 15 min
Übungsinhalte	<ul style="list-style-type: none"> <li>✓ Code-Standards für HTML 4 und XHTML einhalten</li> <li>✓ Code validieren</li> </ul>		
Übungsdatei	<i>hotel11</i>		
Ergebnisdatei	<i>hotel11-e</i>		

1. Öffnen Sie die Webseite *info/ueberuns-html4.html* aus der vorherigen Übung und fügen Sie die Dokumenttyp-Deklaration für HTML 5 ein.
2. Fügen Sie zusätzlich die Codezeile mit der Zeichencodierung ein.
3. Speichern Sie die Webseite unter dem Namen *ueberuns\_html5.html* und validieren Sie sie mit dem Validator (<http://validator.w3.org>).
4. Validieren Sie die Webseite.

# 12

## Webauftritt veröffentlichen

### 12.1 Suchmaschinenoptimierung (Search Engine Optimization – SEO)

#### meta-Element für Suchroboter angeben

Über das Element `meta` im `head`-Bereich können Sie Informationen über den Inhalt der Webseite bereitstellen, die von Software ausgewertet werden kann. Das Attribut `name="description"` (dt.: Beschreibung) zusammen mit dem Attribut `content` (dt.: Inhalt) enthält eine Zusammenfassung Ihrer Seite.

```
<meta name="description" content="Beschreibung">
```

#### Inhaltsbeschreibung eingeben

Diese Angabe hat zwar keine Auswirkung auf die Platzierung (Google), dient aber in vielen Suchmaschinen als Beschreibung der Webseite und wird im Suchergebnis dargestellt.

```
<meta name="description" content="Hotel Vallora in Küstendorf bietet Unterkunft, einen Wellness-Bereich, eine große Auswahl an Ausflügen und ein Restaurant an.">
```

*Beispiel für die Inhaltsbeschreibung im meta-Element*

Die Angaben sollten ...

- ✓ in vollständigen Sätzen angegeben werden,
- ✓ die wichtigsten Suchbegriffe beinhalten,
- ✓ ein Maximum von 156 Zeichen nicht überschreiten,
- ✓ Interesse für Ihre Seite wecken.

## Schlüsselwörter

Über das Attribut `name="keywords"` können Sie Stichwörter angeben, die zum Inhalt Ihrer Webseite passen.

Bis heute hält sich das Gerücht, dass dadurch die Platzierung (Ranking) in Suchmaschinen verbessert werden kann. Da diese in der Vergangenheit oft missbräuchlich verwendet wurden, werten Suchmaschinen die Schlüsselwörter **nicht** mehr aus. Daher sollten Sie auf die Angabe von Schlüsselwörtern verzichten.

## Inhalt, Inhalt, Inhalt

Ihre Inhalte sind das Wichtigste für Google und Co und das gleich auf dreifache Weise.

- ✓ Die Inhalte selber werden von Google direkt bewertet (s. a. „Webseiten für Suchmaschinen vorbereiten“)
- ✓ Gute Inhalte werden gerne verlinkt (führen dadurch zu einem hohen PageRank)
- ✓ Beliebte Inhalte fesseln Besucher und führen dazu, dass
  - ✓ Ihre Webseite in sozialen Medien geteilt wird,
  - ✓ die Absprungrate niedrig bleibt (steigert Ihre Reputation – auch bei den Suchmaschinen),
  - ✓ sich Besucher für weitere Inhalte auf Ihrer Website interessieren und länger bleiben.

## Doppelte Inhalte vermeiden

Wichtig ist, dass Ihre Inhalte einzigartig sind – Google kann doppelte Inhalte (duplicate content) nicht leiden. Nutzen Sie also selbst verfasste Texte und Bilder anstelle von kopierten Inhalten.

Dabei ist zu beachten, dass Ihre Webseiten in der Regel über mehrere Adressen erreichbar sind. Ihre Startseite kann bei Providern, die http und https unterstützen und den Aufruf mit oder ohne www erlauben, wie folgt aufgerufen werden:

- <http://www.example.com>
- <http://example.com>
- <http://www.example.com/index.htm>
- <https://www.example.com>
- <https://example.com>
- <https://www.example.com/index.htm>

Wenn Sie ein Content-Management verwenden, ist Ihre Startseite vermutlich zusätzlich wie folgt erreichbar:

- <http://www.example.com/index.php?pid=1>
- <http://example.com/index.php?pid=1>
- <https://www.example.com/index.php?pid=1>
- <https://example.com/index.php?pid=1>

**Wenn Sie ein Blogsystem einsetzen, können weitere URLs entstehen, z. B. wenn ein Beitrag mehreren Kategorien zugeordnet ist oder syndiziert wird.**

**Für Google sind das alles unterschiedliche Adressen mit demselben Content.** Sie sollten der Suchmaschine daher mitteilen, dass nur eine dieser Seiten die Originalseite ist und für diese das gesamte Ranking gelten soll – was diese Seite entsprechend stark gewichtet und bei Google weiter oben platziert.

Dazu geben Sie in einem Meta-Element die sogenannte canonical URL an – das ist die Adresse, die Sie bewerben wollen und die in den Suchmaschinen angezeigt werden soll (möglichst eine Variante mit https).

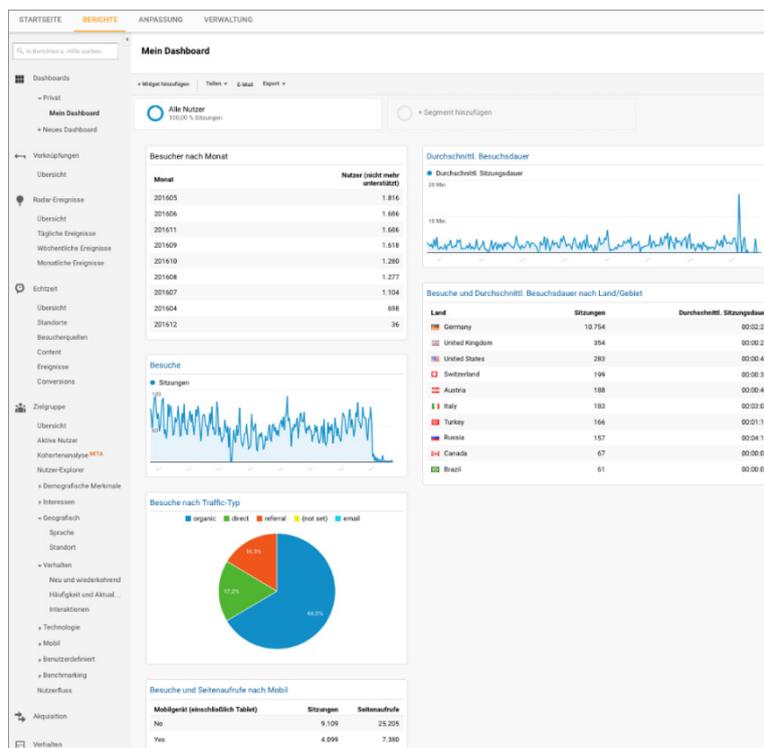
```
<link rel="canonical" href="https://www.example.com/index.htm" >
```

Mehr dazu unter <https://support.google.com/webmasters/answer/139066?hl=de> – Google und andere Suchmaschinenbetreiber stellen umfangreiche Tipps und Hilfen für die Suchmaschinen-Optimierung mit Informationen aus erster Hand bereit, die Sie nutzen sollten.

## Textgestaltung

Außerdem sollten die Texte ausreichend lang sein – zwei bis drei DIN-A4-Seiten sollten es vom Umfang her schon sein (mindestens aber 300 Wörter).

Von besonderer Bedeutung sind Überschriften, Alternativ-Texte von Bildern, als wichtig (strong) markierte Texte, Einträge in der Navigation und natürlich der Dokumentname und die Adresse, unter der ein Beitrag zu erreichen ist. An diesen Stellen sollten Sie die Begriffe verwenden, unter denen Sie gefunden werden wollen.



Umfangreiche Statistiken von Google Analytics

## Google Konsole

Ebenfalls unumgänglich ist die Nutzung der Search Console (auch Webmaster-Tools) unter <https://www.google.com/webmasters/tools> – hiermit analysieren Sie Ihre Webseite hinsichtlich Ihrer SEO-Bemühungen und können Optimierungspotential ausmachen.

Eine gute Unterstützung bietet Ihnen ein zusätzliches Analyse-Werkzeug. Google Analytics ist kostenlos und arbeitet am besten mit Google Ads zusammen, sodass Sie leicht den Erfolg bezahlter Werbemaßnahmen nachvollziehen können. Aber auch Piwiks (kostenlos – muss auf einem eigenen Server installiert werden) oder kommerzielle Tools (die allerdings sehr teuer sind) helfen bei der Auswertung Ihrer Bemühungen.

Einige der kostenpflichtigen Tools haben kurze Laufzeiten oder sogar kostenlose Probezeiträume (z. B. SISTRIX), sodass Sie Ihre Webseite analysieren und Fehler beheben und den Dienst danach wieder abbestellen können. Selbst kurze Zeiträume kosten leicht ein paar hundert Euro und der Erfolg von Optimierungsmaßnahmen stellt sich oft erst nach Wochen ein. Dennoch: Wenn Sie auf Ihrer Webseite kostenpflichtige Dienstleistungen oder Waren anbieten, kann sich diese Investition schnell auszahlen.

Die wichtigsten hier beschriebenen Maßnahmen und deren Erfolg können Sie aber recht einfach feststellen: Steht Ihre Seite nach der Optimierung weiter oben bei Google, sind Sie auf dem richtigen Weg.

Generell sollten Sie sich bei der Erstellung Ihrer Webseite und bei der Nutzung kommerzieller Tools und Hilfen die Frage stellen, ob sich eine Leistung auszahlt. Dann lohnt sich auch eine überschaubare Investition.

Wenn Sie sich für professionelle Hilfe entscheiden, geben Sie keine großen Summen auf einen Schlag aus. Investieren Sie schrittweise und überprüfen Sie den Erfolg der Maßnahmen mittels der oben genannten Tools, so wie Sie Ihre eigene Arbeit auch überprüfen würden. Sonst verpufft Ihr Werbebudget in nutzlosen Maßnahmen.

Alle Maßnahmen zur Verbesserung des Rankings auf den Seiten selber bezeichnet man als on-Page-SEO – es gibt aber auch off-Page-SEO, also Optimierungsmaßnahmen, die außerhalb der eigenen Seite stattfinden.

## Suchmaschinen-Marketing (Search Engine Marketing – SEM) und Online-Marketing

### Für die eigenen Inhalte kostenlos werben

Erstellen Sie weitere Inhalte in den sozialen Medien. Die meisten Menschen erreichen Sie mit Facebook, die Google-Suche freut sich dagegen über Posts auf Google+ – aber auch Twitter ist immer eine Nachricht wert. Weisen Sie in diesen Netzwerken auf Neuigkeiten hin.

Besonders erfolgreich sind nicht unbedingt öffentliche Posts „an alle“, sondern Beiträge in Gruppen, die thematisch zu Ihrem Inhalt passen. Also Motorsportgruppen, wenn Sie einen Artikel über Autos verfasst haben oder eine lokale Gruppe über Ihren Ort, wenn Sie über die nächste Bürgermeisterwahl geschrieben haben.

## Google MyBusiness-Eintrag

Unter Google My Business (<https://www.google.de/business/>) sollten Sie Ihre Firma eintragen. Das gilt ganz besonders, wenn Sie ein lokales Unternehmen haben.

## Kostenpflichtige Werbung schalten

Google verdient sein Geld mit Werbung. Gegen Bezahlung können Sie eine Platzierung oberhalb der organischen Treffer (also die Treffer, die aufgrund ihres Rankings angezeigt werden) kaufen.

Dasselbe gilt für die besonders großflächige Hervorhebung Ihres Google-MyBusiness-Eintrags. Ähnliche Angebote halten auch die anderen Suchmaschinen-Betreiber bereit.

Übrigens beschränken sich mögliche Werbemaßnahmen nicht auf Suchmaschinen. Sehr gezielte Werbung für ganz genau definierbare Nutzergruppen können Sie beispielsweise auf Facebook schalten. Auch hier kann man mit Beträgen deutlich unter hundert Euro experimentieren, um erfolgreiche Strategien zu entwickeln, in die Sie im Erfolgsfall mehr investieren können.

The screenshot shows a Google search for 'herdt verlag für bildungsmedien gmbh'. The search results are as follows:

- Search Results:**
  - Unser Verlag sucht Autoren - Senden Sie Uns Ihr Manuskript
  - Verlag sucht Autoren - novumverlag.com
  - Digitale Bildungsmedien - interactivemedia-foundation.com
  - HERDT-Verlag | Bücher & digitale Medien zu Software und IT Themen
- My Business Listing (highlighted with an orange border):**
  - HERDT-Verlag für Bildungsmedien GmbH**
  - Verlag
  - Adresse: Am Kümmerling 21-25, 55294 Bodenheim
  - Telefon: 06135 922210
  - Buttons: Website, Routenplaner
  - Options: Änderung vorschlagen, Fehlende Informationen hinzufügen, Öffnungszeiten hinzufügen, Bewertungen (Bewertung schreiben, Foto hinzufügen)
  - Section: Andere suchen auch nach (with thumbnails for Klett Verlag GmbH, Schulbuch... der Verlagsgr..., Cornelsen Verlag, Verband Bildungs...)

*Prominente Hervorhebung des Herdt-Verlags für Bildungsmedien als Google My Business-Eintrag.*

## Webseiten für Suchmaschinen vorbereiten

Suchmaschinen analysieren nach ständig verbesserten mathematischen Regeln (**Suchmaschinen-Algorithmus**) die Inhalte Ihrer Website. Dabei werden die diversen Inhalte Ihrer Website unterschiedlich bewertet. Was Sie im `title`-Element angeben, wird dabei stärker gewichtet als Text im `body`-Bereich. Überschriften werden hier besonders stark gewichtet.

Wenn ein Besucher einen Suchbegriff in einer Suchmaschine eingibt, werden verschiedene Kriterien bewertet, um eine Platzierung in der Trefferliste (Ranking) zu berechnen. Die Kriterien und ihre Gewichtung unterscheiden sich von Suchmaschine zu Suchmaschine. Die folgende Tabelle gibt Ihnen einen Überblick darüber, wofür sich Suchmaschinen auf Ihrer Webseite besonders interessieren.

Seitentitel	Befindet sich der Suchbegriff im Seitentitel der Webseite?
Suchwortdichte	Wie oft kommt der Suchbegriff auf der Seite vor? Dieses Kriterium wird unterschiedlich bewertet.
Suchbegriffprominenz	Wie weit oben befindet sich der Suchbegriff – steht er am Anfang der Seite oder ganz unten?
Alternativtext von Bildern	Steht der Suchbegriff im alternativen Text ( <code>alt</code> -Attribut) der Bilder?
Hyperlinktext	Befindet sich der Suchbegriff im „anklickbaren“ Text eines Hyperlinks?
Suchbegriffnähe	Wenn der Suchbegriff aus mehreren Wörtern besteht, wie weit sind diese im Text voneinander entfernt? In welcher Reihenfolge befinden sie sich? – Idealerweise bilden Sie ein zusammenhängendes Konstrukt wie <i>„günstige Schuhe in Musterstadt“</i> .
Überschriften	Befindet sich der Suchbegriff in Überschriften (d. h. innerhalb der Tags <code>&lt;h1&gt;</code> bis <code>&lt;h2&gt;</code> )?
Domainname	Befindet sich der Suchbegriff im Domainnamen?
URL	Befindet sich der Suchbegriff in der URL der Seite?



Suchmaschinen sind nicht dazu da, Ihre Seite zu bewerben. Suchmaschinen wollen den eigenen Besuchern Webseiten empfehlen, die möglichst optimal zu den eingegebenen Suchbegriffen passen und darüber hinaus „gut“ sind – d. h. Seiten, die beliebt sind (auf die häufig verlinkt wird), die schnell geladen werden und die auch mit mobilen Geräten (z. B. Smartphones) gut bedienbar sind.

Kommt ein Besucher „schnell“ zu Google zurück, war die besuchte Seite wohl nicht relevant. So gibt ein Suchmaschinen-Nutzer ein negatives Votum über Ihre Seite ab. Optimieren Sie also für Inhalte, die Sie tatsächlich anbieten!

Es reicht nicht aus, wenn Sie beispielsweise Begriffe, unter denen Sie gefunden werden wollen, möglichst oft in Ihre Seite „stopfen“ (engl.: keyword stuffing). Abhängig vom Grad dieses **Spammings** werden Websites aus dem **Suchmaschinen-Index** gelöscht. Das heißt, Ihre Seiten werden nicht mehr von der entsprechenden Suchmaschine gefunden.

## Bei Suchdiensten anmelden

Viele Suchdienste durchsuchen alle 4–6 Wochen das Web nach neuen Einträgen. Möchten Sie, dass Ihr Webauftritt schneller gefunden wird, bzw. möchten Sie von Suchdiensten gefunden werden, die keine Suchroboter einsetzen, melden Sie Ihren Webauftritt an.

Die meisten Suchdienste bieten eine kostenlose Anmeldung an. Sie müssen nicht alle Webseiten Ihres Webauftritts anmelden, sondern lediglich die Homepage. In der Regel werden Sie aufgefordert, die folgenden Informationen einzugeben:

- ✓ die URL Ihrer Homepage,
- ✓ den Seitentitel der Homepage,
- ✓ eine kurze Beschreibung des Webauftritts sowie einige Stichwörter, die den Inhalt der Webseiten beschreiben,
- ✓ die Kategorie, unter der Ihr Webauftritt aufgelistet werden soll.

Google verlangt hierfür ein Google-Konto, stellt aber im Gegenzug mit den Webmaster Tools, Google Places, Google Analytics, Ad Words und anderen Diensten besonders viele Werkzeuge zur Suchmaschinen-Optimierung, Online-Werbung und statistischen Auswertung bereit.

Die URLs zur Anmeldung bei wichtigen Suchmaschinen im deutschsprachigen Raum sind in der folgenden Tabelle aufgeführt:

Suchmaschine	URL zur Anmeldung des Webauftritts
Bing und Yahoo!	<a href="https://ssl.bing.com/webmaster/SubmitSitePage.aspx">https://ssl.bing.com/webmaster/SubmitSitePage.aspx</a>
Google	<a href="http://www.google.de/addurl/?hl=de&amp;continue=/addurl">http://www.google.de/addurl/?hl=de&amp;continue=/addurl</a>

## Suchmaschinenoptimierung testen

Nachdem Sie Ihr Webangebot beim Provider veröffentlicht haben, können Sie testen, wie gut die Seiten von einer Suchmaschine gefunden werden. Sämtliche Suchmaschinenbetreiber bieten hierzu kostenlose Online-Tools, z. B. Bing Webmaster Tools und Google Webmaster Tools. Änderungen müssen Sie in Ihren lokalen Dateien vornehmen und anschließend erneut veröffentlichen.

## Geduld bei der Suchmaschinen-Optimierung

Die Ergebnisse Ihrer Optimierungen wirken sich erst nach geraumer Zeit aus, denn Suchmaschinen besuchen gerade kleinere Sites nicht täglich. Unter Umständen kommt eine Suchmaschine nur alle zwei Wochen bei Ihnen vorbei. Um eine Seite besser zu positionieren, brauchen Sie einen langen Atem. Nehmen Sie sich hierfür 3 bis 6 Monate Zeit – als Anfänger eher mehr. Andersherum gesprochen: geben Sie nicht auf, wenn Sie etwas an Ihrer Homepage ändern und sie danach nicht sofort deutlich besser gefunden werden.

Hier wurden ein paar wichtige Tipps und generelle Empfehlungen gegeben. Diese gelten größtenteils seit vielen Jahren und werden aller Voraussicht nach auch noch längere Zeit Gültigkeit besitzen. Um tiefer in die Suchmaschinen-Optimierung einzusteigen, sollten Sie sich online informieren, da die Suchmaschinen-Optimierung aufgrund der rasanten Entwicklung der Suchmaschinen-Algorithmen ein sehr kurzlebiges Geschäft ist, das sich nicht gut für die Vermittlung in Druckwerken eignet.

Ein guter Anlauf-Punkt sind zunächst die Hilfe-Seiten der Suchmaschinenbetreiber selbst:

- ✓ **Google Webmasters**  
<https://www.google.com/webmasters/>
- ✓ **Google Youtube-Kanal**  
<https://www.youtube.com/user/GoogleWebmasterHelp>
- ✓ **Bing webmastertools**  
<http://www.bing.com/toolbox/webmaster>
- ✓ **Yahoo! Webmaster Ressourcen**  
<https://de.hilfe.yahoo.com/kb/SLN2248.html>

## 12.2 Einen Domainnamen registrieren

### Brauchen Sie einen eigenen Domainnamen?

Die erste Überlegung auf dem Weg zur Webpräsenz ist das Aussehen der Webadresse (URL) Ihres Webauftritts. Grundsätzlich gibt es zwei Arten von Webadressen:

Eigene Domain	<a href="http://www.hotelvallora.de">http://www.hotelvallora.de</a>
Benutzername in einer bestehenden Domain	<a href="http://home.example.com/hotelvallora">http://home.example.com/hotelvallora</a>

In der ersten Variante wurde ein Domainname in einer bestimmten Top-Level-Domain (*com, org, de, ch, at* usw.) beantragt, hier in der Top-Level-Domain *de* für Deutschland. Für kommerzielle Webauftritte ist diese Variante empfehlenswert, da z. B. die Adresse *www.hotelvallora.de* für Benutzer leicht zu merken ist und hochwertig wirkt.

Für die zweite Variante wurde ein Benutzername auf dem Webserver eines Providers eingerichtet. Diese Möglichkeit ist für private Webauftritte geeignet, da sie von vielen Betreibern kostenlos angeboten wird. Diese Adressen sind länger, da der Name des Webservers mit angegeben werden muss.

### Verfügbarkeit eines Domainnamens prüfen

Die Adresse Ihrer Website setzt sich i. d. R. aus drei Teilen zusammen: *subdomain.domain.tld*

- ✓ Die Top Level-Domain (tld) können Sie sich aus einem endlichen Pool auswählen (zum Beispiel *.de, .com* oder *.org*).
- ✓ Den Domainnamen (domain) können Sie frei wählen, sofern er noch in Kombination mit der Top-Level-Domain verfügbar ist.

- ✓ Als Subdomain verwenden die meisten Seiten *www* – es sind aber auch andere möglich. Zum Beispiel *de* oder *fr* um bei mehrsprachigen Webauftritten die Sprachversionen voneinander zu trennen oder *privat* und *www*, wenn Sie unter der Domain *meinName.de* einen offiziellen Auftritt und einen privaten Auftritt planen.

Wenn Sie sich für einen Namen entschieden haben, prüfen Sie, ob dieser noch verfügbar ist. Die Vergabe und Registrierung von Domainnamen werden von verschiedenen Firmen im Internet angeboten.

Neben den großen Anbietern wie Telekom, Strato, 1&1 oder HostEurope gibt es Dutzende weiterer Anbieter, die sich oft auf qualitativ hochwertigen Service rund um bestimmte Techniken spezialisiert haben. So gibt es Anbieter, die Ihre Server für bestimmte Content Management Systeme wie TYPO3 oder Wordpress optimieren und dazu auch kompetent beraten können.

Die Auswahl eines Anbieters ist ein wichtiger Schritt, denn bei einem Wechsel müssen oft Verträge mit Laufzeiten von bis zu 2 Jahren erfüllt (d. h. bezahlt) werden und der Umzug der Webseite (Domainübernahme) ist nicht immer problemlos.

Die Wahl eines falschen Hosting-Partners kann zu hohen Kosten führen. Das trifft oft auf vermeintlich billige Anbieter zu, die wenig für das Hosten Ihrer Webseite verlangen, aber z. B. hohe Gebühren für die telefonische Beratung berechnen, wobei die Bestellhotline meist kostenlos ist. Deshalb sollten Sie Gebühren und Umfang der Leistung sorgfältig prüfen, bevor Sie sich an einen Provider binden.

Der reguläre Preis für die Bereitstellung von ausreichend Platz für eine Webseite inklusive eigener Domain und E-Mail sollte auch für Qualitätshosting nicht über 5,- EUR je Monat liegen (nach Ablauf von 0-EUR-Monaten für Neukunden).

Hochwertigen Webspace für den Betrieb von Content Management-Systemen (mit viel Platz, PHP, Email-Adressen, ein oder mehreren inklusiv-Domains und mindestens einer MySQL-Datenbank) gibt es für maximal 10,- EUR im Monat.

Darüber hinaus fallen für die Ersteinrichtung oft einmalige Kosten an. Weitere Leistungen werden ebenfalls gesondert berechnet.

Im folgenden Beispiel fragen Sie bei der Registrierungsstelle für deutsche Domains ab, ob der Domainname *hotelvallora.de* zur Verfügung steht. Bei anderen Betreibern ist die Vorgehensweise analog.

- ▶ Laden Sie die Startseite einer DNS-Registrierungsstelle in Ihrem Browser, im Beispiel <http://www.denic.de>.
- ▶ Geben Sie den gewünschten Domainnamen in dem entsprechenden Feld ① ein.
- ▶ Betätigen Sie . Sie erhalten eine Mitteilung darüber, ob der Domainname noch frei oder bereits registriert ist.



**Domainsuche**

Sie können die Registrierung selbst bei einem Domainbetreiber vornehmen oder von einer Firma (Provider oder Host) durchführen lassen, der Speicherplatz im Web anbietet. Viele Firmen bieten die Registrierung günstig oder kostenfrei an, wenn Sie einen Vertrag über die Veröffentlichung Ihres Webauftritts mit dem Provider abschließen. Dort lässt sich in der Regel auch feststellen, ob der gewünschte Name unter exotischeren Top-Level-Domains verfügbar ist. In der Regel finden Sie hier schneller Ihren Wunschnamen als bei der denic und anderen Diensten, die nur für eine oder wenige TLDs Auskunft erteilen.

### Rechte Dritter bei der Domainnamenregistrierung berücksichtigen

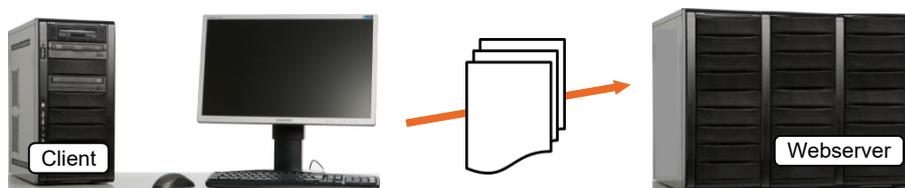
Wenn Sie einen freien Domainnamen gefunden haben, sollten Sie sicherstellen, dass dieser Name von keiner bereits bestehenden Institution beansprucht werden kann. Überprüfen Sie im Internet, ob der von Ihnen gewählte Domainname eventuell mit einem Eintrag im deutschen, im österreichischen bzw. im Schweizer Firmenindex (<http://www.branchenbuch.de>, <http://www.yellowmap.at> bzw. <http://www.yellowmap.ch>) oder im internationalen Markenregister (<http://www.wipo.int>) übereinstimmt.

Auch geografische Bezeichnungen (Städtenamen, Regionsbezeichnungen etc.) sollten Sie nach Möglichkeit nicht verwenden, um spätere Ansprüche Dritter zu vermeiden. Es empfiehlt sich, den Wunschnamen mithilfe verschiedener Suchmaschinen zu suchen, um herauszufinden, ob die Bezeichnung bereits verwendet wird, auch wenn sie noch nicht registriert wurde.

## 12.3 Speicherplatz auf einem Webserver reservieren

### Was ist ein Webserver?

Ein **Webserver** ist Software auf einem Computer, der mit dem Internet verbunden ist. Der Computer selbst wird ebenfalls Webserver genannt. Webseiten werden auf den Webserver hochgeladen. Auf Anfrage des **Clients** (z. B. Browsers) liefert der Webserver Webseiten aus.



Lokale Dateien auf den Webserver hochladen

Die zwei am häufigsten eingesetzten Webserver sind die Internet Information Services von Microsoft, der auf Betriebssystemen von Microsoft eingesetzt werden kann, und der Apache Web Server, der mit fast allen Betriebssystemen kompatibel ist.

Entweder haben Sie selbst einen Webserver oder Sie kaufen Speicherplatz (**Webpace**) auf dem Webserver eines Anbieters (**Provider** oder **Host**). Einige Provider bieten auch kostenlosen Speicherplatz, wobei diese Möglichkeit sich hauptsächlich für den privaten Bereich eignet.

## Einen Provider finden

Die Veröffentlichung eines Webauftritts bei einem Provider wird auch als **Hosting** bezeichnet. Die nachfolgend aufgeführten Kriterien sollen Ihnen bei der Auswahl eines geeigneten Providers helfen.

- ✓ Dient Ihr Webauftritt kommerziellen Zwecken oder ist er rein privater Natur?
- ✓ Wie viel Speicherplatz benötigen Sie auf dem Webserver? Wie groß ist Ihr Webauftritt aktuell? Sind weitere Seiten geplant?
- ✓ Sie können eine eigene Domain über den Provider registrieren lassen und auf diese Weise Zeit und gegebenenfalls auch Geld sparen. Prüfen Sie diesbezügliche Angebote.
- ✓ Falls Sie täglich viele Zugriffe auf Ihre Seite erwarten, wählen Sie einen geeigneten Tarif und eine entsprechend hohe Leistung des Services.
- ✓ Benötigen Sie zusätzliche Dienstleistungen wie Formularauswertung, E-Mail-Adressen, Webauftritt-Statistik, Shop-Funktionen, Skripte (Informationsabfragen, die serverseitig bearbeitet werden)?
- ✓ Welche Sicherheitsvorkehrungen verwendet der Provider, um Ihren Webauftritt vor unberechtigten Zugriffen und damit vor Manipulationen zu schützen?
- ✓ Wie gut sind Beratung und Hilfe bei Problemen? Gibt es eine möglichst kostenlose Hotline, und wie gut und zu welchen Zeiten ist diese erreichbar?
- ✓ Deutsche Gesetze verpflichten Sie unter Umständen zur Datenhaltung in Deutschland. Auch wegen der Überwachung durch ausländische Geheimdienste sind deutsche Server beliebt. Vor allem, wenn Sie nicht nur ohnehin öffentlich zugängliche Webseiten dort bereitstellen, sondern beispielsweise zusätzlich E-Mail- oder Kundendaten.

Vergleichen Sie anhand Ihrer vorgegebenen Kriterien die Preise (monatlich und einmalig anfallende Kosten, sowie Gebühren für Beratungen bei der Hotline) sowie die Mindestlaufzeit der Verträge (meistens 12 Monate) und die Kündigungsfristen.

Die Suchmaschine Ihrer Wahl hilft beim Auffinden der Anbieter, aber auch von Erfahrungsberichten und mehr Informationen zu den verschiedenen Firmen.

Vielleicht haben Sie ja bereits einen Platz zum Speichern Ihrer Website und wissen es noch gar nicht? Oft ist Platz für eine Website und eine oder zwei Domains im Preis von umfangreichen Angeboten von Telekommunikationsanbietern enthalten, bei denen Sie einen Internet-Zugang, Telefon- oder Handy-Vertrag abgeschlossen haben.

## 12.4 Einen Webauftritt veröffentlichen

### Was ist FTP?

Das am häufigsten verwendete Verfahren, um Dateien auf einen Server im Internet zu laden, nutzt das **File Transfer Protocol (FTP)**. Dieses Protokoll ermöglicht die systemunabhängige Datenübertragung zwischen Computern. Sie benötigen die folgenden Zugangsdaten von Ihrem Provider, um Ihren Webauftritt zu veröffentlichen:

- ✓ die Adresse des FTP-Servers,
- ✓ Ihren Benutzernamen und Ihr Kennwort,
- ✓ den Namen der Ordner auf dem Webserver, in den Sie Ihre Dateien ablegen.

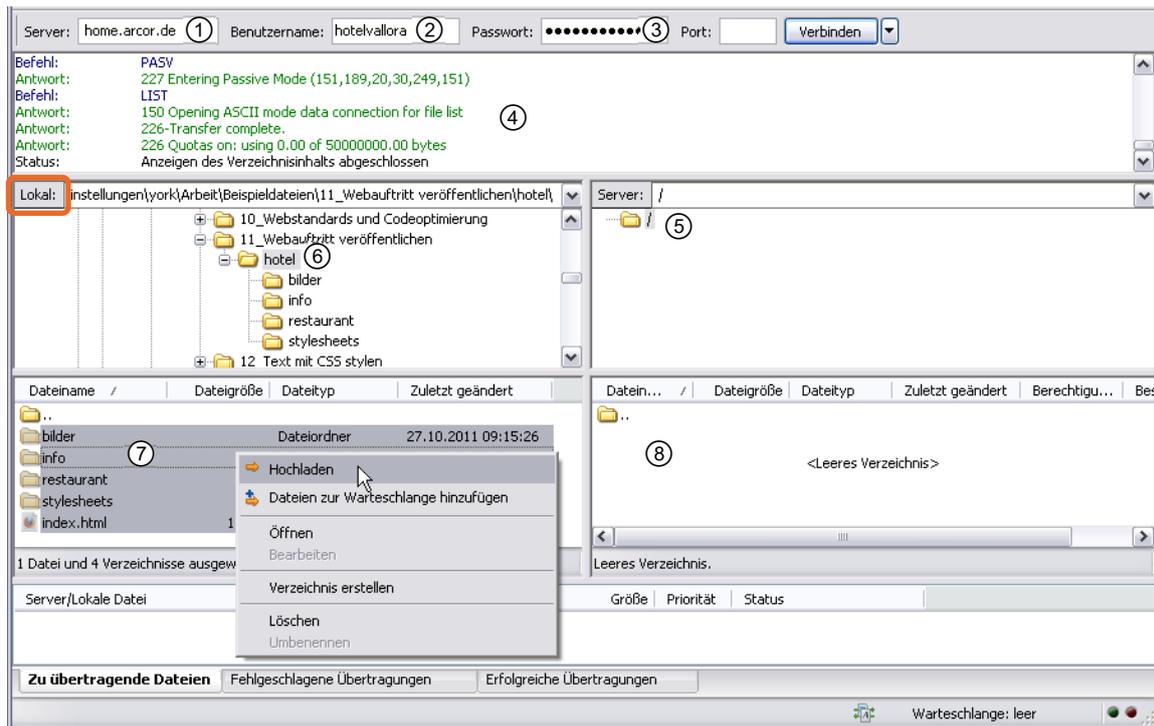
Zusätzlich muss ein **FTP-Client** auf Ihrem Computer installiert werden. Ein FTP-Client ist Software, mit der Sie die FTP-Verbindung herstellen und die Datenübertragung verwalten. Eine Liste deutschsprachiger FTP-Clients ist in der folgenden Tabelle aufgeführt.

FTP-Client	System	Typ
Cyberduck	Mac OS X	Freeware
FileZilla	Linux, Mac OS X, Windows	Freeware
FireFTP	Linux, Mac OS X, Windows	Freeware (Firefox-Plugin)
SmartFTP	Windows	Kommerziell
WISE-FTP	Windows	Kommerziell
WS_FTP	Windows	Kommerziell

### Webauftritt mit FileZilla auf den Webserver hochladen

Im folgenden Beispiel laden Sie die fertigen Webseiten von Hotel Vallora auf einen Webserver. Dazu wird der FTP-Client FileZilla verwendet. Die Vorgehensweise bei anderen FTP-Clients ist ähnlich.

- ▶ Laden Sie FileZilla von der Webseite <http://www.filezilla.de> herunter und installieren Sie das Programm auf Ihrem Computer.
- ▶ Starten Sie FileZilla.
- ▶ Geben Sie im Feld *Server* ① den Namen des Webserver ein.



#### Dateien auf den Webserver hochladen

- ▶ Tragen Sie im Feld ② Ihren Benutzernamen ein.
- ▶ Geben Sie im Feld ③ Ihr Passwort ein.
- ▶ Betätigen Sie die Schaltfläche *Verbinden*, um eine Verbindung zum Webserver herzustellen. Der Verlauf der Verbindung wird im Feld ④ angezeigt. Bei erfolgreicher Verbindung erscheint der Ordner, der für Sie auf dem Webserver erstellt wurde, der sogenannte Remote-Ordner, im Bereich ⑤.
- ▶ Wechseln Sie im Bereich *Lokal* zum Stammordner ⑥ Ihres Webauftritts. Die Unterordner und Dateien werden im Bereich ⑦ aufgelistet.
- ▶ Markieren Sie die Ordner und Dateien, die Sie hochladen möchten.
- ▶ Klicken Sie mit der rechten Maustaste auf die markierten Einträge und wählen Sie den Kontextmenüpunkt *Hochladen*. Die hochgeladenen Dateien werden im Bereich ⑧ angezeigt.
- ▶ Öffnen Sie Ihren Browser und geben Sie die URL Ihres Webauftritts ein, um den Webauftritt im Internet zu besuchen.

**Plus+** Lernvideo: [webseiteMitFileZillaVeroeffentlichen.mp4](#)

! Bei einer FTP-Übertragung werden Benutzernamen und Kennwörter unverschlüsselt übertragen. Benutzen Sie darum möglichst verschlüsselte Verbindungen (SFTP oder SSH).

## 12.5 Übung

### Webseiten für Suchmaschinen vorbereiten

Level		Zeit	ca. 10 min
Übungsinhalte	✓ Inhaltsbeschreibung eingeben ✓ Schlüsselwörter festlegen		
Übungsdatei	<i>hotel12</i>		
Ergebnisdatei	<i>hotel12-e</i>		

1. Öffnen Sie die Webseite *index.html* im Übungsordner *hotel12*.
2. Fügen Sie eine Inhaltsbeschreibung im `head`-Bereich ein.
3. Speichern Sie die Seite.
4. Öffnen Sie die Webseite *ueberuns.html* im Übungsordner *hotel12/info* und geben Sie hier ebenfalls eine passende Inhaltsbeschreibung.
5. Speichern Sie die Seite.

# 13

## Text mit CSS stylen

### 13.1 Das Aussehen von Text mit CSS gestalten

#### Rückschau auf die Syntax von CSS

Sie können das Aussehen von Texten mittels CSS festlegen. Wie Sie schon erfahren haben, legen Sie über einen Selektor fest, welches Element Sie stylen möchten, und dann geben Sie ein Eigenschafts-Werte-Paar ein.

Wenn Sie eine Schriftfarbe für das Element `body` festlegen, gilt diese als Standardschriftfarbe für die ganze Seite, da alle anderen Elemente Nachfahren des `body`-Elements sind und Schriftfarben auf Nachfahren vererbt werden.

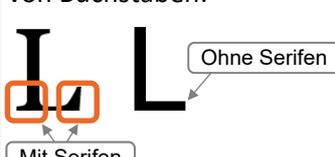
Geben Sie die folgende Regel im Stylesheet-Bereich bzw. in Ihrem externen Stylesheet ein, um eine Standardschriftfarbe zu definieren:

```
body {color: blue; }
```

Mit der CSS-Eigenschaft `color` die Schriftfarbe festlegen

#### Welche Texteingenschaften können mit CSS gesteuert werden?

In der nachfolgenden Tabelle erhalten Sie einen Überblick über einige Texteingenschaften, die von allen gängigen Browsern unterstützt werden.

CSS-Eigenschaft	Werte/Beispiele	Beschreibung
<b>font-family</b>	Arial Arial Black Courier New Georgia serif für Serifenschriften wie Times New Roman sans-serif für serifenlose Schriften wie Arial	Schriftart bzw. Schriftfamilie  <b>Serifen</b> sind Abschlussstriche von Buchstaben: 

CSS-Eigenschaft	Werte/Beispiele	Beschreibung
<b>font-size</b>	Schlüsselwörter, z. B. <code>medium</code> (Standard) Pixelwerte, z. B. <code>12px</code> Prozentwerte, z. B. <code>150%</code> em-Werte, z. B. <code>1.2em</code> (1em entspricht der Größe des Buchstabens „M“ oder 16 Pixeln)	Schriftgröße
<b>font-style</b>	<code>normal</code> (Standard) <code>italic</code> (kursiv) <code>oblique</code> (Manche Schriftarten besitzen den Schnitt Oblique anstatt Italic.)	Schriftschnitt
<b>color</b>	Farbnamen, z. B. <code>black</code> (Standard) Hexadezimalwerte, z. B. <code>#0000FF</code> RGB-Werte, z. B. <code>rgb(0, 0, 255)</code>	Schriftfarbe
<b>font-variant</b>	<code>normal</code> (Standard) <code>small-caps</code>	Kapitalchen
<b>font-weight</b>	<code>normal</code> (Standard) <code>bold</code> 100 bis 800 (sehr schmal bis sehr fett)	Fette Schrift / Gewicht
<b>text-align</b>	<code>start</code> (Standard, links bei uns, rechts z. B. in arabischen Ländern) <code>left</code> <code>right</code> <code>center</code> <code>justify</code> (Blocksatz)	Ausrichtung
<b>text-decoration</b>	<code>none</code> (normale Schreibweise, Standard) <code>underline</code> (unterstrichen) <code>line-through</code> (durchgestrichen) <code>overline</code> (Linie über dem Element)	Texteffekte
<b>text-indent</b>	0 (Standard)      Wir wünschen Ihnen einen netten Aufenthalt.  5px  Wir wünschen Ihnen einen netten Aufenthalt.  -5px  Wir wünschen Ihnen einen netten Aufenthalt.	Einzug der ersten Zeile Der Einzug kann als absoluter bzw. relativer Wert angegeben werden.
<b>text-transform</b>	<code>none</code> (normale Schreibweise, Standard) <code>capitalize</code> (jeder erste Buchstabe groß) <code>lowercase</code> (Kleinschreibung) <code>uppercase</code> (Großschreibung)	Groß-/Kleinschreibung erzwingen

CSS-Eigenschaft	Werte/Beispiele	Beschreibung
<b>line-height</b>	normal (Standard) Zahlen, z. B. 1.5 Prozentwerte, z. B. 150% em-Werte, z. B. 1.5em	Zeilenhöhe (möglichst ohne Einheit angeben, bezieht sich auf die Schriftgröße)
<b>word-spacing</b>	normal (Standard) Absolute und relative Werte. (Ein positiver Wert erhöht und ein negativer Wert verringert den Abstand zwischen Wörtern.)	
<b>letter-spacing</b>	normal (Standard) Absolute und relative Werte. (Ein positiver Wert erhöht und ein negativer Wert verringert den Abstand zwischen Zeichen.)	Zeichenabstand

## 13.2 Schriftfamilien definieren

### Was ist eine Schriftfamilie?

**Schriftschnitte** (z. B. fetter, fett, normal, dünn, dünner, kursiv, fett und kursiv) werden in **Schriftfamilien** zusammengefasst. Mehrere Schriftfamilien werden in der Regel zu **Schriftklassen** zusammengefasst (z. B. Klassizistische Antiqua, Serifenlose Linear-Antiqua oder Schreibschrift).

In CSS können Sie außerdem **generische Schriftfamilien** angeben. Damit fordern Sie den Browser auf, eine beliebige Schrift zu verwenden, welche die angegebene Bedingung erfüllt (z. B. Serifen oder keine Serifen).

Generische Schriftfamilie	Beschreibung	Schriftfamilie (Beispiele)
<b>serif</b>	Serifenschriften wurden für Druckmedien entwickelt, um Zeilen besser voneinander unterscheidbar zu machen und so die Lesbarkeit zu verbessern. Auf Standard-Bildschirmen mit geringer Auflösung werden diese Schriften jedoch unscharf oder sogar unleserlich dargestellt.	Times Times New Roman Georgia
<b>sans-serif</b>	Serifenlose Schriften haben keine Abschlussstriche an den Buchstaben und sind geeigneter für Text auf Standard-Bildschirmen. Für hochauflösende Displays (z. B. Apples Retina-Displays) stellen Serifen kein Problem dar.	Arial Trebuchet MS Verdana

Generische Schriftfamilie	Beschreibung	Schriftfamilie (Beispiele)
<b>monospace</b>	Die Zeichen von Festbreitenschriften haben eine konstante Breite. Der Buchstabe <i>i</i> in einer Festbreitenschrift nimmt beispielsweise genauso viel horizontalen Platz ein wie ein <i>w</i> .	Courier Courier New Lucida Console
<b>cursive</b>	Kursive Schriften sehen wie Handschrift aus und erwecken den Eindruck von Verspieltheit. Die Lesbarkeit leidet zusätzlich bei Schriften mit Serifen.	Comic Sans <i>Montotype Corsiva</i>
<b>fantasy</b>	Diese Schriftarten sind sehr stilisiert. Viele dieser Schriftarten sind nicht auf allen Systemen verfügbar. Diese Schriften sind in der Regel nicht auf gute Lesbarkeit optimiert und daher nicht für längere Texte geeignet.	COMIX Impact

Beachten Sie bei der Auswahl Ihrer Schriftarten, dass nicht alle Schriftarten auf allen Systemen verfügbar sind. Es gibt Möglichkeiten, z. B. über die CSS-Regel `@font-face`, Schriftarten einzubinden. Für den Einstieg ist es jedoch ratsam, sich auf die folgenden Schriftarten zu beschränken, da diese auf fast allen Systemen vorhanden sind:

✓ Arial

✓ Courier

✓ Verdana

## Schriftfamilie angeben

Die **Standardschriftfamilie** für die meisten Browser unter Windows ist Times New Roman. Wenn Sie nichts anderes vorgeben, wird der Text Ihrer Webseite in der Standardschriftart angezeigt. Bei der Festlegung einer Schriftart teilen Sie dem Browser mit, welche Schriftart von Ihnen erwünscht ist. Diese Anweisung kann der Browser nur erfüllen, wenn die vorgegebene Schriftart installiert ist. Findet der Browser die vorgegebene Schriftart nicht, greift er auf die Standardschriftart zurück.

Mit der CSS-Eigenschaft `font-family` (auch Schriftfamilie oder Schriftliste genannt) haben Sie mehr Kontrolle über die zu verwendende Schriftart, da Sie Alternativen angeben können. Über die Eigenschaft `font-family` legen Sie Ihre bevorzugte Schriftart als ersten Eintrag in einer Liste fest, gefolgt von mehreren Alternativen aus der gleichen Schriftfamilie:

```
body {font-family: Verdana, Geneva, Arial, sans-serif; }
```

*Eine Schriftliste definieren*

Im obigen Beispiel teilen Sie dem Browser mit, dass der Text in der Schriftart *Verdana* angezeigt werden soll. Falls diese Schriftart auf dem System des Besuchers nicht vorhanden ist, soll die Schriftart *Geneva* verwendet werden, ansonsten *Arial*. Wenn auch *Arial* fehlt, soll eine serifenlose Schriftart verwendet werden, die der Browser auswählt. Ist keine der angegebenen Schriftarten auf dem Computersystem vorhanden, stellt der Browser den Text in seiner Standardschrift dar.

Bei Schriftarten mit Namen, die aus mehreren Wörtern bestehen, geben Sie den Namen in einfachen bzw. doppelten Anführungszeichen ( ' bzw. " ) an, z. B.

```
body {font-family: "Times New Roman", Georgia, serif; }
```

Aus mehreren Wörtern bestehende Schriftartnamen werden in Anführungszeichen geschrieben

- ✓ Die Anzahl der Alternativen, die Sie für das Element `font-family` eingeben, bleibt Ihnen überlassen. Falls erwünscht, können Sie nur eine Alternative oder zwei Alternativen angeben.
- ✓ Die Schriftartnamen können groß- oder kleingeschrieben werden; ob Sie z. B. *Arial* oder *arial* schreiben, spielt keine Rolle.
- ✓ Als letzten Wert sollten Sie einen Schriftfamilienamen angeben, z. B. *serif* bzw. *sans-serif*. Falls die anderen aufgelisteten Schriftarten nicht vorhanden sind, wählt der Browser eine Schriftart aus der gewünschten Schriftfamilie.

## 13.3 Schlüsselwörter verwenden

### Schriftgrößen mit Schlüsselwörtern definieren

Wie Sie schon erfahren haben, können Sie Schriftgrößen als absolute Werte, z. B. Pixel (px - für Bildschirme) oder Punkt (pt – für den Druck), oder als relative Werte, z. B. Prozente (%) oder Ems (ems), festlegen. Relative Werte eignen sich für Druck und Bildschirm. Zusätzlich haben Sie die Möglichkeit, die Schriftgröße über Schlüsselwörter anzugeben. Die Reihe von Bezeichnungen fängt mit dem Schlüsselwort `xx-small` an und wird mit jedem Schritt nach oben in der Hierarchie ungefähr 20 % größer. Die Darstellung der Schriftgröße kann sich in den verschiedenen Browsern geringfügig unterscheiden, aber in der Regel wird die Schriftgröße `small` 12 Pixel hoch dargestellt.

xx-small  
x-small  
small  
medium  
large  
x-large  
xx-large

Schlüsselwörter

Der Syntax für einen Body-Text, der 12 Pixel hoch ist, sieht so aus:

```
body {font-size: small; }
```

Schriftgröße mit einem Schlüsselwort definieren

Mit dem Schlüsselwort `smaller` bzw. `larger` können Sie das nächstniedrigere bzw. -höhere Maß festlegen. Sie können beispielsweise das Schlüsselwort `small` für den Body-Text festlegen und dann das Schlüsselwort `larger` für h1-Text. In diesem Fall wird der Text der Überschriften 1 in der Größe `medium` dargestellt.

## 13.4 Bestimmte Textstellen mit CSS stylen

### Klassen verwenden

Das Attribut `class` können Sie verwenden, um alle Elemente zu kennzeichnen, die Sie mit einer besonderen Formatierung versehen wollen. Beispielsweise haben Sie eine ungeordnete Liste von Ausflugsmöglichkeiten. Sie möchten besonders zu empfehlende Ausflüge mit Kursivschrift und Dunkelrot hervorheben. Sie weisen den entsprechenden Listenpunkten das Attribut `class` zu:

```
<ul>
  <li class="empfohlen">Ballonglücken</li>
  <li>Stadtrundgang</li>
  <li class="empfohlen">Bootsfahrt</li>
  <li>2-Tage-Ausflug zur Tour de France</li>
</ul>
```

Im nächsten Schritt definieren Sie im `style`-Bereich bzw. in der verlinkten CSS-Datei einen Stil für die Klasse. Achten Sie darauf, dass Sie einen Punkt (.) vor dem Klassennamen eingeben.

```
.empfohlen {font-style: italic; color:
maroon; }
```



- *Ballonglücken*
- Stadtrundgang
- *Bootsfahrt*
- 2-Tage-Ausflug zur Tour de France

In CSS muss den Klassennamen ein Punkt vorangestellt werden.

### Inline Spans einsetzen

Um einzelne Wörter mit CSS hervorzuheben, müssen Sie die hervorzuhebenden Stellen in einem `span`-Element einschließen. Beispielsweise können Sie im folgenden Satz das Wort *sehr* mit Rot wie folgt hervorheben:

```
<p>Das ist <span class="empfohlen">sehr</span> gut!</p>
```

Der Vorteil von Spans in Verbindung mit Klassen wird deutlich, wenn Sie bestimmte Wörter auf einer Webseite bzw. im gesamten Webauftreten einheitlich hervorheben möchten. Beispielsweise möchten Sie den Hotelnamen stets mit Fettdruck und in roter Schrift hervorheben. Erst kennzeichnen Sie die Stellen im Text des HTML-Dokuments wie folgt:

```
<p>Im ruhig gelegenen <span class="hotelname">Hotel
Vallora</span> finden Sie Erholung pur.</p>
```

*Eine Klasse definieren*

Danach legen Sie in Ihrer CSS-Datei die gewünschte Formatierung fest, indem Sie einen Stil für die Klasse `hotelname` definieren:

```
.hotelname {font-weight: bold; color: red }
```

Möchten Sie die Formatierung später ändern, müssen Sie nur die Angaben in der CSS-Datei ändern. Alle entsprechenden Stellen in den HTML-Dokumenten werden automatisch geändert.

Beachten Sie: Die hier verwendeten Namen beziehen sich nicht auf das gewünschte Aussehen (die Klasse lautet also nicht `bold-red`). Gerade im Sinne späterer Änderungen ist das wichtig. Eine Klasse mit solch einem Namen lässt sich später nicht für blaue, kursive Schrift verwenden und Sie müssten alle Stellen in jedem HTML-Dokument anpassen, wenn Sie Ihr Layout ändern wollen.

**Verwenden Sie daher immer logische Klassennamen, die die Bedeutung des zu formatierenden Elementes beschreiben.**

## IDs verwenden

Das Attribut `id` dient dazu, Elemente in einem HTML-Dokument eindeutig identifizierbar zu machen. Sie können es wie eine Klasse verwenden, mit dem Unterschied, dass Sie einer ID nur einmal pro Dokument denselben Wert zuweisen dürfen.

Das Attribut geben Sie wie folgt ein:

```
<ul>
  <li id="besonderheit">Ballonglücken</li>
  <li>Stadtrundgang</li>
  <li>Bootsfahrt</li>
  <li>2-Tage-Ausflug zur Tour de France</li>
</ul>
```

Den Stil legen Sie im `style`-Bereich bzw. in der verlinkten `css`-Datei fest, indem Sie ein Rautenzeichen (`#`) vor den ID-Namen setzen:

```
#besonderheit {color: maroon; }
```

## 13.5 Übung

### Das Aussehen von Text mit CSS definieren

Level		Zeit	ca. 20 min
Übungsinhalte	✓ Schrifteigenschaften mit CSS festlegen und ändern		
Übungsdatei	<i>hotel13</i>		
Ergebnisdatei	<i>hotel13-e</i>		

1. Öffnen Sie im Texteditor die Stylesheet-Datei *vallorastyles.css* und ersetzen Sie die vorhandenen Styles durch die folgenden Angaben:

Selektor	Eigenschaften und Werte
body	Schriftart: Verdana, Geneva, Arial, sans-serif Schriftfarbe: #003366 Schriftgröße: small
h1	Schriftgröße: 150 % vom body-Text Schriftfarbe: #ff9933
h2	Schriftgröße: 120 % vom body-Text Schriftfarbe: #ff9933
blockquote	Schriftschnitt: italic

2. Speichern Sie das Stylesheet und öffnen Sie die Webseite *index.html* im Browser.
3. Wechseln Sie zurück zum Stylesheet und ändern Sie die Standardschriftgröße in `large` um.
4. Speichern Sie das Stylesheet und laden Sie die Webseite im Browser erneut.

# 14

## Layout in CSS

In diesem Kapitel werden Ihnen zwei verschiedene Layouts nähergebracht und Schritt für Schritt erklärt. Die vorgestellten Layout-Methoden sind beide responsiv, das heißt beide funktionieren sowohl auf großen Bildschirmen, als auch auf Smartphones und Tablets.

### 14.1 Zweispalter mit Kopfbereich

#### Das „mobile first“-Prinzip

Webseiten werden in Deutschland mittlerweile ebenso häufig auf mobilen Geräten (Smartphone und Tablet) betrachtet wie auf großen Monitoren. Tendenziell verlieren die großen Monitore weiter an Bedeutung. Obwohl Sie Ihre Webseite wahrscheinlich an einem großen Monitor erstellen, sollten Sie nie vergessen, dass das (bald) nicht mehr die Standard-Darstellung für die meisten Ihrer Besucher ist.

Bevor Sie mit der Erstellung eines Layouts beginnen, hat Ihre Webseite schon eine Darstellung, die auf jedem Gerät funktioniert (schwarzer Text auf weißem Grund, der am Browserrand umbricht).

Fügen Sie erst nur eigene Farben und Schriften hinzu. Alle Elemente stehen weiter untereinander und lassen sich durch Scrollen erreichen.



*Standarddarstellung plus eigene Farben („kap14\2spalter.htm“)*

## Der HTML-Aufbau im Beispiel für ein Zweispalten-Layout

### HTML

```
① <!DOCTYPE html>
<html lang="de">
  <head>
    <meta charset="utf-8">
    <title>Layout: Zweispalter mit Titelzeile</title>
    <link rel="stylesheet" href="twoColumns.css">
  </head>

  <body>
    ② <h1>
      <a href="index.htm">
        
      </a>
      <span>Überschrift</span>
    </h1>
    ③ <div id="container">
      ④ <nav>
        <ul>
          <li><a href="#">Link</a></li>
          <li><a href="#">Link</a></li>
          <li><a href="#">Link</a></li>
        </ul>
      </nav>

      <main>
        <h2>Überschrift des Artikels</h2>
        <p>Fortune ...</p>
        <p>Omnia ...</p>
      </main>
    </div>
  </body>
</html>
```

Struktur der Datei „kap14/2spalter.htm“

- ① Am Anfang der Datei finden Sie die gewohnten Angaben wie Dokumenten-Typ, Sprachangabe, Zeichenkodierung, Titel und die Verknüpfung mit einer CSS-Datei (siehe nächstes Listing).
- ② Die Überschrift enthält das Logo und einen Text in einem eigenen `span`-Element. So können diese getrennt voneinander formatiert und ausgerichtet werden.
- ③ Ein Container-Element gruppiert Navigation und Hauptbereich der Seite, so dass diese in diesem Container ausgerichtet werden können.
- ④ Navigation und Hauptbereich stehen zunächst untereinander. Dies ist das natürliche Verhalten von Blockelementen.

## Farben für die Beispielwebseite in der CSS-Datei angeben

### CSS

```
body {
  font-family: Calibri, Helvetica, Arial, Geneva, sans-serif;
  background-color: #ddd;
  margin: 1em auto;
}
h1>*, #container>* {
  padding: 0.5em;
}
h1>a {
  display: inline-block;
  background-color: #4e95dc;
}
h1>span {
  display: block;
  background-color: #eee;
}
nav {
  background-color: #5fa6ed;
}
main {
  background-color: #fff;
}
```

- ✓ In der CSS-Datei werden zunächst nur Farben notiert. Das Ergebnis ist schon recht ansehnlich, obwohl Sie auf einer echten Webseite eher mit weniger unterschiedlichen Hintergründen arbeiten würden. Diese verursachen, dass das Beispiel sehr kleinteilig und zerstückelt wirkt. Es hilft hier aber die einzelnen Bereiche zu unterscheiden und soll daher im Beispiel beibehalten werden.
- ✓ Die Angabe von `margin: 1em auto;` bewirkt, dass oben und unten ein Rand von 1em frei gelassen wird, rechts und links soll automatisch gleich viel Rand zugewiesen werden. Da sich Blockelemente über die gesamte Seitenbreite erstrecken, wirkt sich diese Angabe nur aus, wenn Sie eine Breitenangabe notieren und der Bildschirm breiter ist, als das Element. Diese Angabe spielt nur auf größeren Bildschirmen eine Rolle und wird im nächsten Abschnitt wichtig.

## Mehrspaltiges Layout mit `display: table`

Nach den Farben legen Sie nun die Aufteilung der Seite in der Datei `twoColumns.css` fest.

```

① @media screen and (min-width:40em) {
②   body {
      display: table;
      width: 40em;
    }
③   h1, #container {
      display: table-row;
    }
④   #container>* {
      display: table-cell;
    }
⑤   h1>a,
      h1>span {
      display: table-cell;
      vertical-align: middle;
    }
      nav {
      vertical-align: top;
    }
      article {
      vertical-align: top;
    }
⑥ }
```

*Schnell erledigt: Dank der Tabellendarstellung reichen für das Layout wenige Zeilen CSS*

- ① Mit der in Kapitel 2 erläuterten `@media`-Regel weisen Sie den Browser an, die folgenden Regeln nur umzusetzen, wenn die genannten Bedingungen erfüllt sind. Alles innerhalb der geschweiften Klammern soll also nur auf Bildschirmen (screen) mit einer Mindestbreite von 40em angewendet werden. Dadurch stellen Sie sicher, dass das Layout auf kleinen Bildschirmen (z. B. Smartphones) einspaltig bleibt – Zwei Spalten haben hier nebeneinander nicht genug Platz.



*Ein paar Zeilen CSS sorgen für nebeneinander liegende Spalten*

- ② Um zwei Spalten bilden zu können, wird der Browser angewiesen, die Seite als Tabelle darzustellen. Die Breite wird auf `40em` festgelegt. Bereits vorher wurde mit `margin: 1em auto;` festgelegt, dass weiterer Platz links und rechts gleichmäßig verteilt werden soll, wodurch die Inhalte zentriert werden.
- ③ Nun bestimmen Sie, welche Elemente wie Tabellenzeilen dargestellt werden sollen. Im vorliegenden Fall bietet sich die Überschrift `h1` für die erste Zeile an, für die zweite Zeile benutzen Sie das Container-`div`.
- ④ In der ersten Zeile (`h1`) befinden sich zwei Elemente (`span` und `a`), die als Zellen fungieren können. So haben Sie eine Tabellenzeile, die zwei Zellen enthält. Im nachfolgenden Container gibt es wiederum zwei Kind-Elemente: `nav` und `main`. Weisen Sie auch diesen die Eigenschaft `display: table-cell` zu, um in dieser zweiten Zeile ebenfalls zwei Zellen zu schaffen. Damit haben Sie ein zweispaltiges Layout, das in allen aktuellen Browsern funktioniert.
- ⑤ Da die Zellen im Beispiel alle eindeutig voneinander unterscheidbar sind (`nav`, `main`, `h1>a`, `h1>span`), können Sie für jede eigene Styles definieren: Schriftarten, Farben, Rahmen usw. lassen sich für jeden Bereich einzeln vergeben. Elemente darin können Sie mittels Kind- oder Nachfahren-Selektor gezielt auswählen und gestalten.
- ⑥ Denken Sie daran, die Klammer der `@media`-Regel wieder zu schließen!

Mit wenigen Zeilen CSS konnten Sie ein mehrspaltiges Layout umsetzen. Das so entstandene Beispiel lässt sich um beliebig viele Spalten erweitern. Fügen Sie dafür einfach pro Zeile ein oder mehrere Kind-Elemente hinzu. Aufgrund der Selektoren `h1>*` bzw. `#container>*` erhalten diese automatisch die Eigenschaft `display: table-cell`. Sie sollten lediglich darauf achten, dass alle Zeilen gleich viele Zellen enthalten.

Den gesamten Code finden Sie in der Beispiel-Datei `kap14/twoColumns.css`.

## 14.2 Flexbox – flexible, responsive Layouts ohne @media-Regel

Mit Flexbox steht eine Layout-Methode bereit, die ideal für das Layout von Webseiten ist. Der größte Unterschied einer Webseite zu Dokumenten aus der realen Welt ist die Tatsache, dass Webseiten keine vorhersehbaren Dimensionen besitzen. Selbst auf großen Bildschirmen, kann ein Fenster klein sein. Oder jemand, der schlecht sehen kann, vergrößert Schriften so stark, dass auch in einem breiten Fenster nur wenige Inhalte nebeneinander passen.

Zwar berücksichtigt das Tabellen-Layout schon einige dieser Aspekte, doch musste mit der `@media`-Regel ein fester Breakpoint bestimmt werden.

Diesen zu setzen erfordert Erfahrung und trotzdem geht es nicht immer ohne viele Tests, bis sich eine Webseite wie gewünscht verhält.



### Überschrift des Artikels

Fortune plango vulnera stilantibus ocellis, quod sua michi munera subtrahit rebellis. Verum est, quod legitur, fronte capillata, sed plerumque sequitur occasio calvata. In Fortune solio sederam elatus, prosperitatis vario flore coconatus; quicquid enim florum felix et beatus, nunc a summo corruui gloria privatus. Fortune rota volvitur descendo minoratus, alter in altum tollitur, nimis exaltatus rex sedet in vertice; caveat ruinam! Nam sub axe legimus hecubam redinam.

Omnia Sol temperat purus et subtilis, novo mundo reserat faciem Aprilis; ad Amorem prosperat animus herilis et iocundis imperat deus niobilis. Rerum tanta novitas in clementi

*Flexbox – Darstellung auf einem Smartphone-Bildschirm*

Flexboxen spielen hier ihre Stärke aus: kombiniert mit einer Basis-Angabe für die Breite, wird „überflüssiger“ Platz vom Browser gleichmäßig auf alle Elemente einer Zeile aufgeteilt.

Passen ein oder mehrere Elemente nicht in eine Zeile, werden diese umgebrochen und auch in dieser neuen Zeile wird überflüssiger Platz wieder gleichmäßig verteilt.

Dieses Verhalten können Sie über unterschiedlichste Angaben fein steuern, so dass immer ein ordentliches Aussehen mit bündigen Kanten erreicht wird.

Neben vielen anderen Vorteilen gegenüber dem letzten Beispiel brauchen Sie sich hierbei nicht darum zu kümmern, wie viele „Zellen“ in einer Zeile stehen.

Um das zweispaltige Layout mit Flexbox umsetzen zu können, benötigen Sie keine andere HTML-Datei, auch die Farb- und Schriftangaben aus dem ersten Beispiel können erhalten bleiben. Nur die Layout-Angaben lauten anders:

```
① body {
    max-width: 40em;
    margin: 1em auto;
}
② h1, #container {
    display: flex;
    flex-wrap: wrap;
    margin: 0;
}
③ h1 > *,
#container > * {
    flex: 2 1 22rem;
}
④ h1 > :first-child,
#container > :first-child {
    flex: 1 2 11rem;
}
```

Die Layout-Angaben aus der Datei „kap14/twoColumns-flex.css“

- ① Wie zuvor wird das Layout auf 40em Breite beschränkt, diesmal aber mittels `max-width`, was ein Schrumpfen des verfügbaren Platzes zulässt.
- ② Die Container, die zuvor die Spalten beinhaltet haben, werden angewiesen, sich wie Flexboxen zu verhalten. `flexwrap: wrap` erlaubt ein Umbrechen der nebeneinander liegenden Inhalte und `margin: 0` sorgt dafür, dass der Außenabstand der Überschrift verschwindet und keine Lücke zwischen `h1` und `div#container` entsteht.
- ③ Den Inhalten der Flexboxen geben Sie mit `flex` drei Werte mit. Diese stehen für Wachsen, Schrumpfen und Basisgröße (meist Breite). Die Angabe für Wachsen und Schrumpfen geben das Verhältnis zu benachbarten Boxen an. Die Basis-Breite wird auf 22em gesetzt.
- ④ Erst mit diesen Angaben wird das Aussehen endgültig festgelegt. Die erste Spalte hat eine Basisbreite von 11rem, also die Hälfte der folgenden Spalten. Bei zwei Spalten ergibt sich dadurch eine Aufteilung von 1 zu 2. Steht genügend Platz zur Verfügung sollen die Spalten auch im Verhältnis 1 zu 2 wachsen (jeweils die erste Angabe der Eigenschaft `flex`).

Das Verhältnis wird beibehalten, weil die Angaben zum Wachsen und Schrumpfen immer das **Verhältnis** angeben, indem freier Platz auf die Boxen verteilt wird. Daher werden auch keine Einheiten angegeben. Um die komplette Berechnung brauchen Sie sich nicht zu kümmern, das übernimmt der Browser und die Seite sieht immer ordentlich aus. Wird das Fenster kleiner als `33rem` (`11rem + 22rem`), werden die Boxen untereinander dargestellt (siehe Smartphone-Darstellung am Anfang dieses Abschnitts).

Ein Problem gibt es noch mit dem Logo. Dessen feste Breite sorgt bei Bildschirmen, die gerade noch groß genug für zwei Spalten sind, für eine unschöne Ecke (siehe Kreis im Bildschirmfoto).

Das können Sie lösen, indem Sie dem Bild eine Breite von maximal 100% mitgeben:

```
img {max-width: 100%;}
```



Den kompletten Quelltext finden Sie in der Datei *kap14/twoColumns-flex.css*.

### 14.3 Vor- und Nachteile der beiden Layout-Methoden

Das Tabellenlayout eignet sich vor allem für einfache Layouts und ist besonders leicht zu erlernen. Flexboxen sind komplexer und haben entsprechend mehr Möglichkeiten. Während Tabellen-Layouts starr sind und immer gleich aussehen – egal wie viele Spalten Sie anlegen: diese bleiben immer erhalten, egal wie breit ein Bildschirm ist.

Flexboxen sind dagegen sehr viel flexibler. Sie spielen gerade im Web ihr Potential aus. Ihre Webseite sieht immer gut aus und wirkt wie für genau die gerade benutzte Seitenbreite gemacht – und das ohne Breakpoints. Mit dem Einsatz von Flexboxen erstellen Sie Webseiten, die so flexibel sind, dass es egal ist, mit welchem Browser Besucher auf Ihre Webseite kommen. Ihre Webseite passt sich an.

Flexboxen benötigen eine neue Denkweise. Begriffe wie „oben“ und „unten“, „rechts“ und „links“ spielen keine Rolle mehr, da sich die Boxen je nach Bildschirmbreite über- oder untereinander anordnen. Das gilt allerdings immer für responsive Webseiten und ist kein Nachteil, sondern gewollt.

Flexboxen machen insofern ein Umdenken beim Konzept eines Layouts erforderlich. Wenn Sie Flexboxen gekonnt einsetzen, werden Sie dafür mit Möglichkeiten belohnt, die Ihnen kein anderes Layout-Konzept bietet: das meiste wird einfacher, vieles überhaupt erst möglich.

## 14.4 Eine responsive Navigation mit Flexbox umsetzen

Wie beim Flexbox-Layout gesehen, können sich Flexboxen untereinander oder nebeneinander anordnen, je nachdem, ob der Platz dafür ausreicht – eine interessante Option für eine responsive Navigation.

Folgender Code sorgt zunächst dafür, dass die Navigationsliste wie ein Menü aussieht:

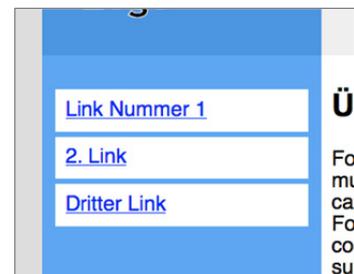
```

① nav ul {
    margin: 0;           /* Listeneinrueckung entfernen (IE) */
    padding: 0;         /* Listeneinrueckung andere Browser */
    list-style-type: none; /* Listenpunkte entfernen */
}
② nav li {
    margin: 0.125rem;   /* Abstaende zwischen den li */
}
③ nav a {
    display: block;     /* Links füllen Elternelement aus */
    padding: 0.5rem;    /* Abstaende um die Links */
④ background: white;  /* Hintergrund der Links */
}

```

*Die Navigationsliste als Menü gestalten*

- ① Listentypische Darstellung wird entfernt.
- ② Zwischen den Listeneinträgen wird etwas Abstand hinzugefügt.
- ③ Die Links bekommen eine größere Fläche: als Blockelemente füllen Sie nun das Elternelement in voller Breite aus.
- ④ Um die klickbare Fläche zu kennzeichnen, bekommen die Links einen weißen Hintergrund.



*Die Liste sieht nun aus wie ein typisches Navigationsmenü*

Mit einem ähnlichen Code wie bei der Seitenaufteilung, können wir die Menüpunkte veranlassen nebeneinander oder untereinander zu stehen.

In unserem Layout hat die Navigation eine Breite von `11rem` (abzüglich von Abständen). Wenn jeder Menüpunkt diese Breite bekommt, stehen alle untereinander. Da Flexboxen automatisch auf die gesamte Breite gezogen werden (überflüssiger Platz wird an alle Elemente komplett vergeben), kann auch eine geringere Breite angegeben werden. Wenn Sie sicherstellen wollen, dass niemals zwei Elemente nebeneinander stehen, achten Sie lediglich darauf, dass zwei Elemente zusammen nicht weniger breit sind, als das Elternelement. Andererseits sollten die Elemente breit genug für ihren Inhalt sein. Eine Breite von `9rem` reicht im vorliegenden Fall aus.

Der obige Code muss nur um drei Zeilen CSS ergänzt werden:

```

① nav ul {
    display: flex;           /* Flexboxcontainer wird definiert */
    flex-wrap: wrap;       /* Umbrechen der Flexboxen erlauben */
    margin: 0;             /* Listeneinrueckung entfernen (IE) */
    padding: 0;           /* Listeneinrueckung andere Browser */
    list-style-type: none; /* Listenpunkte entfernen */
}
② nav li {
    flex: 1 0 9rem;        /* grow shrink base */
    margin: 0.125rem;      /* Abstaende zwischen den li */
}
③ nav a {
    display: block;        /* Links füllen Elternelement aus */
    padding: 0.5rem;       /* Abstaende um die Links */
    background: white;     /* Hintergrund der Links */
}

```

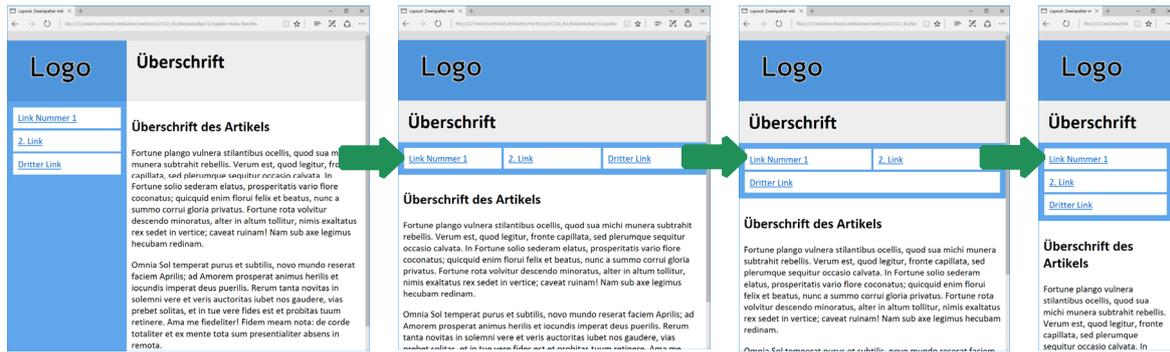
*Nur drei Zeilen CSS sorgen für eine flexible Darstellung der Navigation*

- ① Weisen Sie das Listenelement `ul` an, als Flexbox-Container zu fungieren. Damit die Listeneinträge `li` bei wenig Platz untereinander stehen können, erlauben Sie das Umbrechen mittels `flex-wrap: wrap`
- ② Mit `flex: 1 0 9rem` weisen Sie den Browser an, dass die Listeneinträge wachsen (1), aber nicht schrumpfen dürfen (0 – damit sie nicht kleiner werden als der enthaltene Text). Die Basis-Größe geben Sie mit `9rem` an. Das ist breit genug für die Menüeinträge in diesem Beispiel – in Ihrer eigenen Seite könnte ein anderer Wert notwendig sein oder besser aussehen.
- ③ Der Rest bleibt wie gehabt.

Der entscheidende Unterschied wird erst bei geänderter Bildschirmbreite sichtbar. Auf großen Bildschirmen sieht die Navigation nämlich wie bisher aus, da in die schmale linke Spalte nach wie vor keine zwei Links nebeneinander passen.

Auf schmalere Bildschirmen werden Kopfbereich, Navigation und Inhaltsbereich aber untereinander dargestellt. Jetzt haben die Menüpunkte genügend Platz, um nebeneinander zu stehen.

Auf ganz kleinen Bildschirmen (z. B. Smartphone) reicht der Platz dafür nicht mehr und die Listeneinträge stehen wieder untereinander.



Unterschiedliche Darstellung auf verschiedenen Displays

## 14.5 Benutzerfreundliche Menüs erstellen

Die Gestaltung der Navigationsbereiche ist von großer Bedeutung für die Nutzbarkeit Ihrer Webseite. Eine gute Navigation führt Ihre Besucher zielsicher auf kurzem Weg zu gesuchten Inhalten. Eine undurchdachte Navigation kann den Besucher verwirren. Daher sollten Sie Ihre Navigation nachvollziehbar gestalten: Einerseits sollte die Navigation auffallen und sofort erkennbar sein und sich andererseits optisch harmonisch in das Design Ihrer Seite einfügen.

Obwohl die Gestaltung einer Webseite in vielen Punkten von persönlichen Interessen und Vorlieben abhängt, gibt es einige Punkte, die gerade unerfahrenen Entwicklern helfen können, den Besucher nicht vor Rätsel zu stellen.

- ✓ **Halten Sie Menüs kurz**– pro Ebene sollten nicht mehr als sieben Einträge enthalten sein
- ✓ **Vermeiden Sie Untermenüs.** Wenn das nicht möglich ist, stellen Sie nicht mehr als 5 Menüpunkte pro Ebene dar. Eine zweite oder gar dritte Hierarchie-Ebene muss sich nicht in der Hauptnavigation befinden. Sie können dafür einen weiteren Navigationsblock zum Beispiel in der linken Seitenleiste anbieten.
- ✓ **Machen Sie es wie die anderen.** Eine Navigation befindet sich meistens direkt über oder unter dem Kopfbereich. Da wird sie gefunden und als solche erkannt.
- ✓ **Erfüllen Sie die Erwartungen.** Auf einer Webseite möchte man in aller Regel etwas erledigen: eine E-Mail schreiben, ein Formular ausfüllen, etwas herunterladen oder eine bestimmte Information finden. Dafür sind klare, deutliche Beschriftungen wichtig. Wenn man zum Beispiel auf einen Link „Kontakt“ klickt, möchte man gesammelt möglichst alle Kontaktmöglichkeiten auf einen Blick – von der Anfahrtsskizze über eine E-Mail-Adresse und Telefonnummer bis zur Postanschrift und tatsächlichen Adresse. Nur ein Kontaktformular ist zu wenig oder gar unnötig.
- ✓ **Unterstützen Sie den Nutzer.** Sicher bieten Sie auch Navigationsmöglichkeiten außerhalb des Menüs an. Klickbare Bereiche auf Ihrer Seite sollten sich als solche ausweisen. Von anderen Seiten sind Ihre Besucher es gewohnt, wenn beim Überfahren von Schaltflächen etwas passiert: Farben ändern sich, Bildchen werden eingeblendet, ein anderer Mauszeiger wird verwendet. Links werden sofort als solche erkannt, wenn sie unterstrichen sind. Wenn Sie die Unterstreichung aus ästhetischen Gründen entfernen, sollten Sie unbedingt eine ähnlich **intuitive und deutliche** Kennzeichnung hinzufügen.

Besonders mit der ersten Empfehlung „Halten Sie Menüs kurz“ tut sich mancher zu Recht schwer. Soll eine Navigation mit nur wenigen Einträgen einen Nutzer sinnvoll führen – womöglich unter Verzicht einer weiteren Menüebene – muss eine Webseite gut durchdacht sein.

Ein Ansatz ist es, Beiträge zu verschlagworten und Übersichten aller Artikel zu bestimmten Schlagworten an **passenden** Stellen anzubieten (z. B. über, unter oder neben verwandten Artikeln).

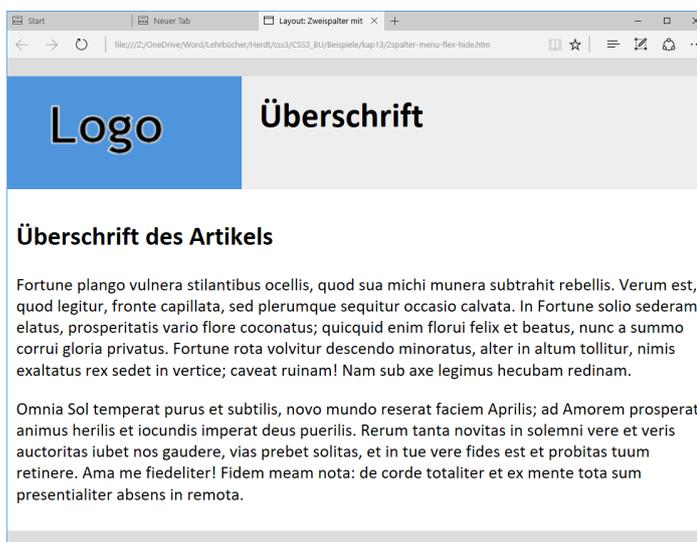
Trotzdem können in der Hauptnavigation schnell mehr Einträge sein, als auf einem Smartphone gut wäre: schließlich sollen Seitenkopf (Logo und Name des Angebotes) plus Menü nicht das Einzige sein, was ein Besucher zu sehen bekommt.

Als kompakte Darstellung für Suche und Menü hat es sich durchgesetzt, diese zu verbergen und erst beim Antippen eines Buttons anzuzeigen. Da sich diese Methode weit verbreitet hat, kann sie verwendet werden. Sie erfüllt die anderen oben genannten Anforderungen an benutzerfreundliche Menüs.

Die Umsetzung lässt sich mit reinem CSS realisieren.

Hierfür benötigen Sie tatsächlich eine Medien-Abfrage, da Sie das Menü nur auf besonders kleinen Bildschirmen verbergen wollen und nur dann, wenn es sich wirklich nicht vermeiden lässt.

Zunächst wird das Menü ausgeblendet. Dazu gibt es mehrere Methoden. Die hier vorgestellte Methode lässt das Menü nur optisch verschwinden, so dass es für Software (z. B. Vorleseprogramme für Blinde) weiterhin zugänglich ist.



Das Menü wird per CSS-Anweisungen vor Sehenden versteckt

## CSS

```
nav ul {
  nav {
    border: 0;           /* Rahmen entfernen */
    clip: rect(0 0 0 0); /* Größe beschneiden auf 0 */
    height: 1px;       /* Größe begrenzen */
    margin: -1px;      /* Um 1px aus dem sichtbaren
                       Bereich verschieben */
    overflow: hidden;  /* Größere Inhalte verbergen */
    padding: 0;        /* Innenabstand entfernen */
    position: absolute; /* Oben links positionieren */
    width: 1px;        /* Größe begrenzen */
  }
}
```

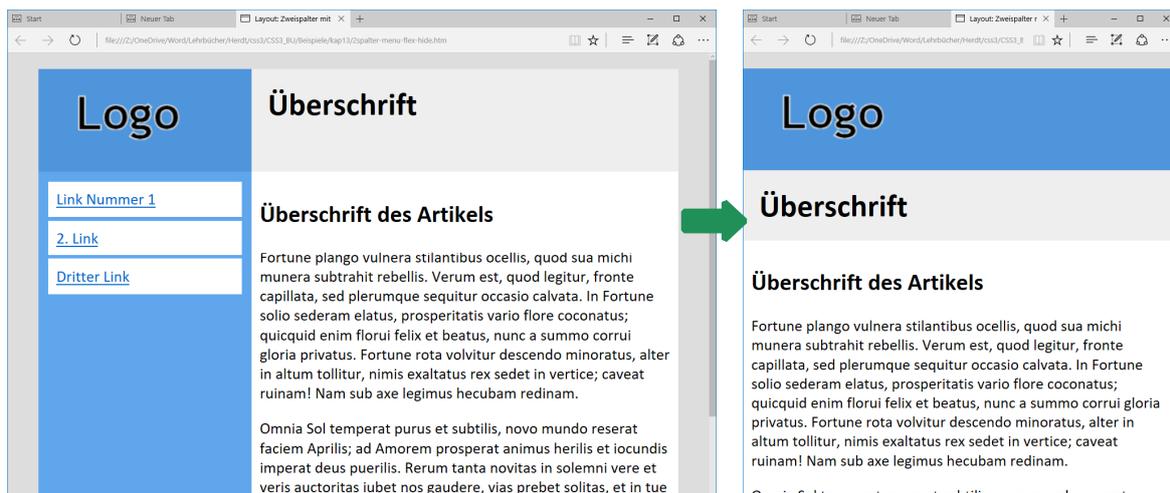
- ✓ Das Menü ist nicht mehr zu sehen, da es aus dem sichtbaren Bereich verschoben wurde. Software wie Vorleseprogramme kümmern sich nicht um die Position und geben es weiterhin aus.

Als nächstes stellen Sie sicher, dass das Menü auf großen Bildschirmen nicht ausgeblendet wird. Dazu nutzen Sie eine Medienabfrage:

### CSS

```
@media screen and (max-width: 30rem) {
  nav ul {
    /* Formate wie oben */
  }
}
```

- ✓ Die gemachten Angaben (die zum Verbergen des Menüs führen) sollen nur für Bildschirme gelten, die maximal 30rem breit sind. Das entspricht bei normaler Schriftgröße 480 Pixeln. Sowohl auf Tablets, als auch auf Desktop-Monitoren wird das Menü angezeigt – wenn die Fenster breit genug sind.



Da das Menü für Smartphone-Nutzer nun gar nicht mehr erreichbar ist, muss eine weitere, vom Benutzer beeinflussbare Bedingung hinzugefügt werden, um das Menü gezielt einblenden und ausblenden zu können.

Eine Technik, die Checkbox-Hack genannt wird, bietet sich hierfür an. Sie macht geschickten Gebrauch von CSS-Selektoren:

### CSS

```
:checked ~ ul
```

- ✓ Hier werden zwei Selektoren miteinander kombiniert. Die Pseudoklasse `:checked` fragt ab, ob eine Checkbox oder ein Radio-Button ausgewählt ist. Mit `~ ul` ist eine Liste gemeint, die der Checkbox folgt (also ein später im Quelltext auftauchendes Geschwister). Verwenden Sie dies statt nur `ul`, so wird die Navigation wieder auf Smartphones angezeigt, da sich vor der Liste keine (ausgewählte) Checkbox befindet.

Diese fügen Sie im nächsten Schritt zur HTML-Datei hinzu:

```
<nav>
  <input type="checkbox">
  <ul>
    <li><a href="#">Link Nummer 1</a></li>
    [...]
```

- ✓ Eine Checkbox vor der Navigation erlaubt das gezielte Ein- und Ausschalten des Menüs.

Jetzt können Sie durch Setzen des Häkchens die Navigation gezielt anzeigen oder verbergen. Da Platz gespart werden soll, verbergen Sie die Navigation standardmäßig, indem Sie der Checkbox das Attribut `checked` mitgeben:

#### HTML

```
<input type="checkbox" checked>
```

Jetzt muss der Nutzer eingreifen, um das Menü anzeigen zu lassen. Es bleibt aber ein Problem: er weiß nicht, wozu das Kästchen dient. Es wird Zeit, es zu beschriften. Diverse frei verfügbare Artikel im Web legen nahe, dass das Wort „Menü“ umgeben von einem Rahmen zumindest im englischsprachigen Raum von den meisten Nutzern als Schalter zum Anzeigen von Menüs erkannt wird (zum Beispiel <http://sitesforprofit.com/mobile-menu-abtest>).

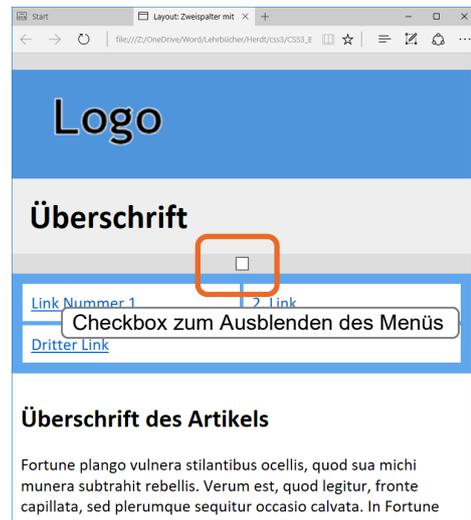
Das lässt sich erreichen, indem Sie ein ohnehin noch fehlendes Element in das HTML-Dokument einfügen: eine Beschriftung für die Checkbox.

#### HTML

```
<input type="checkbox" id="menu-switch">
<label for="menu-switch">Menü</label>
```

- ✓ Beachten Sie, dass Sie der Checkbox eine ID hinzufügen müssen, um die Beschriftung dieser Checkbox mittels `for`-Attribut zuweisen zu können

Das bringt Sie in eine komfortable Lage, denn ein Klick auf die Beschriftung funktioniert genau wie ein Klick auf die Checkbox selber. Das heißt, Sie müssen die Checkbox nicht mehr anzeigen lassen und können sich der Gestaltung des Schriftzuges „Menü“ widmen. Das wiederum geschieht in der CSS-Datei:



## CSS

```
label {  
  border: 1px solid currentColor; /* Rahmen in Schriftfarbe */  
  padding: 0 1em;                /* Innenabstand */  
  color: #0004E8;                /* Dunkelblaue Schriftfarbe */  
  background: #eee;              /* Hellgrauer Hintergrund */  
  cursor: pointer;               /* Mauszeiger als Hand */  
}
```

Nun funktioniert das Menü und es ist verständlich, wie es aufgeklappt werden kann. Ein paar Dinge sind dennoch zu tun. Um Benutzern klar zu machen, dass das Wort „Menü“ klickbar ist, sollte der Button beim Überfahren mit der Maus und beim Antabben sein Aussehen ändern, z. B. Vorder- und Hintergrundfarbe tauschen:

## CSS

```
label:hover {  
  color: #eee;  
  background: #0004E8;  
}
```

Wie eingangs erwähnt, steht das Menü blinden Menschen, die ein Vorleseprogramm nutzen immer zur Verfügung. Sie benötigen den Menü-Button also nicht, ergänzen Sie daher im öffnenden `label`-Tag das Attribut `aria-hidden`:

## HTML

```
<label for="menu-switch" aria-hidden="true">Menü</label>
```

Den vollständigen Code (CSS und HTML) finden Sie im Beispiel-Ordner in den Dateien `kap14/twoColumns-menu-flex-hide.css` und `kap14/2spalter-menu-flex-hide.htm`

 **Lernvideo:** *Flexibles einblendbares Menü.mp4*

## 14.6 Übungen

### Übung 1: Flexbox-Verhalten in verschiedenen Viewportbreiten überprüfen

Level		Zeit	ca. 3 min
Übungsinhalte	✓ Verhalten von Flexboxen kennenlernen		
Übungsdatei	2spalter.htm		
Ergebnisdatei	2spalter-flex.htm		

1. Öffnen Sie die Datei in einem Editor.
2. Passen Sie den Link zur CSS-Datei so an, dass die Datei *twoColumns-flex.css* verwendet wird.
3. Vergrößern und verkleinern Sie das Browserfenster und beobachten Sie das Verhalten der Flexboxen. Führen Sie dasselbe bei starker Vergrößerung oder Verkleinerung der Inhalte durch.

### Übung 2: Mehrspaltiges Tabellen-Layout erzeugen

Level		Zeit	ca. 10 min
Übungsinhalte	✓ Tabellenlayout verändern		
Übungsdatei	2spalter.htm, twoColumns.css		
Ergebnisdatei	uebung3.htm, uebung3.css		

1. Benutzen Sie die Datei *2spalter.htm* als Ausgang für ein 3-spaltiges Layout. Fügen Sie dazu in jeder Zeile der CSS-Tabelle jeweils ein Kind-Element hinzu. Im Hauptbereich benutzen Sie dafür das Element `aside`.
2. Verbreitern Sie die Seite von `40em` auf `50em` und weisen Sie in der ersten Tabellenzeile den Zellen in der CSS-Datei unterschiedliche Breiten zu, um die Breite der Spalten selber festzulegen. Geben Sie die Breiten so an, dass der Artikel den meisten Platz erhält. Beachten Sie bei einer Breitenangabe in `em`, dass die `h1` doppelt so groß ist wie normaler Text.

Logo	<b>Überschrift und Logo</b>	<b>3. Spalte</b>
<ul style="list-style-type: none"> <li>• <a href="#">Link</a></li> <li>• <a href="#">Link</a></li> <li>• <a href="#">Link</a></li> </ul>	<p><b>Artikelüberschrift</b></p> <p>Fortune plango vulnera stillantibus ocellis, quod sua michi munera subtrahit rebellis. Verum est, quod legitur, fronte capillata, sed plerumque sequitur occasio calvata. In Fortune solio sederam elatus, prosperitatis vario flore cocoonatus; quicquid enim florui felix et beatus, nunc a summo cornu gloria privatus. Fortune rota volvitur descendendo minoratus, alter in altum tollitur, nimis exaltatus rex sedet in vertice; caveat ruinam! Nam sub axe legimus hecubam redinam.</p> <p>Omnia Sol temperat purus et subtilis, novo mundo reserat faciem Aprilis; ad Amorem prosperat animus herilis et locundis imperat deus puerilis. Rerum tanta novitas in solemnibus vere et veris auctoritas iubet nos gaudere, vias prebet solitas, et in tue vere fides est et probitas tuam retinere. Ama me fideliter! Fidem meam nota: de corde totaliter et ex mente tota sum presentialiter absens in remota.</p>	<p><b>Weiterführende Informationen</b></p>

Dreispalter mit festgelegten Spaltenbreiten („uebung3.htm“ und „uebung3.css“)

## A

## Anhang: HTML-Element-Referenz

Tag	Verwendung
<a>	Link (Hyperreferenz) und Ziel für Links
<abbr>	Abkürzung
<address>	Angabe zum Autor eines Textes
<area>	Klickbarer Bereich innerhalb einer Grafik
<article>	Ein abgeschlossener Text, der ohne weitere Erklärung oder andere Elemente einer Seite verständlich ist. Zum Beispiel ein Blogeintrag, ein Aufsatz oder eine Nachricht.
<aside>	Text mit zusätzlichen (optionalen) Informationen z. B. zu einem Artikel – der Artikel muss auch ohne diese Informationen Sinn ergeben.
<audio>	Hörbarer Inhalt (z. B. Musik, gesprochene Texte)
<B>	Text, der sich vom umgebenden Text unterscheidet, ohne wichtiger zu sein. Zum Beispiel der zu erklärende Begriff in einem Lexikon-Eintrag.
<base>	Element zur Angabe einer URL-Basis
<bdi>	Bereich für eine Schreibrichtung (z. B. von links nach rechts in Deutschland, von rechts nach links in arabischen Ländern)
<bdo>	Richtungsänderung bei bidirektionalem Text
<blockquote>	Zitat in einem eigenen Block
<body>	Rumpf eines HTML-Dokumentes – die Angaben in diesem Bereich werden vom Browser im Hauptbereich (engl.: Viewport) ausgegeben
 	Zeilenende
<button>	Schaltfläche in Formularen

Tag	Verwendung
<canvas>	Leere Fläche („Leinwand“) zur Nutzung mittels Skripten (z. B. um Animationen ablaufen zu lassen oder einen interaktiven Bereich zu erstellen)
<caption>	Beschriftung von Tabellen
<cite>	Quellenangabe
<code>	Code
<col>	Tabellenspalte
<colgroup>	Gruppe von Tabellenspalten
<datalist>	Liste vordefinierter Einträge für ein Eingabefeld; Einträge werden beim Tippen als Vervollständigungsoption angezeigt
<dd>	Beschreibung eines dd-Elementes innerhalb einer dl-Liste
<del>	Verworfenen Inhalt, macht Änderungen an einem Dokument nachverfolgbar
<details>	Enthält typischerweise einen Langtext und eine Zusammenfassung. Der Langtext soll erst nach einem Klick auf die Zusammenfassung erscheinen
<dfn>	Definition
<dialog>	Dialogfenster (in der Regel ein Overlay oder PopUp)
<div>	Bereich ohne weitere Spezifizierung; wird in der Regel genutzt zur (optischen) Gruppierung von Elementen oder um Bereiche über eine mitgegebene Klasse gemeinsam ansprechen zu können
<dl>	Liste mit dt- und dd-Elementen; zur Bildung von Paaren wie Beschriftung (z. B. Datum) und Information (z. B. 01.01.1970), Glossareintrag und Erläuterung usw.
<dt>	Zu erläuternder Ausdruck in einem dl-Element; Beschriftung, Name eines Glossareintrages o. ä.
<em>	wichtiger, betonter Text
<embed>	Eingebettetes Element; wird über ein Browser-PlugIn abgespielt
<fieldset>	Satz von zusammengehörigen Eingabefeldern; zum Beispiel Angaben zur Person (Name, Vorname, Geburtsdatum) oder zum Anliegen (Betreff, Nachricht, Rückantwort erbeten) können zusammengefasst und mit dem Element legend überschrieben werden
<figcaption>	Beschreibung dessen, was im Element figure enthalten ist (i. d. R. eine Bildunterschrift)
<figure>	Illustration (Bild, Grafik, aber auch Tabelle, Code oder Animation)
<footer>	Fußbereich einer Webseite (body-Element) oder eines der Elemente section, article, aside oder nav
<form>	Formular
<h1 - h6>	Überschrift 1. bis 6. Ordnung

Tag	Verwendung
<head>	Bereich eines HTML-Dokumente, der Metainformationen über das Dokument beinhaltet, also Titel, Autorenangabe, Beschreibung, aber auch technische Informationen zum verwendeten Zeichensatz, zur Darstellung auf unterschiedlichen Geräten sowie Verknüpfungen zu weiteren Dateien, wie Formatvorlagen usw. Dieser Bereich wird vom Browser nicht im Viewport ausgegeben.
<header>	Kopfbereich einer Webseite (body-Element) oder eines der Elemente <code>section</code> , <code>article</code> , <code>aside</code> oder <code>nav</code>
<hr>	Trennelement (dargestellt als horizontale Linie)
<html>	Wurzelement aller HTML-Dokumente. Dieses Element dient als Container für <code>head</code> und <code>body</code> ; alle weiteren Elemente werden innerhalb von <code>head</code> (beschreibende Informationen über das Dokument) oder <code>body</code> (eigentlicher Dokumentinhalt, der im Viewport dargestellt wird) eingesetzt.
<i>	Im Text hervorzuhebender Ausdruck, der meist kursiv dargestellt wird (z. B. in Lexikoneinträgen Angaben zur Herkunft eines Begriffs, lateinische Fachtermini)
<iframe>	In einem Frame eingebettet Webseite
<img>	Inhaltlich relevante Grafik
<input>	Formular-Element, verschiedene Typen möglich, wie (Text-)Eingabefelder, Checkboxes, Absendebutons usw.
<ins>	nachträglich eingefügter Inhalt, macht Änderungen an einem Dokument nachverfolgbar
<kbd>	Über Tastatur einzugebender Text (z. B. in Anleitungen)
<keygen>	Erstellt ein Schlüsselpaar in Formularen. Der lokale Schlüssel wird auf dem absendenden Rechner gespeichert, der öffentliche zum Server gesendet.
<label>	Beschriftung für Formularfelder
<legend>	Überschrift für ein <code>fieldset</code> (Gruppe zusammengehöriger Formularelemente)
<li>	Listeneintrag, muss Kind sein von <code>ol</code> oder <code>ul</code>
<link>	Verknüpft eine Webseite mit anderen Ressourcen, meist Formatvorlagen innerhalb von <code>head</code>
<main>	Hauptbereich einer Seite
<map>	Klickbare Grafik
<mark>	Markiert Text, der hervorgehoben werden soll (z. B. ein gefundener Suchbegriff)
<meta>	Meta-Informationen innerhalb von <code>head</code> . Angaben zum Autor, zum verwendeten Zeichensatz, Kurzbeschreibung usw.
<meter>	Zeigt einen Wert als Balken an, möglicher Höchst- und Tiefstwert müssen mitgegeben werden.

Tag	Verwendung
<nav>	Hauptnavigationbereich einer Seite
<noscript>	Inhalt wird ausgegeben, wenn der Browser kein JavaScript unterstützt oder JavaScript vom Nutzer abgeschaltet wurde
<object>	Eingebettetes Objekt, meist Multimedia-Inhalte, zunehmend durch canvas, audio und video ersetzt
<ol>	Liste, die nachvollziehbar sortiert ist (z. B. nach Datum)
<optgroup>	Gruppierung von Einträgen innerhalb einer Auswahlliste
<option>	Eintrag einer Auswahlliste
<output>	Bereich für Ausgaben von Berechnungen
<p>	Absatz
<param>	Parameter von object-Elementen
<pre>	Vorformatierter Text (wird so ausgegeben, wie geschrieben, d. h. mehrere Whitespaces werden <b>nicht</b> zu einem Leerzeichen zusammengefasst)
<progress>	Fortschrittsbalken, nicht als Balkendiagramm (meter) zu nutzen; muss, damit ein Fortschritt erkennbar ist, mittels JavaScript gesteuert werden
<q>	Wörtliche Rede
<rp>	Inhalt einer Ruby-Annotation
<rt>	Erklärung einer Ruby-Annotation
<ruby>	Ruby-Annotationen, in der mittels rt und rp fernöstliche Zeichen erklärt werden
<s>	Überholter (nicht mehr gültiger) Text
<samp>	Beispiel
<script>	Enthält Skripte oder ruft externe Skripte auf (i. d. R. JavaScript)
<section>	Sinnzusammenhang (enthält z. B. mehrere Artikel zu einem Thema wie Sport, Inland, Ausland oder kennzeichnet innerhalb eines Artikels ein Kapitel)
<select>	Auswahlliste
<small>	„Kleiner Text“ enthält rechtliche Hinweise, Disclaimer usw.
<source>	Mehrere mögliche Quellen zum Abspielen eines multimedialen Inhaltes, aus denen der Browser wählt (je nachdem, welches Format er beherrscht)
<span>	Unspezifischer Inhalt (wird i. d. R. genutzt, um per CSS oder JavaScript über eine mitgegebene Klasse oder ID darauf zugreifen zu können)
<strong>	Wichtiger Text
<style>	Formatierungsangaben
<sub>	Tiefgestellter Text (z. B. in Formeln)
<sup>	Hochgestellter Text (z. B. in Formeln)

Tag	Verwendung
<table>	Tabelle
<tbody>	Tabellenrumpf (enthält die tabellarischen Daten)
<td>	Tabellarisches Datum (wird als Tabellenzelle dargestellt)
<textarea>	Mehrzeiliges Texteingabefeld
<tfoot>	Fuß einer Tabelle, z. B. für Angaben zur Aktualität oder zum Urheber der Daten
<th>	Tabellenüberschrift
<thead>	Tabellenkopf (enthält Überschriften)
<time>	Datums- oder Zeitangabe
<title>	Name eines HTML-Dokuments
<tr>	Tabellenzeile
<track>	Untertitel, Beschriftungen für audio oder video
<ul>	Unsortierte Liste (Einträge sind NICHT nachvollziehbar sortiert)
<var>	Variable
<video>	Filmischer Inhalt
<wbr>	Zeigt eine bevorzugte Stelle für einen Zeilenumbruch an

# B

## Anhang: Editoren und Entwicklungsumgebungen

### B.1 WYSIWYG-Editoren

Der Anspruch von **WYSIWYG-Editoren** lautet: „what you see is what you get“ (engl. für „Was man im Editor sieht, wird man auch im Browser sehen“). Die Idee ist, Menschen ohne Programmierkenntnissen die Erstellung von Webseiten zu ermöglichen. Um dieses Ziel zu erreichen, versuchen die Hersteller solcher Programme die Fähigkeiten von HTML über eine Oberfläche bereit zu stellen, die an eine Textverarbeitung erinnert. So können Sie Überschriften erstellen, indem Sie einen Text markieren und einen Button drücken, der daraus eine Überschrift macht – also so, wie auch in einer Textverarbeitung.

WYSIWYG-Editoren ermöglichen es oft, in eine Codeansicht zu wechseln, in der man den Code wie in einem Texteditor bearbeiten kann.

Das Problem mit dem solche Editoren zu kämpfen haben ist der große Umfang an Gestaltungsmöglichkeiten, die Webentwickler zur Verfügung stehen. Um diese den Nutzern solcher Editoren bereit zu stellen, ist der Funktionsumfang stetig gestiegen. Daher ist das Erlernen solcher Programme in aller Regel aufwändig. Die Ergebnisse sind dennoch nicht so ordentlich, wie händisch erstellte Webseiten.

Aufgrund des hohen Funktionsumfangs sind auch die Kosten für solche Programme oft beträchtlich. Dennoch halten diese oft nicht mit der schnellen Entwicklung von Webtechniken mit.

Programme, die zu dieser Gattung gehören, sind Adobe Dreamweaver oder Microsoft Expression Web (beide kostenpflichtig).

## B.2 Integrierte Entwicklungsumgebungen (IDE)

Wahre Alleskönner sind **integrierte Entwicklungsumgebungen** (Abkürzung **IDE**, von englisch *integrated development environment*).

Diese haben ebenfalls einen immensen Funktionsumfang, der es ermöglicht, nicht nur die HTML-Dokumente zu editieren, sondern auch alle weiteren Arbeiten innerhalb einer Programmoberfläche erledigen zu können. Dazu gehören u. a. das Veröffentlichen der Website, der Abgleich mit einem Versionierungs-Dienst oder eine umfangreiche Projektverwaltung.

**IDEs empfehlen sich für komplexe Projekte und fortgeschrittene Anwender**, Anfänger stehen dem immensen Funktionsumfang zunächst ratlos gegenüber und die Einarbeitungszeit ist hoch.

Bekannte Vertreter für IDEs sind Eclipse, Visual Studio oder NetBeans.

## B.3 Reine Texteditoren

Die saubersten Ergebnisse und den kompaktesten Code (also schnell ladende Seiten) erhalten Sie, wenn Sie Ihren Code selber schreiben. Auch für diese Arbeit gibt es spezielle Editoren, die Ihre Arbeit extrem beschleunigen. Wenn Sie damit erst einmal Erfahrung gesammelt haben und sicher in der Anwendung von HTML-Elementen sind, werden Sie WYSIWYG-Editoren nicht vermissen. Am besten, Sie gewöhnen sich von Anfang an eine saubere Arbeitsweise im Texteditor Ihrer Wahl an.

Es ist schwierig, eine allgemeine Empfehlung für einen Editor auszusprechen, da es von der persönlichen Arbeitsweise und dem Aufgabengebiet eines jeden Entwicklers abhängt, welcher Editor besonders geeignet ist.

Ein guter Editor sollte Ihnen vor allem umfangreiche Möglichkeiten zur Konfiguration bereitstellen, damit Sie ihn an Ihre persönlichen Vorlieben und Arbeitsweise anpassen können.

Texteditoren sind sehr zahlreich, hier ein paar Beispiele beliebter Editoren: *jEdit*, *UltraEdit* oder *Notepad++*. Besonders populär ist der kostenpflichtige *Sublime Text*.

Das GitHub-Projekt baut diesen Editor als OpenSource-Variante unter dem Namen *Atom* nach.

Über Erweiterungen lassen sich Funktionen nachrüsten, die Sie für Ihre Arbeit benötigen, ohne dass Sie unnötigen Ballast mit sich herumschleppen.

## Brackets von Adobe – Spezialist für Frontend-Entwicklung

**Brackets** von Adobe nimmt eine Sonderstellung unter den Texteditoren ein, weil er sich speziell an **Frontend-Entwickler** richtet. Frontend-Entwickler sind jene, die das Aussehen und Verhalten einer Webseite beeinflussen. Dies geschieht mit HTML, CSS und JavaScript.

In diesem Buch und in den Büchern *Cascading Style Sheets (CSS3)* und *JavaScript* werden ausschließlich Frontend-Techniken vermittelt, so dass Sie *Brackets* gut verwenden können, um die Beispiele in diesen Büchern nachzuvollziehen.

Ein paar Features machen die Arbeit mit *Brackets* besonders effizient. Wenn Sie statt einer einzelnen Datei einen Ordner öffnen, nimmt *Brackets* an, dass alle enthaltenen Dateien zu einem Projekt gehören. So entfällt das oft umständliche Anlegen eines Projektes, wie Sie es bei anderen Editoren finden. Trotzdem sind projektweite Aktionen wie *Suchen*, *Ersetzen* usw. möglich.

Für ein solches Projekt können Sie die Live-Vorschau aktivieren (setzt *Google Chrome* als Browser voraus). Auf diese Weise wirkt sich jede Änderung am Dokument direkt auf der Webseite aus, an der Sie gerade arbeiten. Für andere Editoren sind (oft kostenpflichtige) Erweiterungen nötig und zusätzlich eine Erweiterung im unterstützten Browser. Gerade Anfängern ist es eine große Hilfe, wenn Sie unmittelbar sehen, was die durchgeführten Änderungen bewirken.

.					
.htm	17				
.html	17				
<b>&lt;</b>					
<a href="#">	84, 85				
<a href>	77, 84				
<acronym>	55				
<blockquote>	51				
<body>	16				
 	22, 24				
<caption>	101				
<dd>	93				
<details>	55				
<dl>	93				
<dt>	93				
<figcaption>	55, 61				
<figure>	61				
<form>	113				
<h1> bis <h6>	25				
<head>	16				
<html>	16				
<id>	83				
<img>	22, 58				
<li>	91, 92				
<link>	36				
<meta>	139, 150				
<meta>-Tags	150				
<ol>	92				
<optgroup>	126				
<p>	10, 21, 23, 25				
<q>	55				
<select>	123				
<span>	169				
<strong>	55				
<style>	32				
<table>	99				
<tbody>	102				
<tfoot>	102				
<th>	100				
<thead>	102				
<ul>	91				
<b>A</b>					
Absätze	23				
Absolute Pfade	47				
Absolute Werte	40				
alt	58, 59				
Alternativtext für Grafiken	58				
Animation	61				
Anker	83				
ASP	114				
Attribute	22				
Attributnamen	22				
Attributwert	22				
Audio Dateien	67				
Auflösung	62				
Aufzählungen	90				
Aufzählungen, Startwert	92				
Ausrichtung	165				
Auswahllisten	115, 123				
Auszeichnungssprache	10				
<b>B</b>					
Beschreibung der Seiteninhalte	150				
Beschreibungslisten	90, 93				
Bilder, Auflösung	62				
Bildpunkte	62				
Bildschirmlesegeräte	26				
Blockelemente	48, 51				
bold	165				
Browser	10, 14, 16, 18, 19				
Browser, Übersicht	143				
Browserstatistik	145				
Button	115				
<b>C</b>					
Cascading Style Sheets	13				
CGI	114				
checkbox	115, 126				
cite	51				
class	169				
Code	17				
Code validieren	141				
Code, Leerräume	17				
Code, Whitespace	17				
color	116, 165				
colspan	104				
CSS	13				
CSS, Maßeinheiten	40				
CSS, Online-Referenz	14				
CSS-Regeln	28, 31, 32				
CSS-Regeln gruppieren	33				
Cyberduck	161				
<b>D</b>					
datalist	116				
date	116				
Dateiname, HTML	82				
Dateinamen, Konventionen	44				
Dateinamenserweiterungen anzeigen	44				
Deklaration	31				
Dezimaltrennzeichen	40				
DNS-Registrierungsstelle	158				
Dokumentkopf	16				
Dokumentrumpf	16				
Dokumenttyp-Definition	15				
Dokumenttyp-Deklaration	138				
Domainname	82				
Domainnamen registrieren	157				
Domainnamenregistrierung, Rechte Dritter	159				
DTD	138				
DTD Frames	148				
DTD Transitional	148				
Durchgestrichen	165				
<b>E</b>					
Editor	14, 15				
Eigenschaft-Wert-Paar	31				
Eingabefelder	115				
Eingabefelder, einzeilige	117				
Eingabefelder, mehrzeilige	122				
Eingebettete Styles	28				
Einstiegsseite	17				
Einzug, erste Zeile	165				
Elemente	21				
Elemente verschachteln	26, 53				
Elemente, äußere	26				
Elemente, innere	26				
Elemente, leere	22				
Eltern-Elemente	26, 53				
E-Mail-Adressen	78				
Ems	41				
Endtags	19, 21				
Erweiterungen, Dateinamen	44				
Exes	41				
Extensible Hypertext Markup Language	135				
Externe Styles	29				
<b>F</b>					
Farben, websichere	39				
Farbnamen	37				
Farbwerte	32				
Farbwerte in CSS definieren	37				
Favoriten	16				
Fehlerbehebung	141				
Fettdruck	165				
File Transfer Protocol	161				
FileZilla	161				
Firebug	143				
Firefox	144				
FireFTP	161				
font	170				
font-family	167				
font-size	165				
font-weight	165				
Formatierung mit Tags	30				
Formulardaten ausführen	114				
Formulardaten codieren	115				
Formulardaten versenden	114				
Formulare	112				
Foto-Stocks	58				
FTP	161				
FTP-Client	161				

<b>G</b>		Hypertext Markup Language (HTML)	10	mailto	78, 115
Geschwister-Elemente	53	Hypertext Transfer Protocol Secure (https)	47	Maßeinheiten	40
GET	114	<b>I</b>		method-Attribut	113
Gewicht	165	id	170	Microsoft Paint	64
GIF-Format	61	id-Attribut	113	Millimeter	40
GitHub	147	image	115	multiple	123
Grafikdatei, Hyperlink zu	80	Inches	40	Musikdatei, Hyperlink zu	80
Grafiken	58	Inhaltsbeschreibung	150	<b>N</b>	
Grafiken einfügen	58	Inhaltstyp	139	Nach oben in der Ordnerstruktur wechseln	46
Grafiken verkleinern	63	Inline Spans	169	Nachfahren-Elemente	53
Grafiken, Alternativtext	58	Inline Styles	28	number	117
Grafiken, Dateigröße verringern	62, 63	Inline-Elemente	48, 51	Nummerierte Listen	92
Grafikformate	61	<b>J</b>		Nutzerführung	76
Grafikformate, Übersicht	62	JPG-Format	61	<b>O</b>	
Groß- und Kleinschreibung	23, 44	jQuery-Plugins	147	Online-Referenzen	14
<b>H</b>		<b>K</b>		Optionen gruppieren	126
Hauptnavigation	76	keyword stuffing	155	Optionsfelder	127
height	63	keywords	59, 151	Ordner erstellen	45
Hervorhebung einzelner Wörter	169	Kind-Elemente	26, 53, 176	Ordnernamen, Konventionen	44
Hexadezimalwerte	39	Klassen	169	Ordnerstruktur	43
Hex-Codes	39	Kleinschreibung	23	<b>P</b>	
Hoster	159	Kommentare, CSS	57	Paint	64
Hosting	160	Kommentare, HTML	56	password	115
HTML	10	Kontrollkästchen	126	Passwort	114
HTML 4	138	Konventionen, Ordner- und Dateinamen	44	Perl	114
HTML 4.01	134	<b>L</b>		Pfade	45
HTML 4.01 Frameset	139	LaTeX	12	Pfade, absolute	47
HTML 4.01 Strict	138	Leere Elemente	22	Pfade, relative	45
HTML validieren	141	Leerschritte, CSS	31	PHP	114
HTML zu XHTML konvertieren	143	Leerzeichen	44	Pixel	40, 62
HTML, Grundgerüst	15	letter-spacing	166	Plausibilitätsprüfungen, Formulare	115
HTML, Online-Referenz	14	Links	10, 76	PNG-Format	61
HTML, Versionen	134	Links, absolute	47, 81	Portnummer	82
HTML5	13, 135	Links, relative	45, 77, 78	POST	113
HTML5-Boilerplate	145	Links, seiteinterne	83	Protokoll	82
HTML-Code	19	Listen	90	Provider	159
HTML-Code eingeben	19	Listen kombinieren	96	Prozent	41
HTML-Code von Webseiten anzeigen	20	Listen verschachteln	94	Punkt	40
HTML-Dateiname	82	Listen, Beschreibungs-	90	<b>Q</b>	
HTML-Element	10	Listen, nummerierte	92	q, Element	51
HTML-Elemente, Übersicht	54	Listen, sortierte	90, 92	Quellcode anzeigen	20
HTML-Tags	10, 19	Listen, Startwert	92	Quelltext	18
HTML-Tags, veraltete	30	Listen, unsortierte	90, 91	<b>R</b>	
Hyper Text Transfer Protocol (http)	47	Listenfeld, aufklappbares	123	radio	115
Hyperlinks	10	Lizenzbestimmungen	58	Radiobutton	127
Hyperlinks in einer Webseite	76	<b>M</b>		range	116
Hyperlinks zu E-Mail	78	Mac Vorschau	64	Ranking	26, 155
Hyperlinks, absolute	47	Macintosh TextEdit	15	Regeln, CSS	31
Hyperlinks, relative	45	<b>Nach oben in der Ordnerstruktur wechseln</b>		Reihenfolge, Tags	53
Hyperlinks, Ziele	86				
Hyperlinkziel	83				
Hypertext	10				

Relative Pfade	45	Suchdienste, Anmeldung	156	Verläufe	146
Relative Werte	41	Suchergebnisse	26	Veröffentlichen	161
Remote-Ordner	162	Suchmaschinen	11	Verschachtelungsfehler	26
Rendering	11	Suchmaschinen, Ranking	155	Verschachtlung	53
Rendering Engine	11	Suchmaschinenoptimierung	156	Verweise	76
Ressourcen	43			Videodatei, Hyperlink zu	80
RGB-Werte	38	<b>T</b>		Videos	65
root directory	15	Tabelle, CSS	101	Vorfahren-Elemente	53
rowspan	105	Tabellen	99	Vorschau	64
		Tabellen, Tags	100		
<b>S</b>		Tabellenfuß	102	<b>W</b>	
Safari	144	Tabellenkopf	102	W3C	134, 135
Schaltfläche	115	Tabellenkörper	102	W3C-Validator	141
Schaltflächen, grafische	128	Tabellenüberschrift	101	Web Developer	143
Schlüsselwörter	59	Tabellenzellen, leer	100	Webclient	17
Schlüsselwörter, Schriftgrößen	168	Tag, öffnendes	10	Webseite erstellen	16
Schlüsselwörter, Webseite	151	Tag, schließendes	10	Webseite speichern	17
Schriftart, Serifen	164	Tags	10, 19	Webseiten validieren	141
Schriftart, Standard	164	Tags, Reihenfolge	53	Webserver	17, 159
Schriftarten	164	Tags, veraltet	14	Webserver, Verbindung zum	161
Schriftfamilien	164, 166	Tags, veraltete	30	Websichere Farben	39
Schriftfamilien, generische	166	target-Attribut	113	Webspace	159
Schriftgewicht	165	text	115	Webstandards	135
Schriftgröße	165	text-align	165	Werte	40
Schriftgrößen, Schlüsselwörter	168	textarea	122	Whitespace, Code	17
Schriftklassen	166	Textdatei, Hyperlink zu	80	width	63
Schriftliste	167	TextEdit	15	Windows Editor	15
Schriftschnitt	165	Texteditor	14, 15	WISE-FTP	161
Schriftschnitte	166	Texteffekte	165	word-spacing	166
Screenreader	10	Texteigenschaften in CSS		World Wide Web Consortium	134
search	116	festlegen	164	Wörter hervorheben	169
selected	123	time	116	Wurzelverzeichnis	15, 43
Selektor	31	Tooltip für Grafiken	60		
Serifen	164	Transparenz	61	<b>X</b>	
Skalierbarkeit	61			XHTML	135, 138
SmartFTP	161	<b>U</b>		XHTML 1.0 Frameset	139
Sonderzeichen	44	Überschriften	25	XHTML 1.0 Strict	138
Sortierte Listen	90, 92	Überschriften der Tabelle	101	XHTML 1.0 Transitional	138
Spalten verbinden	104	Überschriften, Ebenen	25		
Spans, Inline	169	Überschriften, Spalten	100	<b>Z</b>	
Speicherplatz	159	Umlaute	44	Zeichencodierung	139
Speicherplatz, Webserver	159	Unsortierte Listen	90, 91	Zeichen-Entities	148
src	58	Unterstrichen	165	Zeichensatz	22
Stammordner erstellen	15	URI	81	Zeilen verbinden	105
Standardschriftart	164	URL	47	Zeilennummierung	20
Standardschriftfamilie	167	URL	47	Zeilennumbrüche auf Webseiten	24
Startseite	17, 43	UTF-8	22, 56	Zeilennumbrüche im Code	17
Starttags	19, 21			Zeilennumbrüche, CSS	31
Stichwort	151	<b>V</b>		Zentimeter	40
Styles, eingebettete	28	Validator	141, 148	Ziele von Hyperlinks	86
Styles, externe	29	Vererbung überschreiben	35		
Styles, inline	28	Vererbung, CSS	33		
Stylesheets, eingebettete	32	Verknüpfungen, absolute	47		

---

# Impressum

Matchcode: HTML5

Autor: Marc Haunschild

Produziert im HERDT-Digitaldruck

4. Ausgabe, 2. Aktualisierung, November 2017

HERDT-Verlag für Bildungsmedien GmbH  
Am Kümmerling 21-25  
55294 Bodenheim  
Internet: [www.herdtd.com](http://www.herdtd.com)  
E-Mail: [info@herdtd.com](mailto:info@herdtd.com)

© HERDT-Verlag für Bildungsmedien GmbH, Bodenheim

Alle Rechte vorbehalten. Kein Teil des Werkes darf in irgendeiner Form (Druck, Fotokopie, Mikrofilm oder einem anderen Verfahren) ohne schriftliche Genehmigung des Verlags reproduziert oder unter Verwendung elektronischer Systeme verarbeitet, vervielfältigt oder verbreitet werden.

Dieses Buch wurde mit großer Sorgfalt erstellt und geprüft. Trotzdem können Fehler nicht vollkommen ausgeschlossen werden. Verlag, Herausgeber und Autoren können für fehlerhafte Angaben und deren Folgen weder eine juristische Verantwortung noch irgendeine Haftung übernehmen.

Wenn nicht explizit an anderer Stelle des Werkes aufgeführt, liegen die Copyrights an allen Screenshots beim HERDT-Verlag. Sollte es trotz intensiver Recherche nicht gelungen sein, alle weiteren Rechteinhaber der verwendeten Quellen und Abbildungen zu finden, bitten wir um kurze Nachricht an die Redaktion.

Die in diesem Buch und in den abgebildeten bzw. zum Download angebotenen Dateien genannten Personen und Organisationen, Adress- und Telekommunikationsangaben, Bankverbindungen etc. sind frei erfunden. Eventuelle Übereinstimmungen oder Ähnlichkeiten sind unbeabsichtigt und rein zufällig.

Die Bildungsmedien des HERDT-Verlags enthalten Verweise auf Webseiten Dritter. Diese Webseiten unterliegen der Haftung der jeweiligen Betreiber, wir haben keinerlei Einfluss auf die Gestaltung und die Inhalte dieser Webseiten. Bei der Bucherstellung haben wir die fremden Inhalte daraufhin überprüft, ob etwaige Rechtsverstöße bestehen. Zu diesem Zeitpunkt waren keine Rechtsverstöße ersichtlich. Wir werden bei Kenntnis von Rechtsverstößen jedoch umgehend die entsprechenden Internetadressen aus dem Buch entfernen.

Die in den Bildungsmedien des HERDT-Verlags vorhandenen Internetadressen, Screenshots, Bezeichnungen bzw. Beschreibungen und Funktionen waren zum Zeitpunkt der Erstellung der jeweiligen Produkte aktuell und gültig. Sollten Sie die Webseiten nicht mehr unter den angegebenen Adressen finden, sind diese eventuell inzwischen komplett aus dem Internet genommen worden oder unter einer neuen Adresse zu finden. Sollten im vorliegenden Produkt vorhandene Screenshots, Bezeichnungen bzw. Beschreibungen und Funktionen nicht mehr der beschriebenen Software entsprechen, hat der Hersteller der jeweiligen Software nach Drucklegung Änderungen vorgenommen oder vorhandene Funktionen geändert oder entfernt.