Variablen und Zuweisungen in Python Lösungen

Lernziele

Nach diesem Arbeitsblatt kannst du:

- erklären, was eine Variable in der Informatik ist,
- einfache Zuweisungen in Python lesen und schreiben,
- gültige Variablennamen von ungültigen unterscheiden,
- mehrere Variablen gleichzeitig zuweisen,
- zusammengesetzte Zuweisungsoperatoren wie `+=`, `-=` und `*=` verwenden,
- kurze Python-Programme ohne Ausführung im Kopf nachvollziehen.

1 Was ist eine Variable?

Ein Computer kann während der Ausführung eines Programms viele verschiedene Werte speichern und verändern. Damit wir auf einen solchen Wert später zugreifen können, geben wir ihm einen **Namen**.

Einen solchen Namen nennen wir eine **Variable**.

Eine einfache Vorstellung ist die einer beschrifteten Schachtel:

alter
15

name
"Anna"

preis
12.50

Die Variable `alter` enthält hier den Wert `15`. Die Variable `name` enthält den Text `"Anna"` und `preis` enthält den Wert `12.50`.

Merke

Eine Variable ist ein **Name**, unter dem ein Wert gespeichert bzw. über den auf einen Wert zugegriffen werden kann.
Der Wert einer Variable kann sich während eines Programms ändern.

Der Wert einer Variable kann sich ändern

Zum Beispiel:

```
punkte = 10
punkte = 15
```

Nach der ersten Zeile hat `punkte` den Wert 10.

Nach der zweiten Zeile hat `punkte` den Wert 15.

Der bisherige Wert wurde also durch den neuen Wert ersetzt.

2 Einfache Zuweisungen in Python

In Python wird eine Zuweisung mit dem Gleichheitszeichen `=` geschrieben.

Zum Beispiel:

```
alter = 15
name = "Anna"
preis = 12.50
```

Das bedeutet:

- Der Wert `15` wird der Variable `alter` zugewiesen.
- Der Text `"Anna"` wird der Variable `name` zugewiesen.
- Der Wert `12.50` wird der Variable `preis` zugewiesen.

Achtung

Das Zeichen `=` bedeutet bei einer Zuweisung nicht "ist gleich" im mathematischen Sinn.

```
alter = 15
```

bedeutet:

Speichere den Wert 15 unter dem Namen `alter`.

Zuweisungen mit Berechnungen

Auf der rechten Seite einer Zuweisung darf auch eine Berechnung stehen:

```
a = 10
b = 3
c = a + b
```

Nach diesen drei Zeilen gilt:

$$a = 10, \qquad b = 3, \qquad c = 13$$

Python berechnet zuerst den Ausdruck auf der rechten Seite. Anschliessend wird das Ergebnis der Variablen auf der linken Seite zugewiesen.

Merke

Bei

`variable = Ausdruck`

wird zuerst der Ausdruck rechts ausgewertet. Danach wird sein Wert unter dem Namen links gespeichert.

3 Regeln für Variablennamen

Exkurs: Regeln für Variablennamen

Für Variablennamen gelten in Python einige Regeln.

1. Ein Variablenname darf Buchstaben, Ziffern und den Unterstrich `_` enthalten.
2. Ein Variablenname darf **nicht mit einer Ziffer beginnen**.
3. Leerzeichen sind nicht erlaubt.
4. Gross- und Kleinschreibung wird unterschieden.
5. Bestimmte reservierte Wörter dürfen nicht als Variablennamen verwendet werden. Dazu gehören beispielsweise `if`, `for`, `while`, `class` und `def`.

Beispiele

`Name`	gültig?	Grund
`alter`	Ja	
`mein_alter`	Ja	Unterstrich ist erlaubt
`note1`	Ja	Ziffer am Ende ist erlaubt
`1note`	Nein	beginnt mit einer Ziffer
`mein alter`	Nein	Leerzeichen sind nicht erlaubt
`mein-alter`	Nein	Bindestrich ist nicht erlaubt
`for`	Nein	reserviertes Python-Schlüsselwort
`Alter`	Ja	Grossbuchstaben sind erlaubt

Achtung

`alter`, `Alter` und `ALTER` sind in Python **drei verschiedene Namen**.

Gute Variablennamen

Variablennamen sollten möglichst verständlich sein.

Zum Beispiel:

```
alter = 15
anzahl_punkte = 87
maximale_temperatur = 35
```

Ein Name wie `x` kann in einer kurzen Rechnung durchaus sinnvoll sein. In längeren Programmen sind aussagekräftige Namen jedoch meistens besser.

4 Mehrfachzuweisungen

Python erlaubt es, mehrere Variablen in einer einzigen Zeile zuzuweisen.

Zum Beispiel:

```
a, b = 10, 20
```

Danach gilt:

$$a = 10, \qquad b = 20$$

Man kann auch drei oder mehr Variablen gleichzeitig zuweisen:

```
x, y, z = 1, 2, 3
```

Das entspricht:

```
x = 1
y = 2
z = 3
```

Zwei Variablen vertauschen

Eine interessante Anwendung der Mehrfachzuweisung ist das Vertauschen zweier Werte:

```
a = 5
b = 8

a, b = b, a
```

Nach der letzten Zeile gilt:

$$a = 8, \qquad b = 5$$

Python kann also die beiden Werte direkt vertauschen.

5 Zusammengesetzte Zuweisungsoperatoren

Python bietet sogenannte **zusammengesetzte Zuweisungsoperatoren** (englisch: *compound assignment operators*).

Sie verbinden eine Rechenoperation mit einer Zuweisung.

Zum Beispiel:

```
punkte = 10
punkte += 5
```

Die zweite Zeile bedeutet dasselbe wie:

```
punkte = punkte + 5
```

Danach hat `punkte` den Wert 15.

Die wichtigsten Operatoren

Operator	Beispiel	entspricht
`+=`	`x += 3`	`x = x + 3`
`-=`	`x -= 3`	`x = x - 3`
`*=`	`x *= 3`	`x = x * 3`
`/=`	`x /= 3`	`x = x / 3`
`//=`	`x //= 3`	`x = x // 3`
`%=`	`x %= 3`	`x = x % 3`
`**=`	`x **= 3`	`x = x ** 3`

Merke

Bei einem zusammengesetzten Zuweisungsoperator wird der bisherige Wert der Variable verwendet, verändert und anschliessend wieder derselben Variable zugewiesen.

Beispiel

```
x = 4
x *= 3
x += 2
```

Schritt für Schritt:

$$4 \longrightarrow 12 \longrightarrow 14$$

Also gilt am Ende:

$$x = 14$$

6 Übungen

Übung 1 – Einfache Zuweisungen

Bestimme jeweils den Wert der Variable, die mit `print(...)` ausgegeben wird.

Führe den Code nicht aus!

(a)

```
x = 7
x = 12
print(x)
```

Antwort: 12

(b)

```
a = 4
b = 6
c = a + b
print(c)
```

Antwort: 10

(c)

```
x = 10
y = x + 5
x = 3
print(y)
```

Antwort: 15

(d)

```
a = 8
b = a
a = 20
print(b)
```

Antwort: 8

Übung 2 – Mehrfachzuweisungen

Bestimme den ausgegebenen Wert.

(a)

```
a, b = 3, 8
c = a + b
print(c)
```

Antwort: 11

(b)

```
x, y = 10, 4
x, y = y, x
print(x)
```

Antwort: 4

(c)

```
a, b, c = 2, 4, 6
a = b + c
print(a)
```

Antwort: 10

Übung 3 – Zusammengesetzte Zuweisungen

Bestimme jeweils den ausgegebenen Wert.

(a)

```
x = 10
x += 7
print(x)
```

Antwort: 17

(b)

```
x = 20
x -= 6
print(x)
```

Antwort: 14

(c)

```
x = 5
x *= 4
print(x)
```

Antwort: 20

(d)

```
x = 24
x /= 6
print(x)
```

Antwort: 4.0

Übung 4 – Gemischte Aufgaben

Jetzt werden einfache, mehrfache und zusammengesetzte Zuweisungen gemischt.

(a)

```
x = 5
y = 3
x += y
print(x)
```

Antwort: 8

(b)

```
a = 10
b = 2
a *= b
b += 3
print(a)
```

Antwort: 20

(c)

```
x, y = 4, 7
x += y
y = x - y
print(y)
```

Antwort: 4

(d)

```
a = 6
b = 4
a, b = b, a
a *= 2
print(a)
```

Antwort: 8

(e)

```
x = 3
x *= 4
x -= 2
x += 5
print(x)
```

Antwort: 15

(f)

```
a, b = 2, 5
c = a + b
a *= c
print(a)
```

Antwort: 14

Übung 5 – Welche Variablennamen sind korrekt?

Kreuze an, ob der jeweilige Name in Python als Variablenname verwendet werden darf.

Name	korrekt	falsch
`alter`	☒	☐
`2alter`	☐	☒
`mein_alter`	☒	☐
`mein alter`	☐	☒
`alter2`	☒	☐
`mein-alter`	☐	☒
`Alter`	☒	☐
`for`	☐	☒
`_punkte`	☒	☐
`punkte_gesamt`	☒	☐
`class`	☐	☒
`punkte!`	☐	☒

Übung 6 – Fehler finden

In jeder Zeile steckt ein Problem.

Erkläre, warum der Variablenname nicht verwendet werden kann, und schlage einen passenden Namen vor.

(a) `2preis = 15`

Problem: Name darf nicht mit Ziffer beginnen

Verbesserung: `preis2 = 15`

(b) `mein preis = 20`

Problem: Name darf keine Leerzeichen enthalten

Verbesserung: `mein_preis = 20`

(c) `for = 5`

Problem: Name darf kein Python-Schlüsselwort sein.

Verbesserung: Einen anderen Namen wählen

Übung 7 – Den Code im Kopf ausführen

Versuche, die folgenden Programme **ohne Computer** Schritt für Schritt zu verfolgen.

Notiere dir bei Bedarf nach jeder Zeile den aktuellen Wert der Variablen.

(a)

```
punkte = 10
bonus = 5
punkte += bonus
punkte *= 2
print(punkte)
```

Ausgabe: 30

(b)

```
a, b = 3, 7
a += b
b = a - b
a *= 2
print(b)
```

Ausgabe: 3

(c)

```
x = 4
y = 2
x *= y
x += 3
y = x - 1
print(y)
```

Ausgabe: 10

(d)

```
a, b = 5, 2
c = a * b
a += c
b *= 3
print(a)
```

Ausgabe: 15

Tipp für das Lesen von Zuweisungen

Bei

`variable = Ausdruck`

gilt:

1. Berechne den Ausdruck rechts.
2. Speichere das Ergebnis unter dem Namen links.