

# PHP-Programmierkurs [6] - for-Schleife

## for

Falls sich innerhalb eines Programms einzelne Anweisungen oder Blöcke von **Anweisungen wiederholen**, werden Schleifen verwendet.

Wir verwenden die for-Schleife, wenn dir die **Anzahl der Wiederholungen bekannt** ist oder diese sich eindeutig im Verlauf des Programms vor der Schleife ergibt (Zählschleife). <tabbox Ausprobieren> Ein Beispiel für eine for-Schleife in PHP:

```
<?php
    for ($i=1;$i<=5;$i++)
    {
        echo "Zeile $i <br>";
    }
?>
```



Kopiere den Beispiel-Code und teste ihn auf [http://www.writephponline.com/!](http://www.writephponline.com/)

Wie oft wird die echo-Anweisung im obigen Beispiel wiederholt?

- 1 mal
- 2 mal
- 3 mal
- 4 mal
- 5 mal

Check Answers

You Scored % - /



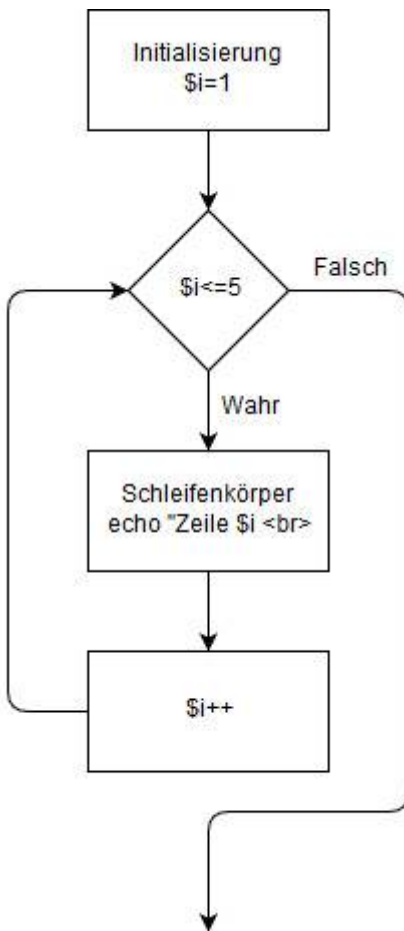
Ein komplexeres Anwendungsbeispiel für eine for-Schleife ist dieser [Notenpunkte-Rechner](#).

## PAP

In PHP sieht unser Beispiel so aus:

```
<?php
for ($i=1;$i<=5;$i++)
{
    echo "Zeile $i <br>";
}
?>
```

Das Beispiel sieht als Programmablaufplan:



Der Programmablaufplan (PAP) ist eine grafische Darstellung zur Umsetzung eines Algorithmus in einem Programm.

Hier soll er uns die Schleife veranschaulichen.

## Info

**Ein weiteres Beispiel; diesmal Schritt für Schritt erklärt:**

```
<?php
    for ($i=0;$i<3;$i++)
    {
        echo "Hello! $i";
    }
?>
```

**Browserausgabe:** Hello! 0 Hello! 1 Hello! 2

Die Zählschleife beginnt mit dem Schlüsselwort **for**. Zur Steuerung der Schleife folgen in den runden Klammern drei **Ausdrücke**:

1. Beim erstmaligen Ausführen dieser Anweisung erhält die **Schleifenvariable** `$i` den **Initialwert** 0 (`$i=0`).
2. Danach wird die **Laufbedingung** `$i<3` überprüft; nur wenn diese Bedingung erfüllt ist, wird die Schleife durchlaufen. Da 0 kleiner 3 ist, werden die Schleifenanweisungen (hier: `echo „Hello! $i“;`) abgearbeitet.
3. Nachdem die Schleife durchlaufen wurde, erhält die Schleifenvariable einen neuen Wert (Re-Initialisierung). In unserem Fall wird `$i` um eins erhöht (siehe: `i`), **erhält als den Wert 1. Danach wird erneut die Laufbedingung geprüft; da 1 kleiner 3 ist, erfolgt ein weiterer Schleifendurchlauf, an dessen Ende `$i` auf 2 erhöht wird. - Dieser Vorgang wiederholt sich so lange, bis `$i` den Wert 3 erhält. Da nun die Laufbedingung nicht mehr erfüllt ist (3 ist nicht kleiner 3), endet die Schleife und das Programm wird nach der schließenden, geschweiften Klammer fortgesetzt.** </panel> <panel type="success" title="#1"> {{url}><https://learningapps.org/watch?v=p3ub2dgon18> 80%,600px iFrame not installed}} </panel>

## #2

Schreibe eine for-Schleife, die Zahlen von 1 bis 10 ausgibt.

Nutze hierzu <http://www.writephponline.com/>!

Wenn deine Ausgabe folgendes zeigt, weißt du, ob du es richtig gemacht hast ;)



```
1
2
3
4
5
6
7
8
9
10
```





Mit `<br>` in einer echo-Anweisung erzwingst du einen Zeilenumbruch.

### #3

Schreibe eine for-Schleife, die Zahlen von 1000 bis 5000 in Tausenderschritten ausgibt.

Nutze hierzu <http://www.writephponline.com/>!

Wenn deine Ausgabe folgendes zeigt, weißt du, ob du es richtig gemacht hast ;)



```
1000
2000
3000
4000
5000
```



Re-Initialisierung: Statt `$i++` kannst du auch `$i=$i+1` schreiben.

### #4

Ein klassisches Beispiel zur gezählten Wiederholung lieferte der 9-jährige Carl Friedrich Gauss: Der Mathematiklehrer gab der Klasse die Aufgabe, die Summe aller Zahlen von 1 bis 100 zu bilden.



**Ermittle die Summe** mit PHP!

Nutze hierzu <http://www.writephponline.com/>!

### #5

Schreibe ein PHP-Programm, das alle natürlichen Zahlen von 1 beginnend bis zu einer vorgebenen Zahl `n` summiert. Eine derartige Summe nennt man auch (endliche) Reihe.



Ermittle die Summe mit PHP!



Nutze hierzu [http://www.writephonline.com/!](http://www.writephonline.com/)

## #6 - Notenschlüssel

### Linearer Notenschlüsselrechner

Schreibe ein PHP-Programm, das den linearen Notenschlüssel für Verrechnungspunkte abbildet. Beim linearen Notenschlüssel verteilen sich Punkte und Noten gleichmäßig auf der Skala.

Formel:  $\text{Note} = 6 - 5 * (\text{Erreichte Punktzahl} / \text{maximale Punktzahl})$

### Beispiel

Es gibt in einer Klassenarbeit maximal 20 Verrechnungspunkte:

```
Verrechnungspunkte maximal 20
VP 20 <--> Note 1
VP 19.5 <--> Note 1.1
VP 19 <--> Note 1.3
VP 18.5 <--> Note 1.4
VP 18 <--> Note 1.5
VP 17.5 <--> Note 1.6
VP 17 <--> Note 1.8
VP 16.5 <--> Note 1.9
VP 16 <--> Note 2
VP 15.5 <--> Note 2.1
VP 15 <--> Note 2.3
usw.
```

Mit der Funktion `round` kannst du Werte in PHP runden:

<http://php.net/manual/de/function.round.php>

Nutze hierzu [http://www.writephonline.com/!](http://www.writephonline.com/)

Ein erweiterte Aufgabenstellung findest du hier: [PHP-Arbeitsauftrag Linearer Notenschlüssel](#)

## #7

Noch mehr Aufgaben zur for-Schleife:

<https://www.w3resource.com/php-exercises/php-for-loop-exercises.php>

</accordion>

[\[informatik, thema, php, html, lernpfad, schleife\]](#)

From: <https://herr-pfeiffer.de/unterrichtswiki/> - **Unterrichtswiki - Herr Pfeiffer**

Permanent link: <https://herr-pfeiffer.de/unterrichtswiki/informatik:php:php-programmierkurs:php-p6?rev=1723191221>

Last update: **2024/08/09 10:13**

