

## Drill 2

### Aufgabe 1: Hello World, Hello World, ...

Schreibt ein Programm, dass 10x Hello World ausgibt - benutzt hierfür einmal eine for-schleife...

```
for _ in _:
```

und einmal eine while-Schleife

```
while _ < _:
```

### Aufgabe 2: Nur unter einer Bedingung...

Neben den klassischen Rechenoperatoren (+, -, \*, /) bringt Python noch zwei "besondere" Operatoren mit (//, %)

$a//b$  macht eine Ganzzahldivision. Wie bei der Division in der Grundschule wird geschaut, "wie oft b in a passt".

So ist etwa  $5 : 2 = 2$  Rest 2, daher ergibt  $5 // 2 == 2$ , da die 2 zwei Mal in die 5 passt.

$a\%b$  ist hingegen der sogenannte Modulo-Operator und gibt den **Rest** einer Division aus.  $5 \% 2 = 1$ , da  $5 : 2 = 2$  **Rest 1**

Wir können den Modulooperator nutzen um zu testen ob eine Zahl eine andere Zahl teilt.

Folgender Code prüft etwa, ob eine Zahl gerade ist, indem geschaut wird ob die Zahl restlos durch 2 dividiert werden kann.

```
zahl = int(input("Gib eine Zahl ein: "))
```

```
if zahl % 2 == 0:
```

```
    print("Deine Zahl ist gerade!")
```

```
else:
```

```
    print("Deine Zahl ist ungerade!")
```

Nutzt obigen Code als Orientierung und vervollständigt folgendes Programm so, dass es alle Zahlen bis 1000 ausgibt, die durch 6 teilbar sind.

```
for _ in _ :  
    if _
```

### Aufgabe 3: Aufgabe 1. in Besser.

Funktionen dienen dazu, Programmteile wiederverwendbar zu machen - dies geschieht dadurch dass man konkrete Use-Cases zu einem allgemeineren Use-Case zusammenfasst.

a) Statt dem konkreten Fall "Gib Hello World 10x aus" wäre es doch viel eleganter, wenn wir sagen könnten "Gib einen beliebigen Text 10x aus"

Vervollständigt die Funktion `printtexttentimes()` so, dass sie einen beliebigen Text 10x ausgibt.

```
def printtexttentimes(text):
```

Noch eleganter wäre es , wenn wir nicht nur den Text, sondern auch die Anzahl frei wählen könnten.

Vervollständigt die Funktion `printtextXtimes()` so, dass sie einen beliebigen Text beliebig oft ausgibt.

```
def printtextXtimes(text, count):
```