

### Datentypen

Typ	Bytes	Bereich
int	2 oder 4	$\pm 32.000$ oder $\pm 2.000.000.000$
double	8	Kommazahl: 15 Stellen und $\pm 10^{308}$
char	1	ASCII Buchstaben
int_Mt	N	$-2^{N-1}$ bis $2^N-1$
uint_Mt	N	0 bis $2^N-1$

### Operatoren

+ , - , * , /	Plus, Minus, Mal, Geteilt
%	Rest ( $9 \% 4 = 1$ )
== , !=	Gleich, Ungleich
> , < , >= , <=	Größer, Kleiner
&& ,    , !	Und, Oder, Nicht

### Zahlen, Zeichen und Strings

char c ='A';	einzelnes Zeichen
char s[6] = "Hallo";	Zeichenkette/String

#### Escape-Sequenzen:

\ , \ " , \ ? , \ \ , \ n , \ t	' , " , ? , \ , neue Zeile, Tab
---------------------------------	---------------------------------

#### Zahlenformate:

int bin= 0b1010110	int octal = 021
int hex = 0x1A	double exponent = 1.5e3

Jeder String endet intern mit \0, "Hallo" == "Hallo\0 "

### Arrays

int numbers[3] = { 4, 5, 6 };	
int numbers[100] = { 0 };	Alle Elemente sind 0
int table[3][3];	2 Dimensionales Array
char *colors[2] = { "blue", "yellow" }	

Intern sind Arrays Zeiger:  $a[0] == *a$  ,  $a[1] == *a(a + 1)$

### Zeiger

&	Adressoperator
*	Dereferenzierungsoperator
int *pi = &i;	Speichern der Adresse von i
*pi = 5;	Zugriff auf i

Adresse zum Kopieren und Teilen von Daten

### Schleifen und Verzweigungen

#### Loops

```
for (int i = 0; i < 4; i++) { ... }
for (int i = 0, j = 10; i < 5; i++, j--){ ... }
while (a == 7) { ... }
do { ... } while (a == 5);
```

#### If...else

```
if (n < 2) { ... }
else if (n < 4) { ... }
else { ... }
```

#### Switch

```
switch (n) {
    case 1: { ... }; break;
    case 2: { ... }; break;
    def ault: { ... }; break;
}
```

#### Break/ Con tinue

break: beendet Schleife  
continue: beendet Durchlauf

### Funktionen

```
int sum(int a, int b) {
    return a + b;
}
void add5(int *a) {
    *a = *a + 5;
    return;
}
// main function
int main() {
    return 0;
}
```

### Präprozessor

#define MAX 100	Konstante
#include <filename>	Importiert Standard-Bibliothek
#include "filename"	Impotiert eigene Header-Datei

In einer Header-Datei werden Funktionen deklariert, die in mehreren Dateien genutzt werden.



### Speicherverwaltung

```
// Datenmenge einlesen
scanf("%i", &dataCount);
// Speicherplatz reservieren
int *dynData;
dynData = (int*) malloc(sizeof(int) * dataCount);
// Speicherplatz freigeben
free(dynData);
```

### Enum

```
typedef enum { SLOW = 5, FAST = 100 } Speed;
Speed carSpeed = SLOW;
// default: C1 = 1, C2 = 2 usw.
```

### Strukturen

```
typedef struct {
    int day;
    char month[4];
} Date;
Date today= { 1, " Jun " }; // Anlegen einer
Instanz
today.day = 2; //Zugriff auf Elemente
Date *pToday = &today //Anlegen eines Zeigers
pToday ->day = 3; // Zugriff auf Elemente
/*Alte rna tiven mit wenig Speich erv erb rauch:
union und bitfield*/
```

### Dateibearbeitung

fopen()	Öffnen einer Datei
fclose()	Schließen einer Datei
fseek()	Verändern der Beareitungsposition
fgetc(), fputc()	Zeichenweise Lesen/Schreiben
fscanf(), fprintf()	Formatiertes Lesen/Schreiben
fgets()	Zeilenweise Lesen
fputs()	Stringweise Scheiben
fread(), fwrite()	Binäres Lesen/Schreiben (byteweise)
sscanf(), sprintf()	Strings Lesen/Schreiben

Details auf <https://cplusplus.com/reference/cstdio>

### Input & Output

#include <stdio.h>	Bibliothek für Ein-/Ausgabe
printf("My age is %d", age);	Ausgabe auf Konsole
scanf("%d", &value)	Tastatureingabe (unsafe)
%d , %c , %s, %lf	int, char, string, double

Formatangabe: %[flags][width][.precision][length]specifier

Details auf <https://cplusplus.com/reference/cstdio>

### Abkürzungen

a++	a = a + 1
a += 2	a = a + 2
q? a1 : a2	if q a1 else a2

