JavaScript – Arrays

Eine Funktion ist eine benannte Gruppe von Codezeilen. Ein ARRAY ist etwas Ähnliches.

- Array werden mit [ ] „eckigen Klammern“ definiert.
- Arrays sind **besondere Arten von Variablen**. Darin kann man nicht nur einen Wert speichern, **sondern mehrere Werten**.
- Es handelt sich dabei um eine benannte Gruppe von Variablen.
- Man kann Arrays verwenden, wenn man mit einer Liste von Werten desselben Datentyps arbeiten möchte.
- Anschließend weißt man der Arrayvariablen so viele Werte zu, wie man möchte, in eckigen Klammern, getrennt durch Kommas. Das Leerzeichen nach dem Komma ist keine Pflicht, aber es hilft bei der Lesbarkeit.

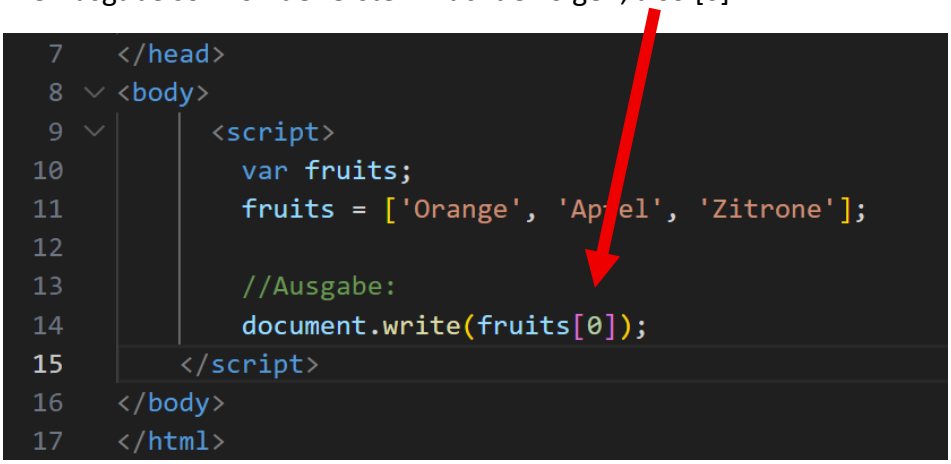
- Hier werden, wie bei Objekten, sogenannte „key – value“ Paare erstellt.
- Der key ist die Reihung beginnend mit „0“.

Beispiel:

Erstelle das File „array.html“:

Erstelle daher wieder eine boiulerplate-Datei (Grundgerüst mit ! + Enter-Taste)

Die Ausgabe soll von der ersten Frucht erfolgen, also [0]

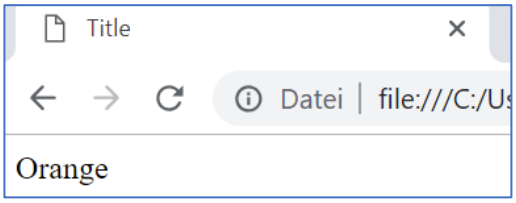


Code:

```
<script>
    var fruits;
    fruits = ['Orange', 'Apfel', 'Zitrone'];

    //Ausgabe:
    document.write(fruits[0]);
 </script>
```

Ergebnis im Browser:



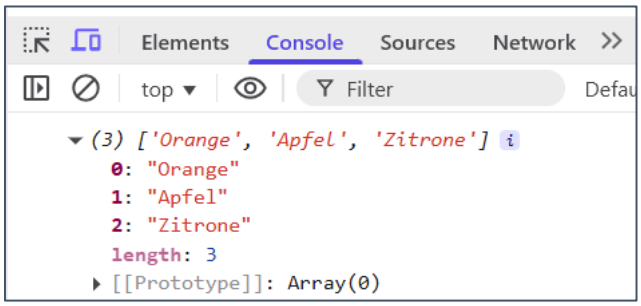
Anzeige in der Konsole:

Ändere die Ausgabe um auf die Konsole. Kommentiere das „document.write“ aus und schreibe

console.log(fruits);

```
10        let fruits;
11        fruits = ['Orange', 'Apfel', 'Zitrone'];
12
13        //Ausgabe:
14        // document.write(fruits[0]);
15        console.log(fruits);
16    </script>
17  </body>
```

Klicke im Browser auf F12 und wähle Console:
Klicke auch auf das kleine Dreieck vor der Zahl zum Aufklappen.



Hier sieht man auch die Nummerierung ab 0 sehr gut.

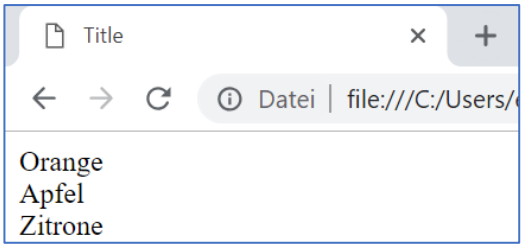
Alle ausgeben: FOR-Schleife verwenden

Nun wieder im Browser weiter und weg von der Konsole. Daher die Konsole auch schließen und den Code wieder mit „document.write“ ausgeben.

In den array-Klammern die 0 gegen das i austauschen nicht vergessen.

```
for (i = 0; i<3; i++)
        document.write(fruits[i] + "<br>");
```

Ergebnis:



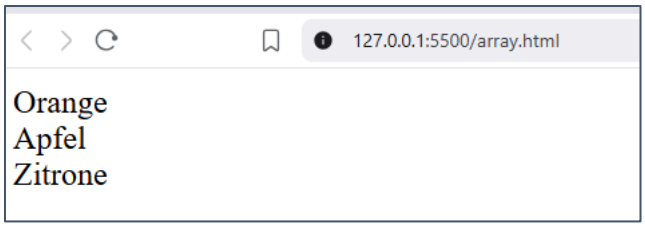
bzw.

for (i = 0; i<fruits.length; i++)

document.write(fruits[i] + "
");

```
8   <body>
9       <script>
10        let fruits;
11        fruits = ['Orange', 'Apfel', 'Zitrone'];
12
13        //Ausgabe:
14        for (i = 0; i<fruits.length; i++)
15        document.write(fruits[i] + "<br>");
16
17      </script>
```

Ergebnis im Browser:



Funktion in einem Array: length

Die Länge des Arrays **(fruits.length)** wird in der Bedingung der for-Schleife verwendet. Der Vorteil ist, dass die Schleife automatisch an die Größe des Arrays angepasst wird, auch wenn man Elemente hinzufügt oder entfernt.

Eigenschaft „length“

Übung:

Nutze „length“ und gib letztes Element aus.

Das letzte Element ist immer die gesamte Länge minus 1, da die Liste mit 0 beginnt.

Zur Abgrenzung zur vorhergehenden for-Schleife soll direkt mit JavaScript-Code eine horizontale Linie eingefügt werden: der Tag dafür ist <hr>

document.write("<hr>"); //horizontale Linie

```
<script>
  let fruits;
  fruits = ['Orange', 'Apfel', 'Zitrone'];

  //Ausgabe:
  for (i = 0; i<fruits.length; i++)
  document.write(fruits[i] + "<br>");

  document.write("<hr>"); //horizontale Linie

  document.write(fruits[fruits.length - 1]);
```

Ergebnis:

Kurzschreibweise:

Die einzelnen Werte werden nacheinander in die eckigen Klammern geschrieben. Getrennt werden sie jeweils durch ein Komma. Das Leerzeichen nach dem Komma ist keine Pflicht, aber es hilft bei der Lesbarkeit.

Nach dem letzten Element sollte kein weiteres Komma geschrieben werden.

```
var dieMannschaft = [ "Hans", "Josef"];
```

Zahlen können ohne Anführungszeichen geschrieben werden. Texte und Zahlen können auch ohne Probleme gemischt vorkommen.

```
var allesMoegliche = [ "Hans", 42, 25, "Patrick"];
```

- **Zugriff auf einzelne Elemente**

Jedes Element, jeder Wert eines Arrays bekommt automatisch eine eindeutige, fortlaufende Nummer. Über diese Nummer kann man jeden Wert auslesen, einer anderen Variablen zuordnen oder sogar direkt als Teil einer Berechnung verwenden.

Abrufen lassen sich die Werte eines Arrays über den Namen der Variablen und die Position innerhalb des Arrays, ebenfalls in eckigen Klammern.

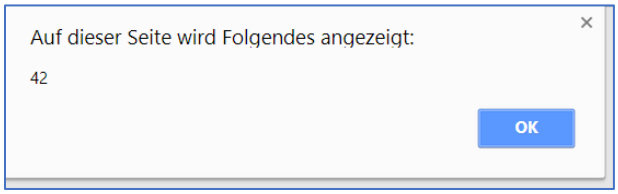
Die Position, also die Nummer in den eckigen Klammern, nennt man Schlüssel.

Jedoch beginnt der Computer mit „0“ zu zählen an, nicht bei „1“.

```
alert( allesMoegliche[1]);
```

Ist die 2. Position!!!

Lösung:



Das Ergebnis ist also nicht „Hans“ sondern „42“.

Info: Zusätzliche Funktionen

- push() – am Ende hinzufügen
- unshift() – am Anfang hinzufügen
- pop() – den letzten Wert löschen
- shift() – den ersten Wert löschen
- join() mehrere Arrays vereinen und
- sort() sortieren.

Beispiel push()

Es soll am Ende der Obstliste eine neue Sorte, nämlich Erdbeere hinzugefügt werden.

Da die alte Ausgabe bestehen bleiben soll, muss darunter die push() erstellt werden und nochmals darunter eine neue Ausgabe aller Früchte.

Zur Abgrenzung dazwischen wurde eine horizontale Linie eingefügt, nicht mit einer Unterbrechung des <script>-Tags, sondern gleich direkt in JavaScript mittels

```
document.write("<hr>"); //horizontale Linie
```

```
<script>
  let fruits;
  fruits = ['Orange', 'Apfel', 'Zitrone'];

  //Ausgabe:
  for (i = 0; i<fruits.length; i++)
  document.write(fruits[i] + "<br>");

  document.write("<hr>"); //horizontale Linie

  fruits.push("Erdbeere");
  // Ausgabe nach dem Hinzufügen eines neuen Elements:
  for (i = 0; i<fruits.length; i++)
  document.write(fruits[i] + "<br>");

</script>
```

Ergebnis:

Quelle:

https://www.youtube.com/watch?v=EenvvRCcVl4

https://www.youtube.com/watch?v=cAiSbkWvBQU ab ca. 56:30 Minuten (Mario,2022)

Florian Franke, Johannes Ippen; in: Apps mit HTML5, CSS3 und JavaScript;
Rheinwerk Verlag, Bonn 2015; S. 106-107

Andy Harris, in: JavaScript für dummies; Wiley-Vch Verlag, 2017, Weinheim; S.128-143