

Programmsteuerung mit PHP - if/else, elseif, switch

Bei der Programmierung geht es meist vor allem darum festzulegen, welche Aktionen wie oft unter welchen Bedingungen ausgeführt werden sollen. Hierzu steht in PHP eine ganze Reihe von Kontrollstrukturen zur Verfügung.

WICHTIG: KEIN Semikolon nach "if()", "else" oder "elseif()".

Mit „if“ kann der Code Entscheidungen fällen

Mit der if-Anweisung von PHP kann der Code Entscheidungen fällen, die darauf basieren, ob etwas wahr oder falsch ist. Somit kann man flexibel auf Bedingungen reagieren.

Beispiel: Bevor er eine Bestellung ausführt, muss der Verkäufer sein Geld erhalten.

Wenn die Kreditkarte des Kunden in Ordnung ist, führe die Bestellung aus. Dieses Szenario können wir mithilfe der if-Anweisung, die genau für derartige Entscheidungen gedacht ist, in PHP-Code überführen.

Eine einfache if-Anweisung hat drei Teile:

- 1 Das Schlüsselwort if**
Mit diesem Schlüsselwort wird die Anweisung eingeleitet.
- 2 Die Testbedingung**
Die Testbedingung oder der **Bedingungsausdruck** folgt in **Klammern** unmittelbar auf das Schlüsselwort **if**. An dieser Stelle geben Sie den Ausdruck an, dessen Gültigkeit oder Wahrheit Sie prüfen wollen.
- 3 Die Aktion**
Die Aktion einer **if**-Anweisung folgt in **geschweiften Klammern** auf die Testbedingung. Hier geben Sie den PHP-Code an, der ausgeführt werden soll, wenn die **Bedingung** tatsächlich **true** ist.

```
1 if (istGueltig($kreditkarte)) {  
    3 bestellungAusfuehren();  
}
```

Mit diesem if beginnt die Anweisung.

Das ist die Bedingung. In ihr wird eine Funktion aufgerufen, die die Gültigkeit dessen prüft, was in ikreditkarte gespeichert ist.

Diese geschweifte Klammer öffnet den Codeblock mit der Aktion.

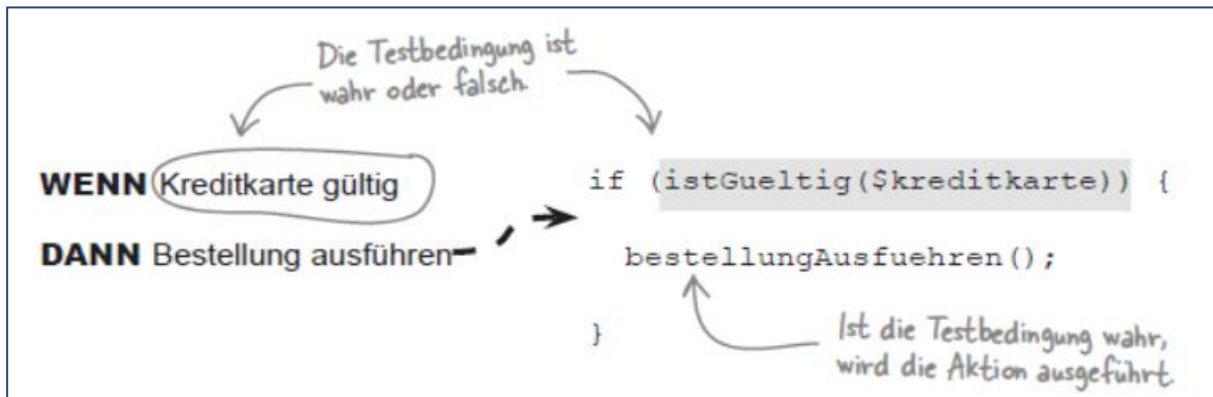
Diese geschweifte Klammer beendet die Anweisung.

Das ist die Aktion – das, was PHP ausführt, wenn die Bedingung erfüllt ist. Sie können an dieser Stelle beliebig viele Codezeilen angeben.

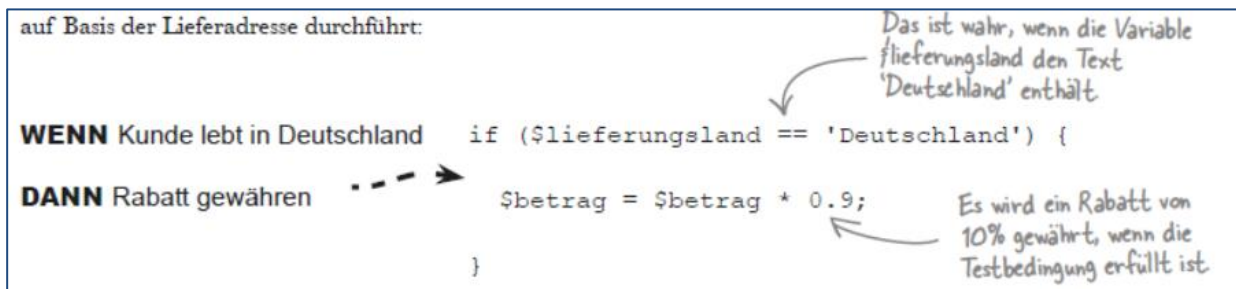
Diese fiktive Funktion liefert true oder false, je nachdem, ob die Kreditkarte gültig ist oder nicht.

Auf Wahrheit prüfen

Das Herz der if-Anweisung ist die Testbedingung, die entweder als true oder als false betrachtet wird. Die Testbedingung kann eine Variable, ein Funktionsaufruf oder ein Vergleich von zwei Dingen sein. Unser Beispiel basiert beispielsweise auf einem Funktionsaufruf. Das bedeutet, dass der von der Funktion zurückgelieferte Wert als Wahr/Falsch-Wert verstanden wird.



Häufig werden Vergleiche als Testbedingungen verwendet. Dabei wird üblicherweise eine Variable mit einem anderen Wert verglichen. Nehmen wir beispielsweise an, der Verkäufer möchte Kunden aus Deutschland einen Rabatt geben. Dann könnte er eine if-Anweisung erstellen, die einen Vergleich auf Basis der Lieferadresse durchführt:



Diese Testbedingung prüft auf Gleichheit. Der entsprechende Vergleichsoperator ist das doppelte Gleichheitszeichen (`==`). Prüfungen auf Gleichheit sind nicht auf Variablen und Strings beschränkt. Sie können auch Variablen mit Zahlen vergleichen, Variablen mit Variablen und sogar Berechnungen durchführen.

1) Fallunterscheidung mit if/else:

Mit der Funktion if() kann eine Variable auf ein bestimmtes Kriterium geprüft werden.

- In der Klammer steht die **Bedingung**, auf die geprüft werden soll,
- im if-Block die **Anweisungen**, die geschehen sollen, wenn das Kriterium zutrifft (das können beliebig viele PHP-Anweisungen sein) und im
- else-Block jene, die ausgeführt werden, wenn alle vorher geschriebenen Bedingungen nicht zutreffen.

Der if- bzw. else-Block muss immer von geschwungenen Klammern eingeschlossen sein.

Operatoren

Ausdrücke mit mehreren Operatoren werden von links nach rechts aufgelöst – unter der Beachtung der Rangordnung. Die Rangordnung ist bei der folgenden Tabelle oben am höchsten.

Operator	Bedeutung
()	Klammern
! -	logisches Nicht, negatives Vorzeichen
* / %	Multiplikation, Division, Modulo-Operation
+ -	Addition, Subtraktion
< <= > >=	kleiner, kleiner oder gleich, größer, größer oder gleich
== !=	gleich, ungleich
&&	logisches Und
	logisches Oder
=	Zuweisung

Beispiel: erstelle eine PHP Datei „if1.php“

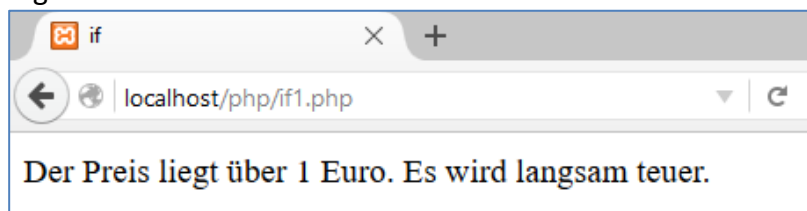
```
<?php
$preis = 1.1;
if ($preis <= 1)
    {
    echo "Der Preis liegt unter 1 Euro bzw. ist eins";
    }
else {
    echo "Der Preis liegt über 1 Euro. Es wird langsam teuer.";
}
?>
```

```

8 <body>
9 <?php
10 $preis = 1.1;
11 if ($preis <= 1)
12 {
13     echo "Der Preis liegt unter bzw. bis 1 Euro";
14 }
15 else {
16     echo "Der Preis liegt über 1 Euro. Es wird langsam teuer.";
17 }
18 ?>
19 </body>

```

Ergebnis:



Übung 1 (Formular) - Zugang

Eintragen eines Passwortes in ein HTML-Formular. Überprüfung in dem passenden PHP-Dokument.

HTML-Dokument: speichern als „zugang.html“

```

<body>
<h1>Bitte das Zugangspasswort eingeben:</h1>
<form action = "zugang.php" method = "post" >
    <p>Passwort <input name = "pw" size = "50" > </p>
    <br>
    <input type="submit" value="Abschicken">
</form>
</body>

```

```

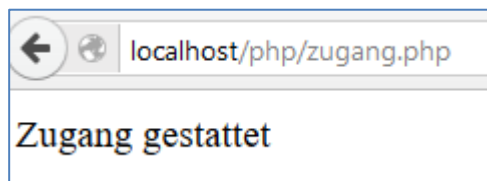
8 <body>
9
10 <h1>Bitte das Zugangspasswort eingeben:</h1>
11 <form action = "zugang.php" method = "post" >
12     <p>Passwort <input name = "pw" size = "50" > </p>
13 <br>
14 <input type="submit" value="Abschicken">
15 </form>
16
17 </body>
18 </html>

```

Auswertungsprogramm: speichern als „zugang.php“

```
8 <body>
9 <?php
10 if ($_POST["pw"] == "Reinhard") {
11     echo "Zugang gestattet";
12 }
13     else {
14     echo "Zugang verweigert";
15 }
16 ?>
17 </body>
```

Ergebnis:



Übung 2 (Formular) - if/else – Tankfüllung

Erstelle ein Eingabeformular (**tanken.html**) und ein passendes PHP-Programm (**tanken.php**). Es soll der Preis für eine Tankfüllung berechnet werden.

Es gibt zwei Sorten:

Benzin (Preis: 1,3 Euro)

Diesel (Preis: 1,2 Euro).

Der Benutzer gibt im ersten Eingabefeld die getankte Literzahl und im zweiten Feld entweder B oder D ein.

B ist Benzin

D ist Diesel

Das PHP-Programm ermittelt in Abhängigkeit von der Sorte und der getankten Menge den zu zahlenden Betrag. Es wird davon ausgegangen, dass der Benutzer keine Fehleingaben macht.

A screenshot of a web browser window titled 'Zahlen-Formular'. The address bar shows 'localhost/php/tanken.html'. The main content area has the heading 'Bitte gib die Menge und die Sorte ein'. Below the heading are two input fields: 'Menge in Liter:' and 'Sorte (B oder D):'. Below these fields is the text 'Es wird der zu zahlende Betrag ausgegeben:'. At the bottom of the form are two buttons: 'Abschicken' and 'Löschen'.

Übung:

Erweitere das Programm „zugang.html“ und „zugang.php“ folgendermaßen:
Es sollen zwei Benutzer Zugang erhalten, nämlich „Reinhard“ und „Mario“.

2)elseif-Anweisung

Mit Hilfe von **elseif** können beliebig viele Bedingungen zusammengefügt werden, indem mit elseif weitere Bedingungen definiert werden.

```
9  <?php
10  if (Bedingung1) {
11      Anweisung1;
12  }
13  elseif (Bedingung2) {
14      Anweisung2;
15  }
16  elseif (Bedingung3) {
17      Anweisung3;
18  }
19  else {
20      Anweisung4;
21  }
```

Der **PHP-Interpreter** geht die Bedingungen der Reihe nach durch, d.h.

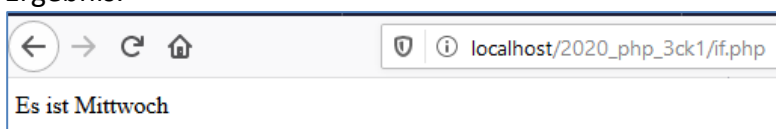
- zuerst wird der if-Fall geprüft, dann der
- erste elseif-Fall, dann der
- nächste elseif-Fall

sobald eine der Bedingungen zutrifft, wird die Anweisung ausgeführt und dann die komplette Fallunterscheidung verlassen.

Beispiel: erstelle „elseif.php“

```
1  <?php
2  $tagDerWoche = 1;
3
4  if($tagDerWoche == 1){
5      echo 'Es ist Montag';
6  } elseif($tagDerWoche == 2){
7      echo 'Es ist Dienstag';
8  } elseif($tagDerWoche == 3){
9      echo 'Es ist Mittwoch';
10 } else {
11     echo 'Das ist kein gültiger Tag!';
12 }
13
14 ?>
```

Ergebnis:




Übung 3- Taschengeld

Erstelle ein Taschengeld-Beispiel nachfolgendem Schema:

Wenn du sofort das Zimmer aufräumst, darfst du heuet fortgehen, räumst du es dagegen später auf, gibt es schon ein Zeitlimit. Wenn du darauf vergisst, musst du zu Hause bleiben und PHP lernen.

Es soll eine Eingabe in einem Formular möglich sein (zimmer.html):



← → ↻ 🏠 localhost/2020_php_3ck1/zimmer/zimmer.html

Geld fürs Aufräumen:

Sofort, gerne
 OK, später
 geht nicht

Absenden

```
1 <html>
2 <head>
3     <meta charset="utf-8">
4     <title>Aufräumen</title></head>
5 <body>
6 <h1>Geld fürs Aufräumen:</h1>
7 <form action="zimmer.php" method="post">
8
9     <input type="radio" name="zimmer" value="sofort" id="1">
10     <label for="1">Sofort, gerne</label><br>
11     <input type="radio" name="zimmer" value="spaeter" id="2">
12     <label for="2">OK, später</label><br>
13     <input type="radio" name="zimmer" value="nein" id="3">
14     <label for="3">geht nicht</label><br><br><br>
15
16     <input type="submit" value="Absenden">
17 </form>
18 </body>
19 </html>
```

Code:

```
<form action="zimmer.php" method="post">
```

```
    <input type="radio" name="zimmer" value="sofort" id="1">
        <label for="1">Sofort, gerne</label><br>
    <input type="radio" name="zimmer" value="spaeter" id="2">
        <label for="2">OK, später</label><br>
    <input type="radio" name="zimmer" value="nein" id="3">
        <label for="3">geht nicht</label><br><br><br>
```

```
    <input type="submit" value="Absenden">
</form>
```

php-Datei:

```
1 <?php
2 $zimmer = $_POST['zimmer'];
3
4 if($zimmer == "sofort")
5 {
6     echo "<h2>Super, du darfst heute fortgehen.</h2>";
7 }elseif($zimmer == "spaeter")
8 {
9     echo "<h2>Ok, dann gehst bis 22 Uhr fort.</h2>";
10 } else
11 {
12     echo "<i>Hausarrest.</i>";
13 }
14 ?>
```

Übung – Tanken

Erweitere das Beispiel von oben (tanken.html und tanken.php)

Es soll der Preis für eine Tankfüllung berechnet werden, bei der man zwischen drei Sorten wählen kann:

B: Benzin:	Preis: 1.3 Euro
D: Diesel:	reis: 1.2 Euro
S: Super:	Preis: 1.35 Euro

3)Entscheidungen mit switch:

Solange man zwei, drei Werte vergleicht, hat man mit if/else und elseif keine Probleme. Sobald es aber darum geht, aus einer Reihe von Möglichkeiten die richtige zu wählen, wird es mit if bald unübersichtlich.

Hier steht die Alternative „switch“ bereit.

Damit lassen sich gewissermaßen **beliebig viele if-Anweisungen** zusammenfassen.

Die switch-Anweisung wertet einen Ausdruck aus und vergleicht anschließend beliebig viele Varianten mit dem Ergebnis. Sollte eine Variante (**case**) zutreffen, wird der dazu gehörige Programmcode ausgeführt und die Anweisung schließlich über das Kommando „break“ verlassen. Ein frei definierbarer „default-Block“ kann zudem Anweisungen enthalten, die ausgeführt werden, wenn keine der Vorgaben zutrifft.

Beachte:

Die case- und default-Zeilen werden mit einem Doppelpunkt und nicht mit einem Strichpunkt abgeschlossen!

Die zu untersuchenden Variablen stehen in geschwungener Klammer nach dem switch-Befehl. Diese „Fälle“ werden auch „**case**“ genannt. Hier wird der Wert angegeben und nach einem Doppelpunkt die Anweisungen.

Man kann beliebig viele cases angeben.

Beispiel: (Übung: switch.php)


```

8 <body>
9 <?php
10 $auswahl = 2;
11 switch($auswahl)
12 {
13     case 1:
14         echo "eins";
15         break;
16     case 2:
17         echo "zwei";
18         break;
19     case 3:
20         echo "drei";
21         break;
22     default:
23         echo "etwas anderes";
24 }
25 ?>
26

```

Wenn der Wert der Variable dem in einem **CASE-Abschnitt** angegebenen Wert entspricht, wird der zugehörige Ausdruck ausgeführt.

- Der **BREAK-Befehl** nach jedem Block bringt das Verlassen der switch-Anweisung. Ansonsten würden ab dem Zutreffen eines Falles, alle folgenden Anweisungen ausgeführt. In unserem Fall würden hintereinander „zweidreietwas anderes“ ausgegeben.
- Der **DEFAULT-Teil** wird nur ausgeführt, wenn keine der definierten case-Bedingungen zutrifft.
- Im obigen Beispiel wird zwei ausgegeben.

Übung: Wochentage mit switch bestimmen

Erstelle die Datei „switch_wochentag.php“

Die Funktion „date()“ mit dem Parameter „1“ den englischen Wochentag des aktuellen Datums. Über eine switch-Anweisung lässt sich die englische Bezeichnung in einen deutschen Wochentag umwandeln.

- zuerst wird mit \$heute = date("1") die englische Bezeichnung in der Variablen \$heute gespeichert
- anschließend prüft eine switch-Anweisung alle möglichen Inhalte und ersetzt die englische durch die deutsche Bezeichnung.
- Die default-Anweisung ergibt sich von selbst: wenn die Entscheidung von Monday bis Saturday alle „false“ sind, dann muss \$heute den Inhalt Sunday haben.

```

8 <body>
9 <?php
10 $heute = date("l");
11 switch ($heute) {
12     case "Monday":
13         $heute = "Montag";
14         break;
15     case "Tuesday":
16         $heute = "Dienstag";
17         break;
18     case "Wednesday":
19         $heute = "Mittwoch";
20         break;
21     case "Thursday":
22         $heute = "Donnerstag";
23         break;
24     case "Friday":
25         $heute = "Freitag";
26         break;
27     case "Saturday":
28         $heute = "Samstag";
29         break;
30     default:
31         $heute = "Sonntag";
32 }
33 echo "Heute ist $heute";
34 ?>

```

Ergebnis:

