**Lektion 6: Bedingungen (Conditions)**

Bedingungen werden benutzt, um Teile des Skriptes nur auszuführen, wenn vordefinierte Voraussetzungen (Bedingungen) erfüllt werden. Beispiele können hier sein, dass ein Datum nach dem 01. Januar 2012 liegen soll oder dass eine Variable größer als 7 ist.

**If ...**

Die erste Art von Bedingungen, die wir uns hier ansehen werden, ist [if](http://php.net/manual/en/control-structures.if.php), mit der folgenden Syntax:

 if (*Bedingung*) {

 Anweisung

 }

Und wieder ist die Syntax ziemlich nah am normalen Englisch: ***If*** *a condition is met, then execute something*, also in etwa ***Wenn*** *eine Bedingung erfüllt ist, dann führe etwas aus*. Lassen Sie uns wieder mit einem einfachen Beispiel beginnen:

 <html>

 <head>

 <title>Bedingungen</title>

 </head>

 <body>

 <?php

 $x = 2;

 if ($x > 1) {

 echo "<p>Die Variable $x ist größer als 1</p>";

 }

 ?>

 </body>

 </html>

**if ... else ...**

Die nächste Art von Bedingung, die wir uns anschauen, ist [else](http://php.net/manual/en/control-structures.else.php), mit folgender Syntax:

 if *(Bedingung)* {

 Anweisung

 }

 else {

 Anweisung

 }

Und wieder, sehr nah am normalen Englisch: ***if*** *a condition is met execute something or* ***else*** *execute something else*, also ***Wenn*** *eine Bedingung erfüllt ist, dann führe etwas aus oder* ***sonst*** *führe etwas anderes aus*.

In [Lektion 4](http://de.html.net/tutorials/php/lesson4.php), haben Sie gelernt, wie man die Nummer eines Monats herausbekommt. Im folgenden Beispiel werden wir diese Nummer in einer [if](http://php.net/manual/en/control-structures.if.php)-[else](http://php.net/manual/en/control-structures.else.php)-Bedingung verwenden, um herauszufinden, welche Jahreszeit ist:

 <html>

 <head>

 <title>Bedingungen</title>

 </head>

 <body>

 <?php

 if (date ("m") == 3) {

 echo "<p>Jetzt ist Frühling!</p> ";

 }

 else {

 echo "<p>Ich weiß nicht, welche Jahreszeit gerade ist!</p> ";

 }

 ?>

 </body>

 </html>

[Beispiel anzeigen](http://de.html.net/tutorials/php/lesson6_ex1.php)

Wie Sie sehen, ist diese Bedingung keine besonders kluge Bedingung - sie funktioniert nur im März!

Wie auch immer, es gibt eine Vielzahl an Möglichkeiten, die Bedingungen zu verbessern und sie präziser zu machen. Nachfolgend sind die Vergleichsoperatoren aufgelistet, die verwendet werden können:

**==** - ist gleich
**<** - kleiner als
**>** - größer als
**<=** - kleiner-gleich
**>=** - größer-gleich
**!=** - ungleich

Außerdem gibt es einige logische Operatoren:

**&&** - UND (AND)
**||** - ODER (OR)
**!** - NICHT (NOT)

Die Operatoren können dazu verwendet werden, präzisere Bedingungen zu entwickeln. Jetzt können wir das obige Beispiel erweitern und alle Frühlingsmonate einbeziehen:

 <html>

 <head>

 <title>Bedingungen</title>

 </head>

 <body>

 <?php

 if (date("m") >= 3 && date("m") <= 5) {

 echo "<p> Jetzt ist Frühling!</p> ";

 }

 else {

 echo "<p> Jetzt ist entweder Winter, Sommer oder Herbst!</p> ";

 }

 ?>

 </body>

 </html>

Schauen wir uns die erweiterte Bedingung etwas genauer an:

 date("m") >= 3 && date("m") <= 5

Die Bedingung kann übersetzt werden mit:

 Wenn (if) der Monat größer-gleich 3 UND der Monat kleiner-gleich 5

Clever oder? Operatoren spielen eine wichtige Rolle in vielen verschiedenen Bereichen von PHP.

Aber dennoch funktioniert unser Beispiel nur mit März, April und Mai. Alle anderen Monate sind noch nicht von der Bedingung abgedeckt. Also lassen Sie uns probieren, sie noch ein wenig weiterzuentwickeln.

**if ... elseIf ... else ...**

Indem wir [elseif](http://php.net/manual/en/control-structures.elseif.php) verwenden, können wir die Bedingung erweitern, so dass sie für alle Monate funktioniert:

 <html>

 <head>

 <title>Bedingungen</title>

 </head>

 <body>

 <?php

 if (date("m") >= 3 && date("m") <= 5) {

 echo "<p>Jetzt ist Frühling!</p>";

 }

 elseif (date("m") >= 6 && date("m") <= 8) {

 echo "<p>Jetzt ist Sommer!</p>";

 }

 elseif (date("m") >= 9 && date("m") <= 11) {

 echo "<p>Jetzt ist Herbst!</p>";

 }

 else {

 echo "<p>Jetzt ist Winter!</p>";

 }

 ?>

 </body>

 </html>

[Beispiel anzeigen](http://de.html.net/tutorials/php/lesson6_ex2.php)

Um Bedingungen zu verfassen, muss man eigentlich nur logisch denken und methodisch vorgehen. Das obige Beispiel ist noch recht überschaubar, aber Bedingungen können sehr kompliziert werden.

**switch ... case**

Eine andere Möglichkeit Bedingungen zu beschreiben ist die [switch](http://php.net/manual/en/control-structures.switch.php)-Methode:

 switch *(Ausdruck)* {

 case 1:

 *Anweisung*

 break;

 case 2:

 *Anweisung*

 break;

 default:

 *Anweisung*

 break;

 }

Diese Methode basiert auf einem **Ausdruck** und listet verschiedene "Antworten" oder "Werte" mit den zugehörigen **Anweisungen** auf. Am einfachsten läßt sich diese Methode an einem Beispiel erklären.

Wie Sie sich vielleicht aus Lektion 4 erinnern, gibt die Funktion [date("w")](http://php.net/manual/en/function.date.php) den aktuellen Wochentag aus. Dies kann in einem Beispiel genutzt werden, wo wir den Namen des Tagen anstatt der Nummer ausgeben:

 <html>

 <head>

 <title>Loops </title>

 </head>

 <body>

 <?php

 switch(date("w")) {

 case 1:

 echo "Heute ist Montag";

 break;

 case 2:

 echo "Heute ist Dienstag";

 break;

 case 3:

 echo "Heute ist Mittwoch";

 break;

 case 4:

 echo "Heute ist Donnerstag";

 break;

 case 5:

 echo "Heute ist Freitag";

 break;

 case 6:

 echo "Heute ist Samstag";

 break;

 default:

 echo "Heute ist Sonntag";

 break;

 }

 ?>

 </body>

 </html>

[Beispiel anzeigen](http://de.html.net/tutorials/php/lesson6_ex3.php)

Oft ist [switch](http://php.net/manual/en/control-structures.switch.php) eine gute Alternative zu [if](http://php.net/manual/en/control-structures.if.php)-[else](http://php.net/manual/en/control-structures.else.php)-Bedingungen. Was Sie in einer bestimmten Situation verwenden sollen, hängt davon ab, welche Methode Sie am einfachsten und logischsten finden. Ihre Skripte logisch und klar aufzubauen, kann eine große Herausforderung sein.

In der nächsten Lektion werden wir uns damit beschäftigen, wie Sie Ihren Scripten erläuternde Kommentare hinzufügen können. Gute Erläuterungen sind äußerst wichtig, wenn Sie oder jemand anderes später Änderungen an Ihrem Code vornehmen möchte.