

## Programmieraufgaben für Python

1. <b>Hello World!</b> Schreibe ein Programm, dass auf der Konsole den Text „Hello World!“ ausgibt. <i>Beachte jeweils die Fußzeilen für weitere Erklärungen.</i>	Textausgabe auf Bildschirm <sup>1</sup>
2. <b>Netto-Brutto-Rechner:</b> Schreibe ein Programm, bei dem der Nutzer einen Preis ohne Steuer eingibt. Der Rechner gibt den Preis mit 19% Mehrwertsteuer aus.	Eingabe über Tastatur <sup>2</sup> , Textausgabe auf Bildschirm
3. <b>Zinsberechnung:</b> Der Nutzer soll einen Geldbetrag und einen Zinssatz eingeben können. Ausgabe sollen die Zinsen für ein Jahr sein.	Division <sup>3</sup>
4. <b>Rabattrechner:</b> Eingabe soll der Normalpreis sein und wie viel Prozent Rabatt es auf den Artikel gibt. Ausgabe soll der Endpreis sein.	
5. <b>Begrüßung:</b> Schreibe ein Programm, das den Nutzer nach seinem Namen fragt und ihn dann mit seinem Namen begrüßt, z.B. „Hallo Johanna! Wie geht's dir?“	Textausgabe auf Bildschirm
6. <b>Quader:</b> Schreibe ein Programm, bei dem der Nutzer Breite, Länge und Höhe eines Quaders angibt. Das Programm berechnet dann das <u>Volumen</u> , die Oberfläche und die Raumdiagonale ( $\sqrt{a^2+b^2+c^2}$ )	Variablendeklaration, Tastatureingabe, Rechnungen <sup>4</sup> , Wurzel <sup>5</sup> )

1 **Ausgabe auf dem Bildschirm:** print „Zeichenkette“, z.B. print „Hallo“

**Probleme mit Umlauten?** Als erste Zeile `# -*- coding: utf-8 -*-` einfügen.

Will man mehrere Dinge in einer Zeile ausgeben, muss man diese mit „+“ aneinander hängen. Z.B. `print „Das Ergebnis beträgt“ + str(x)` Die Funktion `str()` wandelt Zahlen in Zeichenketten um.

2 **Eingabe über Tastatur:** Die Eingabe über die Tastatur erfolgt mit der Zuweisung `netto=input(„Geben Sie den Preis ohne Steuer ein:“)` Achtung: Eingaben mit `input` werden immer als Zahlen interpretiert. Will man das nicht, muss man `raw_input()` benutzen!

3 Teilt man z.B. `x / 100` und `x` ist eine ganze Zahl, dann wird eine ganzzahlige Division ausgeführt. `507 / 100` wäre dann `5`. Um so zu dividieren, dass eine Kommazahl heraus kommt, muss eine der beiden Zahlen eine Kommazahl sein. Also `x / 100.0`. Sind beides Variablen, so wäre `x * 1.0 / y` eine Lösung, um den Rechner zur Typumwandlung zu zwingen.

4 **Berechnungen:** Will man die Inhalte der Variablen `a`, `b`, `c` miteinander multiplizieren und das Ergebnis in der Variablen namens `volumen` speichern, ist der Befehl hierfür `volumen= a* b*c`

5 **Quadratwurzel:** Will man aus `x` die Wurzel ziehen, so rechnet man `x**(0.5)`. Ein doppelt Malzeichen steht also für „hoch“.

7. <b>Quaderberechnung abändern:</b> Ändere das Programm Quaderberechnung so ab, dass es den Nutzer darauf hinweist, falls er negative Eingaben macht, dass dies nicht sein kann.	Verzweigung <sup>6</sup>
8. <b>Rabattrechner abändern:</b> Weise den Nutzer darauf hin, wenn er 100 oder mehr Prozent eingibt, dass dies nicht sein kann.	
9. <b>Rückgeld:</b> Schreibe ein Programm, bei dem man eingibt, wie viel jemand zahlen muss und wie viel er in bar gezahlt hat. Es soll ausgegeben werden, wie viel Rückgeld der Kunde bekommt. Falls er zu wenig bezahlt hat, soll das Programm eine Warnung ausgeben.	
10. Das Programm soll eine <b>quadratische Gleichung</b> der Form $ax^2+bx+c=0$ lösen können. Der Nutzer gibt hierfür die Zahlen a, b, c ein. Das Programm prüft die Diskriminante, also $b^2-4ac$ , ob diese positiv, null oder negativ ist und gibt dann aus wie viele Lösungen es gibt und ggf. deren Werte.	Tastatureingabe, Variablendeklaration, Rechnungen, Wurzel, Verzweigung <sup>7</sup>
11. <b>BMI-Rechner:</b> Eingabe: Körpergröße, Körpergewicht und Geschlecht. Ausgabe: BMI und Hinweis, ob Unter-, Normal- oder Übergewicht. (Bei Frauen ist ein BMI von 19 bis 24 normal, bei Männern 20 bis 25.)	Deklaration, Tastatureingabe, Rechnung, Verzweigung
12. <b>Mittelwert:</b> Schreibe ein Programm, bei dem man 4 Zahlen eingeben kann. Das Programm berechnet den Mittelwert.	
13. <b>Wahlprogramm:</b> Schreibe ein Programm, bei dem man für drei Kandidaten Namen eingeben kann und die Anzahl der erhaltenen Stimmen. Das Programm gibt die prozentualen Anteile der Stimmen aus.	
14. <b>Maklerprogramm:</b> Eingabe Grundstückslänge und -Breite, Quadratmeterpreis, Maklerprovision in %, aktueller Umsatzsteuersatz in %. Ausgabe: Grundstückspreis und Maklerprovision.	
15. <b>Benzin-Rechner:</b> Eingabe: Literpreis Benzin, verbrauchte Liter, gefahrene Kilometer. Ausgabe: Verbrauch pro 100 km, Preis pro 100 km, Preis der Gesamtstrecke.	Deklaration, Tastatureingabe, Rechnung
16. <b>Kopfrechnen:</b> Schreibe ein Programm, das zwei Zufallszahlen	Zufallszahlen erzeugen <sup>8</sup>

6 **Verzweigung:** Soll das Programm prüfen, ob eine Bedingung erfüllt ist, z.B. ob der Wert der Variablen namens **c** negativ ist, dann ist die Python-Syntax folgende: **if d<0:** Eine Zeile tiefer kommen dann **ingerückt** die Zeilen, die nur dann aus geführt werden sollen, falls die Bedingung wahr ist. **Doppelpunkt** nach der Bedingung nicht vergessen!

7 **Verzweigung:** Soll das Programm prüfen, ob eine Bedingung erfüllt ist, z.B. ob der Wert der Variablen namens **diskriminte** negativ ist, dann ist die Python-Syntax folgende: **if Bedingung : Anweisungen (ingerückt)** also z.B. **if diskriminante < 0: print „Es gibt keine Lösung.“**

erzeugt und den Nutzer auffordert diese zu addieren. Falls der Nutzer richtig gerechnet hat, soll dieser gelobt werden.	
17. <b>Potenzrechnung:</b> Schreibe ein Programm, bei dem der Nutzer zwei Zahlen a und b eingeben kann, b ganzzahlig. Das Programm berechnet die Potenz $a^b$ .	For-Schleife <sup>9</sup>
18. <b>Fakultät berechnen:</b> In der Wahrscheinlichkeitsrechnung werdet ihr später das Produkt aller natürlichen Zahlen bis n benötigen „n!“, sprich „n Fakultät“. Z.B. $3! = 1 \cdot 2 \cdot 3 = 6$ und $5! = 1 \cdot 2 \cdot 3 \cdot 4 \cdot 5 = 120$ . Schreibe ein Programm, bei dem man n eingibt und das n! ausgibt.	
19. <b>Größte Zweierpotenz:</b> Schreibe ein Programm, bei dem der Nutzer eine natürliche Zahl eingibt und das Programm gibt die größte Zweierpotenz zurück, die kleiner oder gleich der eingegebenen Zahl ist.	While-Schleife <sup>10</sup>
20. Schreibe ein Programm, das mithilfe einer Schleife n Zeilen mit 10 Sternen schreibt: ***** ***** *****	For-Schleife, print ohne Zeilenumbruch <sup>11</sup>
21. Schreibe ein Programm, das mithilfe von zwei Schleifen n Zeilen mit je m Sternen schreibt. (Nutzer gibt n und m ein.)	geschachtelte For-Schleife
22. <b>Halber Weihnachtsbaum:</b> Schreibe ein Programm das einen „Weihnachtsbaum“ der Höhe n ausgibt. Eingabe: n. * ** *** ****	
23. Schreibe ein Programm, das einen ganzen Weihnachtsbaum ausgibt:	

8 Dazu muss die Funktion **randint** aus der **random**-Bibliothek importiert werden: **from random import randint**. Danach kann man mit **randint(1, 20)** eine Zufallszahl zwischen 1 und 20 erzeugen.

9 For-Schleife: Soll ein Block von Anweisungen wiederholt ausgeführt werden, und steht im Voraus fest, wie oft, benutzt man dafür eine For-Schleife: **for i in range(1, 11): print str(i)** würde alle Zahlen von 1 bis 10 aufschreiben. Beachte: Als letzte Zahl muss man immer eins größer angeben als der letzte Schleifendurchlauf sein soll!

10 Syntax der While-Schleife: **while Bedingung: Anweisungen eingerückt.**

11 Will man nach einem **print**-Befehl keinen Zeilenumbruch, muss man danach ein Komma setzen: **print „\*“,**

<pre> * *** ***** ***** # </pre>	
<p>24. <b>BugTrouble:</b> Wir gehen davon aus, dass ein sehr vermehrungsfreudiger Käfer ein Volumen von <math>0.0002m^3</math> hat und sich pro Woche um 95% vermehrt. Nun ist ein Programm zu schreiben, in das der Nutzer die momentan vorhandene Anzahl Käfer sowie das Volumen des Hauses eingibt. Das Programm soll dann für Woche 1, Woche 2 usw. die Anzahl der vorhandenen Käfer sowie deren Gesamtvolumen angeben – für so viele Wochen, bis das Haus komplett gefüllt ist.</p>	Tastatureingabe, Rechnungen, While-Schleife
<p>25. <b>Kopfrechnen II:</b> Verändere das Kopfrechnenprogramm so, dass es nicht nur eine Additionsaufgabe stellt, sondern 10. Es soll mitzählen, wie viele Aufgaben der Nutzer richtig hatte.</p>	
<p>26. Schreibe ein Programm, dass auf Wunsch des Nutzers eine Multiplikationsreihe ausgibt. Der Nutzer soll angeben können, welche Reihe, z.B. die ger-Reihe, also <math>1*9=9</math>; <math>2*9=18</math>; <math>3*9=27</math> usw. Der Nutzer soll außerdem angeben können, wie weit die Reihe gehen soll.</p>	For-Schleife
<p>27. <b>Fibonacci-Zahlen:</b> Schreibe ein Programm, das die Fibonaccizahlen bis n (Eingabe) ausgibt: 0 1 1 2 3 5 8 13 21 34 55 89 144 usw.  Die nächste Zahl ergibt sich immer als Summe der beiden vorangegangenen.</p>	While-Schleife
<p>28. <b>Rückgeld:</b> Schreibe ein Konsolenprogramm, bei dem der Nutzer einen Geldbetrag mit Euro und Cent eingibt und der Rechner ausgibt mit wie vielen 50-Euro-Scheinen, 20-Euro-Scheinen, 10-Euro-Scheinen usw. bis wie viele 1-Cent-Stücke,</p> <pre> Geben Sie die Anzahl Euros gefolgt von der Anzahl Cent: 89 30 Sie müssen 89 Euro 30 zurückgeben. Hundert-Euro-Scheine: 0. Bleiben 89 Euro. Fünfundzwanzig-Euro-Scheine: 1. Bleiben 39 Euro. Zwanzig-Euro-Scheine: 1. Bleiben 19 Euro. Zehn-Euro-Scheine: 1. Bleiben 9 Euro. Fünf-Euro-Scheine: 1 Zwei-Euro-Stücke: 2 Ein-Euro-Stücke: 0 </pre> <p>dieser Betrag ausbezahlt wird.</p>	Ganzzahlige Division <sup>12</sup>

<sup>12</sup> **Ganzzahlige Division:** Sind die Variablen **euro** und **anzahl** beide ganze Zahlen, führt der Befehl **anzahl = euro/50** eine ganzzahlige Division durch, d.h. in **anzahl** wird gespeichert, wie oft die 50 ganz in euro hineinpasst. Z.B. **96/50** gibt 1. Den Rest erhält man mit **96%50**, also hier 46.

<p>29. Zeitemrechnung: Eingabe: Anzahl Sekunden; Ausgabe: Anzahl Tage, Stunden, Minuten, Sekunden.</p> <pre>Geben Sie die Anzahl der Sekunden ein: 216512 Das sind 2 Tage, 12 Stunden, 8Minuten und 32 Sekunden.</pre>	<p>Ganzzahlige Division</p>
<p>30. Kopfrechnen (nur Addition): Das Programm stellt 10 Kopfrechenaufgaben. Das Programm erzeugt hierfür je zwei Zufallszahlen, die der Nutzer im Kopf addieren soll. Das Programm zählt, wie viele Ergebnisse der Nutzer richtig hat und zeigt dies an.</p>	<p>Zufallszahlen erzeugen</p> <pre>Addition üben: Aufgabe 1: 26+10=36      Prima! Aufgabe 2: 22+30=52      Prima! Aufgabe 3: 30+18=48      Prima! Aufgabe 4: 20+9=29       Prima! Aufgabe 5: 17+26=43      Prima! Aufgabe 6: 17+13=30      Prima! Aufgabe 7: 1+21=22       Prima! Aufgabe 8: 5+3=8         Prima! Aufgabe 9: 11+11=22      Prima! Aufgabe 10: 14+6=1       Leider falsch! Richtig gewesen wäre: 20 9 von 10 Aufgaben richtig.</pre>
<p>31. <b>Kopfrechnen advanced:</b> Erweitere das Programm so, dass der Nutzer zwischen +, -, *, : wählen kann.</p>	<p>For-Schleife, Verzweigung</p>