



### 3. Wiederholungen (die For-Schleife)

Programmiersprachen kommen ohne Schleifen nicht aus. Schleifen sind Wiederholungs-Anweisungen. Ein Text wird z.B. 100 mal geschrieben.

#### 3.1 Vergleich zur Karol

In Karol haben wir Wiederholungen bereits kennengelernt:

Beispiel:

```
wiederhole 10 mal  
schritt  
*wiederhole
```

#### 3.2 Die For-Schleife in PHP

In PHP sieht das Konstrukt anders aus und kann auch mehr, es verwendet eine Variable, die einen Start- und einen Endwert hat und bei jeder Wiederholung ihren Wert verändert.

Als Variable kann ein beliebiger Name herangezogen werden, z.B. \$i. Wir geben der Variable den Startwert 0, also

```
$i=0;
```

Wir definieren, wie oft die Wiederholung ausgeführt werden soll. Dazu bestimmen wir zuerst, wie sich der Wert von \$i bei jeder Wiederholung verändern soll. I.d.R. wird er sich um 1 erhöhen.

```
$i=$i+1;
```

Damit haben wir eine Zählvariable, die sich immer um 1 erhöht.

Zum Schluss müssen wir noch festlegen, wie oft die Wiederholung erfolgen soll, z.B. 10 mal. Da wir bei 0 beginnen werden wir die Wiederholung solange durchführen, solange \$i noch kleiner als 10 ist, also

```
$i<10;
```

Alle Drei Werte, "Startwert", "Abbruchbedingung" und "Veränderung des Startwertes" werden nun benötigt, um eine Wiederholung zu implementieren:

```
for ($i=0; $i<10; $i=$i+1) echo "Hallo";
```

#### 3.3 Anweisungsblöcke

Soll mehr als ein Befehl mehrfach wiederholt werden, so sind die Befehle in geschweifte Klammern zu setzen.

```
for ($i=0; $i<10; $i=$i+1){  
    echo "Hallo";  
    echo "<br>";  
}
```



### 3.4 Beispiele

Wie oft werden die Anweisungen aufgeführt?

```
for ($i=5; $i<10; $i=$i+1) echo "Hallo";  
    Start mit 5, dann mit 6, 7, 8, 9 also 5 mal
```

```
for ($i=5; $i<10; $i=$i+2) echo "Hallo";  
    Start mit 5, dann 7, dann 9 also 3 mal
```

```
for ($i=0; $i<100; $i=$i+10) echo "Hallo";  
    Start mit 0, dann 10, 20, 30, 40, 50, 60, 70, 80, 90 also 10 mal
```

### 3.5 Die Variable kann auch mit verwendet werden:

Die Zählvariable, in unseren Beispielen \$i kann in den Befehl mit integriert werden wie jede andere Variable auch:

```
for ($i=0; $i<10; $i=$i+1) echo "Dies ist die $i. Zahl<br>";
```

Das Ergebnis sieht dann so aus:

```
Dies ist die 0. Zahl  
Dies ist die 1. Zahl  
Dies ist die 2. Zahl  
Dies ist die 3. Zahl  
Dies ist die 4. Zahl  
Dies ist die 5. Zahl  
Dies ist die 6. Zahl  
Dies ist die 7. Zahl  
Dies ist die 8. Zahl  
Dies ist die 9. Zahl
```

Möchte man, dass die Ausgabe mit 1 beginnt so hat man zwei Möglichkeiten, entweder man lässt die For-Schleife mit 1 beginnen und bis \$i<=10 zählen oder man verändert die echo-Anweisung:

Alternative 1:

```
for ($i=1; $i<=10; $i=$i+1) echo "Dies ist die $i. Zahl<br>";
```

Alternative 2 (zur Berechnung mit Konkatination):

```
for ($i=0; $i<10; $i=$i+1)  
    echo "Dies ist die".($i+1)." Zahl<br>";
```