

Assoziative PHP Arrays

Bei grossen Arrays wird es natürlich irgendwann umständlich, zu wissen, welche Nummer/Index zu welchem Wert gehört, darum gibt es assoziative PHP Arrays.

Das heißt, man kann für einen Wert einen Schlüssel/Key zuweisen, dies kann z.B. eine andere Zeichenkette sein.

Die Zuweisung erfolgt per: =>

```
<?php
$wochentage = array(
    "so" => "Sonntag",
    "mo" => "Montag",
    "di" => "Dienstag",
    "mi" => "Mittwoch",
    "do" => "Donnerstag",
    "fr" => "Freitag",
    "sa" => "Samstag");
?>
```

foreach-Schleife:

```
<?php
$mitarbeiter = array("Bob", "Peter", "Lisa");

echo "Foreach-Schleife: <br>";
foreach($mitarbeiter AS $name)
    {
        echo $name."<br>";
    }
?>
```

foreach example \$key => \$value

We have an associative array that stores the names of people in our company as the keys with the values being their age. We want to know how old everyone is at work so we use a Foreach loop to print out everyone's name and age.

PHP Code:

```
$employeeAges;
$employeeAges["Lisa"] = "28";
$employeeAges["Jack"] = "16";
$employeeAges["Ryan"] = "35";
$employeeAges["Rachel"] = "46";
$employeeAges["Grace"] = "34";

foreach( $employeeAges as $key => $value){
    echo "Name: $key, Age: $value <br />";
}
```

Display:

```
Name: Lisa, Age: 28
Name: Jack, Age: 16
Name: Ryan, Age: 35
Name: Rachel, Age: 46
Name: Grace, Age: 34
```

foreach example (eigene Bezeichnung)

The operator "=>" represents the relationship between a key and value. You can imagine that the key points => to the value. In our example we named the key \$key and the value \$value. However, it might be easier to think

of it as \$name and \$age. Below our example does this and notice how the output is identical because we only changed the variable names that refer to the keys and values.

PHP Code:

```
$employeeAges;  
$employeeAges["Lisa"] = "28";  
$employeeAges["Jack"] = "16";  
$employeeAges["Ryan"] = "35";  
$employeeAges["Rachel"] = "46";  
$employeeAges["Grace"] = "34";  
  
foreach( $employeeAges as $name => $age){  
    echo "Name: $name, Age: $age <br />";  
}
```

Display:

```
Name: Lisa, Age: 28  
Name: Jack, Age: 16  
Name: Ryan, Age: 35  
Name: Rachel, Age: 46  
Name: Grace, Age: 34
```

Mehrdimensionale Arrays

In einem Array kann man ein weiteres Array, und in diesem Array wieder ein Array speichern, und so weiter. Solche Arrays nennt man dann mehrdimensionale Arrays.

Die Dimension gibt dabei an, wie Tief diese Verschachtelung geht. Ein normales Array wäre ein 1-dimensionales Array, wenn jetzt in dem Array ein weiteres Array gespeichert ist, ist dies ein 2-dimensionales Array.

Dafür hier ein Beispiel:

```
<?php
$mitarbeiter = array(
    array("Klaus", "Zabel"),
    array("Arnie", "Meier"),
    array("Willi", "Brand")
);

//Daten ausgeben
echo "Vorname: ".$mitarbeiter[0][0];
echo " Nachname: ".$mitarbeiter[0][1];
?>
```

Ausgabe: Vorname: Klaus Nachname: Zabel