

Konstanten sind Werte, die sich nicht ändern.

### 1. Präprozessor-Konstanten

Wenn man einen Wert im Programm häufig benötigt, kann man ihn per Präprozessor-Anweisung als Konstante definieren. In der Regel macht man das am Anfang des Programms nach den Include-Anweisungen.

Beispiel:

```
#define PI 3.1415927
```

Nach dieser Anweisung kann man `PI` schreiben, wenn man 3,1415927 meint. Der Präprozessor ersetzt die Konstante durch den Zahlenwert.

Viele solcher Konstanten, zum Beispiel  $\pi$ , stehen in Headerdateien. Werfen Sie einen Blick in `cmath` und in `climits`!

### 2. Konstante Variablen

Man kann eine Variable konstant machen, indem man bei der Initialisierung das Schlüsselwort `const` davor schreibt. Dann achtet der Compiler darauf, dass ihr Wert nicht mehr verändert wird

Beispiel: In einem Programm wird der Rabatt für Kunden am Anfang eingegeben und darf dann nicht mehr verändert werden.

```
int main()
{
    int rabattEingabe;

    cout << "Rabatt in Prozent eingeben: ";
    cin >> rabattEingabe;

    const int RABATT = rabattEingabe;

    // ... eigentliches Programm ...
}
```

Im Gegensatz zu Präprozessor-Konstanten verbrauchen konstante Variablen Arbeitsspeicher.

### 3. Konstante Parameter

Konstante Variablen kann man auch als Parameter einsetzen.

Beispiel:

```
void steuer(double netto, const double steuersatz)
{
    // steuersatz = 7; Nicht möglich wegen const!

    cout << netto*steuersatz/100 << endl;
}
```

Hier verhindert der Programmierer, dass er versehentlich in der Funktion den Steuersatz verändert.

### Konventionen für die Schreibweise von Variablen und Konstanten

- Es ist üblich, dass man die Namen von **Präprozessor-Konstanten** **komplett in GROSSBUCHSTABEN** schreibt. Halten Sie sich unbedingt daran!
- Konstante Variablen werden von vielen Softwareentwicklern ebenfalls komplett groß geschrieben.
- **Normale Variablen und konstante Parameter** dagegen schreibt man **immer klein**.