# L1\_2 Array

Arrays sind spezielle Variablen, die nicht nur einen Wert, sondern eine Vielzahl von Werten speichern können. Dafür wird der Wert in ein bestimmtes Feld abgelegt. Um auf den Inhalt eines Feldes (das Element) zugreifen zu können, werden Indizes verwendet.

Arrays sind vergleichbar mit Regalen. Wir befüllen die einzelnen Fächer mit Werten, die in einer bestimmten Weise zusammengehören (bspw. die Primzahlen bis 50).

Beispiel: primzahlen

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Index | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
| Feldinhalt | 2 | 3 | 5 | 7 | 11 | 13 | 17 | 19 | 23 | 29 | 31 | 37 | 41 | 43 | 47 |

Im oben dargestellten Array primzahlen sind alle Primzahlen unter 50 aufgeführt. Auffällig ist hierbei, dass der Index (Fachnummer) bei 0 anfängt und nicht bei 1. Auf den Vergleich mit dem Regal bezogen, könnte man sagen, dass im Fach mit der Fachnummer 0 der Wert 2 liegt, im Fach mit der Fachnummer 8 liegt der Wert 23 liegt, usw.

**Deklarierung und Initialisierung in JavaScript**

Es gibt verschiedene Möglichkeiten in JavaScript ein Array zu erzeugen. Zwei Möglichkeiten werden hier aufgeführt:

Möglichkeit 1:

|  |  |
| --- | --- |
| JavaScript | Struktogramm |
| //Deklaration und Initialisierung  var himmelsrichtung = [ ] ;  //Zuweisung  himmelsrichtung[0] = "Nord" ;  himmelsrichtung[1] = "Ost" ;  himmelsrichtung[2] = "Süd" ;  himmelsrichtung[3] = "West" ; |  |

**Hinweis:** Die Größe des Arrays muss bei der Erzeugung nicht angegeben werden, da in JavaScript dynamisch Werte hinzugefügt und entfernt werden können.

Möglichkeit 2

|  |  |
| --- | --- |
| JavaScript | Struktogramm |
| //Deklaration und Initialisierung  var primzahlen = [2, 3, 5, 7]; |  |

**Zugriff auf Arrayelemente mit einem Index**

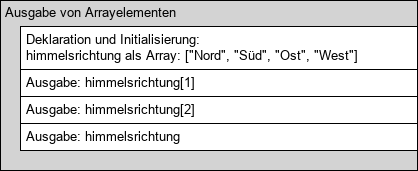
Auf die einzelnen Arrayelemente kann mit dem Index sowie einer Schleife zugegriffen werden. Möchte man nur auf ein bestimmtes Element zugreifen, bietet sich der Zugriff mit dem Index an. Dabei gilt es zu beachten, dass auf das erste Element in einem Array mit dem Index 0 zugegriffen wird. Folglich wird auf das letzte Element (n-tes Element) in einem Array mit dem Index n-1 zugegriffen.

var himmelsrichtung = ["Nord", "Süd", "Ost", "West"];

window.alert(himmelsrichtung[1]) ; //Ausgabe: Süd

window.alert(himmelsrichtung[0]) ; //Ausgabe: Nord

window.alert(himmelsrichtung) ; //Ausgabe: Nord Süd Ost West

Struktogramm

**Zugriff auf Arrayelemente mit einer Schleife**

Sollen alle Elemente eines Arrays ausgegeben werden, ist es bequemer eine Schleife für die Ausgabe zu verwenden. Dafür ist es sinnvoll die Eigenschaft length zu verwenden. Die Eigenschaft length ermittelt die Anzahl der Elemente, die in einem Array gespeichert sind.

var himmelsrichtung = ["Nord", "Süd", "Ost", "West"];

var anzHimmelsrichtungen = himmelsrichtung.length;

(🡺 Ausgabe: 4)

window.alert(anzHimmelsrichtungen);

for(var i = 0; i < anzHimmelsrichtungen; i++) {

(🡺 Ausgabe: Nord

Süd

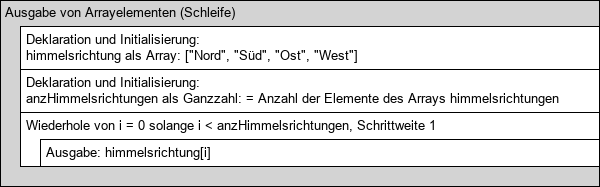
Ost

West)

window.alert(himmelsrichtung[i]) ;

}

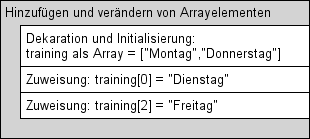
Die Variable i wird für die Navigation im Array genutzt.

Struktogramm

**Hinzufügen und verändern von Arrayelementen**

Im Gegensatz zur allgemeinen Definition ist ein Array in JavaScript nicht abschließend, das heißt, es können weitere Elemente hinzugefügt werden. Außerdem können bereits bestehende Elemente mit neuen Werten überschrieben werden.

|  |  |
| --- | --- |
| **Code** | **Array** |
| var training = ["Montag", "Donnerstag"]; | training["Montag","Donnerstag"] |
| //verändern  training[0] = "Dienstag" ; | training["Dienstag", "Donnerstag"] |
| //hinzufügen  training[2] = "Freitag" ; | training["Dienstag", "Donnerstag",  "Freitag"] |

Struktogramm