

# Aufgabenblatt 3

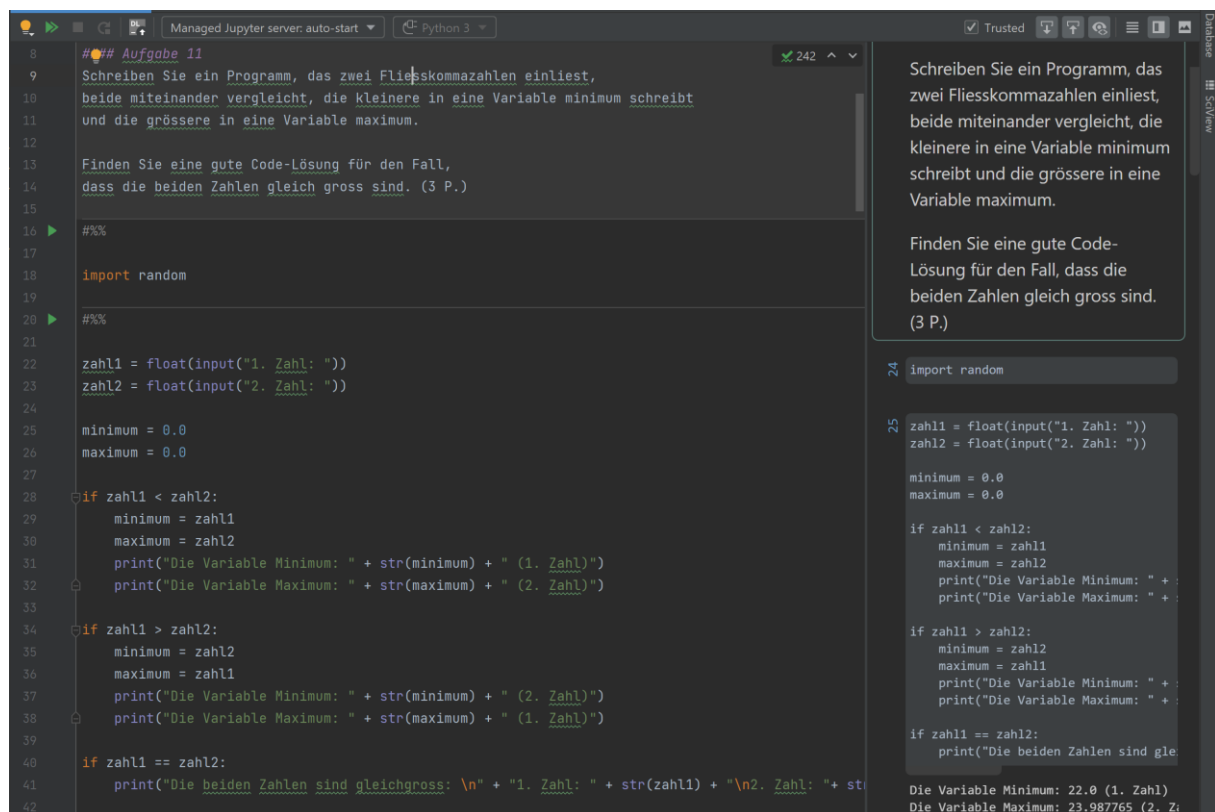
Das sind die Aufgaben der Woche 3. Es handelt sich um Verzweigungen in Programmen, also if und else.

## Aufgabe 11 – Eingaben und Konditionale

Wir wollen lernen, wie man im Programm verzweigen kann, also eine sogenannte Selektion durchführen kann. Dies geschieht mit if-else-Ausdrücken.

Schreiben Sie ein Programm, das zwei Fließkommazahlen einliest, beide miteinander vergleicht, die kleinere in eine Variable **minimum** schreibt und die grössere in eine Variable **maximum**. Finden Sie eine gute Code-Lösung für den Fall, dass die beiden Zahlen gleich gross sind.

Lösung:



```
8  ### Aufgabe 11
9  Schreiben Sie ein Programm, das zwei Fließkommazahlen einliest,
10 beide miteinander vergleicht, die kleinere in eine Variable minimum schreibt
11 und die grössere in eine Variable maximum.
12
13 Finden Sie eine gute Code-Lösung für den Fall,
14 dass die beiden Zahlen gleich gross sind. (3 P.)
15
16 #%%
17
18 import random
19
20 #%%
21
22 zahl1 = float(input("1. Zahl: "))
23 zahl2 = float(input("2. Zahl: "))
24
25 minimum = 0.0
26 maximum = 0.0
27
28 if zahl1 < zahl2:
29     minimum = zahl1
30     maximum = zahl2
31     print("Die Variable Minimum: " + str(minimum) + " (1. Zahl)")
32     print("Die Variable Maximum: " + str(maximum) + " (2. Zahl)")
33
34 if zahl1 > zahl2:
35     minimum = zahl2
36     maximum = zahl1
37     print("Die Variable Minimum: " + str(minimum) + " (2. Zahl)")
38     print("Die Variable Maximum: " + str(maximum) + " (1. Zahl)")
39
40 if zahl1 == zahl2:
41     print("Die beiden Zahlen sind gleichgross: \n" + "1. Zahl: " + str(zahl1) + "\n2. Zahl: " + str(zahl2))
42
```

Schreiben Sie ein Programm, das zwei Fließkommazahlen einliest, beide miteinander vergleicht, die kleinere in eine Variable minimum schreibt und die grössere in eine Variable maximum. Finden Sie eine gute Code-Lösung für den Fall, dass die beiden Zahlen gleich gross sind. (3 P.)

```
import random

zahl1 = float(input("1. Zahl: "))
zahl2 = float(input("2. Zahl: "))

minimum = 0.0
maximum = 0.0

if zahl1 < zahl2:
    minimum = zahl1
    maximum = zahl2
    print("Die Variable Minimum: " + str(minimum) + " (1. Zahl)")
    print("Die Variable Maximum: " + str(maximum) + " (2. Zahl)")

if zahl1 > zahl2:
    minimum = zahl2
    maximum = zahl1
    print("Die Variable Minimum: " + str(minimum) + " (2. Zahl)")
    print("Die Variable Maximum: " + str(maximum) + " (1. Zahl)")

if zahl1 == zahl2:
    print("Die beiden Zahlen sind gleichgross: \n" + "1. Zahl: " + str(zahl1) + "\n2. Zahl: " + str(zahl2))
```

Die Variable Minimum: 22.0 (1. Zahl)  
Die Variable Maximum: 23.987765 (2. Zahl)

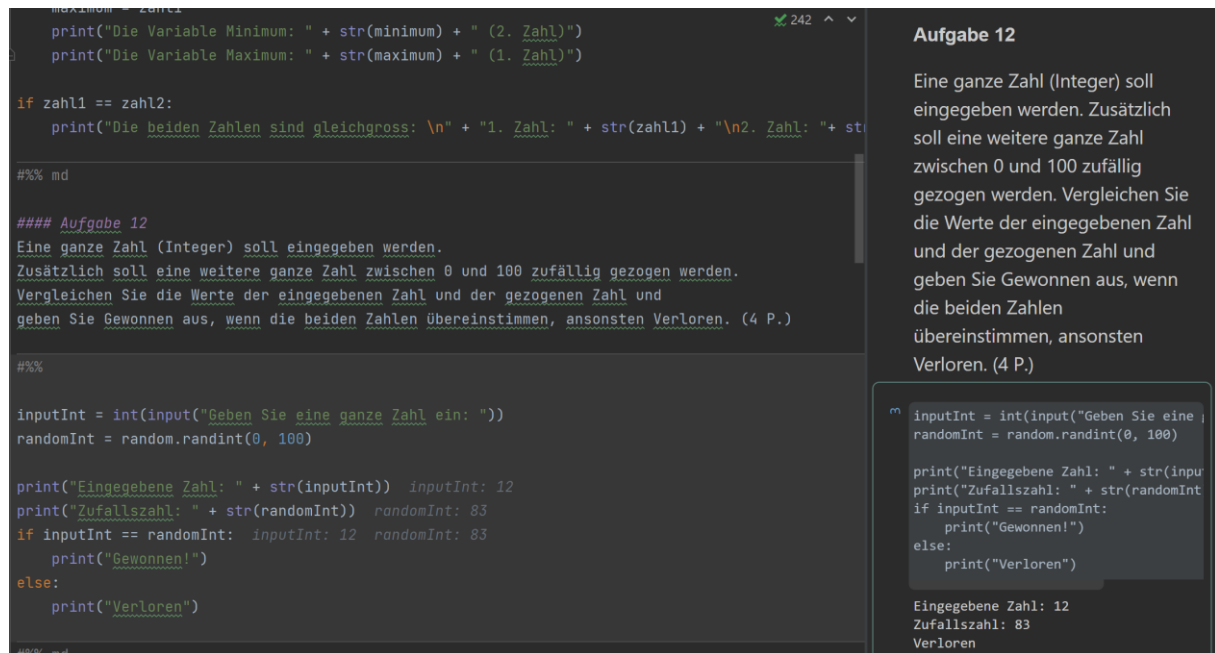
Abbildung 1 - Aufgabe 11

Die Lösung ist auch im python Skript im .zip zu finden.

## Aufgabe 12 – Zufallszahlen und Bibliotheken

Eine ganze Zahl (Integer) soll eingegeben werden. Zusätzlich soll eine weitere ganze Zahl zwischen 0 und 100 zufällig gezogen werden. Vergleichen Sie die Werte der eingegebenen Zahl und der gezogenen Zahl und geben Sie **Gewonnen** aus, wenn die beiden Zahlen übereinstimmen, ansonsten **Verloren**. (4 P.)

Lösung:



```
maximum = zahl1
print("Die Variable Minimum: " + str(minimum) + " (2. Zahl)")
print("Die Variable Maximum: " + str(maximum) + " (1. Zahl)")

if zahl1 == zahl2:
    print("Die beiden Zahlen sind gleichgross: \n" + "1. Zahl: " + str(zahl1) + "\n2. Zahl: " + str(zahl2))

### Aufgabe 12
Eine ganze Zahl (Integer) soll eingegeben werden.
Zusätzlich soll eine weitere ganze Zahl zwischen 0 und 100 zufällig gezogen werden.
Vergleichen Sie die Werte der eingegebenen Zahl und der gezogenen Zahl und
geben Sie Gewonnen aus, wenn die beiden Zahlen übereinstimmen, ansonsten Verloren. (4 P.)

inputInt = int(input("Geben Sie eine ganze Zahl ein: "))
randomInt = random.randint(0, 100)

print("Eingegebene Zahl: " + str(inputInt))
print("Zufallszahl: " + str(randomInt))
if inputInt == randomInt:
    print("Gewonnen!")
else:
    print("Verloren")
```

**Aufgabe 12**

Eine ganze Zahl (Integer) soll eingegeben werden. Zusätzlich soll eine weitere ganze Zahl zwischen 0 und 100 zufällig gezogen werden. Vergleichen Sie die Werte der eingegebenen Zahl und der gezogenen Zahl und geben Sie Gewonnen aus, wenn die beiden Zahlen übereinstimmen, ansonsten Verloren. (4 P.)

```
inputInt = int(input("Geben Sie eine ganze Zahl ein: "))
randomInt = random.randint(0, 100)

print("Eingegebene Zahl: " + str(inputInt))
print("Zufallszahl: " + str(randomInt))
if inputInt == randomInt:
    print("Gewonnen!")
else:
    print("Verloren")
```

Eingegebene Zahl: 12  
Zufallszahl: 83  
Verloren

Abbildung 2 - Aufgabe 12

Die Lösung ist auch als Python Datei (mit Kommentaren) im .zip.

## Aufgabe 13 – Kniffel, das Würfelspiel

Ihre folgende Aufgabe soll es ein, 5 Würfelvariablen zufällig Werte zwischen 1 und 6 zuzuweisen und diese Werte alle auszugeben. **(1 P.)** Schreiben Sie ein Programm, das überprüft, ob:

- (a) Alle 5 Werte gleich sind. **(1 P.)**
- (b) Alle 5 Werte unterschiedlich sind **(2 P.)**
- (c) Die Summe der 5 Werte grösser als 20 und kleiner als 25 ist. **(1 P.)**

Lösung:

Damit diese Aufgabe gelöst werden kann, müssen zuerst die Random Bibliothek und die 5 Zufallszahlen zugewiesen werden:

```
8  import random
9  # Teil 1: Zuweisung Variablen
10 var1 = random.randint(1, 6)
11 var2 = random.randint(1, 6)
12 var3 = random.randint(1, 6)
13 var4 = random.randint(1, 6)
14 var5 = random.randint(1, 6)
15
16 print("1. Zahl: " + str(var1))
17 print("2. Zahl: " + str(var2))
18 print("3. Zahl: " + str(var3))
19 print("4. Zahl: " + str(var4))
20 print("5. Zahl: " + str(var5))
```

Abbildung 3 - Aufgabe 13 Import&Variabelzusweisung

Die Zahlen werden ausgegeben, damit beim «Würfeln» auch etwas spass dabei ist.

- a) Da alle Variablen gleich sein müssen, können diese mit «=» aneinandergehängt werden:

```
23 # Teil 2: a)
24 if var1 == var2 == var3 == var4 == var5:
25     print("Alle Variablen haben die gleiche Zahl drin!")
```

Abbildung 4 - Aufgabe 13 a)

- b) Das Abprüfen der Variablen auf nichtgleichheit, ist etwas komplizierter, da es 5+4+3+2+1=5 Möglichkeiten gibt, also 10.  
Alternativ zu meiner Lösung könnte auch der boolsche Wert in einer Variabel gespeichert werden, anstatt alles im «if» zu haben:

```
22 # für b)
23 gleichWieVar1 = var1 != var2 and var1 != var3 and var1 != var4 and var1 != var5
24 gleichWieVar2 = var2 != var3 and var2 != var4 and var2 != var5
25 gleichWieVar3 = var3 != var4 and var3 != var5
26 gleichWieVar4 = var4 != var5
27
28 # Teil 2: a)
29 if var1 == var2 == var3 == var4 == var5:
30     print("Alle Variablen haben die gleiche Zahl drin!")
31
32 # Teil 3: b)
33 elif gleichWieVar1 and gleichWieVar2 and gleichWieVar3 and gleichWieVar4:
34     print("Alle Variablen sind unterschiedlich!")
```

Abbildung 5 - Aufgabe 13 b) Alternative

Dieses Beispiel ist auch im Python Skript Aufgabe13\_02 zu finden. Generell wäre es eventuell sinnvoll das mit einer Schleife zu überprüfen, ich habe mich aber dagegen entschieden, aufgrund von Überkomplizierung des Codes.

Zu beachten ist, dass der Code vor Zeile 29 stehen muss, da es sich um ein «elif» handelt und die Zeilen 31 und 32 nur im «if» stehen dürfen und nicht ausserhalb.

- c) Im letzten Schritt wird die Summe der Variablen in einer neuen Variabel gespeichert und danach im «if» überprüft, ob diese Summe zwischen 20 und 25 liegt:

```
34 # Teil 4: c)
35 summe = var1 + var2 + var3 + var4 + var5
36 if 20 < summe < 25:
37     print("Die summe der Zahlen: ")
38     print(var1)
39     print(var2)
40     print(var3)
41     print(var4)
42     print(var5)
43     print("ist: " + summe)
44 else:
45     print("Die Summe ist nicht zwischen 20 und 25")
```

Abbildung 6 - Aufgabe 13 c)

Im «else» wird nur noch ausgegeben, falls die Summe nicht zwischen 20 und 25 liegt.

## Aufgabe 14 – Boolesche Ausdrücke

Zu was werten die folgenden Ausdrücke aus (im Falle, dass Sie syntaktisch korrekt sind)? Begründen Sie Ihre Antwort kurz.

- (a) True and False and True or False (1 P.)
- (b) (3 < 3.0) or (not(not(True or False))) (1 P.)
- (c) 3 != 4 != 5 != 6 != 3 (1 P.)
- (d) a = 3  
b = 4  
c = 5  
print(a\*\*2 + b\*\*2 == c\*\*2) (1 P.)

Lösung:

a) Das Resultat ist: False

```

128 a)
129
130 #%%
131
132 #Der erste Teil ist schon "False" da Wahr und Falsch nicht Wahr sind
133 print(True and False and True or False)
134

```

a)

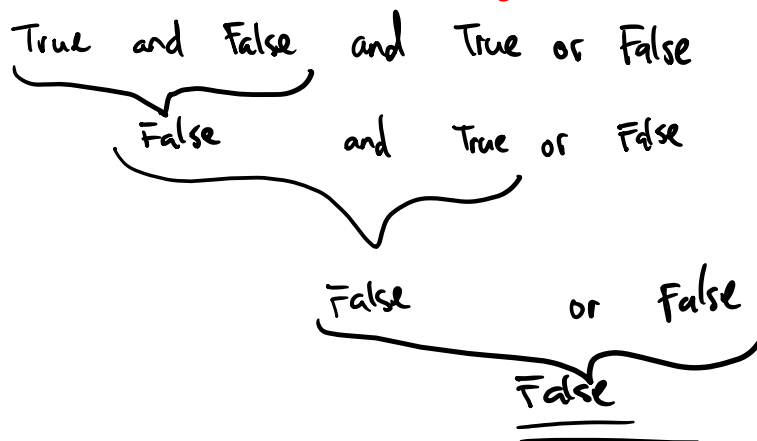
```

#Der erste Teil ist schon "False" da l
print(True and False and True or Falsi
False

```

Abbildung 7 - Aufgabe 14 a)

Der Ausdruck wird von links nach rechts ausgewertet.



b) Das Resultat ist: True

```

137 b)
138
139 #%%
140
141 # 3 < 3.0 ist false
142 # or
143 # not not negiert sich selber
144 # True or False true
145 print((3 < 3.0) or (not(not(True or False))))]
146

```

b)

```

# 3 < 3.0 ist false
# or
# not not negiert sich selber
# True or False true
print((3 < 3.0) or (not(not(True or F
True

```

Abbildung 8 - Aufgabe 14 b)

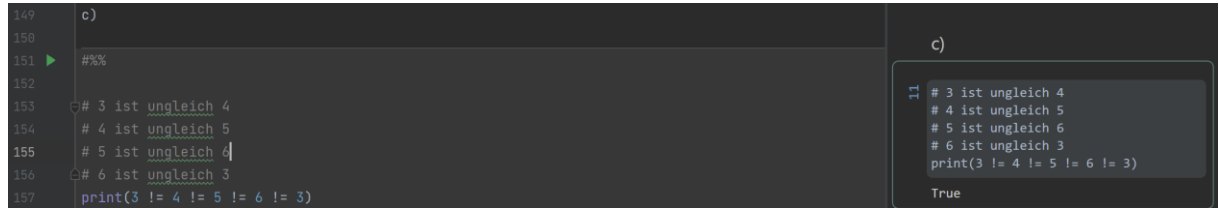
Die Klammern machen, dass zuerst 3 mit 3.0 verglichen wird, was False ist und erst danach die verschachtelten «not».

Da eine Doppelverneinung vorhanden ist, können diese auch wieder gelöscht werden bzw. sind diese unnötig/überflüssig.

True or False gibt True zurück

Also schlussendlich heisst der Ausdruck: False or True → Resultat: True

## c) Das Resultat ist: True



```
149 c)
150
151 #%%
152
153 # 3 ist ungleich 4
154 # 4 ist ungleich 5
155 # 5 ist ungleich 6
156 # 6 ist ungleich 3
157 print(3 != 4 != 5 != 6 != 3)
```

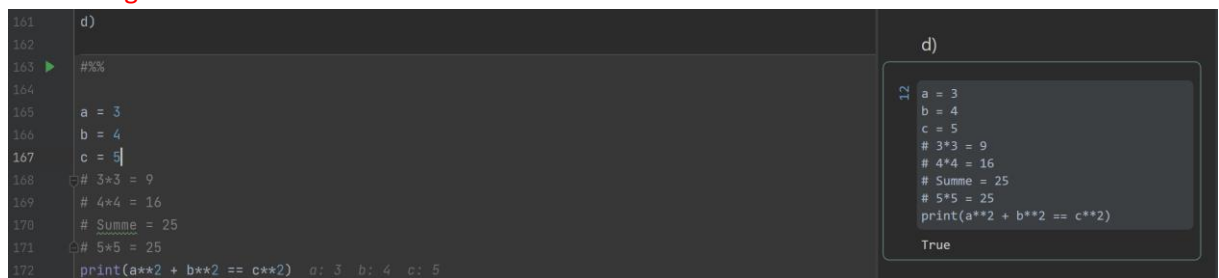
```
11 # 3 ist ungleich 4
# 4 ist ungleich 5
# 5 ist ungleich 6
# 6 ist ungleich 3
print(3 != 4 != 5 != 6 != 3)

True
```

Abbildung 9 - Aufgabe 14 c)

Da von links nach rechts überprüft wird, wird das erste «3» nicht mit dem letzten «3» verglichen.

## d) Die Lösung ist: True



```
161 d)
162
163 #%%
164
165 a = 3
166 b = 4
167 c = 5
168 # 3*3 = 9
169 # 4*4 = 16
170 # Summe = 25
171 # 5*5 = 25
172 print(a**2 + b**2 == c**2) a: 3 b: 4 c: 5
```

```
12 a = 3
b = 4
c = 5
# 3*3 = 9
# 4*4 = 16
# Summe = 25
# 5*5 = 25
print(a**2 + b**2 == c**2)

True
```

Abbildung 10 - Aufgabe 14 d)

Die Antwort ist True, weil uns das Pythagoras bewiesen hat. (In einem Dreieck)

$$a^2 + b^2 = c^2$$

$$3^2 + 4^2 = 5^2$$

$$9 + 16 = 25 \rightarrow \text{Korrekt!}$$

Die Lösung ist auskommentiert im Python-Skript Aufgabe14.py.

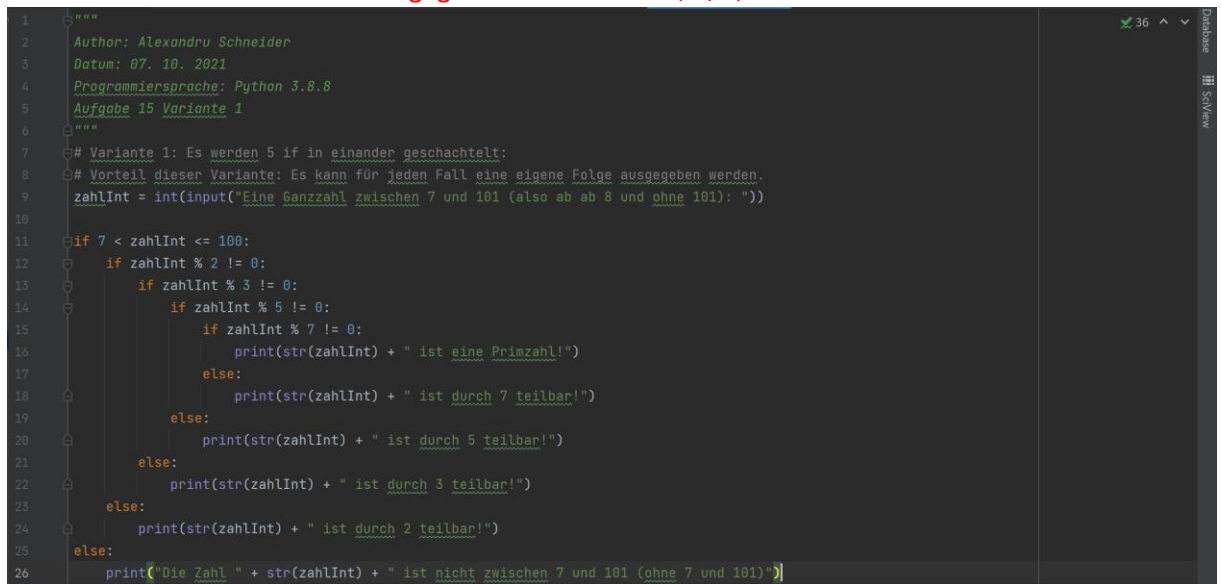
## Aufgabe 15 – Primzahlentest > 7

Schreiben Sie ein Programm, das von einer eingegebenen Integerzahl grösser 7 und kleiner gleich 100 testet, ob diese eine Primzahl ist oder nicht. Eine Zahl zwischen 8 und 100 ist eine Primzahl genau dann, wenn sie nicht durch 2, 3, 5 und 7 teilbar ist. **(4 P.)**

Lösung:

Ich habe 2 Lösungen gefunden.

1. Testet in einzelnen «if» ob die eingegebene Zahl durch 2, 3, 5, 7 teilbar ist.



```
1 """
2 Author: Alexandru Schneider
3 Datum: 07. 10. 2021
4 Programmiersprache: Python 3.8.8
5 Aufgabe 15 Variante 1
6 """
7 # Variante 1: Es werden 5 if in einander geschachtelt:
8 # Vorteil dieser Variante: Es kann für jeden Fall eine eigene Folge ausgegeben werden.
9 zahlInt = int(input("Eine Ganzzahl zwischen 7 und 101 (also ab ab 8 und ohne 101): "))
10
11 if 7 < zahlInt <= 100:
12     if zahlInt % 2 != 0:
13         if zahlInt % 3 != 0:
14             if zahlInt % 5 != 0:
15                 if zahlInt % 7 != 0:
16                     print(str(zahlInt) + " ist eine Primzahl!")
17                 else:
18                     print(str(zahlInt) + " ist durch 7 teilbar!")
19             else:
20                 print(str(zahlInt) + " ist durch 5 teilbar!")
21         else:
22             print(str(zahlInt) + " ist durch 3 teilbar!")
23     else:
24         print(str(zahlInt) + " ist durch 2 teilbar!")
25 else:
26     print("Die Zahl " + str(zahlInt) + " ist nicht zwischen 7 und 101 (ohne 7 und 101)!")
```

Abbildung 11 - Aufgabe 15 Variante 1

Zuerst wird überprüft, ob der eingegebene Integer zwischen 7 und 100 ist. Dabei zu beachten ist, dass der eingegebene Integer auch 100 sein darf. (aber erst bei 8 beginnen)  
Die 4 verschachtelten «if» werden dann auf die oben angegebenen Zahlen überprüft. Dabei wird mit dem Modulo ermittelt, ob ein Rest vorhanden ist. Wenn der Rest nicht «0» ist, dann ist dieser Integer keine Primzahl.

Der Vorteil dieser Variante ist, dass zu jeder Zahl ein eigener «else»-Fall erstellt werden kann. In diesem Fall ist das nicht nötig, deshalb habe ich auch die Variante 2 erstellt:

2. Testet in einem «if» ob die Zahl zwischen 7 und 100 ist, sowohl auch ob die Zahl eine Primzahl ist. (Zur Hilfe der Lesbarkeit wurde das in Klammern gesetzt).



```
1 """
2 Author: Alexandru Schneider
3 Datum: 07. 10. 2021
4 Programmiersprache: Python 3.8.8
5 Aufgabe 15 Variante 2
6 """
7 # Variante 2: Es wird zuerst geprüft ob die Zahl zwischen 7 und 101 ist und
8 # dann in einem if alles abgefragt (mit and-Operator)
9 zahlInt = int(input("Eine Ganzzahl zwischen 7 und 101 (also ab ab 8 und ohne 101): "))
10
11 if (7 < zahlInt <= 100) and (zahlInt % 2 != 0 and zahlInt % 3 != 0 and zahlInt % 5 != 0 and zahlInt % 7 != 0):
12     print(str(zahlInt) + " ist eine Primzahl!")
13 else:
14     print("Die Zahl " + str(zahlInt) + " ist keine Primzahl oder nicht zwischen 7 und 101!")
```

Abbildung 12 - Aufgabe 15 Variante 2

Nun kann beim «else» auch nicht ermittelt werden, warum bzw. wann false (oder in anderen Fällen der Fehler) eingetroffen ist.