

Variablen

Typ	Bereich	Beispielanwendungen
char	-127 bis 127 (auch ASCII)	Ganzzahlen und ASCII
short int	-32.768 bis 32.767	Ganzzahlen
int	-2.147.483.648 bis 2.147.483.647	Ganzzahlen
long	-2.147.483.648 bis 2.147.483.647	Ganzzahlen
float	1.2E-38 bis 3.4E38	Fließkommazahlen
double	2.2E-308 bis 1.8E308	Fließkommazahlen

Include Headers

#include <iostream>	
#include <conio.h>	getch(), getch(), kbhit()
#include <stdlib.h>	Memory, Random...
#include <math.h>	Mathe Funktionen (Pow...)

Namespaces

```
using namespace std;
```

Kommentare

```
// Einzeiliger Kommentar
/* Mehrzeiliger
Kommentar */
```

Eingabe / Ausgabe

```
cin >> var;

cout << "TEXT" << var << endl;
```

Escape Sequenzen

\n	Zeilenumbruch
\t	horizontaler Tabulator
\v	vertikaler Tabulator
\\, \", \'	;', ", \'

Ausgabe-Manipulatoren

setw(int)	Breite der Ausgabe
setfill('char')	Füllzeichen
left, right	Rechts- oder Linksbündig
scientific	wissenschaftliche Darstellung
fixed	standard Kommazahl
hex, oct, dec	Umwandlung in Zahlensystem
showpos	Zeigen des Vorzeichens
boolalpha	boolescher Wert als Text

Conditionals

A == B	Gleich
A != B	Ungleich
A < B	Kleiner als
A > B	Größer als
A <= B	Kleiner gleich
A >= B	Größer gleich
A && B	Und
A B	Oder

Kontrollstrukturen

Funktionen

Regeln:

- Die Funktion muss bevor sie aufgerufen wird bereits deklariert sein (Aber nicht definiert).
- Der Funktionsname darf nicht mit einer anderen Variable übereinstimmen.

Der **Rückgabewert** ist eine Variable die zum Funktionsaufruf mithilfe `return` `var`; zurückgegeben wird.

Als "**call by value**" Funktionsargument bezeichnet man eine Variable die aus dem Funktionsaufruf kopiert wird. In der eigentlichen Funktion ist das eine komplett neue Variable

Aufruf: `kgv(x, y)`;

Deklaration: `void Funktionsname (int ggt, int kgv)`

Als "**call by reference**" Funktionsargument bezeichnet man eine Variable die dem Funktionsaufruf sowie der eigentlichen Funktion bekannt ist. Diese werden in der Funktionsdeklaration mit einem **&** vor dem Variablennamen deklariert.

Aufruf: `kgv(x, y)`;

Deklaration: `void Funktionsname (int &, int &)`

Funktionsprototyp zum deklarieren von Funktionen

Rückgabewert Funktionsname

(Funktionsargumente);

```
int kgv(int z1, int z2, int &, int &);
```

Funktionsdefinition

Rückgabewert Funktionsname

(Funktionsargumente);

```
int kgv(int z1, int z2, int &, int &)
```

```
{
// Code
return var;
}
```

if - Abfrage

```
if (Kondition) {  
  //Code  
}  
else if (Andere_Kondition) {  
  // Mach was anderes  
}  
else {  
  // Mach was standartmäßig  
}
```

Switch - Case

```
switch (var) {  
case val1:  
  // Code  
break;  
[default:  
  // Code  
break;]
```

While - Schleife

```
while (Solange) {  
  // Code  
}
```

For - Abfrage

```
for (i=0; i<n; i++) {  
  // Code  
}
```

do while - Abfrage

```
do {  
  // Code  
} while (Solange)
```

