

# 9 – while-Schleife in PHP

## Infoblatt



### ChatGPT erklärt: Wie funktioniert eine While-Schleife?

Stell dir vor, du hast eine Aufgabe, die du mehrmals wiederholen möchtest, solange eine bestimmte Bedingung erfüllt ist. Hier kommt die *While-Schleife* ins Spiel!

#### Beispiel:

Stell dir vor, du hast eine Kugel aus Eis, die 10 kg wiegt. Jede Stunde schmilzt die  *Hälfte*  davon. Wie lange dauert es, bis die Kugel nur noch 1 kg wiegt?

```
$gewicht = 10; // Anfangsgewicht der Eis-Kugel in Kg
$stunden = 0;

while ($gewicht > 1)
{
    $gewicht = $gewicht / 2;
    $stunden = $stunden + 1;
    echo "Nach Stunde $stunden wiegt die Kugel noch " .
        $gewicht . " kg.\n";
}
```

#### Was ist der Unterschied zur Zählerschleife (FOR)?

Die while-Schleife wird verwendet, wenn du *nicht im Voraus weißt*, wie oft du die Schleife durchlaufen möchtest.

### Beispielaufgabe: Kanister

Du hast einen Kanister mit 6 Litern Wasser. Du entleerst ihn in 2-Liter-Schritten. Zeige, wie viele Entleerungen nötig sind, bis der Kanister leer ist.

*Wir sollten als Ergebnis die Zahl 3 erhalten:*

- Entleerung 1: noch 4 Liter im Kanister
- Entleerung 2: noch 2 Liter im Kanister
- Entleerung 3: Kanister leer!

## Programmiertechnisches Vorgehen:

### a) Welche Variablen brauchen wir?

- Wir brauchen eine Variable mit dem aktuellen Füllstand (`$fuellstand_liter`).
- Wir brauchen eine Variable, mit der wir die Entleerungsvorgänge zählen (`$anzahl_entleerungen`).

```
$fuellstand_liter = 6; // Anfangsmenge im Kanister
$anzahl_entleerungen = 0; // Zähler für die Schleife
```

Code

### b) Schleife entwerfen

Der Füllstand soll sich mehrmals um 2 Liter verringern. Jedes mal, wenn das passiert, erhöht sich die Anzahl der Entleerungen um 1:

```
$fuellstand_liter = $fuellstand_liter - 2;
$anzahl_entleerungen++;
```

Code

Aber wie lange soll die Schleife laufen?

Antwort: So lange, sich noch Wasser im Kanister befindet – also solange `$fuellstand_liter > 0`

```
while($fuellstand_liter > 0) {
    // zwei Liter ausleeren
    // Anzahl der Entleerungen hochzählen
}
```

Code

Damit ergibt sich folgender Gesamtcode:

```
<?php
    $fuellstand_liter = 6; // Anfangsmenge im Kanister
    $anzahl_entleerungen = 0; // Zähler für die Schleife

    while ($fuellstand_liter > 0)
    {
        $fuellstand_liter = $fuellstand_liter - 2;
        $anzahl_entleerungen++;
        echo "<p>$anzahl_entleerungen mal entleert: $fuellstand_liter
            Liter verbleiben.</p>";
    }
    echo "<p>---</p>";
    echo "<p>Kanister leer - $anzahl_entleerungen mal entleert.</p>";
?>
```

Code

## Elemente einer While-Schleife

Eine while-Schleife besteht aus folgenden Elementen:

Code

```
$tag = 0;
$alter = 7512;

while ($tag < 10)
{
    $alter = $alter + 1;
    echo "<p>Ich bin schon wieder einen Tag älter ...</p>";
    $tag++;
}
```

<code>\$tag = 0;</code> <code>\$alter = 7512;</code>	Initialisierung der Variablen.
<code>\$tag &lt; 10</code>	Die Schleifenbedingung. So lange diese Bedingung wahr ist, läuft die Schleife.
<code>\$alter = \$alter + 1</code>	Beliebige Rechnung, die für jeden Schleifendurchlauf ausgeführt wird.
<code>\$tag++</code>	Die Zählervariable wird hochgezählt.

### ! WICHTIG! !

Achte darauf, dass du die Variable, die in der Bedingung steht, innerhalb der Schleife änderst. Andernfalls stoppt die Schleife nie, was zu einer Endlosschleife führen kann.

*Stell dir vor, du würdest im obigen Beispiel das `$tag++`; weglassen. Dann hätte `$tag` immer den Wert 0. Damit würde `$tag` niemals 10 erreichen. Dann stoppt die Schleife nie, weil die Bedingung (`$tag < 10`) immer wahr bleibt.*

## Begriffe

<b>Schleifenbedingung</b>	Die Bedingung, die <i>true</i> sein muss, damit die Schleife ausgeführt wird. Sobald die Bedingung <i>false</i> zurückgibt, endet die Schleife. Wird diese niemals <i>false</i> , läuft die Schleife ewig.
<b>Kopfgesteuerte Schleife</b>	Die while-Schleife prüft die Bedingung <b>am Anfang</b> eines jeden Durchlaufs. Ist die Bedingung von Anfang an nicht erfüllt, wird der Code innerhalb der Schleife <b>überhaupt nicht ausgeführt</b> . <code>while(... Bedingung ...)</code> <code>do { ... }</code>
<b>Fußgesteuerte Schleife</b>	Mit der befassen wir uns aktuell <b>nicht</b> . Bei ihr wird die Schleife zuerst ausgeführt, bevor die Bedingung überprüft wird. <code>do { ... }</code> <code>while(... Bedingung ...)</code>