

# Einführung in die computerlinguistische Programmierung mit Python

## 01-02: Was ist Python? 🐍 🐍 🐍

Python existiert seit 1991 und ist heutzutage eine mächtige und verbreitete Sprache in der **Wirtschaft**, aber auch im **wissenschaftlichen Bereich**. Die Fähigkeiten, die ihr in diesem Kurs erwerbt, könnt ihr also in beiden Bereichen später anwenden!

Python funktioniert auf Windows-, Mac- und Linuxsystemen und ist **vielseitig einsetzbar**. Die Sprache ermöglicht verschiedene Programmierstile, z.B. objektorientiert, logisch, oder funktional. Ziel dieses Kurses ist es, euch einen Einblick in die wichtigsten Funktionalitäten von Python zu bieten.

Eine Besonderheit der Sprache ist, dass in Python geschriebene Programme relativ **einfach zu lesen** sind -- verglichen mit anderen verbreiteten Programmiersprachen. Die folgenden Codeausschnitte geben jeweils die Worte "Hallo Welt!" auf dem Bildschirm aus, wobei die Java-Variante verschachtelter ist und etwas mehr Boilerplate-Code (<https://de.wikipedia.org/wiki/Boilerplate#Programmierung>) enthält:

### Python

```
print("Hallo Welt!")
```

---

### Java

```
public class Main {  
    public static void main(String[] args) {  
        System.out.println("Hello, World!");  
    }  
}
```

---

# Der Python-Interpreter

Gängige Programmiersprachen werden entweder **interpretiert** oder **kompiliert**. Die Unterscheidung bezieht sich darauf, in welcher Weise der geschriebene Programmcode (z.B. unser Hallo-Welt-Programm) in Maschinensprache übersetzt wird, die dann dafür sorgt, dass der Computer unsere Anweisungen befolgt. (Mehr Informationen gibt es z.B. [hier \(https://www.dev-insider.de/der-unterschied-von-compiler-und-interpreter-a-742282/\)](https://www.dev-insider.de/der-unterschied-von-compiler-und-interpreter-a-742282/).)

Wir nutzen Python als interpretierte Programmiersprache. Um Pythoncode ausführen zu können, müssen wir also einen **Python-Interpreter** installieren.

Schreiben können wir den Code in einem beliebigen **Texteditor**. Professionelle Editoren haben meist eingebaute Tools zum Ausführen von Interpretern.

Dieses Jupyter-Notebook kann übrigens Pythoncode direkt **ausführen!** Probiert es einfach mal aus. Im Browser klickt ihr dazu auf den Programmiercode im Kästchen unter dieser Zeile und drückt dann `Ctrl + Enter`. Falls ihr das Notebook in VSCode öffnet, könnt ihr auf den "Play"-Knopf an der Seite des Kästchens mit dem Code drücken. Unterhalb des Kästchens erscheint dann das Ergebnis des Pythonbefehls.

In [ ]:

```
print("Führ dich aus!")
```

Wenn ihr mögt, könnt ihr den Code auch **verändern**. Dazu klickt ihr auf das Kästchen und tippt etwas anderes ein, bevor ihr das Kästchen noch einmal ausführt.

Der Befehl `print()` sorgt dafür, dass der Interpreter den Inhalt der Klammern anzeigt (wir sagen: Der Interpreter **gibt den Inhalt aus**). Im Fall oben war das ein vordefinierter Text. Wir können aber auch komplexe Ausdrücke `print()` en: Dann wird der Inhalt der Klammern zuerst ausgewertet, und danach wird `print()` auf das Resultat angewendet. Probiert es aus und ändert wieder den Inhalt der Klammern, damit ein anderes Ergebnis angezeigt wird:

In [ ]:

```
print(3 + 8)
```

In [ ]:

```
print("Hallo " + "Welt")
```

In [ ]:

```
print(1)
print(2)
print(3)
```

Der Interpreter versucht nacheinander **jede Zeile unseres Codes** zu übersetzen und auszuführen. Das ist problematisch, wenn wir uns verprogrammieren...

In [ ]:

```
print(Hallo Welt!)
```

... oder wenn wir uns Notizen in unserem Programm machen möchten...

In [ ]:

```
meinen Namen ausgeben:  
print("Esther Seyffarth")
```

Notizen im Code sind aber dringend zu empfehlen, gerade in der ersten Zeit, damit man sich merken kann, was der jeweilige Code bedeutet.

Notizen in Programmcode heißen **Kommentare** und können auf zwei verschiedene Weisen markiert werden:

In [ ]:

```
# [das hier ist ein Kommentar:] meinen Namen ausgeben  
print("Ben Burkhardt")
```

In [ ]:

```
""  
[das hier ist ein mehrzeiliger Kommentar:]  
Ausgabe:  
- mein Name  
- meine Emailadresse  
""  
print("Esther Seyffarth")  
print("esther.seyffarth@hhu.de")
```

Beim Interpretieren (also beim Ausführen unseres Programms) werden die Kommentare vom Interpreter einfach **übersprungen**. Kommentare, die mit # beginnen, gelten bis zum nächsten Zeilenumbruch. Kommentare, die mit "" oder ''' beginnen, gelten über Zeilenumbrüche hinweg, bis das nächste Mal "" bzw. ''' gefunden wird. An der Farbe der Schrift erkennen wir übrigens, dass Kommentare nicht wie Programmcode interpretiert werden:

In [ ]:

```
print("kein Kommentar! (1)")  
#print("kein Kommentar! (2)")  
  
#####  
# print("kein Kommentar! (3)") #  
#####
```

Einige Gründe, Kommentare in unserem Pythoncode aktiv zu nutzen:

- Als **Erinnerung** für später, was der Effekt einer Zeile ist.
- Als **Erklärung** für die anderen Gruppenmitglieder, wenn die Zeile nicht selbsterklärend ist.
- Als **Platzhalter** für Codeabschnitte, die schwierig zu schreiben sind.
- Um während des Programmierens den **Überblick** zu wahren: Ein Kommentar ist schnell geschrieben und hilft dabei, sich zu merken, was an welcher Stelle passieren soll.

# Und hier noch einige Tipps...

- Benennt eure Pythondateien mit **sprechenden Namen**! Ihr könnt beispielsweise das Datum im Dateinamen angeben, Stichworte aus der Aufgabenstellung verwenden oder die Übungen durchnummerieren. Wir werden das ganze Semester über immer wieder auf alte Dateien zurückgreifen. Je übersichtlicher eure Dateien sortiert sind, umso einfacher wird das.
- Sammelt eure Pythondateien an einem **zentralen Ort**! Denkt euch zu Beginn des Studiums eine Ordnerstruktur aus, die ihr die ganze Zeit verwenden könnt. Wichtig ist, dass ihr eure Dateien wiederfindet.
- Wenn ihr in eurer Gruppe die Lösung der Aufgaben aufteilt, sorgt dafür, dass trotzdem **alle die Lösungen der Gruppe speichern**. Das ist hilfreich, wenn ihr später Dinge wiederholen wollt und dann einfach in der jeweiligen Datei nachsehen könnt.

## Zusammenfassung

- Python ist eine verbreitete Programmiersprache, die in Wissenschaft und Wirtschaft verwendet wird.
- Wir schreiben Python-Programme als Textdateien und führen sie dann mit dem Python-Interpreter aus.
- Der Editor Visual Studio Code (VSCoDe) kann uns das Leben leichter machen und hat gute Funktionen zum Arbeiten mit Python.
- Mit `print()` sorgen wir dafür, dass das, was in den Klammern eingefügt wird, während des Ausführens ausgegeben wird.
- Kommentare werden in Python mit einem `#` oder dreifachen Anführungszeichen markiert. Kommentare werden beim Ausführen des Programms übersprungen.
- Bei verschachtelten Befehlen in Pythoncode wird immer zuerst der innere Befehl interpretiert, dessen Ergebnis dann für die Interpretation der äußeren Befehle verwendet wird (Beispiel: `print(3 + 5)` )

## Weitere Themen dieser Woche

### Vorlesung:

- 01-01: Wie läuft der Kurs?

### Anleitungen:

- Python installieren
- VSCoDe installieren
- VSCoDe liveshare verwenden
- Jupyter Notebooks verwenden
- Moodle verwenden