

Übersicht Datentypen und Rechenoperationen

- ganze Zahl: **int**
- reelle Zahl: **double**
- einzelnes Zeichen: **char**
- Zeichenkette: **char** ... [] oder alternativ string
- Wahrheitswert: **bool**

```
#include<iostream>
using namespace std;

int main()
{
    //Ganze Zahlen sind vom Typ 'int':
    int ganzeZahl = 5, ganzeZahl2 = -1;
    //Allgemeine reelle Zahlen sind vom Typ 'double':
    double reelleZahl = -0.7, reelleZahl2 = 1.0;

    //Einzelne Zeichen sind vom Typ 'char':
    char zeichen = '!';
    //Ganze Zeichenketten (z.B. ein Satz) sind vom Typ 'char ... []':
    char zeichenkette[] = "Hallo Welt!";

    //Ein Wahrheitswert (true oder false) ist vom Typ 'bool':
    bool stimmts = false, wahrheit = true;

    //Ausgabe einer Variable mit 'cout':
    cout << zeichenkette << endl;
    cout << zeichen << endl;
    cout << reelleZahl << endl;
    cout << "Ein Text." << endl;

    //Addieren zweier Variablen:
    reelleZahl = reelleZahl + reelleZahl2; //'reelleZahl' hat nun Wert 0.3

    //Subtrahieren zweier Variablen:
    reelleZahl = reelleZahl - reelleZahl2; //'reelleZahl' hat nun Wert -0.7

    //Dividieren zweier Variablen:
    reelleZahl = reelleZahl / reelleZahl2; //'reelleZahl' hat nun Wert 1.0

    //Multiplizieren zweier Variablen:
    reelleZahl = reelleZahl2 * reelleZahl2; //'reelleZahl' hat nun Wert 1.0

    return 0;
}
```

Variablen ausgeben und einlesen

(bei eingebundenem `std`-Header)

- Ausgabe: `cout << ...`
- Einlesen: `cin >> ...`
- In die nächste Zeile gehen: `endl`

```
#include<iostream>
using namespace std;

int main()
{
    //Ganze Zahl initialisiert:
    int ganzeZahl = 5;
    //Ganze Zahl nicht initialisiert:
    int ganzeZahl2;
    //Zeichen nicht initialisiert:
    char zeichen;

    //Ausgabe der ganzen Zahl mit 'cout':
    cout << ganzeZahl << endl;

    //Einlesen (und Ueberschreiben) der ganzen Zahl mit 'cin':
    cin >> ganzeZahl;

    //Einlesen des Zeichens mit 'cin':
    cin >> zeichen;

    //Ausgabe des Zeichens mit 'cout':
    cout << zeichen << endl;

    return 0;
}
```