

1 Einführung

Aufgabe 1.1

Beschreibe den Begriff *Quellcode*.

Ein Quellcode ist ein für Menschen lesbarer Text in einer Programmiersprache.

Aufgabe 1.2

Welches Zeichen verwendet Python für Kommentare im Quellcode?

Python verwendet das Doppelkreuzt # für Kommentare.

Aufgabe 1.3

Welchen Typ von Programm benötigt man, um ein Python-Programm zu schreiben?

Um ein Python-Programm zu schreiben, benötigt man einen *Texteditor*.

Aufgabe 1.4

Welches Programm benötigt man, um Python-Quellcode auszuführen?

Einen Python-Interpreter; das ist ein Programm, das den Python-Quellcode zeilenweise ausführt.

Aufgabe 1.5

Welche Bedeutung hat das Wort *Modul* in der Programmiersprache Python?

Ein (Python-)Modul ist eine Datei, die Python-Quelltext enthält.

Aufgabe 1.6

Zähle die drei wesentlichen Typen von Programmierfehlern auf und gib jeweils ein Beispiel (Python) dazu an.

Syntaxfehler betreffen die Orthographie der Programmiersprache. *Beispiele:* `3 = a` statt `a = 3`; fehlender Doppelpunkt nach *if*, *for*, ...; falsche Einrückung

Laufzeitfehler treten erst zur Laufzeit des Programms auf. *Beispiel:* Division durch Null

Semantische (oder logische) Fehler bewirken, dass ein ansonsten fehlerfrei laufendes Programm falsche Ergebnisse liefert. *Beispiel:* Operanden werden addiert statt subtrahiert.

2 Einfache Datentypen

Aufgabe 2.1

```
print(3 + 4 * 1 + 2)
```

9

Aufgabe 2.2

```
print(27 // 6)
```

4

Aufgabe 2.3

```
print(15 / 3)
```

5.0

Aufgabe 2.4

```
print(15 // 3)
```

5

Aufgabe 2.5

```
print(23593901 % 2)
```

1

Aufgabe 2.6

```
print(123456 % 25)
```

6

Aufgabe 2.7

```
print(5.2 + 4.8)
```

10.0

Aufgabe 2.8

```
print((-3)**3)
```

-27

Aufgabe 2.9

```
print(10 / 2)
```

5.0

Aufgabe 2.10

```
print(abs(5-9))
```

4

Aufgabe 2.11

```
print(True and not False)
```

True

Aufgabe 2.12

```
print(False or False or False or False or False or True)
```

True

Aufgabe 2.13

```
print(True or not False and True)
```

True

Aufgabe 2.14

```
print(not False and not True or False)
```

False

Aufgabe 2.15

```
print(4 < 3 or 4 != 5)
```

True

Aufgabe 2.16

```
print(7 > 5 and 7 < 12)
```

True

Aufgabe 2.17

```
print(4 < 7 < 7)
```

False

Aufgabe 2.18

```
print(bool(0))
```

False

Aufgabe 2.19

```
print(True + False + True)
```

2

Aufgabe 2.20

```
print(int(19.9))
```

19

Aufgabe 2.21

```
print(float(255))
```

255.0

Aufgabe 2.22

```
print(bool(-7.2))
```

True

Aufgabe 2.23

```
print(bool(''))
```

False

3 Variablen

Aufgabe 3.1

```
a = 7
a = a - 2
a = 3 * a
print(a)
```

15

Aufgabe 3.2

```
a = 4
b = 5
a = b
b = a
print(b)
```

5

Aufgabe 3.3

```
a = 7
a += 10
a //= 3
print(a)
```

5

Aufgabe 3.4

```
a = 7
a *= -1
a *= a
print(a)
```

49

Aufgabe 3.5

```
a, b = 3, 7
a, b = b+1, a-3
print(b)
```

0

Aufgabe 3.6

```
a, b, c = 9, 8, 3  
b, a, c = c, b, a  
print(a+b-c)
```

2

4 Bedingte Anweisungen und Verzweigungen

Aufgabe 4.1

```
a = 3
if a > 1:
    a = a + 1
a = 2*a
print(a)
```

8

Aufgabe 4.2

```
a = 5
if a < 4:
    a = a + 1
a = 2*a
print(a)
```

10

Aufgabe 4.3

```
a = 5
if a < 7:
    a += 1
else:
    a -= 1
a += 10
print(a)
```

16

Aufgabe 4.4

```
x = 4
if x > 5:
    x *= -1
else:
    x *= 2
x += 10
print(x)
```

18

Aufgabe 4.5

```
a = 4
if a < 3:
    a = a + 1
elif a < 4:
    a = a + 2
elif a < 5:
    a = a + 3
else:
    a = a + 4
print(a)
```

7

Aufgabe 4.6

```
a = 7
if a < 3:
    a = a + 1
    if a > 3:
        a = a + 2
    else:
        a = 2*a
else:
    a = a - 4
    if a > 3:
        a = a + 2
    else:
        a = 2*a

print(a)
```

6

5 Schleifen

Aufgabe 5.1

```
for i in range(3, 7):  
    print(i)
```

3
4
5
6

Aufgabe 5.2

```
for a in [3, 5, 9]:  
    print(a)
```

3
5
9

Aufgabe 5.3

```
for x in range(1, 7, 3):  
    print(x)
```

1
4

Aufgabe 5.4

```
s = 0  
for x in [2, -3, 8]:  
    s += x  
print(s)
```

7

Aufgabe 5.5

```
for k in range(7, 3, -1):  
    print(k)
```

7
6
5
4

Aufgabe 5.6

```
s = 8
for a in range(2, 5):
    s += a
    print(a)
```

2
3
4

Aufgabe 5.7

```
a = 9
while a > 5:
    a -= 1
print(a)
```

5

Aufgabe 5.8

```
s = 0
a = 1
while s > 8:
    s += a
    a += 1
print(a)
```

1

Aufgabe 5.9

```
for x in [3, 8, -4, 9]:
    if x < 0:
        break
print(x)
```

-4

Aufgabe 5.10

```
s = 0
for x in [3, 1, -4, 7]:
    if x < 0:
        continue
    s = s+x
print(s)
```

11

6 Listen

Aufgabe 6.1

```
L = [3, 8, 9, 5, -2, 1]
print(L[3])
```

5

Aufgabe 6.2

```
L = [3, 8, 9, 5, -2, 1]
print(L[-4])
```

9

Aufgabe 6.3

```
L = [3, 8, 9, 5, -2, 1]
print(L[1:4])
```

[8, 9, 5]

Aufgabe 6.4

```
L = [3, 8, 9, 5, -2, 1]
print(L[:3])
```

[3, 8, 9]

Aufgabe 6.5

```
L = [1, 2]
print(3*L)
```

[1, 2, 1, 2, 1, 2]

Aufgabe 6.6

```
A = [1, 2]
B = [5, 3, 7]
print(B + A)
```

[5, 3, 7, 1, 2]

Aufgabe 6.7

```
L = [3, 8]
L.append(5)
print(L)
```

[3, 8, 5]

Aufgabe 6.8

```
L = [3, 8]
L.pop()
print(L)
```

[3]

Aufgabe 6.9

```
L = [3, 8]
x = L.pop()
L = [x] + L
print(L)
```

[8, 3]

Aufgabe 6.10

```
L = [3, 8, 1, 2, 5, 7]
L.insert(2, 4)
print(L)
```

[3, 8, 4, 1, 2, 5, 7]

Aufgabe 6.11

```
L = []
print(len(L))
```

0

7 Funktionen

Aufgabe 7.1

```
def f(x):  
    return 2*x + 1  
print(f(3))
```

7

Aufgabe 7.2

```
def f(x):  
    2*x + 1  
  
print(f(4))
```

None

Aufgabe 7.3

```
def f(x):  
    return 7  
  
print(f(4))
```

7

Aufgabe 7.4

```
a = 3  
print(a)  
  
def f(x):  
    a = x  
    print(a)  
  
f(7)  
print(a)
```

3

7

3

Aufgabe 7.5

```
def f(a, b):  
    c = 3  
    return 10*a + b + c  
  
print(f(2,1))
```

24

Aufgabe 7.6

```
def f(x):  
    return x + 1  
  
print(f(f(f(f(3)))))
```

7

Aufgabe 7.7

```
def f(a, b):  
    return 2*a+b  
  
print(f(f(3,4), f(1,-1)))
```

21

Aufgabe 7.8

```
def f(x):  
    if x == 1:  
        return 1  
    else:  
        return f(x-1)+1  
print(f(4))
```

4

8 Zeichenketten

Aufgabe 8.1

```
a = 'na'
b = 'nu'
print(a + b)
```

nanu

Aufgabe 8.2

```
a = 'x'
b = 'y'
print(2*a + 3*b)
```

xyyyy

Aufgabe 8.3

```
text = "Das kostet {0} Franken."
print(text.format(17))
```

Das kostet 17 Franken.

Aufgabe 8.4

```
text = "{2}-{0}-{1}"
print(text.format(3,7,4))
```

437

9 Ein- und Ausgabe

10 Dictionaries

11 Objektorientierte Programmierung