Variablen - eine Typfrage

Socken, T-Shirts usw. packen Sie wahrscheinlich nicht in die gleiche Schublade. So ist das bei den Python Variablen auch. Es gibt verschiedene Typen von Werten, wie zum Beispiel Zahlen oder Texte. Für diese verschiedenen Typen gibt es daher passende Variablen. Normalerweise braucht man den Typ der Variablen in Python nicht angeben, er wird automatisch erkannt.

String: Im folgenden Beispiel wird ein Text in eine Variable gepackt:

name = ***"Max Mustermann"***

Beim Coding redet man eher von Zeichenketten oder *Strings* und nicht von Texten. Sie werden an den Anführungszeichen automatisch von Python erkannt.

Integer: Ganze Zahlen nennt man in der Softwareentwicklung dabei *Integer*. Sie legen diese Variable in Python wie folgt an:

a = **8**

Float: In diesem Beispiel wird eine Variable für die Zahl Pi mit ihrem Wert belegt. Python erkennt, dass es sich hier um eine Kommazahl handelt und nutzt eine Variable vom Typ *Float* (Fließkommazahlen):

pi = **3.14**

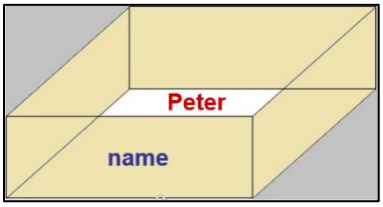
Bool: Etwas Besonderes sind die Wahrheitswerte des Typen *Bool*. Diese können nur die Werte *True* oder *False*, also *Wahr* oder *Falsch* aufnehmen. Man trifft mit den Wahrheitswerten Entscheidungen, um den Programmablauf zu steuern:

isEnabled = **True**

1. Variable als Zeichenkette

Werte wie **Zeichenketten**, **Zahlen** oder **Wahrheitswerte** werden in **Variablen** gespeichert, um diese Werte für spätere Operationen verfügbar zu halten. Sie sind also ein Speicher für Werte, die man im Laufe des Programms wieder benötigt.

Möchte man eine Zeichenkette in einer Variablen speichern, dann muss dafür ein Speicherplatz geschaffen werden. Diesen Speicherplatz kann man sich wie eine Schublade vorstellen. Außerdem braucht man einen Namen für die Variable, damit man auf den Inhalt später wieder zugreifen kann. Bildlich gesprochen bekommt die Schublade ein Schild, damit man weiß, was in der Schublade ist.



Beispiel für die Verwendung einer Variablen:

```
1  name = "Peter"    # Deklaration und Initialisierung
2  print(name)       # Ausgabe der Variablen "name" auf dem Bildschirm
```

Zeile 1: Eine Variable mit dem Namen *name* wird erzeugt und es wird ihr der Wert „Peter“ zugewiesen.
Zeile 2: Mit der Funktion print() soll der Inhalt der Variablen name auf dem Bildschirm ausgegeben werden: print(name).

Ausgabe:

```
Kommandozeile
  Peter
```

Dieses kleine Python-Programm enthält zwei unterschiedliche Anweisungen. Zu beachten ist,

- dass eine Zeichenkette immer zwischen doppelten (oder einfachen) Anführungsstrichen geschrieben wird,
- dass mit dem Zeichen „***#***“ ein Kommentar zur Beschreibung der einzelnen Teile des Programms eingeleitet wird. Kommentare werden nicht als Programmcode interpretiert.

Namensregeln für Variablen:

- Am Anfang sollte ein Kleinbuchstabe stehen. Ziffern sind **nicht** zulässig.
- Danach dürfen ***Groß***- und ***Kleinbuchstaben*** sowie ***Ziffern*** und ***Unterstriche*** verwendet werden.
- Umlaute wie ä, ö, ü das ß, Leerzeichen und weitere Sonderzeichen sind nicht erlaubt.
- Schlüsselwörter aus Programmiersprachen, z.B. while, if, else dürfen nicht verwendet werden.
- Der Name sollte etwas über den Inhalt der Variablen aussagen. Dabei hat sich in Python als Standard ergeben, dass bei zusammengesetzten Wörtern die einzelnen Wörter durch einen Unterstrich getrennt werden, also z.B. alter_in_jahren oder gewicht_in_kg.

Ausgabe mehrerer Variablen in einer Bildschirmausgabe

Beispiel:

Variablen_zusammengesetzt.py

```
1  subjekt = "Ich"
2  praedikat = "spiele"
3  objekt = "Fußball"
4  print(subjekt, praedikat, objekt, "auf der Wiese")
```

Zeilen 1-3: Es werden drei Variablen *subjekt*, *praedikat* und *objekt* erzeugt, denen jeweils ein Wert zugewiesen wird.

Zeile 4: Die drei Variablen sowie der Text „auf der Wiese“ werden mit Hilfe der Funktion print() ausgegeben, indem die Variablen und das Textstück durch Kommata getrennt hintereinander aufgelistet werden. Python listet dann diese Variablen und das Textelement bei der Ausgabe hintereinander auf.

Ausgabe:

```
Kommandozeile
  Ich spiele Fußball auf der Wiese
```

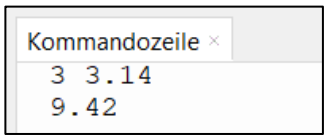
2. Variable mit Zahlen

Möchte man Berechnungen durchführen, benötigt man Zahlen, die wiederum in Variablen gespeichert werden können. Es können sowohl Ganzzahlen als auch Dezimalzahlen verwendet werden.

Beispiel:

```
1  erste_zahl = 3
2  zweite_zahl = 3.14
3  print(erste_zahl, zweite_zahl)
4  print(erste_zahl * zweite_zahl)
```

Ausgabe:



Programmerläuterung:

1. Zuerst wird die Variable *„erste_zahl“* erzeugt und ihr der Wert 3 zugewiesen. Diese Variable ist eine Ganzzahl.
2. Die zweite Variable *„zweite_zahl“* ist auch eine Zahlvariable, aber eine Dezimalzahl. Die Nachkommastellen werden nach der englischen Schreibweise mit einem Punkt abgetrennt.
3. Dann werden beide Variablen mit einer print-Anweisung in einer Zeile ausgegeben.
4. Am Ende werden beide Zahlen miteinander multipliziert und mit einer print-Anweisung ausgegeben.

3. Variablen mit Wahrheitswerten

Neben der Zeichenkette und Zahlen können in Variablen auch Wahrheitswerte vom Typ boolean gespeichert werden. Es gibt zwei Wahrheitswerte: True und False (deutsch: wahr oder falsch).

Vorsicht: Groß- und Kleineinschreibung ist wichtig: Es muss genau True und False heißen, true oder false wäre falsch!

Beispiel:

```
1  welt_ist_schoen = True
2  frieren_ist_schoen = False
3  print("Ist die Welt schön?", welt_ist_schoen)
4  print("Ist Frieren schön?", frieren_ist_schoen)
```

Ausgabe: