JavaScript – LOOPS mit For-Schleife

1)For-Schleifen
2)Beispiele For-Schleife

1)Programmteile wiederholen - Schleifen

Neben den Verzweigungen gehören die Schleifen zu den wichtigen Kontrollstrukturen. Viele Vorgänge, die sich auf dieselbe oder recht ähnliche Weise wiederholen, können mit Hilfe von Schleifen effizient programmiert werden. Sie nutzen die Fähigkeit eines Rechners, sehr viele Schritte in sehr kurzer Zeit durchzuführen.

Schleifen mit „for“

Eine Schleife mit „for“ nutzt man, wenn man als Entwickler weiß, **wie oft ein Programmteil** wiederholt werden soll. Sie ist auch dann sinnvoll, wenn ein Programmteil für eine regelmäßige Abfolge von Zahlen, die von einem Startwert bis zu einem Endwert laufen, wiederholt werden soll. Bei for-Schleifen wird zur Steuerung meist eine Variable eingesetzt. Sie wird Schleifenvariable genannt.

Der Aufbau einer for-Schleife gliedert sich in drei Teile (Parameter):

- eine Anweisung für den Startwert der Schleifenvariablen
- eine Bedingung, die während des gesamten Ablaufs der Schleife für die Schleifenvariable gelten muss
- eine Anweisung für die Änderung

bzw. man kann auch sagen:
for mit zwei runden Klammern und dann die geschweiften Klammern für den Code. Die runde Klammer besteht aus drei Bestandteilen, zwischen denen jeweils ein Semikolon steht
for (;;)

- am Beginn die Initialisierung
- an zweiter Stelle steht eine Bedingung
- es folgt eine Aktion

```
<script>
for (initialisierung; bedingung; aktion) {
    //Code
}
</script>
```

Was passiert nun:
Ganz am Anfang wird die Initialisierung ausgeführt. Dann Schritt 2: Wird bei der Überprüfung der Bedingung ein „wahr“ also ein „true“ ausgegeben, wird der Code in der geschweiften Klammer ausgeführt. Nachdem dieser ausgeführt wurde, wird die Aktion ausgeführt. Dann wiederholen sich die Bedingung, bei „true“ der Code und die Aktion usw.

So könnte die Reihenfolge aussehen: von 1 bis 7

```
<script>
for (initialisierung (1); bedingung (2)(5); aktion (4)(7)) {
    //Code (3)(6)
}
</script>
```

Tipp: Eigentlich sollte man die Variable mit "let" definieren und nicht mit „var". Der Vorteil dabei ist nämlich, dass „let" nur in der Schleife gültig ist und nicht außerhalb.

Beispiel:

Eigentlich hat es sich eingebürgert, die Variable für die Initialisierung „i" zu nennen, daher:

1. Die Zahl wird mit „1" festgelegt: **let i = 1;**
2. Damit nun das Hinaufzählen um eins nicht endlos läuft, kommt die Bedingung nun ins Spiel: die „Bedingung" steht an zweiter Stelle, damit sie vorher überprüft wird, bevor der Teil 3, nämlich die „Aktion" ausgeführt wird: **i <= 10;**
   Hier wird sozusagen die Gültigkeit angegeben.
3. Die Aktion soll sein, dass die Zahl „i" um eins erhöht wird: **i++**

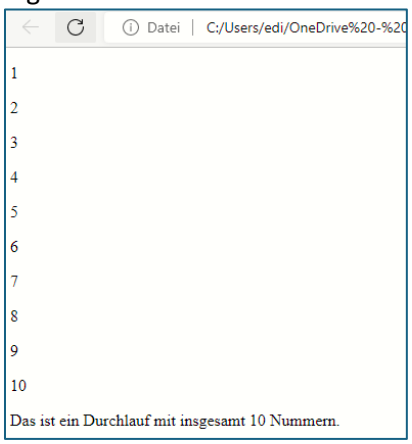
Dann müssen die Daten ausgegeben werden: mit „alert" nicht sinnvoll, da müsste man 10mal das Fenster schließen. Daher wähle „document.write".

Damit die Zahlen ausgegeben werden, muss man **IN DER FOR-Schleife die Ausgabe** schreiben.

Die Ausgabe des Textes erfolgt außerhalb der Schleife!

```
 8 <body>
 9   <script>
10     for (var i = 1; i <= 10; i++) {
11         document.write("<p>" + i + "</p>");    //Ausgabe aller Zahlen
12
13         text = "Das ist ein Durchlauf mit insgesamt " + i  + " Nummern.";
14     }
15     document.write(text);
16   </script>
17 </body>
18 </html>
```

Ergebnis:



Weiteres Beispiel:

```
<body>
 <script>
        let i;
        // 1: Aufwärts
        document.write("<p>1: ");
        for(i=1; i<=5; i++)
        document.write(i + " ");

        // 2: Abwärts
        document.write("<br>2: ");
        for(i=20; i>=10; i--)
        document.write(i + " ");
 </script>
</body>
```

Erklärung:
Die Variable i soll zunächst als Schleifenvariable dienen.

\\1:
In der ersten Schleife erhält i den Startwert 1. Als Bedingung für die gesamte Schleife gilt: Sie läuft, solange i kleiner als 5 oder gleich 5 ist. Nach jedem Durchlauf wird der Wert von i mit Hilfe des Zuweisungsoperators ++ um 1 erhöht. Damit ergibt sich eine Zahlenfolge von 1 bis 5, in Schritten von 1. Startwert, Bedingung und Änderung werden jeweils durch Semikolon voneinander getrennt.
Innerhalb der ersten Schleife steht nur eine Anweisung: Die Ausgabe der Schleifenvariablen.

\\2:
Die zweite Schleife läuft abwärts, im Gegensatz zur ersten Schleife. Sie startet bei 20 und läuft, solange der Wert von i größer als 10 oder gleich 10 ist. Nach jedem Durchlauf wird der Wert von i um 1 vermindert. Damit ergibt sich die Zahlenfolge 20, 19, 18 ... 10.
Die verschiedenen Teile einer Schleife müssen aufeinander abgestimmt sein. Eine Schleife mit for(i=10; i>=5; i++) läuft endlos. Eine Schleife mit for(i=10; i<=5; i--) läuft nie. Solche Schleifen sollten Sie natürlich vermeiden.

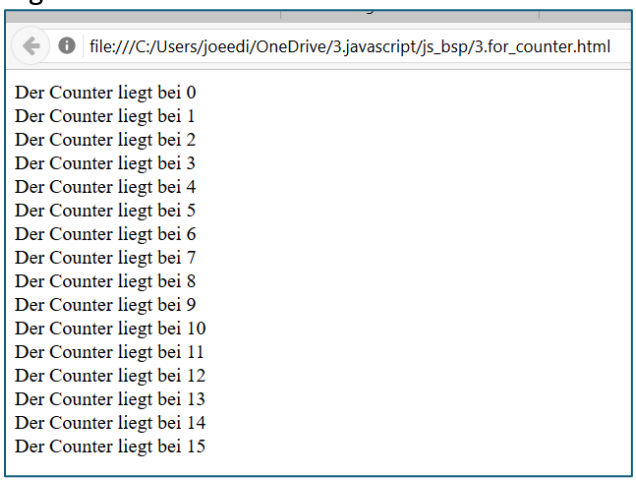
Übung mit einem „counter“

Solange die Variable kleiner ist als der Wert (hier 15), wird der Code innerhalb der Schleife ausgeführt. Es soll der untenstehende Satz solange ausgegeben werden, bis der Wert vom „counter“ erreicht wird.

Erstelle das File „3.for_counter.html“:

```
<!doctype html>
<head>
<meta charset="UTF-8">
<title>counter</title>
</head>
<body>
  <script>
    for(var counter = 0; counter <=15; counter++) {
        document.write("Der Counter liegt bei " + counter + "<br>");
    }
  </script>
</body>
</html>
```

Ergebnis:



2)Mehrfachauswahl mit einem Listenfeld, IF-Anweisung und FOR-Schleife

Auch hier wird das select-Objekt verwendet. Das Standardverhalten einer Drop-Down-Liste ist nicht sinnvoll anwendbar, wenn mehrere Auswahlkriterien zulässig sind, da der User alle Optionen auf einmal sehen muss.

Erstelle das File „2.js_event_mehrfachauswahl.html“.

- Das select-Objekt benötigt eine ID, damit man es in JavaScript auslesen kann.
- Das Attribut multiple=“multiple“ weist den Browser an, mehrere Eigenschaften zu akzeptieren, für die sich ein Benutzer entschieden hat.
- Größe mit size angeben: dieser Parameter gibt die Anzahl der anzuzeigenden Zeilen an und sorgt dafür, dass alle Elemente der Liste angezeigt werden.
- Das „output-div“ bietet den Platz, wo die Antwort angezeigt wird.

```
<html>
<head>
    <meta charset="UTF-8">
    <title>Title</title>
</head>
<body>
<h1>Merhfachauswahl</h1>
    <form>
    <fieldset>
        <label>Diese Sprachen kenne ich. Markiere mit gedrückter STRG-Taste mehrere
        Elemente: <br><br></label>
        <select id="spracheAuswaehlen" multiple ="multiple" size="5">
            <option value="HTML">HTML</option>
            <option value="CSS"> CSS</option>
            <option value="PHP"> PHP</option>
            <option value="JavaScript">JavaScript</option>
            <option value="Python">Python</option>
        </select>
            <button type="button" onclick="auswahlAnzeigen()">übernehmen</button>
    </fieldset>
    </form>

    <div id="output"></div>

</body>
```

Code:

```
<html>
<head>
  <meta charset="UTF-8">
  <title>Title</title>
</head>
<body>
  <h1>Merhfachauswahl</h1>
  <form>
    <fieldset>
      <label>Diese Sprachen kenne ich. Markiere mit gedrückter STRG-Taste mehrere
Elemente <br><br></label>
      <select id="spracheAuswaehlen" multiple ="multiple" size="5">
      <option value="HTML">HTML</option>
      <option value="CSS"> CSS</option>
      <option value="PHP"> PHP</option>
      <option value="JavaScript">JavaScript</option>
      <option value="Python">Python</option>
      </select>
      <button type="button" onclick="auswahlAnzeigen()">übernehmen</button>
    </fieldset>
  </form>

  <div id="output"></div>

</body>
</html>
```

Nun folgt der JavaScript Code:

Eigentlich ist eine Liste mit option-Objekten in einem select-Objekt eine Art Array.

- Die Variable repräsentiert das gesamte select-Objekt:
  var spracheAuswaehlen = document.getElementById("spracheAuswaehlen");

- Für eine komplexe HTML_Ausgabe ist es einfacher mit einer String-Variablen zu arbeiten, als direkt auf die Elemente Code zu schreiben:
  var ergebnis = "<h2>Deine Sprache: </h2>";

- Eine unsortierte Liste, die die Ergebnisse anzeigt
  ergebnis += " \n";

- Die for-Schleife durchläuft das Listenfeld. spracheAuswaehlen besitzt wie ein Array eine Eigenschaft „length“.
  for (i=0; i < spracheAuswaehlen.length; i++){

- Die Variable „aktuelleAusgabe“ ist eine temporäre Variable, die beim Abarbeiten der Schleife alle Verweise auf die einzelnen option-Elemente im ursprünglichen select-Objekt aufnimmt:

aktuelleAuswahl = spracheAuswaehlen[i];

- Überprüfe, ob das aktuelle Element ausgewählt wurde:
  die „aktuelleAuswahl“ ist ein Objekt, das eine Eigenschaft „selected“ besitzt. Diese Eigenschaft sagt, ob das Objekt vom Benutzer markiert worden ist. „selected“ ist eine Boolsche Eigenschaft, wodurch die einzelnen zulässigen Werte „true“ oder „false“ sind.
  if (aktuelleAuswahl.selected == true){

- Wenn das Element markiert wurde, wird in der unsortierten Liste, die in der Variable „ergebnis“ zuhause ist, ein Eintrag angelegt.
  ergebnis += " " + aktuelleAuswahl.value + " \n";

- Das Attribut „innerHTML“ von „output“ bietet sich für die Ausgabe der unsortierten Liste an:
  output = document.getElementById("output");
  output.innerHTML = ergebnis;

Die Optionen eines Auswahlfeldes verhalten sich hier wie ein Array.

```
<script>
    function auswahlAnzeigen(){
        //Daten empfangen
        var spracheAuswaehlen = document.getElementById("spracheAuswaehlen");

        //String für die Ausgabe erstellen
        var ergebnis = "<h2>Deine Sprache: </h2>";
        ergebnis += "<ul> \n";

        //die Optionen durchlaufen
        for (i=0; i < spracheAuswaehlen.length; i++){
            //die aktuelle Auswahl untersuchen
            aktuelleAuswahl = spracheAuswaehlen[i];
            //falls markiert, drucken
            if (aktuelleAuswahl.selected == true){
                ergebnis += " <li>" + aktuelleAuswahl.value + "</li> \n";
            } //ende if
        } //ende for-schleife

        //die liste abschließen und ausgeben
        ergebnis += "</ul> \n";

        output = document.getElementById("output");
        output.innerHTML = ergebnis;
    } //ende auswaehlenAnzeigen
</script>
</html>
```

Code:

```
<script>
  function auswahlAnzeigen(){
    //Daten empfangen
    var spracheAuswaehlen = document.getElementById("spracheAuswaehlen");
```

```
      //String für die Ausgabe erstellen
      var ergebnis = "<h2>Deine Sprache: </h2>";
      ergebnis += "<ul> \n";

      //die Optionen durchlaufen
      for (i=0; i < spracheAuswaehlen.length; i++){
         //die aktuelle Auswahl untersuchen
         aktuelleAuswahl = spracheAuswaehlen[i];
         //falls markiert, drucken
         if (aktuelleAuswahl.selected == true){
            ergebnis += " <li>" + aktuelleAuswahl.value + "</li> \n";
         } //ende if
      } //ende for-schleife

      //die liste abschließen und ausgeben
      ergebnis += "</ul> \n";

      output = document.getElementById("output");
      output.innerHTML = ergebnis;
   } //ende auswaehlenAnzeigen
</script>
```