

The worst is when you run out of monospaced fonts and have to use variable-width variables.

4. Variablen

Für einfachere und vielseitigere Programme müssen wir in der Lage sein, einen Integer oder einen String zu speichern. In Python gibt es dafür **benannte Speicherplätze**. Das Speichern in benannten Speicherplätzen ist nicht zu verwechseln mit dem Speichern z.B. eines Textdokuments. Ein Speicherplatz steht nur zur Verfügung, solange das Programm läuft. Danach werden alle Speicherplätze gelöscht. Die Lernziele für dieses Kapitel sind:

- ☐ Sie definieren die Begriffe Variable und Zuweisung.
- ☐ Sie verwenden Variablen, um Python-Programme einfacher und vielseitiger zu machen.
- ☐ Sie wissen, welche Variablennamen in Python erlaubt sind.
- ☐ Sie berücksichtigen die Clean-Code-Regeln im Umgang mit Variablen.

4.1 Quadrate mit unterschiedlicher Länge

Betrachten wir das Programm in Listing 6 (Quadrat, Seitenlänge 100). Wenn wir nun ein Quadrat mit der Seitenlänge 275 zeichnen wollen, müssen wir das Programm an **vier** Stellen anpassen. Listing 7 zeigt das angepasste Programm. Optimal wäre es, wenn wir das Programm nur an **einer** Stelle anpassen müssten. Dies kann durch die Verwendung einer **Variablen** erreicht werden.

```

1  import turtle
2
3  turtle.fd(100)
4  turtle.lt(90)
5  turtle.fd(100)
6  turtle.lt(90)
7  turtle.fd(100)
8  turtle.lt(90)
9  turtle.fd(100)
10 turtle.lt(90)
11 turtle.done()
12

```

Listing 6: Quadrat (Länge 100).

```

1  import turtle
2
3  turtle.fd(275)
4  turtle.lt(90)
5  turtle.fd(275)
6  turtle.lt(90)
7  turtle.fd(275)
8  turtle.lt(90)
9  turtle.fd(275)
10 turtle.lt(90)
11 turtle.done()
12

```

Listing 7: Quadrat (Länge 275).

Wir speichern den Integer für die Seitenlänge des Quadrats in einer Variablen ab und verwenden diese Variable dann als Argument für die vier Funktionsaufrufe. Listing 8 zeigt das verbesserte

Programm. Wenn das Programm ausgeführt wird, dann wird in Zeile **drei** der Integer 100 in der Variablen mit dem Namen `a` gespeichert. In den Zeilen vier, sechs, acht und zehn wird die Variable `a` dann durch den gespeicherten Integer, das heisst 100 ersetzt.

```

1  import turtle
2
3  a = 100
4  turtle.fd(a)
5  turtle.lt(90)
6  turtle.fd(a)
7  turtle.lt(90)
8  turtle.fd(a)
9  turtle.lt(90)
10 turtle.fd(a)
11 turtle.lt(90)
12 turtle.done()
13

```

Listing 8: Die Variable `a` speichert den Integer 100 (`quadrat_var.py`).

```

1  import turtle
2
3  a = 275
4  turtle.fd(a)
5  turtle.lt(90)
6  turtle.fd(a)
7  turtle.lt(90)
8  turtle.fd(a)
9  turtle.lt(90)
10 turtle.fd(a)
11 turtle.lt(90)
12 turtle.done()
13

```

Listing 9: Zeile drei wurde angepasst - nun wird der Integer 275 gespeichert.

Jetzt können wir das Programm leicht ändern. Wenn wir ein Quadrat mit der Seitenlänge 275 haben wollen, dann müssen wir nur Zeile drei ändern. Listing 9 zeigt das angepasste Programm.

4.2 Was ist eine Variable?

Wir können uns eine Variable als einen Behälter mit einem Etikett vorstellen, in dem ein Wert gespeichert wird. In Abbildung 6 sind zwei Variablen als Behälter dargestellt.

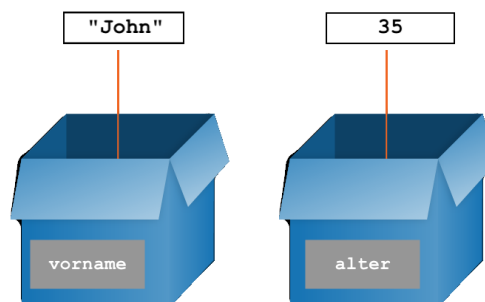


Abbildung 6: Zwei Variablen, dargestellt als Behälter mit Etiketten.¹

Jeder Behälter in Abbildung 6 hat ein Etikett und einen gespeicherten Wert. Der Behälter mit dem Etikett `vorname` entspricht einer Variablen mit dem Namen `vorname` und dem gespeicherten Wert `"John"` (ein String). Die andere Variable heisst `alter` und speichert den Wert `35` (einen Integer).

Definition 9 — Variable. Eine Variable ist ein **Speicherplatz mit einem Namen**. In der Variablen kann ein **beliebiger Wert** gespeichert werden. Dies geschieht in Python mit einer **Zuweisung**. Wir können den gespeicherten Wert auch wieder auslesen. Dazu notieren wir den Namen der Variablen. Den Namen einer Variablen können wir selbst wählen.

¹Bildquelle: Angepasst von <http://www.pyworld.in/Python/pythonVariables.html?i=1>.

4.3 Was ist ein Wert?

Ein Wert stellt einen grundlegenden Baustein dar, mit dem ein Programm operiert. Wir haben Werte bereits als Argumente für Funktionsaufrufe verwendet, und es ist auch möglich, Werte in Variablen abzuspeichern. Werte können in verschiedene **Typen** unterteilt werden. Abbildung 7 zeigt eine Übersicht einiger Typen mit Beispielen.

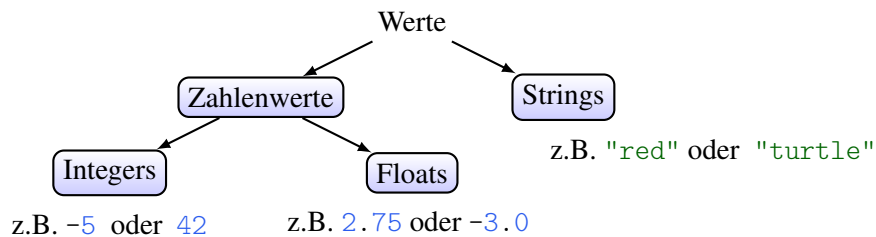
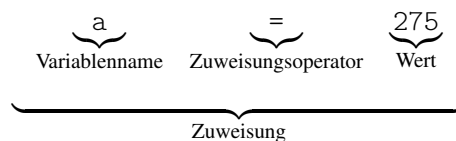


Abbildung 7: Diese Darstellung ist nicht abschliessend. Es gibt noch mehr Typen.

Einen Wert können wir **direkt darstellen**, in dem wir den Wert in das Programm eintippen.

4.4 Was ist eine Zuweisung?

Für uns ist `a = 275` ein neuer Befehl. Der Befehl gehört zur Kategorie „Zuweisung“ (engl. assignment). `a = 275` ist also ein Beispiel für eine **Zuweisung**.



Wir können die Zuweisung auch als „a soll 275 speichern“ lesen. Eine Zuweisung erkennen wir am **Zuweisungsoperator**² (engl. assignment operator). Da in Python das Gleichheitszeichen (=) verwendet wird, um Werte in einer Variablen zu speichern (und nicht wie in der Mathematik im Sinne einer Gleichung), verwenden wir für dieses Zeichen den Fachbegriff Zuweisungsoperator.

Wichtig! Bei einer **Zuweisung** muss auf der **linken** Seite des Zuweisungsoperators eine Variable stehen. ■

Clean Code — **Leerzeichen 2.** Links **und** rechts eines **Zuweisungsoperators** fügen wir je ein **Leerzeichen** ein.

Die Leerzeichen rund um den Zuweisungsoperator sind nicht obligatorisch, gehören aber zu einem guten Programmierstil. In den Listings werden sie daher auch mit dargestellt.

■ **Beispiel 8** Wir können natürlich auch einen String in einer Variablen speichern.



²Sie kennen den Begriff bereits aus der Mathematik. Bei der Addition von zwei Zahlen, z.B. $2 + 3$, stellt das „Pluszeichen“ einen mathematischen Operator dar. Oft wird in der Mathematik auch der Begriff Rechenzeichen verwendet.

4.5 Welche Variablennamen sind erlaubt?

Der Name einer Variablen können wir nahezu frei wählen. Wir können Gross- und Kleinbuchstaben, Ziffern (0 – 9) und den Unterstrich³ (engl. underscore) verwenden. Es gibt nur sehr wenige Einschränkungen:

- Der Name darf **nicht** mit einer **Zahl beginnen**.
- Der Name darf **keinem** Python-Schlüsselwort (engl. keyword)⁴ entsprechen.
- Der Name darf **keine** Leerzeichen enthalten.

Die Namen `1farbe`, `import` und `meine_Farbe` sind daher **nicht** erlaubt. `farbe1`, `warenimport` und `meineFarbe` sind dagegen erlaubt.

Wichtig! Python **unterscheidet** zwischen Gross- und Kleinbuchstaben. Wir sagen, die Programmiersprache ist **case sensitive**. Die korrekte Verwendung der Gross- und Kleinbuchstaben gilt auch für Schlüsselwörter (wie `import`). ■


■ **Beispiel 9** Wenn Sie eine Variable `tmp`⁵ verwenden und an anderer Stelle `TMP` schreiben, dann sind das für Python **zwei unterschiedliche** Variablen. ■

4.6 Wie wählen wir Variablennamen?


Eigentlich können wir recht frei einen Namen wählen, zum Beispiel `Meine_LiebliNgs_figur` oder `birne`. Jedoch **sollten** wir einen Variablennamen so wählen, dass wir durch den Namen erkennen können, was für ein Wert darin gespeichert ist. Dies entspricht der **Clean-Code-Idee**.

 **Clean Code** — **Sinnvolle Variablennamen**. Wir wählen **Variablennamen** so, dass wir möglichst direkt verstehen, was darin gespeichert wird.

■ **Beispiel 10** In Listing 8 haben wir den Namen `a` für die Variable gewählt. Der Name bezeichnet die Seitenlänge des Quadrats. Dies kann als ein sinnvoller Name betrachtet werden, da die Seitenlänge eines Quadrats typischerweise in der Geometrie mit *a* bezeichnet wird. Wir können es auch noch expliziter machen und den Namen `seitenlänge` verwenden. ■

 **Clean Code** — **Snake Case**. In Python ist es üblich, für Variablennamen nur **Kleinbuchstaben** und **Ziffern** zu verwenden. Besteht der Name aus mehreren Wörtern, dann „trennen“ wir diese Wörter durch einen **Unterstrich** (`_`). Diese Schreibweise wird Snake Case genannt.

■ **Beispiel 11** Die Variablennamen `meine_farbe`, `zahl_1` oder `temperatur_in_celsius` sind in Snake Case notiert. ■

 **H** Andere Programmiersprachen empfehlen andere Schreibweisen für Variablennamen. Abbildung 8 zeigt auf humorvolle Art die gebräuchlichsten Schreibweisen.

³Auch Bodenstrich oder Tiefstrich genannt.

⁴Auch reserviertes Wort genannt. Dies sind Wörter, die in Python eine bestimmte Bedeutung haben. Zum Beispiel ist `import` ein Schlüsselwort. Schlüsselwörter werden in einer IDE meist farblich hervorgehoben.

⁵Abkürzung für **temporary**.

4.7 Wie benutzen wir den Inhalt einer Variablen?

Wir können jederzeit den Inhalt einer Variablen benutzen, in dem wir den entsprechenden **Variablennamen** notieren. Python ersetzt dann während der Ausführung den Variablennamen durch den gespeicherten Inhalt. Wir sagen, die Variable wird **ausgelesen**.

■ **Beispiel 12** Im Programm aus Listing 8 wird der Inhalt der Variablen `a` an vier Stellen verwendet. In Zeile vier wird die Variable als Argument für den Funktionsaufruf verwendet. Während der Ausführung wird die Variable durch den gespeicherten Integer (100) ersetzt. Dies wiederholt sich in den Zeilen sechs, acht und zehn. ■

Wichtig! Das Auslesen einer Variablen löscht den gespeicherten Inhalt der Variablen **nicht**. ■

H Wenn die entsprechende Variable beim Auslesen nicht existiert (z. B. weil vorher kein Wert darin gespeichert wurde), dann wird eine **Fehlermeldung** erzeugt. Typischerweise wird ein `NameError` erzeugt.

4.8 Wo ist der Inhalt einer Variablen gespeichert?

Eine Variable ist ein Speicherplatz mit einem Namen. Das Computerbauteil, in dem der Speicherplatz für eine Variable reserviert ist, wird **Arbeitsspeicher** genannt. Häufig wird diese Komponente auch als `RAM`⁶ bezeichnet, da der Arbeitsspeicher in Computern fast ausschliesslich aus diesem Speichertyp besteht. Der Arbeitsspeicher von Standard-Laptops ist typischerweise zwischen 8 GB und 32 GB gross. Nur ein **Teil** davon wird für die Speicherung von Variablen verwendet.

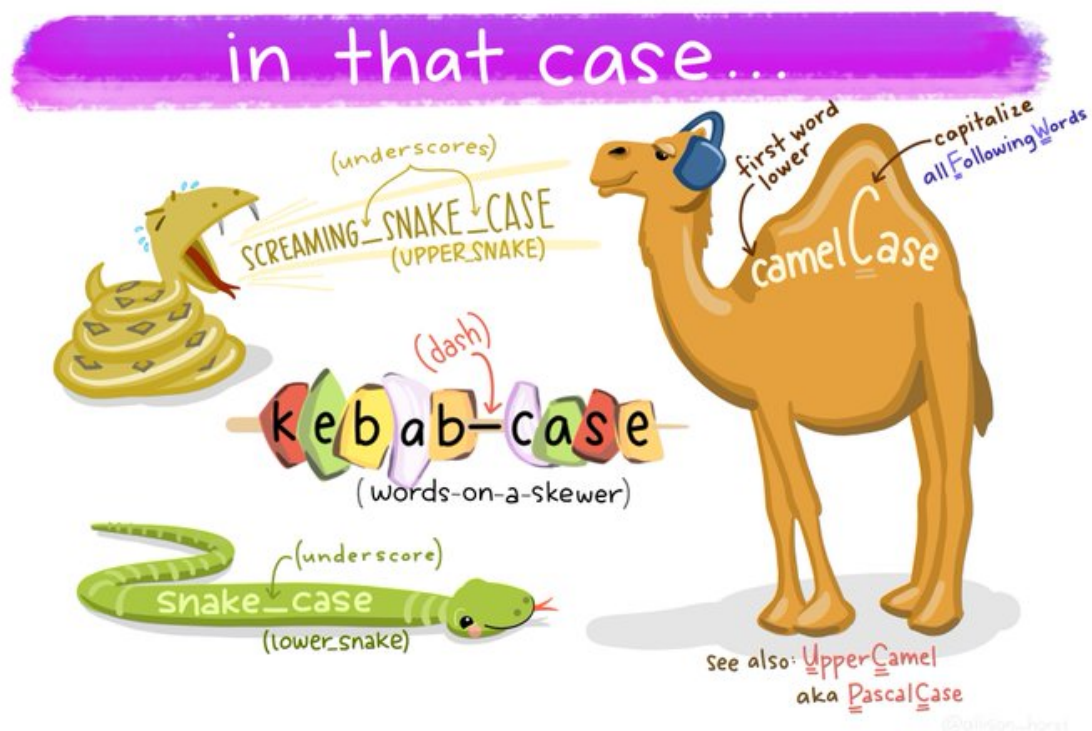


Abbildung 8: In Python ist „Snake Case“ üblich.⁷

⁶Random Access Memory (dt. Direktzugriffsspeicher)

⁷Bildquelle: <https://allisonhorst.com/everything-else>