

Programmieren mit Python

Hilfreiche Links und Prüfungsvorbereitung

Richard Conrardy

Inhalt

1 Variablen, Datentypen und Type Casting	2
1.1 Hilfreiche Links	2
1.2 Prüfungsvorbereitung	2
2 Conditionals (<code>if</code> , <code>elif</code> , <code>else</code>) und Flussdiagramme	3
2.1 Hilfreiche Links	3
2.2 Prüfungsvorbereitung	3
3 Loops (<code>while</code> , <code>for</code>) und Flussdiagramme	4
3.1 Hilfreiche Links	4
3.2 Prüfungsvorbereitung	4
4 Listen	5
4.1 Hilfreiche Links	5
4.2 Prüfungsvorbereitung	5
5 Funktionen	6
5.1 Hilfreiche Links	6
5.2 Prüfungsvorbereitung	6

Dieses Dokument ergänzt die Folien zur Sitzung **Programmieren: textbasiert / Python**. Die Kapitelstruktur und die Übungs- und Prüfungsvorbereitungsaufgaben orientieren sich an den Materialien von **Peter Giger**. Die verlinkten Lösungsvorschläge stammen aus dem **öffentlichen GitHub-Repository**.¹

Die wichtigste Empfehlung gilt für alle Kapitel: Verbringen Sie nicht zu viel Zeit mit Theorie. Programmieren lernt man vor allem durch eigenes Ausprobieren, Fehlermeldungen, kleine Änderungen und viele Übungen.

¹Dieses Werk basiert auf Peter Gigers «Programmieren» und steht unter der Lizenz **CC BY 4.0**.

1 Variablen, Datentypen und Type Casting

1.1 Hilfreiche Links

Auf [YouTube](#) finden Sie Erklärvideos zu [Variablen](#), [Datentypen](#), der [Input-Funktion](#) und [Type Casting](#).

Die [W3Schools-Übersicht zu Variablen](#), [Datentypen](#) und [Type Casting](#) eignet sich als englisches Nachschlagewerk.

Weitere lineare Übungsaufgaben finden Sie bei [Sylvia Lange](#), insbesondere die Aufgaben 2, 3, 4, 6 und 12. Die Lösungsvorschläge sind [Netto/Brutto](#), [Zins](#), [Rabatt](#), [Quader](#) und [Mittelwert](#).

1.2 Prüfungsvorbereitung

Lösungsvorschlag zum Folienauftrag: [cm_to_feet.py](#)

1. Nennen Sie die vier wichtigsten Datentypen und geben Sie je ein Beispiel an.
2. Was ist der Output?

```
x = int(2.6)
y = 2
print(x * 2)
```

3. Was ist der Output?

```
x = "Hello "
y = "World"
print(x + y)
```

4. Was ist der Output?

```
a = 6
b = 1.5
c = a + b + (a / b)
print("Output: " + str(c))
```

5. Was macht dieses Programm?

```
x = float(input("Geben Sie eine Zahl ein: "))
print(x * x)
```

6. Erstellen Sie ein Programm, das den BMI berechnet:

$$BMI = \text{Gewicht}(kg) / \text{Grösse}(m)^2$$

Lösungsvorschlag BMI

2 Conditionals (if, elif, else) und Flussdiagramme

2.1 Hilfreiche Links

Die [Wikipedia-Zusammenfassung zu Flussdiagrammen](#) erklärt die wichtigsten Symbole. Zum Zeichnen eignet sich [draw.io](#) (Vorlage: *Flowchart*).

Auf YouTube finden Sie Erklärungen zur [If-Anweisung](#) und zu [If-Elif-Else](#).

Die [W3Schools-Übersicht zu Conditions](#) ist ein englisches Nachschlagewerk.

Weitere Übungsaufgaben (7, 8, 9, 10 und 11) finden Sie bei [Sylvia Lange](#), mit Lösungen zu [Quader](#), [Rabatt](#), [Rückgeld](#), [pq-Formel](#) und [BMI](#).

2.2 Prüfungsvorbereitung

Lösungsvorschläge zu den Folienaufträgen: [Login I](#), [Login II](#) und [Flussdiagramm Login II](#).

1. Was ist der Output? Erstellen Sie zusätzlich ein Flussdiagramm.

```
x = "Hello"
y = "Tom"

if x != "Hello":
    y = "Tim"
else:
    y = "Tina"

print(y)
```

[Lösungsvorschlag Flussdiagramm](#)

2. Was ist der Output? Erstellen Sie ein Flussdiagramm.

```
x = 5

if x < 0:
    print("a")
elif x > 0:
    if x == 5:
        print("b")
    print("c")
else:
    print("d")
```

[Lösungsvorschlag Flussdiagramm](#)

3. Schreiben Sie ein Programm, das Schweizer Noten in das [amerikanische Notensystem](#) umrechnet:

Schweiz	US
5.5–6	A
4.5–5.4	B
4.0–4.4	C
< 4	F

[Lösungsvorschlag GPA-Rechner](#)

3 Loops (**while**, **for**) und Flussdiagramme

3.1 Hilfreiche Links

Die [Wikipedia-Zusammenfassung zu Flussdiagrammen](#) und [draw.io](#) helfen beim Darstellen von Schleifen.

Auf YouTube finden Sie Erklärungen zur [While-Schleife](#) und zur [For-Schleife](#).

Die [W3Schools-Übersicht zur While-Schleife](#) und zur [For-Schleife](#) eignet sich als Nachschlagewerk.

Weitere Übungsaufgaben (17–20) finden Sie bei [Sylvia Lange](#), mit Lösungen zu [Potenz](#), [Fakultät](#), [Zweierpotenz](#) und [Sterne-Rechteck](#).

3.2 Prüfungsvorbereitung

Lösungsvorschläge zu den Folienaufträgen: [Primarschule](#), [Fahrplan](#), [Prüfungssoftware](#) und [Flussdiagramm Prüfungssoftware](#).

1. Was ist der Output? Übersetzen Sie das Programm, wenn möglich, in eine For-Schleife und erstellen Sie ein Flussdiagramm.

```
i = 2
while i < 5:
    print(i)
    i = i + 1
    print(i)
```

[Lösungsvorschlag Flussdiagramm](#)

2. Was ist der Output? Übersetzen Sie das Programm, wenn möglich, in eine For-Schleife und erstellen Sie ein Flussdiagramm.

```
i = 5
while i < 10:
    i += 1
    if i == 8:
        print("Katze")
    else:
        print(i)
```

[Lösungsvorschlag Code · Flussdiagramm](#)

3. Erstellen Sie ein Programm, das eine Zahl quadriert und so lange läuft, bis die Benutzerin oder der Benutzer `quit` eingibt.

[Lösungsvorschlag](#)

4. Erweitern Sie das Login aus dem Conditionals-Kapitel so, dass das Konto nach drei Fehlversuchen gesperrt wird.

[Lösungsvorschlag](#)

5. Erstellen Sie ein Programm, das aus einer Eingabezahl die Fibonacci-Folge berechnet. Erstellen Sie zusätzlich ein Flussdiagramm. Eine kurze Einführung bietet [SimpleClub](#).

[Lösungsvorschlag Code · Flussdiagramm](#)

4 Listen

4.1 Hilfreiche Links

Auf YouTube finden Sie Erklärungen zu [Listen I](#), [Listen II](#) und zum [Iterieren über Listen](#).

Die [W3Schools-Übersicht zu Listen](#) eignet sich als englisches Nachschlagewerk.

Die [Aufgabe 13 von Sylvia Lange](#) mit dem [Lösungsvorschlag](#) bietet zusätzliche Übung.

4.2 Prüfungsvorbereitung

Lösungsvorschläge zu den Folienaufträgen: [Einkaufsliste](#) und [Notendurchschnittsrechner](#).

1. Was macht dieses Programm und was ist der Output?

```
x = [6, 3, 8, 5, 7, 2, 4, 0, 5]
z = 0
for y in x:
    if y > z:
        z = y
print(z)
```

2. Eine Liste kann Unterlisten enthalten. Schreiben Sie ein Programm, das diese Liste ausgibt:

```
x = [[1, 2], [3, 4], [5, 6]]
```

[Lösungsvorschlag Unterlisten](#)

3. Wie oft kommt das Wort `lorem` (Gross- und Kleinschreibung) vor? Zählen Sie es im folgenden Text:

```
text = ["Lorem", "ipsum", "dolor", "sit", "amet",
        "consetetur", "sadipscing", "elit", "lorem",
        "sed", "lorem", "diam", "nonumy", "eirmod", "tempor",
        "invidunt", "Lorem", "ut", "labore", "et", "dolore",
        "magna", "aliquyam", "erat", "sed", "diam", "voluptuz",
        "At", "lorem", "vero", "eos", "et", "accusam", "et",
        "justo", "duo", "lorem", "dolores", "et", "ea"]
```

[Lösungsvorschlag](#) · [Quelle des Beispieltexts](#)

5 Funktionen

5.1 Hilfreiche Links

Auf YouTube finden Sie Erklärungen zu [Funktionen I](#), [Funktionen II](#) und [Funktionen III](#).

Die [W3Schools-Übersicht zu Funktionen](#) eignet sich als englisches Nachschlagewerk.

5.2 Prüfungsvorbereitung

Lösungsvorschläge zu den Folienaufträgen: [Briefadresse](#) und [Uhrzeit-Rechner](#).

1. Was ist der Output?

```
def meine_funktion(x, y, z):  
    if x > y:  
        return z  
    else:  
        return z - 1  
  
x = meine_funktion(1, 2, 3)  
print(x)
```

2. Was macht diese Funktion? Beschreiben Sie sie in Worten.

```
def meine_funktion(x, y):  
    if x > y:  
        return x  
    else:  
        return y
```

3. Programmieren Sie einen Würfel und würfeln Sie 1000-mal. Für eine Zufallszahl können Sie `random()` verwenden:

```
from random import random  
  
def wuerfle():  
    zufallszahl = random()  
    # Ihr Code kommt hier hin.
```

Lösungsvorschlag Würfel