Die while-Schleife

Verschiedene Vorgänge benötigen mehrere Durchgänge, bis ein bestimmter Zustand erreicht ist. Dabei sollen ähnliche Anweisungen bei jedem "Schleifendurchlauf" ausgeführt werden. PHP bitet mehrere Möglichkeiten, solche Wiederholungen zu implemenmtieren. Die erste Möglichkeit ist die while-Schleife.

Syntax der while-Schleife

Die Syntax einer while-Schleife ist wie folgt aufgebaut:

while.php

```
while(Bedingung) {
   Anweisungen
}
```

- Es wird zunächst geprüft, ob die Bedingung wahr ist, wenn ja wir der Code im Schleifenkörper ausgeführt, wenn nein wird das Prgramm am ende des Schloeifenkörpers fortgesetzt.
- Wenn die Bedingung wahr ist, springt der Programmfluss nach dem Durchlaufen des Schleifenkörpers wieder zu Schleifenkopf, wo erneut die Bediungung geprüft wird. Auf diese Weise wird der Schleifenkörper so lange wiederholt, bis die Bedingung falsch wird.

Kopfgesteuerte while-Schleife

Zum Beispiel kann man, um die Zahlen 1 bis 10 auszugeben, de folgenden PHP-Code verwenden:

```
<?php
echo "1 <br />";
echo "2 <br />";
echo "3 <br />";
echo "4 <br />";
echo "5 <br />";
echo "6 <br />";
echo "7 <br />";
echo "8 <br />";
echo "9 <br />";
echo "10 <br />";
?>
```

Der verwendete Befehl ist stets derselbe (`echo`), nur das Argument ändert sich bei jedem Aufruf (Die Zahl ist eins größer als zuvor). Diesen Effekt kann man mit einer **while**-Schleife sehr viel effizienter erreichen:

zaehlen.php

```php
<?php
$i = 1;
while ($i <= 10) {
    echo $i . "<br />";  // es wird $i ausgegeben,
    $i++;  // Wert wird um 1 erhöht
}
?>
```

Dabei werden alle Befehle im **Schleifenblock** (innerhalb der geschweiften Klammern) solange wiederholt, wie die Bedingung im Argument ($i < = 10) wahr ist. Das Beispiel zeigt eine **kopfgesteuerte** while-Schleife.

```
$i++
```

ist der sogenannte Inkrement Operator, er erhöht die Variable `$i` um eins. Alternativ und länger könnte man `$i=$i+1;` schreiben.

Im Beispiel wird die Variable `$i` ausgegeben und anschließend die Zählvariable um den Wert 1 erhöht. Die Erhöhung des Wertes ist wichtig, damit die Bedingung irgendwann nicht mehr erfüllt ist. Falls die Bedingung immer erfüllt ist, beispielsweise weil du vergessen hast die Zählvariable `$i` zu erhöhen, resultiert das in eine "Endlosschleife" - das Programm verlässt den Schleifenkörper nicht mehr und bleibt scheinbar "hängen". In diesem Fall bricht der Webserver die Ausführung nach einer gewissen Laufzeit deines Scripts ab.

Aufgabe

- Erstelle im Unterverzeichnis "zaehler" deines Webspace eine Formular mit einem Verarbeitungsskript. mit dem man durch Eingabe einer Zahl bestimmen kann, bis zu welcher Zahl das Skript "zählt".
- Erweitere das Programm so, dass alle Quadratzahlen bis zu einer bestimmten Zahl berechnet werden.
- Erweitere das Programm um ein weiteres Formularfeld, so dass Start– und Emndwert angegeben werden können.

Fußgesteuerte while-Schleife: do-while

Manchmal möchte man erreichen, dass der Schleifenblock mindestens einmal durchlaufen wird, sogar dann, wenn die Bedingung von Beginn an nicht erfüllt ist. Dies kann man mit einer fußgesteuerten while-Schleife erreichen:

```php
<?php
$i = 11;
do {
```

```
    echo $i . "<br />";  // es wird $i ausgegeben,
    $i++;  // Wert wird um 1 erhöht
  } while ($i <= 10);
  ?>
```

Hier wird die Bedingung am Ende überprüft, der Schleifenblock läuft mindestens einmal durch.

Aufgabe

Verändere deine Programm von oben, so dass es fußgesteuert ausgeführt wird.

Schleifenabläufe beeinflussen: break und continue

Abbrechen einer Schleife mit ''break''

Mittels `break` können wir den Schleifenablauf in der Schleife beenden. Dies kann nützlich sein, falls wir beispielsweise etwas suchen. Zum Beispiel suchen wir einen bestimmten Nutzer. Dann könnten wir per while-Schleife alle Benutzer durchlaufen. Falls wir feststellen, den gesuchten Nutzer gefunden zu haben, beenden wir mittels `break` den Schleifendurchlauf.

beispiel.php

```
$max = 100;
$zaehler = 0;
$increment = 4;

while($zaehler < $max) {
   if($zaehler == 16) {
      echo "Bei der Zahl 16 hören wir auf";
      break;
   }
   echo "$zaehler , ";
   $zaehler += $increment; //Erhöht den $zaehler um den Wert $increment
}
echo "Ich bin raus";
```

Das Beispiel zählt in Schritten von `$increment` bis 100. Wenn in der Abfolge der Zahlen 16 auftaucht, wird die Schleife verlassen und das Programm nach dem Schleifenkörper fortgesetzt.

A1

Lasse das Programm auf deinem Webspace laufen und experimentiere mit verschiedenen werten von `$increment`.

Überspringen einzelner Schleifendurchläufe mit ''continue''

`continue` beendet nicht die gesamte Schleife, sondern überspringt den restlichen Schleifenkörper. Folgendes Beispiel veranschaulicht das:

beispiel.php

```
$max = 100;
$zaehler = 0;
$increment = 4;

while($zaehler < $max) {
   if($zaehler == 16) {
      echo "Die Zahl 16 wird übersprungen...";
      continue;
   }
   echo "Nicht übersprungen: $zaehler <br /> ";
   $zaehler += $increment; //Erhöht den $zaehler um den Wert $increment
}

echo "Ich bin raus";
```

A2

Lasse das Programm auf deinem Webspace laufen und experimentiere mit verschiedenen werten von `$increment`.

A3

Berechne in einer while-Schleife solange eine ganzzahlige Zufallszahl[1] zwischen 1 und 6, bis eine 6 gewürfelt wurde. Gebe auf der Konsole aus, wieviele Versuche hierzu benötigt wurden.

A4

Verändere das Programm aus der Aufgabe oben Aufgabe, indem du nun eine do/while - Schleife verwendest.

1)
Schau in der PHP Dokumentation nach, wie man Pseudozufallszahlen erzeugen kann