

### Arrays

**Definition** Datentyp Arrayname[Anzahl\_der\_Elemente]

**Definition mit Initialisierung**  
`int Zahlen[10] = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10};`  
*Die Indizes liegen im Bereich von 0 - 15*

**Alternativ**  
`int Zahlen[] = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10};`  
*Die Indizes liegen im Bereich von 0 - 15*

**Mehrdimensionale Arrays**  
 Datentyp Arrayname[Menge][Menge]

**Zugriff auf Array (Arrayschreibweise)**  
`int Array[3] = {1, 2, 3};`  
`int i = Array[1]; //i = 2`

**Zugriff auf Array (Zeigerschreibweise)**  
`int Array[3] = {1, 2, 3};`  
`int *ptr = Array;`  
`int i = *(ptr + 1); //i = 2`

### Strukturen

**Definition**  
`struct [Typname]`  
`{`  
`Aufbau`  
`}` [Variable nliste];

**Beispiel**  
`struct Buch`  
`{`  
`char Titel[ 100];`  
`char Autor [100];`  
`}` Bueche r[50]; // 50 mal struct Buch

**Strukturen in Verbindung mit Typedef**  
`typedef struct sBuch`  
`{`  
`char* Titel;`  
`char* Autor;`  
`}` TBuch;  
 Definiert einen neuen Datentyp "TBuch"

**Zugriff auf Daten eines Zeigers auf eine Struktur**  
`TBuch* buch = malloc(sizeof(TBuch));`  
`buch->Titel = "Buchtitel";`  
`buch->Autor = "Autor";`

### Strukturen (cont)

**Zugriff auf Daten einer Struktur (kein Zeiger)**  
`TBuch buch;`  
`buch.Titel = " Buch titel`  
 `";`  
`buch.Autor = " Aut or";`

### Zeichenketten

**Definition**  
`char Text[] = {'D','i','e','s',' ',' ','i','!',' ','\0'};`

**Bessere Art der Definition**  
`char Text[] = "Dies ist ein Text!";`

Am Ende der Zeichenkette steht das Abschlusszeichen \0 In einer Schleife kann man damit das Ende der Zeichenkette finden.

*Beispiel:*

`while (Kette[i] != '\0')`  
 Anweisung;

### Aufzählungstitel (enum)

**Definition**  
`enum [Typname] {Aufzählung} [Variable nliste]`

**Beispiel**  
`enum Wochentag {Sonntag, Montag, Dienstag,`

**Variablen des Typs Wochentag definieren**  
`enum Wochentag Werktag, Feiertag, Heute =`

**Beispiele (richtig & falsch)**  
`Werktag = Montag; //richtig`  
`Feiertag = Sonntag; //richtig`  
`int i = Freitag; //Richtig (impl. Typumwa`  
`Heute = 4; //falsch! keine Typumw. möglich`  
`int i = Montag + Dienstag; // richtig (abe`  
`Heute = Montag + Dienstag; //falsch`  
`Heute++; //falsch`

**enums mit alternativen Zahlenwerten**  
`enum Farben {weiss = 0, blau = 2, gruen =`



### Unions

**Definition**    `union [Typname]`  
                  `Aufbau`  
                  `} [Variable nliste];`

**Beispiel**      `union Ueberlagerung`  
                  `int Zahl;`  
                  `unsigned char c[size of( int)];`  
                  `} u;`



By **TimSch**  
[cheatography.com/timsch/](https://cheatography.com/timsch/)

Published 24th July, 2016.  
Last updated 17th March, 2018.  
Page 2 of 2.

Sponsored by **Readable.com**  
Measure your website readability!  
<https://readable.com>