

# Grundlagen - Übung

---

Richard Müller, Tom Felber

20. Oktober 2021

Python-Kurs

1. Übung 1 - Hello World
2. Übung 2 - Strings
3. Übung 3 - String-Methoden

# Übung 1 - Hello World

---

# 1. Aufgabe

## Schritt 1:

Starte den Interpreter mit dem Befehl *python3* auf der Konsole und gebe 'Hello World' mithilfe der Funktion *print()* aus. Mit der Funktion *exit()* kannst du den Interpreter wieder verlassen.

## Schritt 2:

Schreibe ein Python-Script im Editor, welches die gleiche Funktion wie aus Schritt 1 enthält.

Führe das Script mit dem Befehl *python3 script.py* aus. Was passiert?

# 1. Aufgabe

## Schritt 3:

Schreibe das *print()*-Statement nun in eine Funktion.

Führe das Script erneut aus. Was hat sich verändert?

Führe das Script mit dem Befehl *python3 -i script.py* interaktiv aus.

Sobald der Interpreter gestartet ist, rufe deine Funktion auf.

## Schritt 4:

Rufe im Script die Funktion im Top Level auf.

Führe das Script erneut aus.

Öffne den Interpreter mit *python3* im selben Verzeichnis, wo sich dein Script befindet und importiere es dort: *import yourscrip*t (Beachte, dass am Ende kein *'.py'* steht). Was passiert?

# 1. Aufgabe

## Schritt 5:

Füge im Script den Boilerplate Code hinzu und rufe deine Funktion im *if*-Block auf. Beachte die Einrückung!

```
1 if __name__ == '__main__':  
2     my_function()
```

Führe das Script noch einmal aus.

Starte den Interpreter und importiere dein Script erneut. Was hat sich verändert?

## Übung 2 - Strings

---

## 2. Aufgabe

Strings, wie verbindet und formatiert man sie?

Tipp: möglichst oft den Code testen, um früh Fehler zu finden

### **Schritt 1:**

Mit dem `input()` Befehl kann eine Eingabe aus der command line gelesen und in einer Variable abgelegt werden: `var = input()`. An der Stelle, wo `input()` aufgerufen wird, pausiert das Script, bis eine Eingabe gegeben wurde.

Schreibe ein Script, dass einen input string von der command line liest und `'Hello '` ausgibt, wobei der eingelesene String angehängen wird. Du kannst f-Strings verwenden, oder die strings einfach mit `+` verbinden.



## Schritt 2:

Erweitere das Script nun um eine Startnachricht, die vor der input Abfrage ausgegeben wird und nach einem Namen fragt:

`input("message")`. Gib eine Willkommensnachricht aus. Zum Beispiel: "Hello NAME, welcome to python", wobei NAME durch den input ersetzt werden soll. Dafür können f-Strings oder die format Funktion verwendet werden.

f-Strings funktionieren so: `f'hallo {'oma'}, willkommen"` -> `"hallo oma, willkommen"`

**Schritt 3** Frage nun nach dem Vor und Nachnamen separat und baue die Werte in den Ausgabestring ein.

## Übung 3 - String-Methoden

---

## 3. Aufgabe

### Schritt 1

Mit der Dokumentation auf dieser Seite

<https://docs.python.org/3/library/stdtypes.html#string-methods>

Schreibe ein Script, dass einen Input-String annimmt und prüft, ob folgende Bedingungen wahr sind:

- (1.) der String ist großgeschrieben
- (2.) der String besteht aus Zahlen
- (3.) im String kommt diese sequenz vor: ':-)'
- (4.) keine der vorherigen Bedingungen ist wahr

Gib aus welcher der Fälle eingetreten ist. Wenn mehrere zutreffend sind, reicht es einen anzugeben.

Substrings erkennen: mit *in* kann nach Teilelementen und auch Teilstrings gesucht werden.

Python hat kein switch case statement, nur if, else, elif (statt else if).

## Schritt 2

Gib im 1. Fall, wenn der String großgeschrieben ist, zusätzlich den String als kleingeschriebene Variante aus.

## Schritt 3

Im 2. Fall, wenn eine Zahl eingegeben wurden, lass das Script nach einer weiteren fragen und gib die Summe aus.

Der Konstruktor eines Typs kann benutzt werden, um einen anderen Wert in diesen Typ umzuwandeln. z.B. `str()`, `int()`

## Schritt 4

Im 4. Fall, wenn kein anderer Fall eingetreten ist, trenne den Eingabestring an jedem 'a', und füge die Teile wieder zusammen, und ergänze ein Semikolon zwischen jedem der Teile.

Strings können mit *split(separator)* an einem Separator getrennt werden.

Und mit *separator\_string.join(['list', 'of', 'strings'])* zusammengefügt werden.