

1. Welche Aussage beschreibt den Begriff Maschinencode?

Instruktionen, die vom Prozessor eines Computers direkt ausgeführt werden können.

Eine Möglichkeit, um Algorithmen graphisch darzustellen.

Ein spezielles Passwort, das bei jedem Start des Computers abgefragt wird.

Eine Beschreibung, die erklärt, was ein Programm tun soll und wie es funktioniert.

2. Welcher Begriff bezeichnet die Tätigkeit, bei der man das Problem genau definiert, wenn man ein neues Programm entwickelt?

Entwurf

Erweiterung

Programmierung

Analyse

3. Welcher Begriff steht für eine Methode, um ein komplexes Problem in kleinere, leichter verständliche Probleme zu zerteilen?

Analyse

Abstraktion

Problemzerlegung

Debugging

4. Welcher Begriff kennzeichnet einen Programmcode, der unabhängig von einer Programmiersprache den Ablauf beschreibt?

Funktion

Maschinencode

Pseudocode

Boolesche Logikaussage

5. Welcher der folgenden Ausdrücke ist ein gültiger Kommentar in Python?

`/* Das ist ein Kommentar`

`// Das ist ein Kommentar`

`# Das ist ein Kommentar`

`print("Das ist ein Kommentar")`

6. Welcher Vergleichsoperator wird für „ungleich“ verwendet?

`==`

`<=`

`<`

`!=`

7. Welcher Begriff bezeichnet eine Schleife?

Integer

**While**

Gleitkommazahl

Liste

8. Welcher Begriff bezeichnet ein Unterprogramm?

Debugging

Select

For-Schleife

**Funktion**

9. Wie heißen Schreibfehler in einem Programm, die zu einem ungültigen Programm und einer Fehlermeldung führen?

**Syntaxfehler**

Rekursionsfehler

Logikfehler

Schleifenfehler

10. Welcher der folgenden Aussagen beschreibt den Begriff Computing?

Eingabe und Bearbeitung von Daten mit Hilfe einer Anwendungssoftware

Entwicklung von Hardware

**Berechnung oder Verarbeitung von Daten unter Verwendung eines Computersystems**

Entwicklung von Anwendungssoftware

11. Welcher der folgenden Begriffe bezeichnet das Analysieren von Problemen und Finden von möglichen Lösungen?

Logisches Denken

**Computational Thinking**

Analytisches Denken

Computer-based Thinking

12. Wofür steht der Begriff Programmspezifikation?

Beschreibung der Vorgangsweise, wie ein Programm bedient werden soll

Beschreibung des Programmcodes

Beschreibung der Entwicklungsumgebung

**Beschreibung der detaillierten Anforderungen, was ein Programm machen soll**

13. Welcher der folgenden Namen ist am aussagekräftigsten und sollte daher für eine Variable verwendet werden, welche die Anzahl der Mitarbeiter der Abteilung Einkauf speichert?

mitarbeiter\_einkauf  
mitarbeiter  
einkäufer  
angestellte

14. Womit kann man erreichen, dass ein Anweisungsblock wiederholt ausgeführt wird?

Wiederholung  
Verzweigung  
Schleife  
Funktion

15. Der folgende Pseudocode beschreibt das Kochen von Nudeln. Wo liegt der Fehler?

```
START
Nudelwasser zum Kochen bringen
Solange Nudeln nicht fertig sind
    Weiterkochen
Nudeln aus dem Wasser nehmen
STOP
```

Fehlende Anweisung  
Falsche Sequenz  
Falsches Entscheidungskriterium  
Inkorrekte Einrückung

16. Welche Anweisung definiert in Python die Variable z als Integer?

int(z)  
z=0  
z=int  
z(int)

17. Im folgenden Code wird nach dem Wohnort einer Person gefragt. Wie muss der Code in der zweiten Zeile ergänzt werden, damit die Frage wie folgt ausgegeben wird: "Wo wohnst du Franz?"

```
frage = "Wo wohnst du" + name + "?"
frage = "Wo wohnst du" ,+ name , + "?"
frage = "Wo wohnst du " + name + "?"
frage = "Wo wohnst du, name?"
```

18. Mit welcher Anweisung wird das dritte Element („drei“) der Liste number\_list ausgegeben?

```
number_list = [„eins“, „zwei“, „drei“, „vier“]
```

```
print(number_list(3))
print(number_list[3])
print(number_list(2))
print(number_list[2])
```

19. Welcher Code muss an der Stelle ... eingefügt werden, sodass eine Funktion mit Rückgabewert mit der Bezeichnung „subtrahiereZahlen“ definiert wird, die zwei eingegebene Zahlen subtrahiert und dann das Ergebnis ausgibt?

```
Zahl1 = int(input(„Gib bitte die erste Zahl ein: „))
Zahl2 = int(input(„Gib bitte die zweite Zahl ein: „))
...
Ergebnis = Zahl1 – Zahl2
return Ergebnis
Antwort = subtrahiereZahlen(Zahl1, Zahl2)
print(„Die Antwort ist:“,Antwort)
```

```
funktion subtrahiereZahlen(Zahl1, Zahl2):
def subtrahiereZahlen(Zahl1, Zahl2):
funktion subtrahiereZahlen(Z1 , Z2):
def subtrahiereZahlen(Z1, Z2):
```

20. Folgender Code enthält einen Syntaxfehler. Welche Zeile muss korrigiert werden?

```
import random
alter = random.randint(1,100)
if alter > 16:
    print("Bitte Erwachsenenkarte kaufen")
else
    print("Bitte Kinderkarte kaufen")

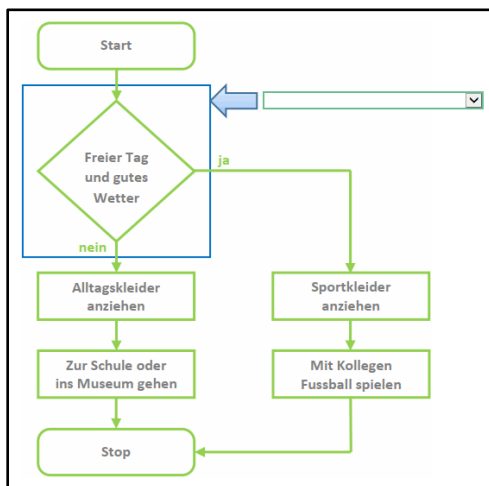
else
import random:
alter = randint(1,100)
if alter > 16
```

21. Der folgende Code soll zwischen den beiden print-Anweisungen eine Pause von 5 Sekunden einfügen. Mit welcher Codezeile wird an der Stelle ... das dafür benötigte Modul eingebunden?

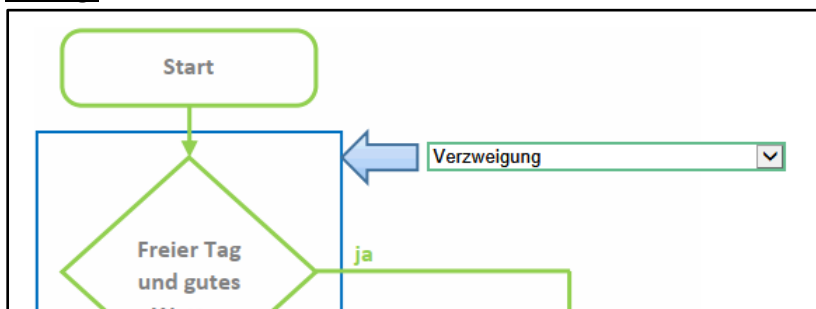
```
...
print("5 Sekunden Pause")
time.sleep(5)
print('Pause beendet')
```

```
from time import sleep
from time import *
import time
import sleep
```

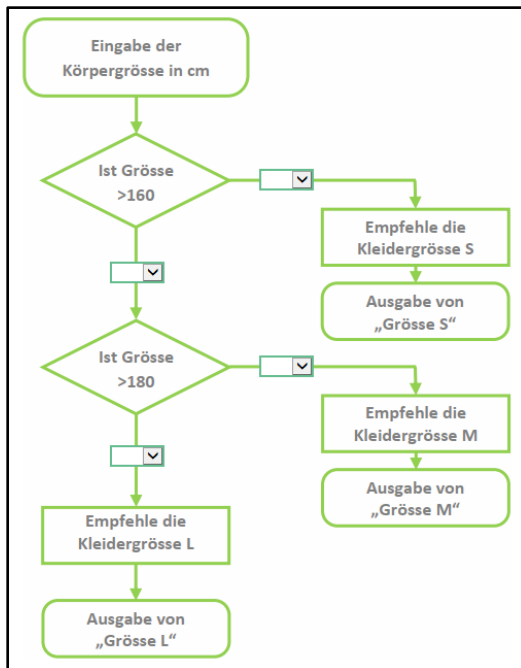
22. Was bedeutet das Symbol einer Raute (s. Bild) In einem Flussdiagramm?



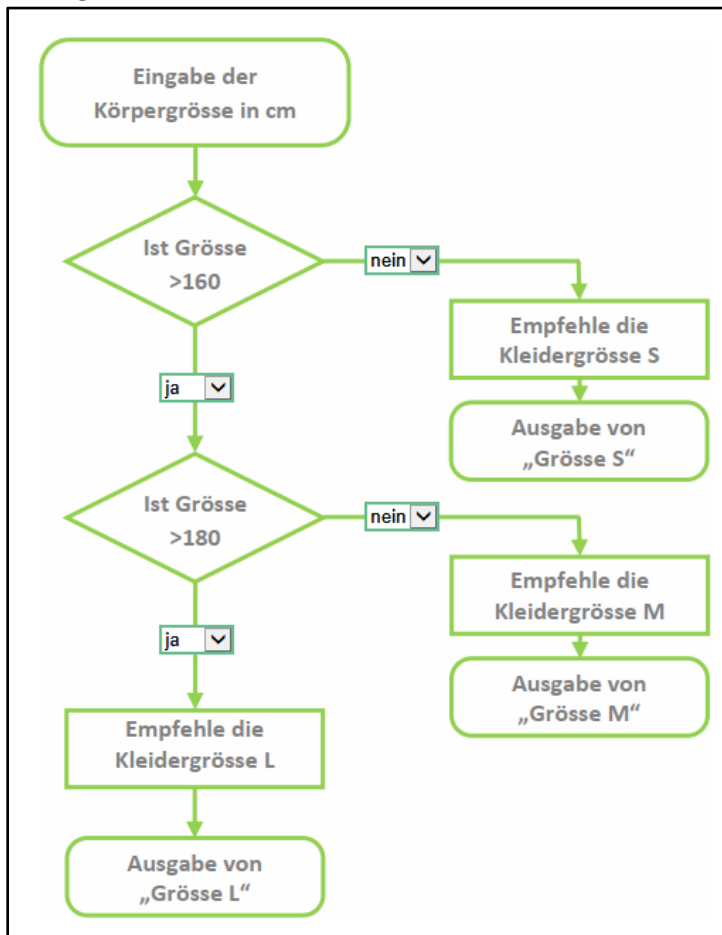
**Lösung:**



23. Vervollständigen Sie das Flussdiagramm über die Dropdown-Menüs, sodass die richtige Kleidergröße empfohlen wird.



**Lösung:**



24. Das angezeigte Python Programm berechnet den größten gemeinsamen Teiler zweier ganzen Zahlen a und b. Vervollständigen Sie das Flussdiagramm.

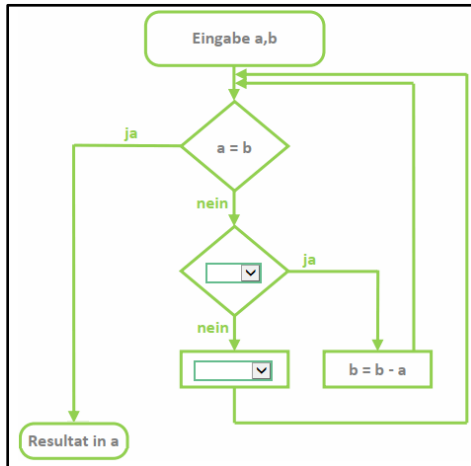
while a != b :

if a > b:

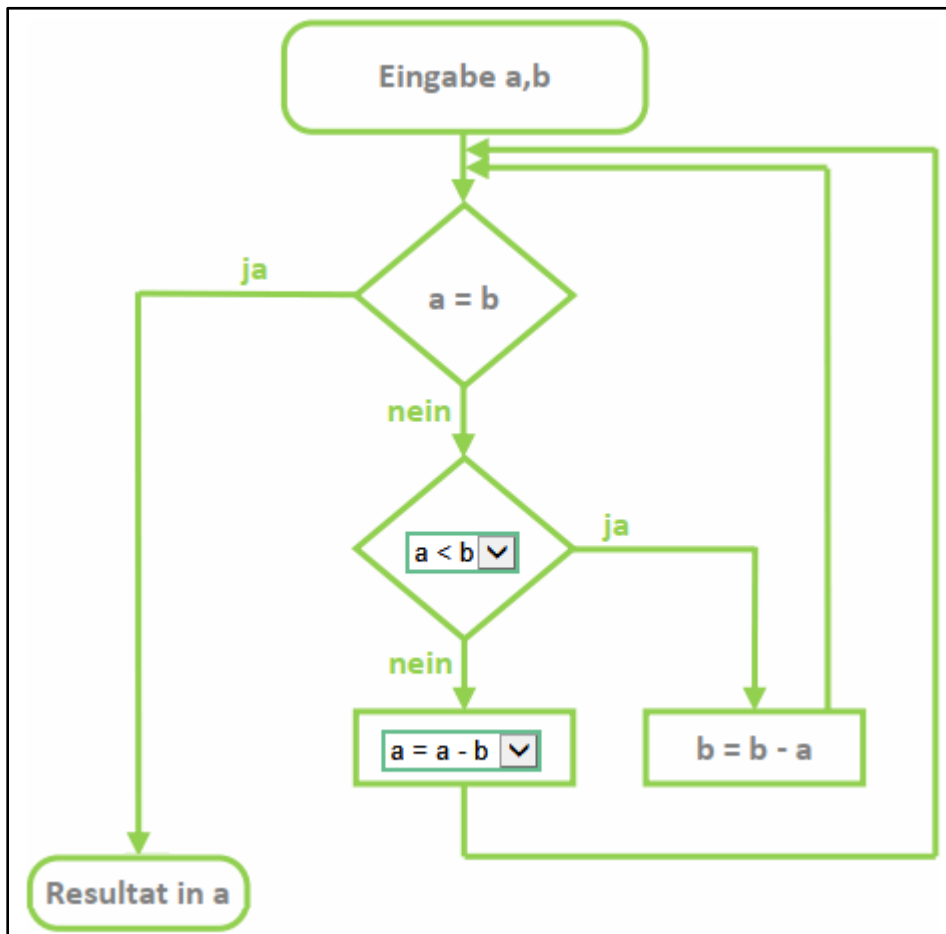
a = a - b

else:

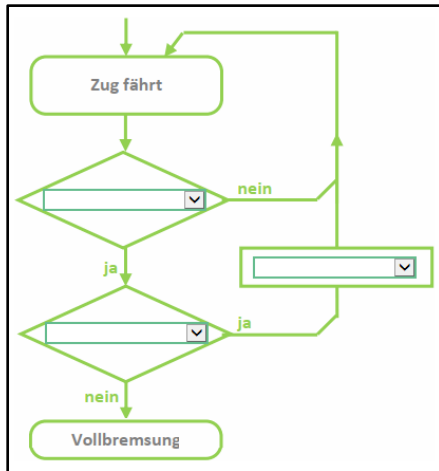
b = b - a



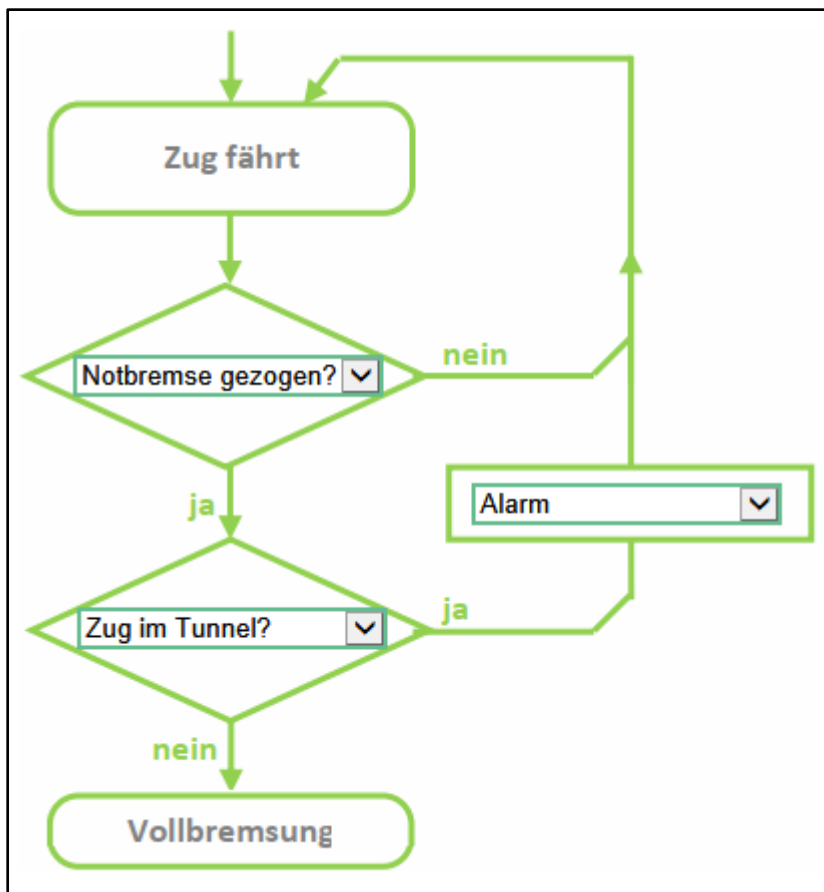
**Lösung:**



25. Ein brennender Personenzug darf in einem Eisenbahntunnel keinesfalls angehalten werden, auch wenn die Notbremse gezogen wird. Ergänzen Sie das Flussdiagramm.




**Lösung:**






26. Entfernen Sie die unnötigen Klammern im angezeigten Ausdruck.  
(Hinweis zur Kontrolle: Das Resultat des Ausdrucks soll gleich 1444 sein.)


 Ausführen

```
1 print (((5 + 2) * 5 + 3) ** 2)
2
```

**Lösung:**

 Ausführen

```
1 print (((5 + 2) * 5) + 3) ** 2)
2 print ((5 + 2) * 5 + 3) ** 2)
```

 Ausgabe zurücksetzen

1444  
1444

27. Die Rechnung soll zu gleichen Teilen auf die Anzahl der Gäste aufgeteilt werden.  
Ergänzen Sie das angezeigte Programm mit dem passenden Code.

```
sophia PYTHON IDE
Ausführen ? Hilfe
1 # Verfahren um eine Rechnung gleichmässig auf die Gäste zu verteilen.
2
3 # Eingabe von Total und Anzahl Gäste
4 total = float(input('Rechnungstotal: '))
5 anzahl = int(input('Anzahl Gäste: '))
6
7 # Dividiert zwei Zahlen
8 ...
9
10 # Anzeige des Resultats
11 print ("Macht pro Person: ",anteil)
12
13
```

**Lösung:**

```
sophia PYTHON IDE
Ausführen ? Hilfe
1 # Verfahren um eine Rechnung gleichmässig auf die Gäste zu verteilen.
2
3 # Eingabe von Total und Anzahl Gäste
4 total = float(input('Rechnungstotal: '))
5 anzahl = int(input('Anzahl Gäste: '))
6
7 # Dividiert zwei Zahlen
8 anteil = total / anzahl
9
10 # Anzeige des Resultats
11 print ("Macht pro Person: ",anteil)
12
13
```

funktioniert dann z.B. so

Explorer-Benutzereingabeaufforderung

Skript-Eingabeaufforderung:

Rechnungstotal:

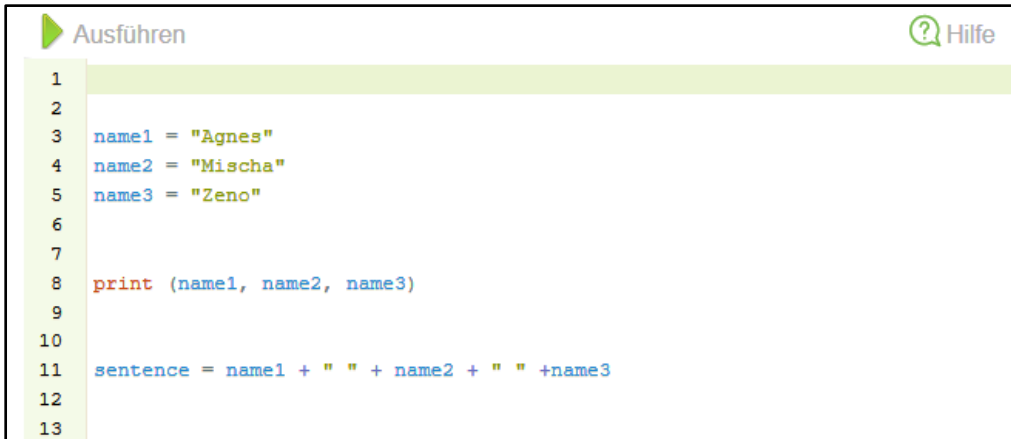
Explorer-Benutzereingabeaufforderung

Skript-Eingabeaufforderung:

Anzahl Gäste:

Macht pro Person: 20.0

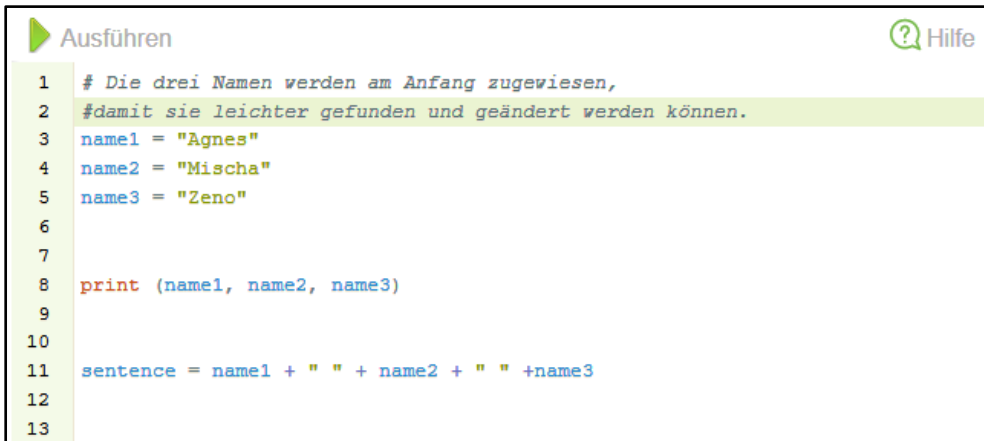
28. Fügen Sie unmittelbar vor der Zuweisung der drei Namen eine Kommentarzeile ein. Erklären Sie im Kommentar, warum Sie diese drei Namen gerade hier definieren.



```
1
2
3 name1 = "Agnes"
4 name2 = "Mischa"
5 name3 = "Zeno"
6
7
8 print (name1, name2, name3)
9
10
11 sentence = name1 + " " + name2 + " " + name3
12
13
```



**Lösung:**

Text ist nur ein Vorschlag – jeder x-beliebige Text kann eingegeben werden.



```
1 # Die drei Namen werden am Anfang zugewiesen,
2 #damit sie leichter gefunden und geändert werden können.
3 name1 = "Agnes"
4 name2 = "Mischa"
5 name3 = "Zeno"
6
7
8 print (name1, name2, name3)
9
10
11 sentence = name1 + " " + name2 + " " + name3
12
13
```



29. Ergänzen Sie das angezeigte Programm mit dem passenden Code unterhalb der Zeile  
**# Deklariert einen leeren String**  
Damit soll ein leerer String deklariert werden, der später mit Einträgen ergänzt werden kann.

```
1  # Code um einen leeren String zu initialisieren.
2  # Einem leeren String können später beliebige Ziffern hinzugefügt werden.
3
4
5  # Deklariert einen leeren String
6  ...
7
8  # Fügt Ziffern hinzu
9  for i in range(0,9) :
10     s=s+str(i)
11
12 # Druckt den String
13 print ("Der Inhalt des Strings: ",s)
```


**Lösung:**

s = ""

```
1  # Code um einen leeren String zu initialisieren.
2  # Einem leeren String können später beliebige Ziffern hinzugefügt werden.
3
4
5  # Deklariert einen leeren String
6  s = ""
7
8  # Fügt Ziffern hinzu
9  for i in range(0,9) :
10     s=s+str(i)
11
12 # Druckt den String
13 print ("Der Inhalt des Strings: ",s)
```


30. Deklarieren Sie eine Ganzzahlvariable **antwort** und weisen Sie ihr die Zahl 42 zu.

 Ausführen


```
1 # Ganzzahlvariable deklarieren und den Wert 42 zuweisen.  
2  
3 # Zuweisung der Antwort  
4 ...  
5  
6 print ("Die Antwort ist", antwort)  
7 print ("... aber was zum Teufel war die Frage ...")  
8
```

**Lösung:**

antwort = 42

 Ausführen

```
1 # Ganzzahlvariable deklarieren und den Wert 42 zuweisen.  
2  
3 # Zuweisung der Antwort  
4 antwort = 42  
5  
6 print ("Die Antwort ist", antwort)  
7 print ("... aber was zum Teufel war die Frage ...")  
8
```

 Ausgabe zurücksetzen

Die Antwort ist 42  
... aber was zum Teufel war die Frage ...

31. Verwenden Sie eine Ganzzahlvariable, um den Originalpreis in einen ganzzahligen Wert zu ändern (z.B. zur Ermittlung des Preises in einem 1,2,3 Euro/Franken Billigladen).

```
Ausführen
1  # Beliebige Zahl als Originalpreis
2  original_preis = 3.75
3
4  # Neuer Verkaufspreis als ganze Zahl zuweisen
5  ...
6
7  # Resultat ausgeben
8  print (discount_preis)
9
10
```

**Lösung:**

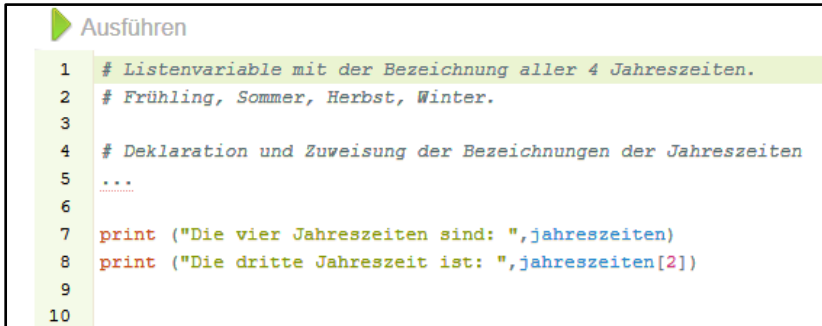
discount\_preis = int(original\_preis)

```
Ausführen
1  # Beliebige Zahl als Originalpreis
2  original_preis = 3.75
3
4  # Neuer Verkaufspreis als ganze Zahl zuweisen
5  discount_preis = int(original_preis)
6
7  # Resultat ausgeben
8  print (discount_preis)
9
10
```

## 32. Deklarieren Sie die Listen-Variable "jahreszeiten".

Sie soll die Bezeichnungen der vier Jahreszeiten Frühling, Sommer, Herbst, Winter als Elemente enthalten.

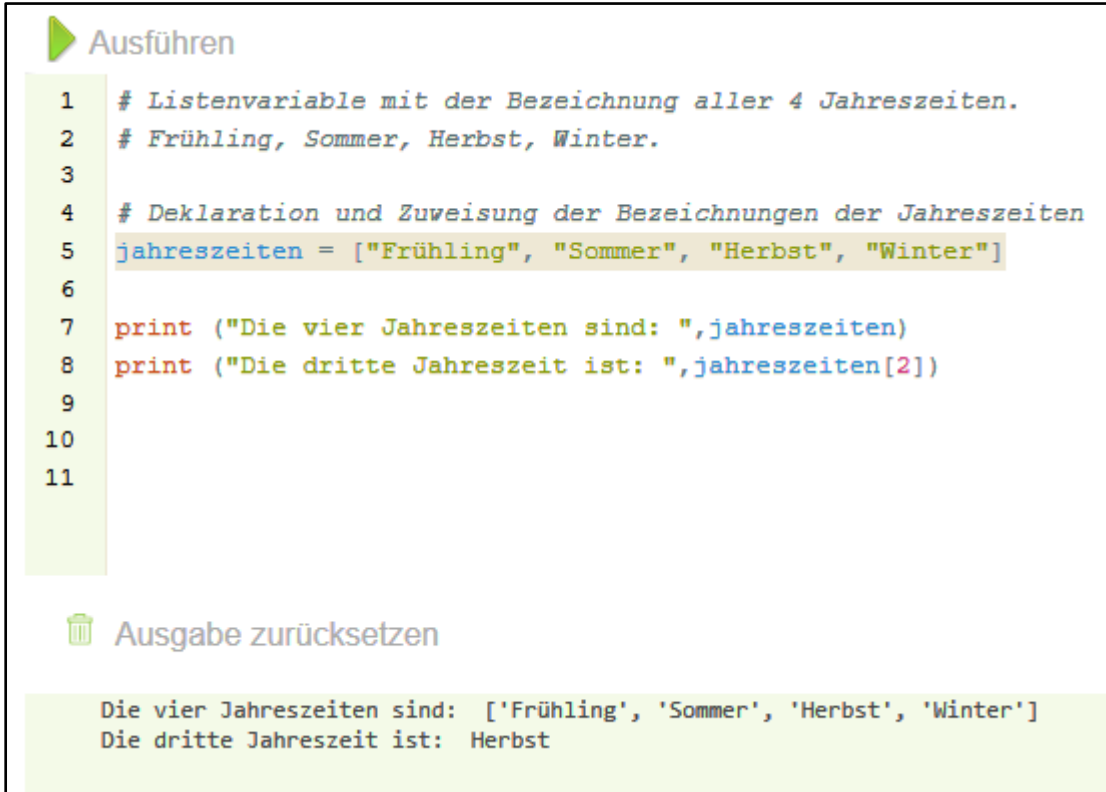
Das Programm soll am Ende die vier Jahreszeiten sowie die dritte Jahreszeit (Herbst) drucken.




```
1 # Listenvariable mit der Bezeichnung aller 4 Jahreszeiten.
2 # Frühling, Sommer, Herbst, Winter.
3
4 # Deklaration und Zuweisung der Bezeichnungen der Jahreszeiten
5 ...
6
7 print ("Die vier Jahreszeiten sind: ",jahreszeiten)
8 print ("Die dritte Jahreszeit ist: ",jahreszeiten[2])
9
10
```

**Lösung:**

```
jahreszeiten = ["Frühling", "Sommer", "Herbst", "Winter"]
```



```
1 # Listenvariable mit der Bezeichnung aller 4 Jahreszeiten.
2 # Frühling, Sommer, Herbst, Winter.
3
4 # Deklaration und Zuweisung der Bezeichnungen der Jahreszeiten
5 jahreszeiten = ["Frühling", "Sommer", "Herbst", "Winter"]
6
7 print ("Die vier Jahreszeiten sind: ",jahreszeiten)
8 print ("Die dritte Jahreszeit ist: ",jahreszeiten[2])
9
10
11
```

 Ausgabe zurücksetzen

```
Die vier Jahreszeiten sind: ['Frühling', 'Sommer', 'Herbst', 'Winter']
Die dritte Jahreszeit ist: Herbst
```

33. Ergänzen Sie das angezeigte Programm so, dass es den Benutzer zu Beginn nach seiner Herkunftsstadt und seinem Herkunftsland fragt.

```
Ausführen
1  # Programm fragt nach Herkunftsstadt und Herkunftsland
2
3  # Eingabe Anweisungen für Stadt und Land
4  ...
5  ...
6
7  herkunft = stadt + ', ' + land
8
9  print ("Herkunft des Benutzers:",herkunft)
10
11
```

**Lösung:**

```
Ausführen
1  # Programm fragt nach Herkunftsstadt und Herkunftsland
2
3  # Eingabe Anweisungen für Stadt und Land
4  stadt = input("Herkunftsstadt?")
5  land = input("Herkunftsland?")
6
7  herkunft = stadt + ', ' + land
8
9  print ("Herkunft des Benutzers:",herkunft)
10
11
```

funktioniert dann z.B. so

Explorer-Benutzereingabeaufforderung

Skript-Eingabeaufforderung:

Herkunftsstadt?

Vöcklabruck

OK Abbrechen

Explorer-Benutzereingabeaufforderung

Skript-Eingabeaufforderung:

Herkunftsland?

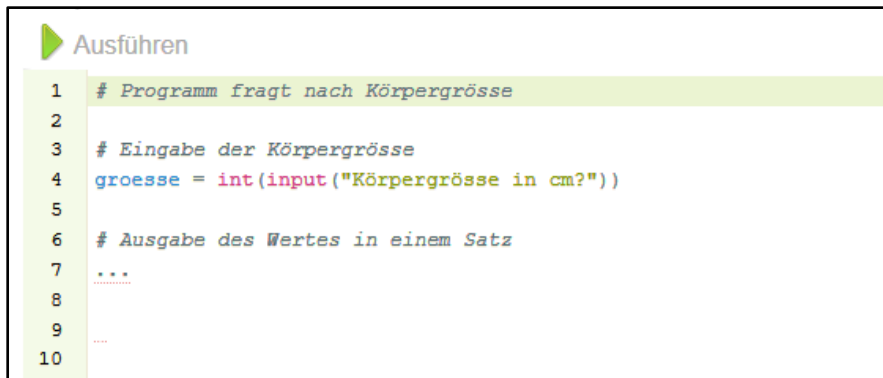
Österreich

OK Abbrechen

Herkunft des Benutzers: Vöcklabruck, Österreich



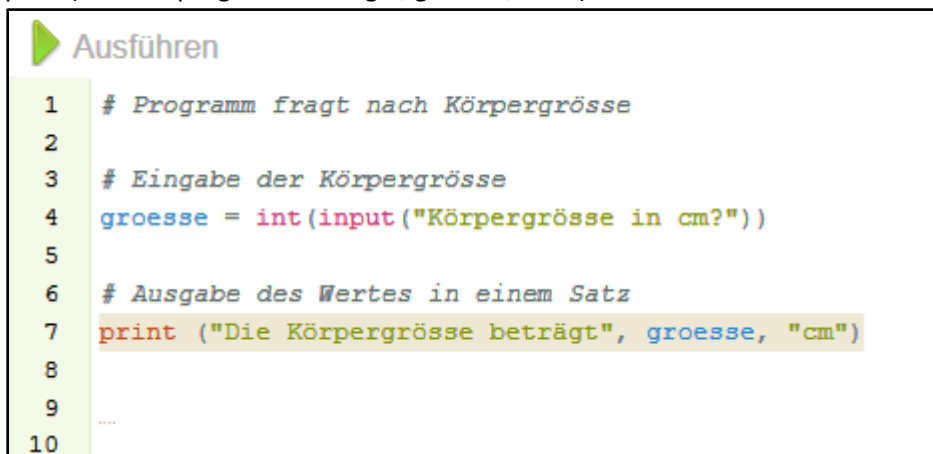
34. Ergänzen Sie das angezeigte Programm so, dass es am Schluss in einem Satz "Die Körpergröße beträgt...cm" ausgibt.



```
Ausführen
1 # Programm fragt nach Körpergröße
2
3 # Eingabe der Körpergröße
4 groesse = int(input("Körpergröße in cm?"))
5
6 # Ausgabe des Wertes in einem Satz
7 ...
8
9
10
```

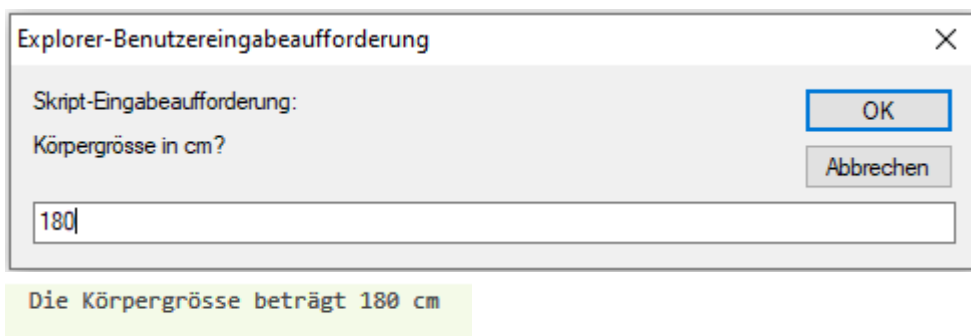
**Lösung:**

`print ("Die Körpergröße beträgt", groesse, "cm")`



```
Ausführen
1 # Programm fragt nach Körpergröße
2
3 # Eingabe der Körpergröße
4 groesse = int(input("Körpergröße in cm?"))
5
6 # Ausgabe des Wertes in einem Satz
7 print ("Die Körpergröße beträgt", groesse, "cm")
8
9
10
```

funktioniert dann z.B. so



35. Ein Rohrsystem enthält die Ventile V1, V2, V3 und V4.

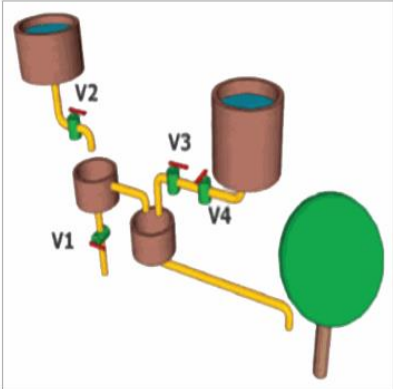
Die zu den Ventilen gehörenden Variablen v1 ... v4 können mit den Werten „wahr“ (Ventil offen) oder „falsch“ (Ventil geschlossen) belegt sein.

Ergänzen Sie das angezeigte Programm um einen Booleschen Ausdruck mit den Ventil-Variablen, der genau dann wahr ist, wenn der Baum Wasser kriegt.

Ausführen
? Hilfe

```

1 # Eingabe Ventil Stellungen
2 v1 = int(input("Ist das Ventil 1 offen, geben Sie 1 ein, sonst 0.")==1;
3 v2 = int(input("Ist das Ventil 2 offen, geben Sie 1 ein, sonst 0.")==1;
4 v3 = int(input("Ist das Ventil 3 offen, geben Sie 1 ein, sonst 0.")==1;
5 v4 = int(input("Ist das Ventil 4 offen, geben Sie 1 ein, sonst 0.")==1;
6
7 if ... :
8     print ("Baum erhält Wasser!")
9 else :
10    print ("Baum bleibt trocken!")
11
12
    
```

🗑️ Ausgabe zurücksetzen


### Lösung:

if not v1 or v3 and v4 :

Ausführen
? Hil

```

1 # Eingabe Ventil Stellungen
2 v1 = int(input("Ist das Ventil 1 offen, geben Sie 1 ein, sonst 0.")==1;
3 v2 = int(input("Ist das Ventil 2 offen, geben Sie 1 ein, sonst 0.")==1;
4 v3 = int(input("Ist das Ventil 3 offen, geben Sie 1 ein, sonst 0.")==1;
5 v4 = int(input("Ist das Ventil 4 offen, geben Sie 1 ein, sonst 0.")==1;
6
7 if not v1 or v3 and v4 :
8     print ("Baum erhält Wasser!")
9 else :
10    print ("Baum bleibt trocken!")
11
    
```

funktioniert dann z.B. so

Explorer-Benutzereingabeaufforderung
✕

Skript-Eingabeaufforderung:

Ist das Ventil 1 offen, geben Sie 1 ein, sonst 0.

1|

OK

Abbrechen

Explorer-Benutzereingabeaufforderung

Skript-Eingabeaufforderung:

Ist das Ventil 2 offen, geben Sie 1 ein, sonst 0.

OK

Abbrechen

2

Explorer-Benutzereingabeaufforderung

Skript-Eingabeaufforderung:

Ist das Ventil 3 offen, geben Sie 1 ein, sonst 0.

OK

Abbrechen

0

Explorer-Benutzereingabeaufforderung

Skript-Eingabeaufforderung:

Ist das Ventil 4 offen, geben Sie 1 ein, sonst 0.

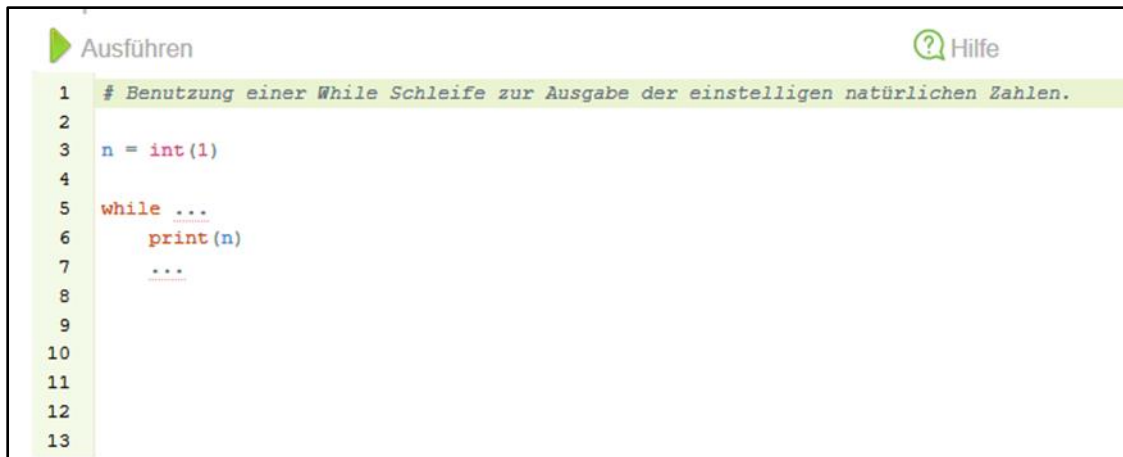
OK

Abbrechen

1

Baum bleibt trocken!


36. Geben Sie durch einen Befehl alte einstelligen natürlichen Zahlen (ganze Zahlen von eins weg) unter Verwendung einer While Schleife auf der Konsole aus.



A Python IDE snippet with a light green background. At the top, there is a green play button icon followed by the text 'Ausführen' and a green speech bubble icon with a question mark followed by 'Hilfe'. The code is as follows:

```
1  # Benutzung einer While Schleife zur Ausgabe der einstelligen natürlichen Zahlen.
2
3  n = int(1)
4
5  while ...
6      print(n)
7      ...
8
9
10
11
12
13
```

**Lösung:**



A Python IDE snippet with a light green background. At the top, there is a green play button icon followed by the text 'Ausführen' and a green speech bubble icon with a question mark followed by 'Hilfe'. The code is as follows:

```
1  # Benutzung einer While Schleife zur Ausgabe der einstelligen natürlichen Zahlen.
2
3  n = int(1)
4
5  while n < 10:
6      print(n)
7      n+= 1
8
9
10
11
12
13
```

Below the code editor, there is a trash can icon followed by the text 'Ausgabe zurücksetzen'.

The output area at the bottom shows the numbers 1 through 9 printed on separate lines:

```
1
2
3
4
5
6
7
8
9
```

37. Verwenden Sie eine **if else** Anweisung, um das Resultat einer Prüfung zu testen.  
Das Programm soll eine Meldung ausgeben mit "**Bestanden**" für Werte mit 75 oder mehr Punkten und sonst "**Nicht bestanden**".

```
Ausführen Hilfe
1  # Programm zur Bestimmung und Ausgabe des Prüfungsergebnisses je nach Punktzahl
2  # Bestanden mit 75 Punkten und höher.
3
4  # Eingabe der Punktzahl
5  punkte = int(input("Bitte Punktzahl eingeben:"))
6
7
8  if ...
9
10 print ("Bestanden - Herzliche Gratulation!")
11 print ("Nicht Bestanden - Prüfung Wiederholen!")
12
13
```

**Lösung:**

**Wichtig:** die beiden print-Zeilen müssen eingerückt werden

```
Ausführen Hilfe
1  # Programm zur Bestimmung und Ausgabe des Prüfungsergebnisses je nach Punktzahl
2  # Bestanden mit 75 Punkten und höher.
3
4  # Eingabe der Punktzahl
5  punkte = int(input("Bitte Punktzahl eingeben:"))
6
7
8  if punkte >= 75:
9      print ("Bestanden - Herzliche Gratulation!")
10 else:
11     print ("Nicht Bestanden - Prüfung Wiederholen!")
12
13
```

funktioniert dann z.B. so

Explorer-Benutzereingabeaufforderung

Skript-Eingabeaufforderung:

Bitte Punktzahl eingeben:

80

OK

Abbrechen

Bestanden - Herzliche Gratulation!

38. Bei der Planung von Informatikmitteln wird mit Zweierpotenzen d.h. ( $2^n$ ) gerechnet. Schreiben Sie eine Funktion `pot2(n)` und verwenden Sie diese, um die Zweierpotenzen von 1, 2, 4, 8, 10, 16, 20, 30, 32, 40, 48 auszugeben.

Ausführen Hilfe

```

1  # Programm berechnet die Zweierpotenzen in Funktion pot2(n)=2**n
2
3  # Definition einer Funktion pot2(n) zur Bestimmung der Anzahl Paarungen mit der genannte
4  ...
5
6  # Auflistung der "interessanten" Zweierpotenzen
7  for i in (1,2,4,8,10,16,20,30,32,40,48) :
8      pot2(i)
9  # Was ist daran interessant?
10 # 2**30 ist die genaue Zahl von Bytes im Gigabyte
11 # 2**40 ist die genaue Zahl von Bytes im Terabyte
12 # 2**48 ist die maximale Anzahl möglicher Netzwerkgeräte mit MAC Kennung.
13

```

**Lösung:**

Ausführen Hilfe

```

1  # Programm berechnet die Zweierpotenzen in Funktion pot2(n)=2**n
2
3  # Definition einer Funktion pot2(n) zur Bestimmung der Anzahl Paarungen mit der genannte
4  def pot2(n):
5      print (2**n)
6  # Auflistung der "interessanten" Zweierpotenzen
7  for i in (1,2,4,8,10,16,20,30,32,40,48) :
8      pot2(i)
9  # Was ist daran interessant?
10 # 2**30 ist die genaue Zahl von Bytes im Gigabyte
11 # 2**40 ist die genaue Zahl von Bytes im Terabyte
12 # 2**48 ist die maximale Anzahl möglicher Netzwerkgeräte mit MAC Kennung.
13

```

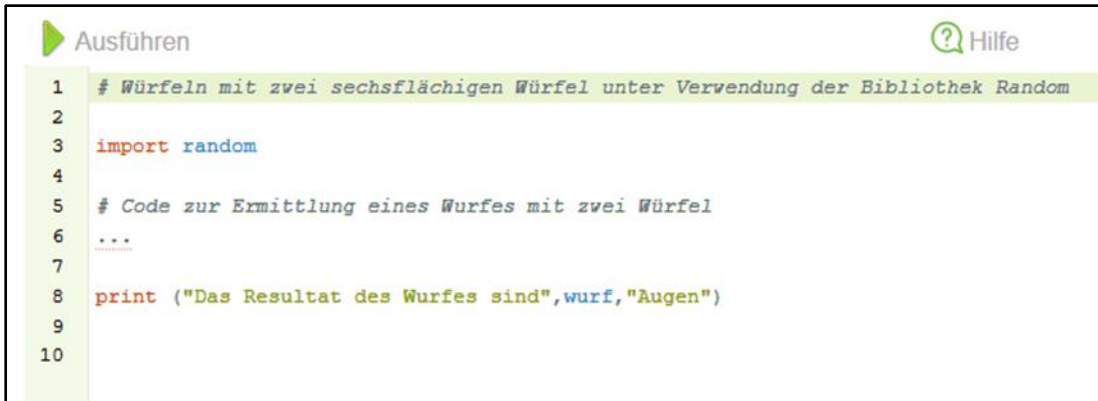
Ausgabe zurücksetzen

```

2
4
16
256
1024
65536
1048576
1073741824
4294967296
1099511627776
281474976710656

```

39. Ergänzen Sie das angezeigte Programm zum Würfeln mit zwei Würfeln so, dass es mit der Bibliothek „Random“ die Anzahl der Augen zweier Würfel berechnet.

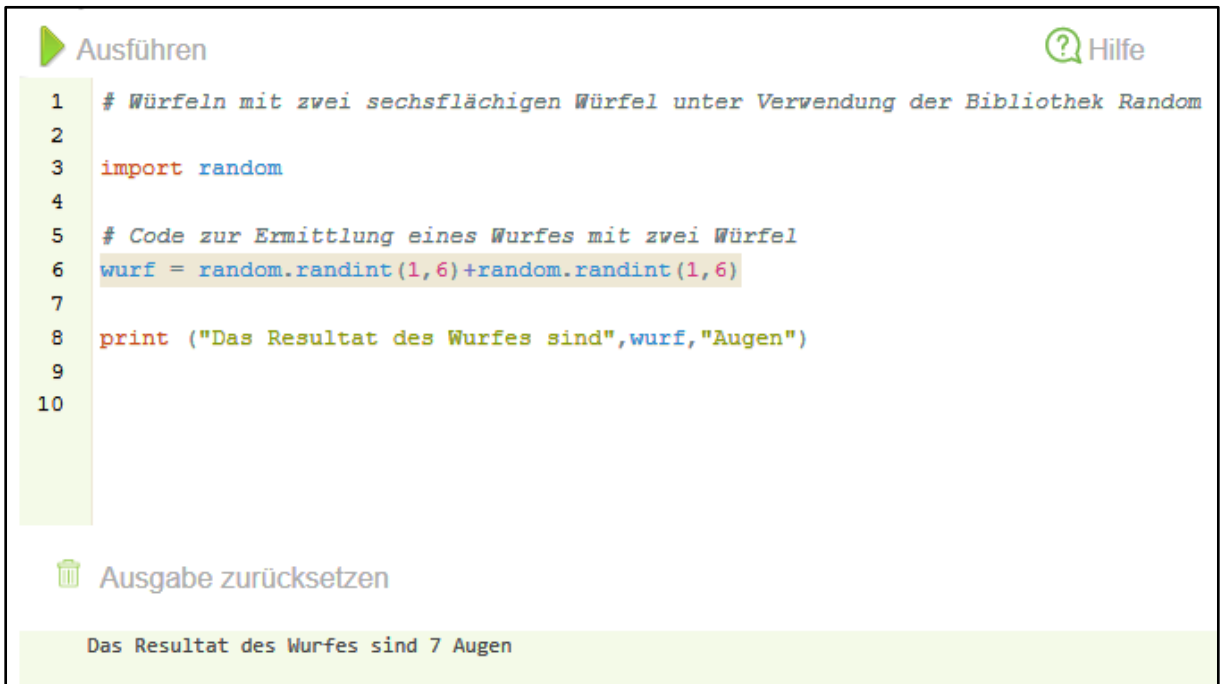


The image shows a Python code editor window titled "Ausführen" (Execute) with a "Hilfe" (Help) icon. The code is as follows:

```
1 # Würfeln mit zwei sechsflächigen Würfel unter Verwendung der Bibliothek Random
2
3 import random
4
5 # Code zur Ermittlung eines Wurfes mit zwei Würfel
6 ...
7
8 print ("Das Resultat des Wurfes sind",wurf,"Augen")
9
10
```

**Lösung:**

wurf = random.randint(1,6)+random.randint(1,6)



The image shows the same Python code editor window, but now with the completed code and the output. The code is:

```
1 # Würfeln mit zwei sechsflächigen Würfel unter Verwendung der Bibliothek Random
2
3 import random
4
5 # Code zur Ermittlung eines Wurfes mit zwei Würfel
6 wurf = random.randint(1,6)+random.randint(1,6)
7
8 print ("Das Resultat des Wurfes sind",wurf,"Augen")
9
10
```

Below the code, there is a button labeled "Ausgabe zurücksetzen" (Reset output) with a trash icon. At the bottom, the output of the program is displayed: "Das Resultat des Wurfes sind 7 Augen".

40. Das angezeigte Programm enthält den ursprünglichen Algorithmus von Euklid zur Bestimmung des größten gemeinsamen Teilers.  
Setzen sie einen Haltepunkt vor die Zeile 5 und bestimmen Sie mit der wiederholten Ausführung zum Haltepunkt die Werte von a und b In Jedem Schleifendurchgang.  
Tragen Sie die beobachteten Werte in die Tupel Liste am Schluss des Programmes ein.

```

1  # Ermittlung des größten Gemeinsamen Teilers nach Euklid
2
3  a,b=15,21
4
5  while a != b :
6      if a > b :
7          a = a - b
8      else :
9          b = b - a
10
11 print (a)
12
13 # Werte von (a,b) bei jedem Schleifendurchgang als Antwort hier eintragen
14 werte=((15,21),...)
15

```

**Lösung:**

werte=((15,21),(15,6),(9,6),(3,6),(3,3))

```

3  a,b=15,21
4
5  while a != b :
6      if a > b :
7          a = a - b
8      else :
9          b = b - a
10
11 print (a)
12
13 # Werte von (a,b) bei jedem Schleifendurchgang als Antwort hier eintragen
14 werte=((15,21), (15,6), (9,6), (3,6), (3,3))
15

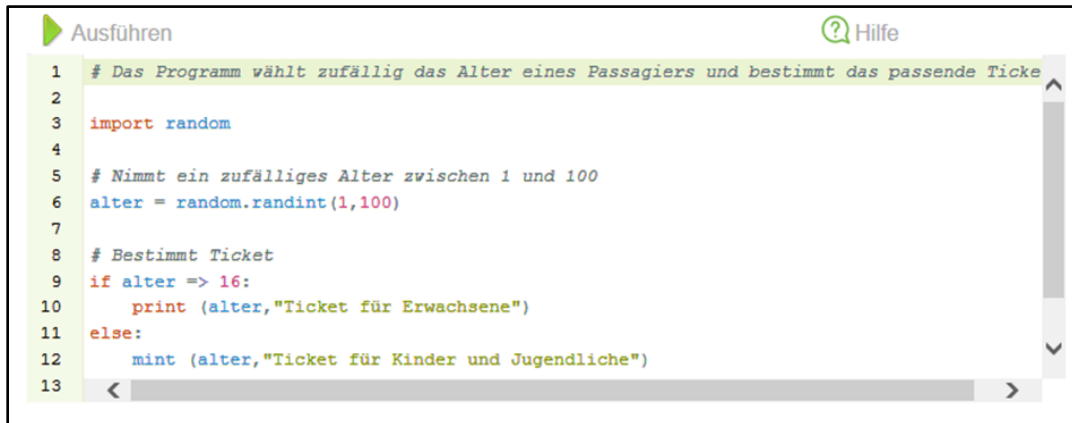
```

Ausgabe zurücksetzen

3



41. Das gezeigte fehlerhafte Programm soll aus einem zufällig generierten Alter einer Person das richtige Ticket bestimmen.  
Personen bis und mit 15 Jahren erhalten ein Kinder-Ticket.  
Beseitigen Sie den Schreibfehler und den Syntax-Fehler aus dem Programm, damit es sich ausführen lässt.

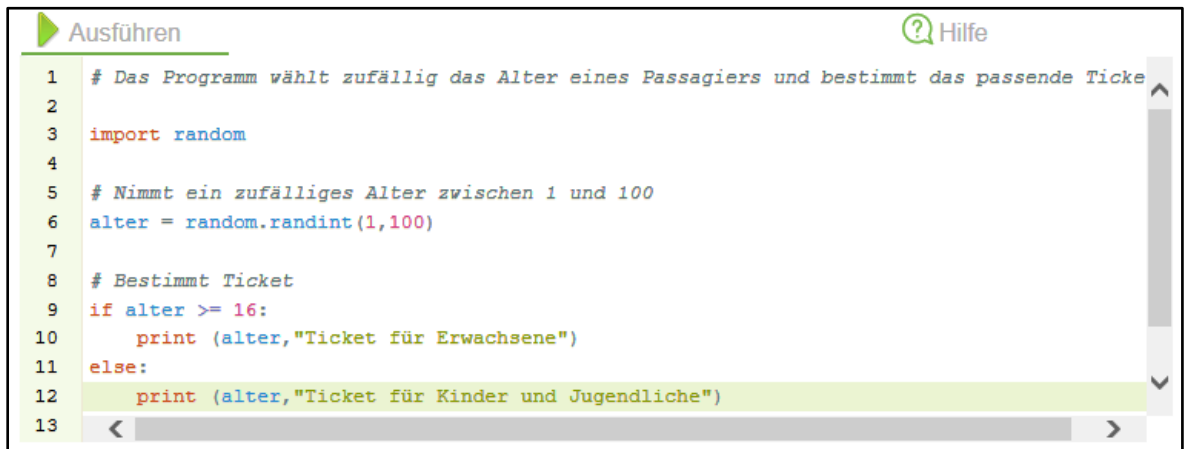


```
1 # Das Programm wählt zufällig das Alter eines Passagiers und bestimmt das passende Ticket
2
3 import random
4
5 # Nimmt ein zufälliges Alter zwischen 1 und 100
6 alter = random.randint(1,100)
7
8 # Bestimmt Ticket
9 if alter => 16:
10     print (alter,"Ticket für Erwachsene")
11 else:
12     mint (alter,"Ticket für Kinder und Jugendliche")
13
```

**Lösung:**

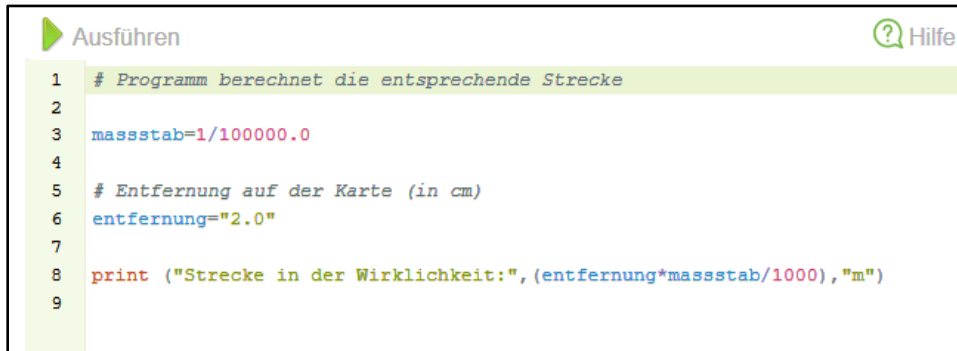
Die Schreibfehler befindet sich in

- Zeile 9 (`=> 16:`) richtig ist: `>= 16:`
- Zeile 12 (`mint`) richtig ist: `print`



```
1 # Das Programm wählt zufällig das Alter eines Passagiers und bestimmt das passende Ticket
2
3 import random
4
5 # Nimmt ein zufälliges Alter zwischen 1 und 100
6 alter = random.randint(1,100)
7
8 # Bestimmt Ticket
9 if alter >= 16:
10     print (alter,"Ticket für Erwachsene")
11 else:
12     print (alter,"Ticket für Kinder und Jugendliche")
13
```

42. Auf einer Karte im Maßstab 1:100 000 soll das Programm aus einer angezeigten Entfernung (z. B. 2 cm) die entsprechende Strecke in der Wirklichkeit (z. B. 200 m) berechnen. Beseitigen Sie den Fehler in der Verwendung der Datentypen sowie den mathematischen Fehler, um am Ende das richtige Resultat von 200 m zu erhalten.

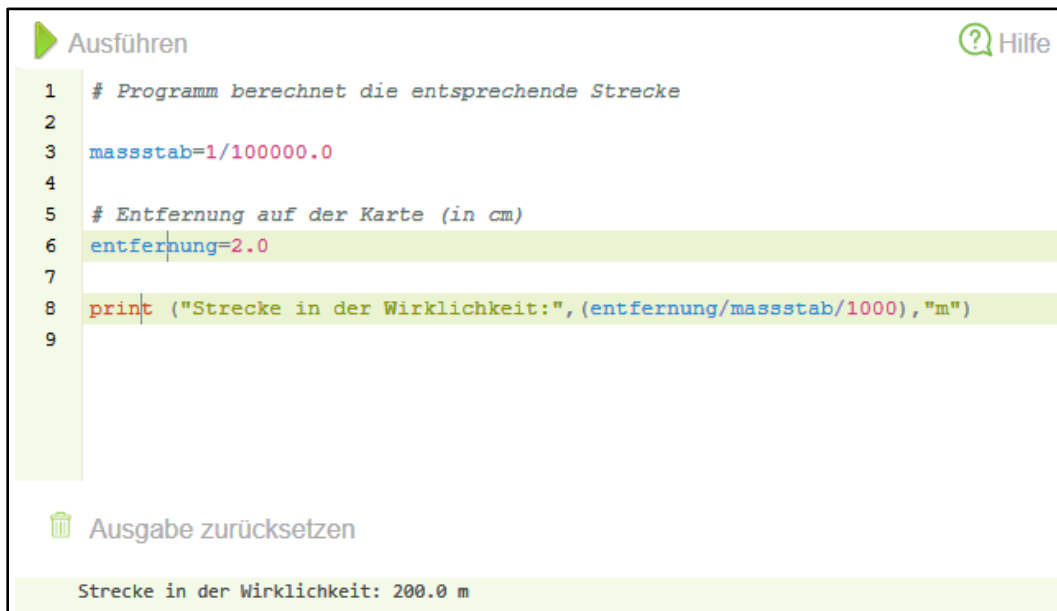


```
1 # Programm berechnet die entsprechende Strecke
2
3 massstab=1/100000.0
4
5 # Entfernung auf der Karte (in cm)
6 entfernung="2.0"
7
8 print ("Strecke in der Wirklichkeit:", (entfernung*massstab/1000), "m")
9
```

**Lösung:**

Die Fehler befinden sich

- in Zeile 6 ("2.0") – richtig ist **2,0** ohne Anführungszeichen, da Zahl und kein Text
- in Zeile 8 (entfernung\*massstab) -richtig ist **entfernung/massstab**



```
1 # Programm berechnet die entsprechende Strecke
2
3 massstab=1/100000.0
4
5 # Entfernung auf der Karte (in cm)
6 entfernung=2.0
7
8 print ("Strecke in der Wirklichkeit:", (entfernung/massstab/1000), "m")
9
```

Ausgabe zurücksetzen

Strecke in der Wirklichkeit: 200.0 m