

# Python Grundlagen

## 1. Grundelemente einer Programmiersprache

Wie jede Programmiersprache folgt auch Python bestimmten Regeln und Mustern, damit wir sinnvoll mit dem Computer arbeiten können. Diese Elemente werden wir uns genauer anschauen.

### Grundelemente einer Programmiersprache

- **Variablen und Zuweisungen**  
Der Computer muss sich Sachen merken können.
- **Ein- und Ausgabe**  
Etwas muss in den Computer rein, etwas kommt raus.
- **Datentypen**  
Der Computer hat Regeln, was er mit welchen Variablen machen darf (mit Text kann er nicht dasselbe machen wie mit Zahlen).
- **Operatoren**  
Der Computer muss wissen, was er mit den Variablen machen kann (rechnen, vergleichen,...).
- **Kontrollstrukturen**  
Der Computer kann ein Programm (oder Teile davon) wiederholen oder je nach Resultat sich für einen von mehreren Wegen entscheiden.
- **Funktionen**  
Wiederverwendbare Unterprogramme, die eine bestimmte Aufgabe erfüllen.

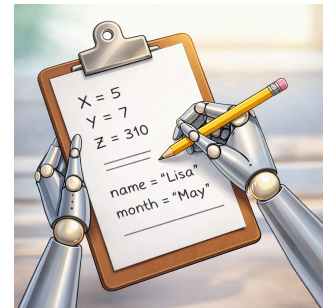
Diese Elemente sind eng miteinander verbunden und manchmal ist es schwierig das eine zu verstehen ohne die anderen.

## 2. Variablen und Zuweisungen

Damit der Computer ein Programm ausführen kann, muss er sich Sachen merken können. Wie wenn wir Menschen beim Einkaufen eine Einkaufsliste brauchen. Die gespeicherten Werte nennt man **Variablen**.

### Variablen und Zuweisungen

Eine **Variable** benutzt man, um einen Wert zu speichern (wie ein Platzhalter).  
Sie wird mit einer **Zuweisung** (=) erstellt, wobei links der **Name** und rechts der **Wert** der Variable steht.



### Zuweisungen

```
1 x = 5
2 monat = "Januar"
3 summe = 1 + 256 + 42
```

Die **Werte der Variablen** können Zahlen, Text oder auch Berechnungen sein.

In der ersten Zeile erhält die Variable x den Wert 5. In der zweiten Zeile bekommt die Variable monat den Wert "Januar". Das ist ein Text, der immer in Anführungszeichen stehen muss, damit der Computer ihn als solchen versteht.

Den **Namen der Variablen** darf man (fast) frei wählen:

- erlaubt sind Buchstaben (klein und gross), Zahlen und der Unterstrich \_
- der Name darf nicht mit einer Zahl beginnen

Im Verlauf des Programms können sich die Werte von Variablen ändern (man kann sie also überschreiben).

### Variablen überschreiben

```
1 x = 5
2 x = 7 # der Wert von x ist neu 7
```

Mit dem Symbol # können wir Code in Python **kommentieren**. Alles, was nach dem # auf derselben Zeile folgt, wird vom Computer ignoriert. **Kommentare** helfen uns, den Code lesbarer (verständlicher) zu machen.

## Aufgabe

Gegeben ist eine Liste von möglichen Variablennamen. Welche sind laut den Regeln für Python gültig?

Summe\_1 = 433

summe\_1 = 123

summe1 = 63

\*myname = "Glurak"

\_my\_name = "Pikachu"

my name = "Flegmon"

8Bit = 256

L337Code = 42

Byte8Num3r = 16

Wie kann man schnell überprüfen, ob ein Variablenname okay ist oder nicht?

## 3. Ein- und Ausgabe

Damit wir mit dem Computer arbeiten können, braucht eine Programmiersprache zwei Dinge: Wir müssen **Eingaben** (**Input**) machen können, und der Computer muss diese verarbeiten und uns danach **Resultate** (**Output**) zurückgeben.

In Python gibt es zwei Funktionen (Befehle) dafür.

`print()` Damit können Sie beliebige Sachen in die Kommandozeile (Ausgabefenster) ausdrucken lassen.

`input()` Damit fragt der Computer nach einem Input vom Benutzer.

```
1 erste_zahl = input("Erste Zahl:")
2 zweite_zahl = input("Zweite Zahl:")
3 print(erste_zahl + zweite_zahl)
```



## 4. Datentypen

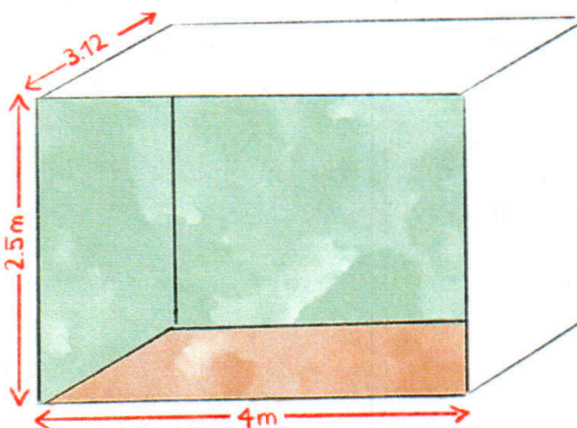
Wenn der Computer mit Variablen zu tun hat, muss er wissen, zu was für einem Typ sie gehören. Wenn er weiss, dass es sich um einen Text handelt, dann weiss er auch sofort, dass nicht alle Operatoren funktionieren werden.

## 5. Operatoren

Die Operatoren bestimmen, was man mit den Variablen machen kann. Sie verknüpfen Variablen oder Werte und liefern ein Resultat.

## Aufgabe

1. Sie haben folgendes ausgemessenes Zimmer, bei dem an den Wänden langsam die Farbe abblättert.



Erstellen Sie in WebTigerPython passende Variablen und berechnen Sie die Gesamtfläche, die neu gestrichen werden muss.

Sie können mit folgendem Code starten und dann weiter ausbauen:

```
1 laenge = 4
2 hoehe = 2.5
3 breite = ...
4 wand_links = hoehe * breite
5 ...
6 gesamtflaeche = ...
7 print(gesamtflaeche)
```



## 6. Kontrollstrukturen

Mit Verzweigungen oder Schleifen kann der Programmablauf gesteuert werden.

## 7. Funktionen

Wir können in Funktionen einen Ablauf speichern und diese dann mit verschiedenen Parametern aufrufen.

Auch hier ist das Einrücken wichtig.