

Technologien für das Erstellen von Webseiten

D. A. Waldvogel

Version 5.0

Inhalt

1	TBZWIKI	3
2	HTML Tutorial	3
2.1	Kurzreferenz: HTML	3
2.2	Tutorials HTML	4
2.3	Grundgerüst einer HTML-Datei	5
3	CSS Tutorial	6
4	Dynamische Webseiten	7
4.1	Was ist JavaScript	8
4.2	Einbinden von JavaScript in HTML-Dokumente	9
4.3	Eingabe überprüfen	10
4.4	JavaScript Tutorial	12
4.5	jQuery Tutorial	13
5	Formulare und Auswertung	14
5.1	Server-seitig PHP	15
5.2	PHP Tutorial	15
5.3	PHP – Tutorial für Fortgeschrittene	16
5.4	Datenbankzugriff mit MySQLi oder PDO	17
5.5	fertige Beispiele zur PHP-Programmierung	17
5.6	Empfehlenswertes zum PHP-Tutorial	17
6	Ajax	18
7	Verwenden von Bildern	19
8	Webseiten validieren lassen	20
9	Selbsttest	20
10	weitere Interessante Links	21
10.1	Beispiele von eigenen Projekte	21

1 TBZWIKI

Viele Informationen finden Sie auf dem TBZ Wiki-System Modul 307:

http://www.tbzwiki.ch/index.php?title=Modul_307

2 HTML Tutorial

<http://de.html.net/>

- [Einführung](#)
Eine kurze Einführung in das Tutorial und was Sie lernen können werden.
- [Lektion 1: Fangen wir an!](#)
Welche Werkzeuge brauchen Sie um Ihre eigene Webseite zu erstellen.
- [Lektion 2: Was ist HTML?](#)
Verstehen Sie was HTML ist und bedeutet.
- [Lektion 3: Elemente and Tags?](#)
Was sind Elemente und Tags und wofür werden Sie benötigt.
- [Lektion 4: Ihre erste Webseite](#)
Erstellen Sie Ihre erstes HTML-Dokument – die Grundlage aller Ihrer zukünftigen Seiten.
- [Lektion 5: Was Sie bisher gelernt haben!](#)
Wiederholung von dem, was Sie bisher gelernt haben und Ausblick auf das, was Sie in den nächsten Lektionen erwartet.
- [Lektion 6: Einige weitere Elemente](#)
Machen Sie sich mit den sieben meistgebrauchten Elementen vertraut.
- [Lektion 7: Attribute](#)
Lernen Sie den Tags zusätzliche Angaben hinzuzufügen und Kommandos direkt zu formulieren.
- [Lektion 8: Links](#)
Finden Sie heraus, wie Sie Links zu Ihren eigenen und anderen Seiten im Netz setzen können.
- [Lektion 9: Bilder](#)
Entdecken Sie, wie einfach es ist, Bilder in Ihre Webseiten einzubinden.
- [Lektion 10: Tabellen](#)
Konstruieren Sie HTML-Tabellen, um strukturierte Inhalte zu präsentieren.
- [Lektion 11: Mehr über Tabellen](#)
Erstellen Sie noch anspruchsvollere Tabellen.
- [Lektion 12: Layout \(CSS\)](#)
Verstehen Sie, wie Cascading Style Sheets (CSS) verwendet werden können, um Ihren Seiten ein fantastisches Layout zu verpassen.
- [Lektion 13: Seiten ins Netz laden](#)
Finden Sie heraus, wie Sie Ihre Seiten publizieren können, damit auch andere Leute diese ansehen können.
- [Lektion 14: Web-Standards und Validierung](#)
Lesen Sie über den HTML-Standard und wie Sie die Richtigkeit Ihres Codes kontrollieren können.
- [Lektion 15: Abschließende Hinweise](#)
Ein paar gute Ratschläge für Ihre Webprojekte.

2.1 Kurzreferenz: HTML

<http://de.selfhtml.org/navigation/html.htm>

2.2 Tutorials HTML

Bei *w3schools* gibt es sehr gute Tutorial. Das Tutorial ist in einem leichtverständlichen Englisch geschrieben. Die Beispiele können gerade mit einem Online-Editor ausprobiert werden

<http://www.w3schools.com/html/>

w3schools.com

HTML CSS JAVASCRIPT SQL PHP BOOTSTRAP JQUERY ANGULAR XML TUTORIALS

HTML Tutorial

HTML HOME

HTML Introduction
HTML Editors
HTML Basic
HTML Elements
HTML Attributes
HTML Headings
HTML Paragraphs
HTML Styles
HTML Formatting
HTML Quotations
HTML Computercode
HTML Comments
HTML Colors
HTML CSS
HTML Links
HTML Images
HTML Tables
HTML Lists
HTML Blocks
HTML Classes
HTML Layout
HTML Responsive
HTML iframes
HTML JavaScript
HTML Head
HTML Entities
HTML Symbols
HTML Charset
HTML URL Encode
HTML XHTML

HTML Forms

HTML Forms
HTML Form Elements
HTML Input Types
HTML Input Attributes

Zu HOST EUROPE wechseln
Aus über 50 Domain-Endungen wählen. Jetzt Markennamen im Netz sichern!

HTML(5) Tutorial
« W3Schools Home

 With HTML you can create your own Web site.
This tutorial teaches you everything about HTML.
HTML is easy to learn - You will enjoy it.

Examples in Every Chapter
This HTML tutorial contains hundreds of HTML examples.
With our online HTML editor, you can edit the HTML, and click on a button to view the result.

Example

```
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
</head>
<body>

<h1>This is a Heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

Try it Yourself »

Das Tutorial für Anfänger von *seminar.de* ist ebenfalls sehr empfehlenswert. In diesem Tutorial lernt man, wie man eigentliche Webprojekte plant und erstellt.

<http://www.html-seminar.de/einsteiger.htm>

Startseite

HTML Tutorial für Anfänger

- benötigte Software
- erste HTML-Seite
- Grundlagen - HTML-TAG
- Absatz erstellen - p
- Überschriften
- HTML-Editor
- HTML-Grundgerüst
- Text fett b - strong
- Text kursiv i oder em
- Schachteln von HTML-TAGs
- unterstreichen
- validieren HTML-Seite
- Sauberer HTML-Code
- Listen, Aufzählungen
- Definitionsliste
- Attribute bei HTML-TAGs
- Links, Hyperlinks
- erzwungener Umbruch
- Trennlinie, Attribute
- Bilder, Fotos u. Grafiken integrieren
- HTML Farben
- Tabellen
- Online stellen
- HTML Befehle Übersicht

HTML Tutorial für Anfänger

Und nun geht es los. Zu Beginn kommen die Grundlagen - dabei wird teilweise mit Vereinfachungen gearbeitet. Wichtig ist mir, dass Sie schnell einen Erfolg sehen und dass die Zusammenhänge verständlich werden.

Im Verlauf des Kurses vervollständigen wir nach und nach unser Wissen, damit korrekte Seiten herauskommen.

Aufbau vom Kurs

Was wird benötigt

Im ersten Schritt sehen wir uns an, welches „Werkzeug“ wir benötigen. Dann geht es zum eigentlichen Lernen von HTML und CSS.

Hier schauen wir uns den aktuellen Standard an. Dies hat den Vorteil, dass wir keine Zeit für veraltete Befehle und deren Attribute verschenden. Diese veralteten HTML-Befehle sollten auch nicht mehr eingesetzt werden.

Inhalt dieser Seite:

Vorstellung des Kursaufbaus

- [HTML Tutorial für Anfänger](#)
- [Aufbau vom Kurs](#)
 - [Was wird benötigt](#)
 - [HTML für Inhalt & Struktur](#)
 - [Design für Websites](#)
 - [Projekt-Websites](#)
 - [Provider eigener Domainnamen](#)
 - [Online stellen](#)
- [Hilfe bei Fragen](#)

2.3 Grundgerüst einer HTML-Datei

<http://www.html-seminar.de/html-seitenaufbau.htm>

<http://de.selfhtml.org/html/allgemein/grundgeruest.htm>

Sehr guter Aufbau mit Kopiervorlage!

3 CSS Tutorial

<http://www.css4you.de/wscss/css02.html>

<http://de.html.net/tutorials/css/>

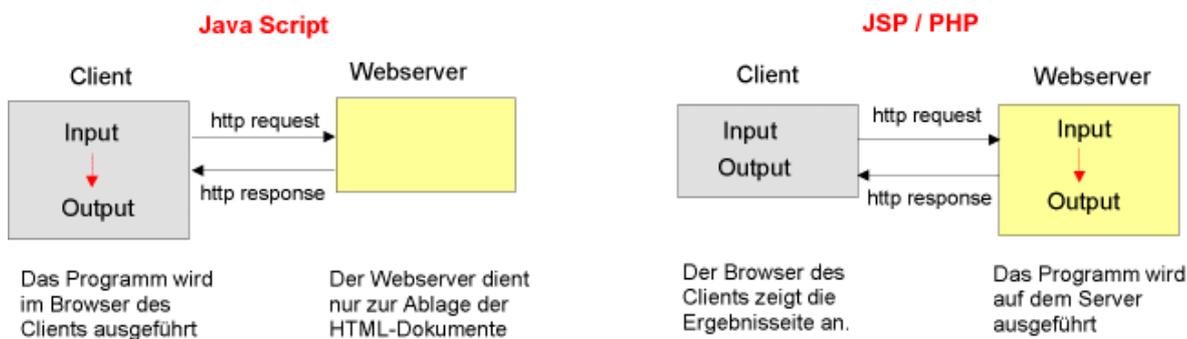
- [Einführung](#)
Eine kurze Einführung in das Tutorial und was Sie lernen können werden.
- [Lektion 1: Was ist CSS?](#)
Ein klein wenig darüber, warum CSS entwickelt wurde und warum es clever ist, CSS anstatt HTML beim layouten und designen zu verwenden.
- [Lektion 2: Wie funktioniert CSS?](#)
Lernen Sie die grundlegende Syntax und erstellen Sie Ihr erstes Stylesheet.
- [Lektion 3: Farben und Hintergründe](#)
Lernen Sie, wie sie Ihrer Webseite Farben und Hintergrundfarben hinzufügen und wie Sie Hintergrundbilder verwenden.
- [Lektion 4: Zeichensätze \(Fonts\)](#)
In dieser Lektion werden Sie etwas über Schriftarten erfahren und wie Sie diese mit Hilfe von CSS einsetzen können.
- [Lektion 5: Text](#)
In dieser Lektion werden Sie in die unglaublichen Möglichkeiten eingeführt, die Ihnen CSS bietet, um Ihren Text zu layouten.
- [Lektion 6: Links](#)
Wie können Sie Ihren Links tolle und nützliche Effekte geben und wie arbeitet man mit Pseudoklassen.
- [Lektion 7: Identifikation und Gruppierung von Elementen \(class und id\)](#)
Ein genauer Blick auf die Verwendung von `class` und `id`, um Eigenschaften für bestimmte Elemente festzulegen.
- [Lektion 8: Gruppierung von Elementen \(span und div\)](#)
Ein genauer Blick auf die Verwendung von `span` und `div`, weil genau diese beiden HTML-Elemente von zentraler Bedeutung für CSS sind.
- [Lektion 9: Das Box-Modell](#)
Das Box-Modell in CSS beschreibt Kästen, die um HTML-Elemente herum generiert werden.
- [Lektion 10: Das Box-Modell - Außen- und Innenabstand \(margin und padding\)](#)
Ändern Sie die Präsentation von Elementen durch den Einsatz der Eigenschaften `margin` und `padding`.
- [Lektion 11: Das Box-Modell - Umrandung \(border\)](#)
Lernen Sie die endlosen Möglichkeiten kennen, wenn Sie Umrandungen auf Ihren Seiten einsetzen.
- [Lektion 12: Das Box-Modell - Breite und Höhe](#)
In dieser Lektion schauen wir uns genauer an, wie einfach man Breite und Höhe eines Elementes definieren kann.
- [Lektion 13: Schwimmende Positionierung von Elementen \(float\)](#)
Ein Element kann mit Hilfe der Eigenschaft `float` auf der linken oder rechten Seite eines Dokumentes positioniert werden.
- [Lektion 14: Positionierung von Elementen](#)
Mit der CSS-Positionierung können Sie ein Element an einer beliebigen Stelle auf Ihrer Seite platzieren.
- [Lektion 15: Ebene auf Ebene mit z-index \(Layer\)](#)
In dieser Lektion zeigen wir, wie sich verschiedene Elemente überlappen können.
- [Lektion 16: Web-Standards und Validierung](#)
Diese letzte Lektion ist über W3C-Standards und wie Sie prüfen können, ob Ihr CSS korrekt ist.

4 Dynamische Webseiten

Bei statischen Webseiten werden HTML-Dateien auf dem Server abgelegt und zur Darstellung unverändert vom Server zum Client übertragen.

Bei dynamischen Webseiten werden die Seiteninhalte erst zur Laufzeit, d.h. beim Aufrufen der Webseite generiert. Die Webseiten enthalten neben dem HTML-Code weitere Programmanweisungen, die auf dem Webserver ausgeführt werden. Dabei kann der Server zum Aufbau der Seite Informationen von verschiedenen Quellen, beispielsweise einer Datenbank beziehen.

Verschiedene Technologien



Java Script

Java Script ist ein **Scriptsprache**, die als Erweiterung von HTML dient. Java Script - Anweisungen ermöglichen es, Elemente einer Webseite durch Einwirken des Benutzers zu ändern.

Die Programme werden als Quelltext im Browser des Clients zur Laufzeit interpretiert. Der Programm-Code ist dadurch als Quelltext für jedermann ersichtlich, ein Vorteil für Lernende, aber ein Nachteil für professionelle Programmierer. Java Script eignet sich daher nicht für komplexe Applikationen. Sie verfügt aber über unzählige Event-Handler- und Windows-Methoden, mit denen sich Web-Seiten optimieren lassen.

JSP (Java Server Pages)

JSP sind HTML-Seiten, in denen Java Code eingebettet ist. HTML-Code dient zur Darstellung der Webseite, mit Java Code wird der Inhalt und die Anwendungslogik beschrieben. Im Unterschied zum Java Script, wo der Programmcode im Browser des Clients ausgeführt wird, wird der JSP-Programmcode auf dem Webserver kompiliert und ausgeführt. Die fertiggestaltete Antwortseite wird über HTTP an den Browser zurückgeschickt. Da Java-Code von JSP auf umfangreiche Java-Bibliotheken zugreifen kann, stellt JSP im Vergleich zu anderen Script-Sprachen viel mächtigeres und eleganteres Instrumentarium zur Verfügung und eignet sich gut für komplexe Applikationen und Online-Datenbanken. Für Arbeit mit JSP sind Grundkenntnisse des Programmierens mit Java erforderlich.

4.1 Was ist JavaScript

JavaScript ist eine objektbasierte Skriptsprache. Obwohl der Name etwas anders suggeriert, hat JavaScript sehr wenig mit der Programmiersprache Java zu tun. JavaScripts werden direkt im HTML-Dokument oder in einer separaten Datei notiert. Sie werden zur Laufzeit vom Web-Browser interpretiert. Dazu besitzen moderne Web-Browser entsprechende Interpreter-Software. JavaScripts ermöglichen es HTML Dokumente interaktiv zu machen, d.h. die Eingaben von Benutzern zu registrieren und zu verarbeiten.

JavaScript läuft in einer so genannten "*Sandbox*". Das ist eine Art Sicherheitskäfig, in dem die Programmiersprache eingesperrt ist. Sie wird dabei um typische Möglichkeiten anderer Programmiersprachen beschnitten, vor allem um die Möglichkeit, beliebig Daten aus Dateien zu lesen und in Dateien schreiben zu können. So wird verhindert, dass JavaScript-Programmierer auf den Rechnern von Benutzern Unfug treiben können. Nach einigen Anfangsproblemen wurde die Browser Software stark verbessert. Eine Deaktivierung von JavaScript ist nicht mehr notwendig.

Typische Anwendungsgebiete von JavaScript

- Überprüfung (Validierung) von Formulareingaben vor dem Absenden
- *Rollover*-Grafiken, die beim Überfahren mit der Maus wechseln
- Banner oder Laufschriften
- dynamische Manipulation von Webseiten über das *Document Object Model (DOM)*
- Senden und Empfangen von Daten, ohne dass der Browser die Seite neu laden muss.

Nachteile:

- Schwache Typisierung von Variablen
- Der Programm-Code ist als Quelltext für jedermann ersichtlich und kopierbar, ein Vorteil für Lernende, aber ein Nachteil für professionelle Programmierer
- JavaScript-Programme werden auf dem Rechner des Benutzers ausgeführt. Deshalb deaktivieren einige Internet-Benutzer JavaScript in ihrem Browser aus Angst vor Hackerangriffen.

4.2 Einbinden von JavaScript in HTML-Dokumente

Ein JavaScript-Code beginnt immer mit dem Tag `<SCRIPT LANGUAGE = "JavaScript">` und endet mit dem Tag `</SCRIPT>`.

Direkt im HTML-Dokument

```
<HTML>
<HEAD>
</HEAD>
<BODY>
<SCRIPT LANGUAGE="JavaScript">
    name = prompt("Geben sie ihren Namen an "," ");
    document.write("Hallo " + name);
</SCRIPT>
</BODY>
</HTML>
```

In einer separaten Datei

```
<HTML>
<HEAD>
</HEAD>
<BODY>
<SCRIPT LANGUAGE = "JavaScript" src = "eingabe.js"></SCRIPT>
</BODY>
</HTML>
```

```
(eingabe.js)
name = prompt("Geben sie ihren Namen an "," ");
document.write("Hallo " + name)
```

Die Datei *eingabe.js* sollte im gleichen Verzeichnis gespeichert sein, wie die HTML Datei, andererseits muss der Pfad zum gewünschten Verzeichnis angegeben werden.

4.3 Eingabe überprüfen

Die Eingabe in den Formularfeldern muss in der Regel überprüft werden, da leere oder fehlerhafte Eingaben bei der Formularauswertung Fehler verursachen können.

`alert("...")` erzeugt eine Dialogbox, in der eine Fehlermeldung angezeigt werden kann.

Beispiel 1: [Beispiel anzeigen](#) (checkText.htm)

Die Eingabe im Textfeld wird auf "leer" überprüft. Falls die Eingabe leer ist, erscheint ein *Alert*-Fenster mit einer Fehlermeldung.

```
<HTML>

<HEAD>

<SCRIPT LANGUAGE = "JavaScript">

    function antwort()
    {
        name = document.form1.eingabe.value;
        if (name.length == 0)
            alert("Bitte Name eingeben!");
        else
            document.form1.ausgabe.value = "Sie heissen " + name + ".";
    }
</SCRIPT>

</HEAD>

<BODY>

<FORM NAME = "form1">
  <P>Name: <INPUT TYPE = "text" NAME = "eingabe" VALUE = "" SIZE = 15 MAXLENGTH =
10></P>
  <P><INPUT TYPE = "button" VALUE = "Hier klicken"onClick = "antwort();"></P>
  <P><INPUT TYPE = "text" NAME = "ausgabe" VALUE = "" SIZE = 40></P>
</FORM>

</BODY>

</HTML>
```

Erklärungen zum Programmcode:

`name.length` gibt die Anzahl Zeichen des Textes, der in der Variable `name` gespeichert wurde. Wurde im Textfeld Name kein Name eingegeben, ist `name.length = 0`)

Beispiel 2: [Beispiel anzeigen](#) (checkZahl.html)

Überprüft, ob eine Zahl eingegeben wurde.

```

<HTML>

<HEAD>

<SCRIPT LANGUAGE = "JavaScript">

    function antwort()
    {
        zahl = document.form1.eingabe.value;
        z = parseInt(zahl);
        if (zahl.length == 0)
            alert("Bitte Alter eingeben!");
        else
            if (isNaN(z))
                alert("Die Zahleneingabe ist ungültig!");
            else
                document.form1.ausgabe.value = "Sie sind " + z + "Jahre alt.";
    }
</SCRIPT>

</HEAD>

<BODY>

<FORM NAME = "form1">
    <P>Zahl: <INPUT TYPE = "text" zahl = "eingabe" VALUE = "" SIZE = 15 MAXLENGTH =
10></P>
    <P><INPUT TYPE = "button" VALUE = "Hier klicken"onClick = "antwort();"></P>
    <P><INPUT TYPE = "text" NAME = "ausgabe" VALUE = "" SIZE = 40></P>
</FORM>

</BODY>

</HTML>
    
```

Achtung: siehe auch [Kap. 5 Formulare und Auswertung](#). HTML5 bietet einige Möglichkeiten, Eingaben von input-Elementen bereits während der Eingabe zu validieren!

4.4 JavaScript Tutorial

<http://www.w3schools.com/js/>

THE WORLD'S LARGEST W

HTML CSS **JAVASCRIPT** SQL PHP BOOTSTRAP JQUERY TUTORIALS REFERENCES EXA

JS Tutorial
JS HOME
 JS Introduction
 JS Where To
 JS Output
 JS Syntax
 JS Statements
 JS Comments
 JS Variables
 JS Operators
 JS Arithmetic
 JS Assignment
 JS Data Types
 JS Functions
 JS Objects
 JS Scope
 JS Events
 JS Strings
 JS String Methods
 JS Numbers
 JS Number Methods
 JS Math
 JS Dates
 JS Date Formats
 JS Date Methods
 JS Arrays
 JS Array Methods
 JS Booleans
 JS Comparisons

CAS Front End Engineering »

Weiterbildung Hochschule Rapperswil
 JavaScript Engineering *richtig*
hsr.ch/weiterbildung

JavaScript Tutorial

« [W3Schools Home](#) [Next Chapter »](#)

JavaScript is the programming language of HTML and the Web.
 Programming makes computers do what you want them to do.
 JavaScript is easy to learn.
 This tutorial will teach you JavaScript from basic to advanced.

Examples in Each Chapter

With our "Try it Yourself" editor, you can change all examples and view the results.

Example

My First JavaScript

Click me to display Date and Time

[Try it Yourself »](#)

http://clab1.phbern.ch/webdesign/index.php?inhalt_links=jscript/nav_jscript.inc.php&inhalt_mitte=jscript/home.inc.php

4.5 jQuery Tutorial

jQuery ist eine JavaScript Library, die das Programmieren in JavaScript einfacher macht.

<http://www.w3schools.com/jquery/>

w3schools.com THE WORLD'S LARGEST W

HTML CSS JAVASCRIPT SQL PHP BOOTSTRAP **JQUERY** TUTORIALS REFERENCES EXA

jQuery Tutorial

jQuery HOME

jQuery Intro

jQuery Get Started

jQuery Syntax

jQuery Selectors

jQuery Events

jQuery Effects

jQuery Hide/Show

jQuery Fade

jQuery Slide

jQuery Animate

jQuery stop()

jQuery Callback

jQuery Chaining

jQuery HTML

jQuery Get

jQuery Set

jQuery Add

jQuery Remove

jQuery CSS Classes

jQuery css()

jQuery Dimensions

jQuery Traversing

jQuery Traversing

jQuery Ancestors

Domain Anbieter wechseln »
Ausgezeichnete Qualität und Top-Service zu fairen Preisen!

CAS Front End Engineering »
Weiterbildung Hochschule Rapperswil. JavaScript Engineering *richtig*

Französisch Sprachreisen »
Lernen Sie Französisch im Ausland. Gratis Offerte & Katalog Online!

jQuery Tutorial

« W3Schools Home Next Chapter »

jQuery is a JavaScript Library.
jQuery greatly simplifies JavaScript programming.
jQuery is easy to learn.

"Try it Yourself" Examples in Each Chapter

With our online editor, you can edit the code, and click on a button to view the result.

Example

```
$(document).ready(function(){
  $("p").click(function(){
    $(this).hide();
  });
});
```

[Try it Yourself »](#)

5 Formulare und Auswertung

https://wiki.selfhtml.org/wiki/HTML/Tutorials/Formulareingaben_mit_HTML5_validieren

HTML/Tutorials/Formulareingaben mit HTML5 validieren

HTML Tutorials

Es ist sinnvoll HTML-basierte [Formulare](#) nicht erst serverseitig, sondern zusätzlich schon vor dem Absenden clientseitig auf Validität zu überprüfen. HTML5 bietet einige Möglichkeiten, Eingaben von [input-Elementen](#) bereits während der Eingabe zu validieren. Auch wenn diese Validierungen eine spätere serverseitige Überprüfung nicht ersetzen, werden falsche Eingaben sofort durch CSS sichtbar gemacht und können ohne weitere HTTP-Requests korrigiert werden.

Für die Grundlagen zu diesem Tutorial eignet sich der Artikel [Formulare erstellen und gestalten](#).

Beachten Sie: Eine clientseitige Validierung von Formulardaten kann eine [serverseitige Validierung](#) keinesfalls ersetzen, sondern nur den Bedienkomfort eines Nutzers erhöhen, indem das Absenden ungeeigneter Daten im Vorfeld zu vermeiden versucht wird. Diese Filterung kann jedoch umgangen werden, weshalb man sich auf der Serverseite niemals auf solche Prüfmechanismen verlassen darf.

Inhaltsverzeichnis [\[Verbergen\]](#)

- 1 Hinweise zur Eingabe
 - 1.1 Beschreibungen mit label
 - 1.2 Hinweise mit placeholder
- 2 browsereigene Validierung mit HTML5
 - 2.1 Eingabemuster mit pattern
 - 2.2 Ober- und Untergrenzen
 - 2.3 Pflichtfelder
- 3 Anwendungsbeispiel
 - 3.1 Validierung der Eingabe
- 4 Weblinks
- 5 Siehe auch

5.1 Server-seitig PHP

PHP ist eine in HTML-eingebettete Skriptsprache. Ihre Stärke liegt in der Gestaltung von dynamischen Webseiten und im Programmieren von online Datenbank-Applikationen mit MySQL. PHP ist wesentlich einfacher als JSP und verfügt über eine grosse Anzahl von vordefinierten Funktionen, die die Programmierarbeit vereinfachen.

PHP-Skripte werden auf dem Server ausgeführt, sie werden nicht im Vorfeld kompiliert wie JSP, sondern bei jedem HTML-Request neu durch den sog. PHP-Parser auf dem Webserver. Dadurch sind PHP-Applikationen langsamer als JSP, dies spielt aber erst bei umfangreicheren Applikationen eine Rolle.

[Ajax \(Asynchronous JavaScript and XML\)](#)

Ajax ist eine Technologie, die durch die asynchrone Übertragung der Daten zwischen Client und Server charakterisiert wird. Ajax-Anwendungen ermöglichen es, innerhalb einer HTML-Seite Anfrage durchzuführen ohne die gesamte Webseite neu laden zu müssen. Die zusätzlichen Daten werden beim Bedarf vom Server angefordert und nur Teile der aktuellen Webseite aktualisiert. Zum Programmieren von Ajax Webseiten wird eine Kombination mehrerer Webtechniken verwendet: JavaScript, XML, CSS, PHP usw.

5.2 PHP Tutorial

<http://de.html.net/tutorials/php/>

- [Einführung](#)
Eine kurze Einführung in das Tutorial und was Sie hier lernen können.
- [Lektion 1: Was ist PHP](#)
Ein wenig darüber, wie PHP funktioniert und was es bedeutet, dass PHP eine serverseitige Technologie ist und was Sie in den folgenden Lektionen lernen werden.
- [Lektion 2: Server](#)
Wir beleuchten verschiedene Möglichkeiten, PHP auf dem eigenen Computer oder einem gehosteten Webserver zu verwenden.
- [Lektion 3: Ihre erste PHP-Seite](#)
In dieser Lektion erstellen Sie Ihre erste, sehr einfache PHP-Seite. Hier können Sie auch testen, ob Ihr Server für das Ausführen von PHP konfiguriert ist.
- [Lektion 4: Arbeiten mit Zeit und Daten](#)
Einführung in Functions (Funktionen), die zum Arbeiten mit Zeiten und Daten genutzt werden können.
- [Lektion 5: Schleifen](#)
Schleifen können Teile eines Skriptes wiederholen. In dieser Lektion sehen wir uns die *while*- und *for*-Schleifen genauer an.
- [Lektion 6: Conditions \(Bedingungen\)](#)
Sog. Conditions können dazu verwendet werden, die Ausführung eines PHP-Skriptes zu kontrollieren. Wir behandeln hier *if ... elseif ... else ...* und *switch ... case*.
- [Lektion 7: Skripte erläutern/kommentieren](#)
Kommentare machen Ihre PHP-Skripte klarer und einfacher verständlich. Erläuterungen können eine große Hilfe sein, wenn Sie oder jemand anderes, später Änderungen an Ihrem Code vornehmen muss.
- [Lektion 8: Arrays \(Felder\)](#)
In dieser Lektion lernen Sie, was ein sog. Array ist, wie und wofür er genutzt werden kann.
- [Lektion 9: Functions \(Funktionen\)](#)
In den vorherigen Lektionen haben Sie gelernt, unterschiedliche PHP-eigene Functions zu benutzen. Jetzt werden Sie lernen, Ihre eigenen zu erstellen.
- [Lektion 10: Variablen über die URL weitergeben](#)
Lernen Sie, wie man Variablen und deren Werte von einer Seite an eine andere mit Hilfe des sog. 'HTTP query string' weitergibt.

- [Lektion 11: Verarbeitung von Formular-Variablen](#)
Interaktive Webseiten verlangen Eingaben von Nutzern. Eine der üblichsten Arten, an diese Eingaben zu gelangen, sind Formulare.
- [Lektion 12: Sessions \(Sitzungen\)](#)
Sog. Sessions können dazu verwendet werden, Informationen während eines Besuches eines Nutzers auf Ihrer Seite zu speichern und wieder abzurufen.
- [Lektion 13: Cookies](#)
Cookies können verwendet werden, um Informationen über einen Nutzer über einen Besuch hinaus zu speichern und (z.B. beim nächsten Besuch) wieder abzurufen.
- [Lektion 14: Filesystem-Functions \(Dateisystem-Funktionen\)](#)
Hiermit können Sie auf das Dateisystem des Servers zugreifen. Dies erlaubt Ihnen, Dateien, Ordner oder Laufwerke mit PHP-Scripten zu verändern.
- [Lektion 15: Aus einer Text-Datei lesen](#)
In dieser Lektion werden wir die Filesystem-Functions benutzen, um aus einer Text-Datei zu lesen. Text-Dateien können sehr nützlich sein, um Daten verschiedener Art zu speichern.
- [Lektion 16: In eine Text-Datei schreiben](#)
Diese Lektion handelt von den Möglichkeiten, mit den Filesystem-Functions in eine Text-Datei zu schreiben. Text-Dateien können sehr nützlich sein, um Daten verschiedener Art zu speichern.
- [Lektion 17: Datenbanken](#)
In diesem Tutorial verwenden wir MySQL-Datenbanken. MySQL ist das am häufigsten verwendete Datenbanksystem, wenn man anfängt Datenbanken mit PHP nutzen wollen.
- [Lektion 18: Datenbanken und Tabellen erstellen](#)
In dieser Lektion betrachten wir zwei Arten, wie man Datenbanken und Tabellen anlegen kann. Zuerst, wie es mit PHP gemacht wird und dann mit dem benutzerfreundlicheren Software-Tool: PhpMyAdmin.
- [Lektion 19: Daten in eine Datenbank einfügen](#)
Lernen Sie, wie Sie SQL-Anweisungen benutzen, um Daten in eine Datenbank einzufügen. Wir betrachten auch Datentypen und weisen auf die häufigsten Anfängerfehler hin.
- [Lektion 20: Daten aus einer Datenbank abrufen](#)
Lernen Sie, wie man eine SQL-Anfrage nutzt, um Daten aus einer Datenbank abzurufen.
- [Lektion 21: Daten aus einer Datenbank löschen](#)
Lernen Sie, wie man mit SQL Datensätze aus einer Datenbank löscht.
- [Lektion 22: Daten in einer Datenbank aktualisieren/ändern](#)
In dieser letzten Lektion werden Sie lernen, wie man Daten mit einer SQL-Anweisung aktualisiert/ändert.

Ein weiteres Tutorial, welches ebenfalls die Grundlagen sehr gut abdeckt. Zudem kann man den Code gerade auch online ausprobieren.

<https://www.w3schools.com/php/default.asp>

5.3 PHP – Tutorial für Fortgeschrittene

- https://www.w3schools.com/php/php_forms.asp
Formulare
- https://www.w3schools.com/php/php_form_validation.asp
- Formulare validieren

5.4 Datenbankzugriff mit MySQLi oder PDO

- https://www.w3schools.com/php/php_mysql_connect.asp
Beispiel mit mysqli und PDO
- https://www.w3schools.com/php/php_mysql_prepared_statements.asp
Prepared Statements
- https://www.w3schools.com/php/php_mysql_select.asp
SelectData
- https://www.w3schools.com/php/php_mysql_insert.asp
InsertData
-

5.5 fertige Beispiele zur PHP-Programmierung

- https://www.w3schools.com/php/php_examples.asp
- https://www.w3schools.com/php/showphp.asp?filename=demo_form_validation_complete
-

5.6 Empfehlenswertes zum PHP-Tutorial

<http://www.php-kurs.com/>

Besonders die Kapitel:

- PHP für Fortgeschrittene (Sessions, require, include)
- Objektorientierte Programmierung in PHP
- MySQL Tutorial Datenbank (Gästebuch, MySQL sauber schreiben)
- PHP und Sicherheit (Session Hijacking, SQL-Injection, Cross-Site Scripting)

http://www.php-kurs.info/tutorial-sql_injection.html

http://www.php-kurs.info/tutorial-sessions_cookies.html

http://www.php-kurs.info/tutorial-variablen_uebergeben_include.html

6 Ajax

Inhalte von Seiten auf datieren, ohne die ganze Seite neu zu laden.

Ajax setzt sich "Asynchronous JavaScript und XML" zusammen und vertritt das Konzept einer asynchronen Datenübertragung zwischen Browser und Server.

<http://www.w3schools.com/ajax/default.asp>

The screenshot shows the w3schools.com website. At the top, there is a navigation bar with links for HTML, CSS, JAVASCRIPT, SQL, PHP, BOOTSTRAP, and JQUERY. Below this is a sidebar menu for the 'AJAX Tutorial' section, with 'AJAX HOME' selected. The main content area features a banner for 'HOST EUROPE .CH DOMAIN' with a 'JETZT UMZIEHEN' button and a price of '1,24 CHF mtl.'. The title 'AJAX Tutorial' is followed by a 'Next Chapter' link. Below this is a green 'AJAX' logo and a list of capabilities: 'Update a web page without reloading the page', 'Request data from a server - after the page has loaded', 'Receive data from a server - after the page has loaded', and 'Send data to a server - in the background'. A section titled 'Try it Yourself Examples in Every Chapter' explains that examples can be edited online. At the bottom, an 'AJAX Example' section contains a text area with 'Let AJAX change this text', a 'Change Content' button, and a 'Try it Yourself' button.

7 Verwenden von Bildern

Um Ihre eigenen Bilder zu erstellen, benötigen Sie ein Bildbearbeitungsprogramm. Ein Bildbearbeitungsprogramm ist eines der wichtigsten Werkzeuge um schöne Webseiten zu erstellen.

Leider ist weder im Lieferumfang von Windows, noch bei anderen Betriebssystemen, ein annehmbares Bildbearbeitungsprogramm inklusive.

Für unsere Zwecke genügt es, wenn Sie sich [Irfan View](#), ein exzellentes Bildbearbeitungsprogramm herunterladen. Irfan View ist sogenannte Freeware und kostet aus diesem Grunde nichts.

Oder Sie borgen sich Bilder von anderen Seiten, indem Sie diese downloaden. Aber bitte seien Sie vorsichtig, dass Sie mit dem Download und einer späteren Verwendung auf Ihren Seiten keinerlei Copyrights verletzen. Trotz allem ist es sicherlich nützlich zu wissen, wie man Bilder aus dem Internet herunterlädt. So wird's gemacht:

Klick mit der rechten Maustaste auf das Bild (irgendein Bild im Internet)

Wählen Sie "Bild speichern als ..." in dem Menü, welches erscheint.

Suchen Sie einen Ordner, in den Sie das Bild speichern möchten und klicken Sie auf "Speichern".

8 Webseiten validieren lassen

<http://validator.w3.org/>

9 Selbsttest

- <https://www.w3schools.com/quiztest/quiztest.asp?Qtest=HTML>
HTML Test
- <https://www.w3schools.com/quiztest/quiztest.asp?qtest=CSS>
CSS Test
- <https://www.w3schools.com/quiztest/quiztest.asp?qtest=JavaScript>
Java Script
- <https://www.w3schools.com/quiztest/quiztest.asp?qtest=jQuery>
- jQuery

10 weitere Interessante Links

<https://www.codecademy.com/learn/javascript>

<https://www.codecademy.com/learn/php>

<https://www.codecademy.com/learn/learn-sql>

10.1 Beispiele von eigenen Projekte

Webseitengestaltung am Beispiel einer Schreinerei

<https://wiki.selfhtml.org/wiki/HTML/Tutorials/HTML-Einstieg>

Beispielprojekt

<http://www.html-seminar.de/komplette-website-erstellen.htm>

- Startseite
- HTML Tutorial für Anfänger
- CSS lernen - Cascading Style Sheets für Webdesign
- CSS3
- Responsive Webdesign
- Projekt Beispielwebsites
 - Vorstellung Projekte
 - 1-Spalten-Layout für Websites
 - 2 Spalten Layout
 - 3-Spalten-Layout mit CSS erstellen
- Bildbearbeitung für Websites
- JavaScript, Ajax und jQuery
- Fortgeschrittene
- Web Apps erstellen
- Forum
- Diverses
- Impressum
- Kontakt
- Sitemap

Beispielprojekte: HTML und CSS lernen Schritt für Schritt anhand Nachbaus von Websites

Vorstellung der Projekte

Anhand des Nachbaus der Beispiel-Websites lernt man die einzelnen HTML-Befehle und CSS-Anweisungen im Zusammenspiel (quasi in freier Wildbahn) und sieht die Umsetzung im HTML-Tutorial.

Nachdem am besten das Lernen durch ein eigenes Umsetzen erfolgt, wird im Folgenden das bisherige Wissen praktisch umgesetzt. Dazu werden komplette Beispielprojekte von verschiedenen Websites Schritt für Schritt nachgebaut.

- einfaches einspaltiges Layout (mit Responsiven Design) mit der Website "Säulenbauten ..."
- 2-Spaltiges Layout anhand Website "Tübingen im Winter"
- 3-Spaltiges Layout mit der Fennek-Website

Die Schritt für Schritt Anleitung für das 2- und 3-spaltige Design findet sich in den Videos. Alle notwendigen Einzelteile wie Grafiken und Bildern können von hier untergeladen werden.

Sie können sich natürlich auch ein eigenes Projekt überlegen in dieses ähnlich der Beispielprojekte umsetzen: z.B. eine Website für den (eigenen?) Verein.

Wenn Sie dieses Kapitel vollständig durchgearbeitet haben, haben Sie eine eigene Website, die Sie nach Belieben auch noch erweitern können. Hier werden alle grundlegenden Schritte nacheinander vorgestellt und sofort umgesetzt.



HTML5 & CSS3 lernen:
mit Beispielen & Videos
www.HTML-Seminar.de