

Python Aufgaben

1. Variablen, Berechnungen und Schleifen

- Schreibe ein Programm, das zwei Zahlen addiert und das Ergebnis ausgibt. Verwende zwei Variablen, um die Zahlen zu speichern.
- Schreibe ein Python-Skript, das folgende Aufgaben löst:
 - Definiere eine Variable `name` und setze sie auf deinen Namen.
 - Definiere eine Variable `age` und setze sie auf dein Alter.
 - Gib eine Ausgabe in folgendem Format aus (statt den eckigen Klammern bitte den Wert der Variablen eintragen ;) :
`Mein Name ist [name] und ich bin [age] Jahre alt.`
- Schreibe ein Programm, das die Anzahl an Sekunden von 36 Minuten berechnet.
- Schreibe ein Programm, das einen Countdown von 5 bis 1 ausführt, die jeweilige Zahl und am Ende "Start!" ausgibt, mögliche Ausgabe: 5 4 3 2 1 Start! (es kann auch auf mehreren Zeilen ausgegeben werden).

2. Sequenzen

- Verwende die Indizierung, um den Wert `"finde_mich"` aus der folgenden Liste zu extrahieren:
`my_list = [1, 2, 3, [4, 5], [6, ["finde_mich"], 7], 8]`
Hinweis: kopiere die Liste aus der Aufgabenstellung und füge sie in deinen Code ein.
- Verwende die Schlüsselauswahl, um den Wert `"finde_mich_dict"` aus dem folgenden Dictionary zu extrahieren:

```
my_dict = {  
    "a": 1,  
    "b": {  
        "c": 2,  
        "d": {  
            "e": "finde_mich_dict"  
        }  
    }  
}
```

```

    },
    "f": 3
}

```

- Erstelle eine Liste der Zahlen von 1 bis 10 und berechne die Summe aller Zahlen in dieser Liste und gib sie aus.
- Erstelle eine Liste mit den Zahlen [3, 7, 2, 9, 5]. Finde die kleinste Zahl in der Liste und gib sie aus.
- Schreibe ein Programm, das eine Liste mit den Zahlen 1 bis 10 erzeugt und dann das Produkt aller **geraden** (durch zwei teilbaren) Zahlen in der Liste berechnet und dieses ausgibt.
- Teste für jedes Element in der Liste `my_list2 = ["ein", "beispiel", "ring"]` ob es im String `s = "MeinBeispielString"` vorhanden ist und gib das Element und das Ergebnis des Tests für jedes Element aus.
- Erzeuge aus dem String `Eins Zwei Drei` eine Liste mit den 3 Wörtern, die im String durch Leerzeichen getrennt sind und gib sie aus.
- Gegeben ist eine Liste von Wörtern:
`words = ["apple", "banana", "apple", "orange", "banana", "apple"]`
 Schreibe ein Programm, das die Anzahl der Vorkommen jedes Wortes in der Liste zählt und das Ergebnis als Dictionary ausgibt.
 Beispielausgabe:
`{"apple": 3, "banana": 2, "orange": 1}`

3. Funktionen

- Schreibe eine Funktion `fizzbuzz(n)`, die die Zahlen von 1 bis `n` durchgeht und dabei Folgendes ausgibt:
 - Für Vielfache von 3 und 5 gib "FizzBuzz" aus.
 - Für Vielfache von 3 gib "Fizz" aus.
 - Für Vielfache von 5 gib "Buzz" aus.
 - Für alle anderen Zahlen gib die Zahl selbst aus.
 -
- Schreibe eine Funktion `is_prime(n)`, die überprüft, ob eine Zahl `n` eine Primzahl ist.

Die Funktion soll `True` zurückgeben, wenn `n` eine Primzahl ist, ansonsten `False`.

Eine **Primzahl** ist eine natürliche Zahl größer als 1, die genau zwei positive Teiler hat: 1 und sich selbst. Das bedeutet, dass eine Primzahl nicht durch eine andere natürliche Zahl ohne Rest teilbar ist, außer durch 1 und die Zahl selbst.

4. OOP

Erstelle eine Klasse `Kreis`, die die folgenden Eigenschaften und Methoden besitzt:

- Eine Eigenschaft `radius`, die den Radius des Kreises speichert.
- Eine Methode `umfang()`, die den Umfang des Kreises berechnet (Formel: $2 * \pi * r$).
- Eine Methode `fläche()`, die die Fläche des Kreises berechnet (Formel: $\pi * r^2$).
- für π können wir den gerundeten Wert 3,14 benutzen

Erstelle ein Objekt der Klasse `Kreis` mit einem Radius von 5 und berechne den Umfang und die Fläche.

5. Rekursion

Schreibe eine rekursive Funktion `fibonacci(n)`, die die n-te Zahl der Fibonacci-Folge zurückgibt. Die Fibonacci-Folge beginnt mit 0 und 1, und jede nachfolgende Zahl ist die Summe der beiden vorhergehenden.