



Technische
Universität
Braunschweig



Einführung in die Programmierung für Nicht Informatiker*innen

Phil Keier
Institut für Nachrichtentechnik (IfN)
Technische Universität Braunschweig



Lösungen 1. Tutorial

31 mögliche Punkte

Aufgabe 1

Gebe den Text Hallo Python aus, Nutze dabei die print-funktion. 1 Punkt

```
print("Hallo Python")
```

Aufgabe 1-1

Aufgabe 2 Punkte:

Definieren Sie zunächst die zwei Variablen a und b und initialisieren diese mit einem Integerwert ungleich 0:

```
a = 1  
b = -2
```

Aufgabe 1-1

Aufgabe 2 Punkte:

Definieren Sie zwei Variablen s und t und initialisieren diese mit einem Floatwert ungleich 0:

```
s = 1.5  
t = -2.7
```

Aufgabe 1-2

Aufgabe 2 Punkte:

Addieren Sie die Werte der Variablen a und b und speichern Sie das Ergebnis in der Variable c:

```
c = a + b
```

Aufgabe 1-2

Aufgabe 5 Punkte:

Nutzen Sie die Variablen a & b und Speichern Sie die Ergebnisse für die Multiplikation, Division, Ganzzahldivision, Exponentiation und den Modulo-Operator in den unten angegebenen Variablen:

$$m = a \cdot b$$

$$d = \frac{a}{b}$$

$$i = \left\lfloor \frac{a}{b} \right\rfloor$$

$$e = a^b$$

$$r = a \bmod b$$

$$m = a * b$$

$$d = a / b$$

$$i = a // b$$

$$e = a ** b$$

$$r = a \% b$$

Aufgabe 2-1

Aufgabe 2 Punkte:

Ein String-Objekt (Text) können sie mit Hilfe von 'Some Text' oder "Some Text2" definieren. Definieren sie die Variable text mit einem beliebigen Text.

```
text = "Hi Mom, I am on TV!"
```

Aufgabe 2-1

Aufgabe 1 Punkte:

Geben Sie die Variablen a & b aus Aufgabe 1 im format "a = 12 und b = 12" (Die Werte sollen dann den Werten aus ihrer Definition entsprechen. 12 ist hier nur ein Beispiel) aus.

```
print("a = {} und b = {}".format(a, b))
```

Aufgabe 2-2

Aufgabe 1 Punkte:

Definieren Sie die Variable `l` und weisen Sie dieser Variable eine Liste mit aufsteigenden Integerwerten von 0 bis 4 zu.

```
l = list(range(5))  
l = [0, 1, 2, 3, 4]
```

Aufgabe 2-2

Aufgabe 1 Punkte:

Hängen Sie der Liste l noch den Wert 42 an.

Hinweis: Nutzen Sie dafür die Methode .append.

```
l.append(42)
```

Aufgabe 2-2

Aufgabe 1 Punkte:

Geben Sie das dritte Element der Liste `l` aus.

Hinweis: Achten Sie darauf das der erste Index immer `0` ist.

```
print(l[2])
```

Aufgabe 2-2

Aufgabe 1 Punkte:

Geben Sie das vorletzte Element der Liste ` ` aus.

Hinweis: Achten Sie darauf das der letzte Index mit -1 ausgegeben wird.

```
print(l[-2])
```

Aufgabe 2-3

Das Dictionary ist ein Datentyp, welcher Schlüssel-Werte-Paare speichert. Dabei wird ein Dictionary mit `dict()` oder `{"Schlüssel1": "Wert1"}` initialisiert. Wichtig ist hierbei das ein Dictionary nicht mit `{}` initialisiert werden kann da dies die Notation für das Set Objekt ist.

Aufgabe 1 Punkte:

Initialisieren Sie die Dictionary Variable ``my_dict`` mit folgendem Mapping:

Key	Value
„apple“	„Apfel“
„banana“	„Banane“
„cherry“	„Kirsche“

```
my_dict = {  
    "apple": "Apfel",  
    "banana": "Banane",  
    "cherry": "Kirsche"  
}
```

Aufgabe 2-3

Aufgabe 1 Punkte:

Fügen Sie nun das Key-Value Paar "pear": "Birne" zu my_dict hinzu.

```
my_dict["pear"] = "Birne"  
my_dict.update({"pear": "Birne"})
```

Aufgabe 2-3

Aufgabe 2 Punkte:

Speichere die Werte des Dictionaries `my_dict` in der Variablen `values`.

Speichern Sie die Elemente des Dictionaries `my_dict`, welche mit der Funktion `.items()` ausgegeben werden, in der Variablen `items`.

```
values = my_dict.values()  
items = my_dict.items()
```

Aufgabe Funktionen

Aufgabe 1 Punkte*:

Schreibe eine Funktion successor die auf jede Eingabe +1 rechnet.

Schreibe eine Funktion add mit den Eingabeparametern a & b, welche die Werte von a & b miteinander addiert.

```
def successor(n):  
    return n+1  
  
def add(a,b):  
    return a+b
```

Aufgabe 3-1

Aufgabe 1 Punkte:

Schreibe eine Funktion `is_odd` mit einem Eingabeparameter `n` die prüft ob die eingegebene Zahl ungerade ist.

Wenn die Zahl gerade ist gebe den Text "Gerade Zahl" und bei ungerade "Ungerade Zahl" zurück.

```
def is_odd(n):  
    if n % 2 == 0:  
        return "Gerade Zahl"  
    else:  
        return "Ungerade Zahl"
```

Aufgabe 3-2

7 Punkte.

Aufgabe: Schreibe eine Funktion fubar mit Eingabeparameter n.
Die Funktion soll wie folgt definiert sein:

- Der Eingabeparameter n ist ein Integer, Floats geben False zurück
- Negative zahlen & 0 beenden die Funktion und geben False zurück
- Die Funktion zählt bis einschließlich dem Eingabeparameter n = 9 -> 1, 2, 3, ..., 9
- Bei jedem Schleifendurchlauf soll die Zahl bei der sich die Schleife gerade befindet mittels print ausgegeben werden werden.
- Ist der zurzeitige Schleifendurchlauf durch 3 teilbar, gebe mittels print denn String Foo aus.
- Ist der zurzeitige Schleifendurchlauf durch 5 teilbar, gebe mittels print denn String Bar aus.
- Ist der zurzeitige Schleifendurchlauf durch 3 & 5 teilbar, gebe mittels print den String FooBar aus.

Aufgabe 3-2

```
def fubar(n: int):  
    if isinstance(n, float) or n < 1:  
        return False  
  
    count = 1  
    while count <= n:  
        msg = count  
        if count % 3 == 0:  
            msg = "Foo"  
        if count % 5 == 0:  
            msg = "Bar"  
        if count % 15 == 0:  
            msg = "FooBar"  
  
        count += 1  
        print(msg, end=', ')
```



Lösungen 2. Tutorial

17+1 mögliche Punkte

Aufgabe - Kontrollstrukturen

3 Punkte

Schreibe eine Funktion `sum_up` mit Eingabeparameter `n`, welcher die Zahlen von `1...n` aufsummiert.

Nutze dafür einen `for-loop`.

```
def sum_up(n: int) -> int:
    count = 0
    for i in range(1, n+1):
        count += i
    return count
```

Aufgabe - Kontrollstrukturen

2 Punkte

Gegeben ist das Dictionary dict2.

Ändere jeden Wert in diesem Dictionary entsprechend der Formel $f(x) = x^3 - 1$ mittels for-loop.

```
dict2 = {"a": 56, 5: 12, "python": 9, 3.14: 1.141414}

# Möglichkeit 1 - Dictionary Comprehension
dict2 = {key: value**3-1 for key, value in dict2.items()}

# Möglichkeit 2 - For Loop
for key, value in dict2.items():
    dict2[key] = value**3 - 1
```

Zusatzaufgabe - Kontrollstrukturen

Keine Punkte (Extra Punkt möglich)

Erstelle eine Liste mittels List Comprehension, welche die Zahlen 1...6 auf deren kubische Zahl 1...216 also der Funktion $f(x) = x^3$ abbildet.

```
cubics = [n**3 for n in range(1,7)]
```

Aufgabe – System Interactions

2 Punkte

Erstelle und Öffne eine Datei testfile.txt mit der open Funktion, nutze dafür das with - Statement.

Schreibe in diese Datei 100 mal den String "Python\n".

```
# Möglichkeit 1
with open('testfile.txt', 'w') as f:
    for _ in range(100):
        f.write("Python\n")
```

```
# Möglichkeit 2
with open('testfile.txt', 'w') as f:
    f.write("Python\n" * 100)
```

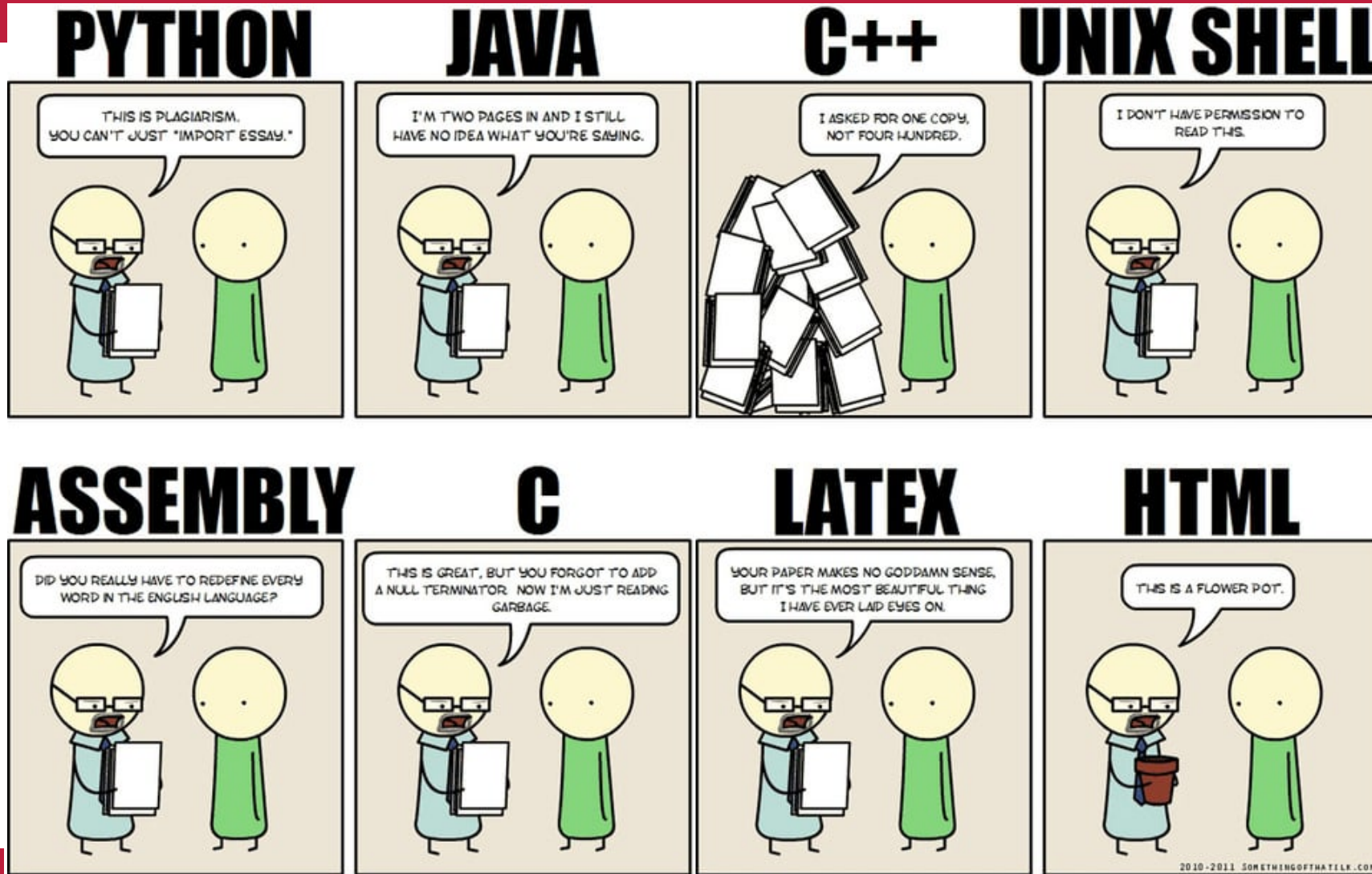
Aufgabe – System Interactions

2 Punkte

Öffne die zuvor erstellte Datei testfile.txt im Lesemodus und weiße den Inhalt der .readlines() Funktion der Variablen lines zu.

```
with open('testfile.txt', 'r') as f:  
    lines = f.readlines()
```

Memes



Aufgabe - Importing

3 Punkte

Importiere Pythons Built-In Library random und rufe zuerst aus dem Modul die Funktion seed auf mit dem Eingabewert 42, und weiße danach der Variable rand den Wert des Funktionsaufrufes von randint(1,100) zu.

```
import random
random.seed(42)
rand = random.randint(1,100)
```

Aufgabe - Importing

1 Punkt

Importiere die Built-In Library datetime als dt.

```
import datetime as dt
```

Aufgabe - Importing

2 Punkte

Importiere die Funktion `sqrt` aus dem Built-In Modul `math`.
Berechne `sqrt(4)`. Speicher das Ergebnis in der variablen `s4`.

```
from math import sqrt  
s4 = sqrt(4)
```

Aufgabe - Importing

2 Punkte

Importiere das Objekt TextCalendar aus dem Built-In Module calendar als TC.

```
from calendar import TextCalendar as TC

# Output
    November 2025
Mo Tu We Th Fr Sa Su
|  |  |  |  | 1 2
3 4 5 6 7 8 9
10 11 12 13 14 15 16
17 18 19 20 [21] 22 23
24 25 26 27 28 29 30

21.11.2025 - Nächste Vorlesung
```