

09b_Miniuebungen_Loesungen

0.1 Mini-Aufgaben zur Überprüfung des Verständnis: while Schleifen

Programmieren Sie die folgenden Mini-Aufgaben, um Ihr Verständnis von while Schleifen zu überprüfen und zu verbessern. Achten Sie jeweils auf eine sauber formatierte Ausgabe.

Berechnen Sie die Summe der Zahlen von 1 bis n :

$$S = \sum_{i=1}^n i$$

Berechnen Sie die Summe der geraden Zahlen von 1 bis 100:

$$S = 2 + 4 + 6 + \dots$$

Berechnen Sie die Summe der ungeraden Zahlen von 50 bis 100:

$$S = 51 + 53 + \dots$$

```
[3]: # Summe der Zahlen von 1 bis n
n = 5
summe = 0
i = 1
while i <= n:
    summe = summe + i
    i = i + 1
print("Summe der Zahlen von {} bis {}: {}".format(1,n,summe))

# Summe der geraden Zahlen von 1 bis 100
n = 100
summe = 0
i = 2          # wir beginnen mit der 2
while i <= n:
    summe = summe + i
    i = i + 2 # Inkrement mit 2, um die ungeraden Zahlen zu überspringen
print("Summe der geraden Zahlen von {} bis {}: {}".format(1,n,summe))

# Summe der ungeraden Zahlen von 50 bis 100
n = 100
```

```

summe = 0
i = 51          # wir beginnen mit der 51, da 50 eine gerade Zahl ist
while i <= n:
    summe = summe + i
    i = i + 2 # Inkrement mit 2, um die geraden Zahlen zu überspringen
print("Summe der ungeraden Zahlen von {} bis {}: {}".format(1,n,summe))

```

Summe der Zahlen von 1 bis 5: 15

Summe der geraden Zahlen von 1 bis 100: 2550

Summe der ungeraden Zahlen von 1 bis 100: 1875

Geben Sie das kleine Einmaleins aus für die Zahlen zwischen 1 und 10. Hinweis: Ein Block in einer while Schleife darf wieder eine while Schleife sein, deren Block dann natürlich "doppelt" eingerückt werden muss. Man nennt dies **geschachtelte Schleife**.

```

[9]: i = 1
n = 10
while i <= n:
    # die Kontrollvariable für die innere Schleife muss bei jedem
    # Durchlauf der äußeren Schleife neu initialisiert werden
    j = 1
    # die innere Schleife läuft nur bis i um doppelte Eintragungen zu vermeiden
    while j <= i:
        print("{:2} x {:2} = {:3}".format(i,j,i*j))
        j = j + 1 # Inkrement nicht vergessen
    i = i + 1     # Inkrement für die äußere Schleife wegen Einrückung
print("-"*14)   # damit es einigermaßen hübsch aussieht

```

1 x 1 = 1

2 x 1 = 2

2 x 2 = 4

3 x 1 = 3

3 x 2 = 6

3 x 3 = 9

4 x 1 = 4

4 x 2 = 8

4 x 3 = 12

4 x 4 = 16

5 x 1 = 5

5 x 2 = 10

5 x 3 = 15

5 x 4 = 20

5 x 5 = 25

6 x 1 = 6

6 x 2 = 12
6 x 3 = 18
6 x 4 = 24
6 x 5 = 30
6 x 6 = 36

7 x 1 = 7
7 x 2 = 14
7 x 3 = 21
7 x 4 = 28
7 x 5 = 35
7 x 6 = 42
7 x 7 = 49

8 x 1 = 8
8 x 2 = 16
8 x 3 = 24
8 x 4 = 32
8 x 5 = 40
8 x 6 = 48
8 x 7 = 56
8 x 8 = 64

9 x 1 = 9
9 x 2 = 18
9 x 3 = 27
9 x 4 = 36
9 x 5 = 45
9 x 6 = 54
9 x 7 = 63
9 x 8 = 72
9 x 9 = 81

10 x 1 = 10
10 x 2 = 20
10 x 3 = 30
10 x 4 = 40
10 x 5 = 50
10 x 6 = 60
10 x 7 = 70
10 x 8 = 80
10 x 9 = 90
10 x 10 = 100

0.2 Impressum

0.2.1 Programmierkurs Python, Dominik Göddeke <https://www.ians.uni-stuttgart.de>,
Universität Stuttgart

Version vom April 2023

Lizenziert unter der Creative Commons Namensnennung 4.0 International Lizenz



Veröffentlicht auf <https://zoerr.de>, (alle Rechte am Logo vorbehalten)



Gefördert durch die Stiftung Innovation in der Hochschullehre. (alle Rechte am Logo vorbehalten)



Gefördert mit Mitteln der Deutschen Forschungsgemeinschaft (EXC 2075 - 390740016) im Rahmen der Exzellenzstrategie.

[]: