

# Woche 1: Einführung in Python

In der ersten Woche des Kurses tauchen wir in die Welt von Python ein. Es beginnt mit der Installation und Einrichtung der Python-Entwicklungsumgebung. Hierbei wird besonderer Wert auf die praktische Arbeit gelegt, um die Teilnehmer schnellstmöglich mit der Handhabung vertraut zu machen. Im Anschluss daran wird die grundlegende Syntax von Python behandelt, einschließlich der Definition und Verwendung von Variablen, sowie der grundlegenden Datentypen.

Ein weiterer wichtiger Aspekt dieser Woche ist die Einführung in die Steuerungsstrukturen von Python, wie Verzweigungen und Schleifen. Diese sind essentiell, um die Abläufe in Programmen zu kontrollieren. Gleichzeitig wird die Benutzereingabe und -ausgabe behandelt, um eine Interaktion mit dem Benutzer zu ermöglichen.

Die praktischen Aufgaben zielen darauf ab, die neu erlernten Kenntnisse zu vertiefen und zu festigen. Dabei wird von einfachen Berechnungen bis hin zu grundlegenden Schleifen und Verzweigungen eine breite Palette abgedeckt.

Der Übungstag rundet die erste Woche ab. Dabei werden die zuvor erlernten Themen wiederholt und vertieft, und es werden erste Einblicke in die Dateioperationen und die Datenstrukturen von Python gegeben. Dies gibt einen Ausblick auf die folgenden Wochen und soll die Lernenden motivieren, sich weiter mit Python zu beschäftigen.

Ziel der ersten Woche ist es, ein solides Fundament in Python zu schaffen, auf dem in den folgenden Wochen aufgebaut wird.

# Gesamtüberblick

Hier ein Überblick über die Inhalte und Aktivitäten der ersten Woche:

- Selbststudium:
  - Installation
  - Syntax
  - Variablen
  - Datentypen
  - Einfache Ein-/Ausgabe
  - Verzweigung
  - Schleife
- Aufgaben:
  - Einfache Berechnungen
  - Benutzereingabe und -ausgabe
  - Einfache Schleifen
  - Einfache Verzweigungen
- Übungstag 1:
  - Wiederholung
  - Vertiefung: Verzweigung
  - Vertiefung: Schleifen
  - Ergänzung: Dateioperationen
  - Ausblick: Datenstrukturen

## Inhalte und thematische Abgrenzung

Die folgende Auflistung zeigt detailliert, welche Themen Sie in der Woche behandeln und bearbeiten. Sie sind eine Voraussetzung für die folgenden Wochen und sollten gut verstanden worden sein.

Wenn es Verständnisprobleme gibt, machen Sie sich Notizen und fragen Sie am Präsenztage nach, so dass wir gemeinsam zu Lösungen kommen können. Und denken Sie bitte immer daran: es gibt keine „dummen“ Fragen!

### 1. Installation

- Python-Download von der offiziellen Website
- Installation auf verschiedenen Betriebssystemen (Windows, macOS, Linux)
- Überprüfung der Installation (Python-Version, pip)
- Einrichtung einer Entwicklungsumgebung (VS Code, PyCharm)
- Erstes Python-Programm erstellen, ausführen

### 2. Python-Syntax

- Kommentare
- Einrückung
- Anweisungen und Ausdrücke
- Fehler und Ausnahmen

### 3. Variablen

- Definition und Zuweisung
- Variablennamen (Regeln und Konventionen)
- Mehrere Zuweisungen
- Unveränderliche vs. veränderliche Variablen

### 4. Datentypen

- Zahlen (Ganzzahlen, Fließkommazahlen)
- Zeichenketten (Erstellen, Formatieren, Methoden)
- Booleans
- Typumwandlung

### 5. Einfache Ein-/Ausgabe

- print()-Funktion
- input()-Funktion
- Formatierung der Ausgabe

### 6. Verzweigung (Bedingte Anweisungen)

- if-Anweisung
- if-else-Anweisung
- if-elif-else-Anweisung
- Verschachtelte if-Anweisungen

### 7. Schleifen

- while-Schleife
- for-Schleife
- Kontrollstrukturen (break, continue)
- Verschachtelte Schleifen

#### 8. Einfache Berechnungen

- Arithmetische Operationen (Addition, Subtraktion, Multiplikation, Division)
- Vergleichsoperationen (>, <, ==, !=, >=, <=)
- Logische Operationen (and, or, not)

## Lernpfad

Der Lernpfad ist ein Vorschlag, in welcher Reihenfolge Sie die Inhalte der Woche angehen können. Betrachten Sie ihn gerne als eine Todo-Liste, die Sie von oben nach unten abhaken. So können Sie sicher sein, dass Sie alle wichtigen Themen bearbeitet haben und sind gut vorbereitet für die folgenden Wochen.

1. Installieren Sie Python und eine Entwicklungsumgebung
2. Machen Sie sich mit der Python-Syntax vertraut: Lernen Sie, wie man Kommentare schreibt, wie die Einrückung funktioniert, und was Anweisungen und Ausdrücke sind.
3. Lernen Sie, wie man Variablen in Python definiert und zuweist. Machen Sie sich mit den Regeln für Variablennamen und der Unterscheidung zwischen unveränderlichen und veränderlichen Variablen vertraut.
4. Erkunden Sie die verschiedenen Datentypen in Python: Zahlen, Zeichenketten und Booleans. Lernen Sie, wie man den Typ einer Variablen ändert.
5. Lernen Sie, wie man in Python Ein- und Ausgaben handhabt. Üben Sie das Verwenden der `print()`- und `input()`-Funktionen.
6. Machen Sie sich mit bedingten Anweisungen vertraut: `if`, `if-else` und `if-elif-else`. Üben Sie das Schreiben von verschachtelten `if`-Anweisungen.
7. Lernen Sie, wie man Schleifen in Python erstellt: `while`- und `for`-Schleifen. Machen Sie sich mit den Kontrollstrukturen `break` und `continue` vertraut.
8. Üben Sie das Schreiben von Programmen, die einfache Berechnungen durchführen. Verwenden Sie arithmetische, Vergleichs- und logische Operatoren.
9. Lernen Sie, wie man Benutzereingaben in Python erhält und ausgibt. Üben Sie das Schreiben von Programmen, die Daten eingeben und ausgeben.
10. Üben Sie das Schreiben von Programmen mit einfachen Verzweigungen.
11. Üben Sie das Schreiben von Programmen mit einfachen Schleifen.
12. Überprüfen Sie Ihr Verständnis durch das Lösen von Übungsaufgaben und Projekten, die die oben genannten Konzepte nutzen.

## Programmieraufgaben

Die folgenden Programmieraufgaben sollen Ihnen eine Anregung geben. Haben Sie eigene Ideen und Themen, die Sie ausprobieren wollen, dann sollten Sie diesen nachgehen. Wichtig ist vor allem, dass Sie „Dinge ausprobieren“. Und auch, dass Sie Fehler machen, sowohl syntaktische als auch semantische. Versuchen Sie diese Fehler zu finden und aufzulösen, dann gerade aus den Fehlern lernen Sie am Ende am meisten.

- Erstellen und Ausführen eines einfachen Python-Programms, das "Hallo Welt!" ausgibt.
- Berechnen der Summe, Differenz, Multiplikation und Division von zwei Zahlen, die vom Benutzer eingegeben wurden („Taschenrechner“).
- Erstellen eines Programms, das den Benutzer nach seinem Namen und Alter fragt und dann eine Nachricht ausgibt, die beides enthält.
- Schreiben eines Programms, das überprüft, ob eine eingegebene Zahl gerade oder ungerade ist.
- Erstellen eines Programms, das zwei vom Benutzer eingegebene Zahlen vergleicht und angibt, welche größer ist oder ob sie gleich sind.
- Erstellen eines Programms, das vom Benutzer Zahlen und Kommandos für Berechnungen (siehe „Taschenrechner“) entgegennimmt und die Ergebnisse ausgibt. Wenn der Benutzer „exit“ eingibt soll das Programm enden.
- Erstellen eines Programms, welches vom Benutzer Daten entgegennimmt, den BMI errechnet und eine Bewertung des BMI-Wertes ausgibt.

## Abschluss-Quiz

Das Quiz soll Ihnen einen ersten Hinweis auf Ihren Lernfortschritt geben. Nach unserer Einschätzung sollten Sie diese Fragen alle beantworten können, wenn Sie den Stoff der Woche durchgearbeitet und verstanden haben. Natürlich gibt es noch sehr viel mehr mögliche Fragen, dazu wollen wir auf die Literatur und das Internet verweisen. Geben Sie gerne einmal „python quizzes“ bei Google ein.

1. Wie ordnen Sie in Python eine Ganzzahl der Variablen "x" zu?
  1. `x = 5.0`
  2. `x = "5"`
  3. `x = 5`
  4. `x == 5`
2. Welche Funktion verwenden Sie, um Benutzereingaben zu lesen?
  1. `read()`
  2. `get()`
  3. `input()`
  4. `fetch()`
3. Welcher der folgenden Operatoren ist der richtige Vergleichsoperator für "gleich"?
  1. `=`
  2. `==`
  3. `:=`
  4. `!=`
4. Welche der folgenden Anweisungen ist korrekt, um eine bedingte Anweisung in Python zu erstellen?
  1. `if x = 5: print("x ist 5")`
  2. `if x == 5 print("x ist 5")`
  3. `if x == 5: print("x ist 5")`
  4. `if x: 5 print("x ist 5")`
5. Welche der folgenden Methoden ist die richtige Methode, um eine for-Schleife in Python zu erstellen?
  1. `for i < 5: print(i)`
  2. `for i = 0; i < 5; i++: print(i)`
  3. `for i in range(5): print(i)`
  4. `for i to 5: print(i)`

6. Wie erstellt man in Python eine Zeichenkette, die mehrere Zeilen umfasst?
1. Verwenden Sie einfache Anführungszeichen: `' '`
  2. Verwenden Sie doppelte Anführungszeichen: `" "`
  3. Verwenden Sie dreifache einfache Anführungszeichen: `''' '''`
  4. Verwenden Sie dreifache doppelte Anführungszeichen: `""" """`
7. Was ist der Ausdruck `10 % 3` in Python?
1. Modulo
  2. Ganzzahlige Division
  3. Exponentiation
  4. Division
8. Welcher der folgenden Ausdrücke ist in Python falsch?
1. `print('Hallo' + ' Welt!')`
  2. `print('Hallo' * 3)`
  3. `print('Hallo' / 3)`
  4. `print('Hallo' - ' Welt!')`
9. Was wird ausgegeben, wenn der folgende Code ausgeführt wird: `print('Hallo\nWelt!')`?
1. `Hallo\nWelt!`
  2. `Hallo`  
`Welt!`
  3. `Hallo Welt!`
  4. `SyntaxError`
10. Was ist der Typ des folgenden Ausdrucks in Python: `"5" + "7"`?
1. Integer
  2. Float
  3. String
  4. Boolean

Lösungen:

1/3 2/3 3/2 4/3 5/3 6/3 7/1 8/3 9/2 10/3



## Ressourcen

Hier nun die Verweise auf Lernquellen, die uns für diese Woche und ihre Inhalte geeignet erscheinen. Je nachdem, welcher Lerntyp Sie sind, wählen Sie sich ihre bevorzugte Quelle, es ist nicht zwingend notwendig alle durchgearbeitet zu haben. Allerdings sollten die Inhalte des Lernpfads angesprochen und verstanden worden sein.

- Buch: Visual Studio Code for Python Programmers
  - Hier suchen Sie sich die Kapitel heraus, die Sie benötigen, um Visual Studio Code einzurichten, wenn Sie mögen. Achten Sie u.a. darauf, dass auch die notwendigen Module installiert wurden.
- Buch: Python Crash Course
  - **Installation:** Dieses Thema wird in der "Preface to the Third Edition" und "Appendix A: Installation und Fehlerbehebung" behandelt.
  - **Syntax:** Die Grundlagen der Python-Syntax werden im gesamten Buch eingeführt und verwendet, beginnend mit "Kapitel 1: Einstieg".
  - **Variablen:** Dieses Thema wird in "Kapitel 2: Variablen und einfache Datentypen" eingeführt.
  - **Datentypen:** Datentypen werden ebenfalls in "Kapitel 2: Variablen und einfache Datentypen" behandelt.
  - **Einfache Eingabe/Ausgabe:** Benutzereingaben und while-Schleifen werden in "Kapitel 7: Benutzereingabe und while-Schleifen" behandelt.
  - **Verzweigung** (if-Anweisungen): Dieses Thema wird in "Kapitel 5: if-Anweisungen" behandelt.
  - **Schleifen:** While-Schleifen werden in "Kapitel 7: Benutzereingabe und while-Schleifen" eingeführt. Für andere Arten von Schleifen, wie zum Beispiel for-Schleifen, möchten Sie vielleicht "Kapitel 4: Arbeiten mit Listen" konsultieren
- Video: CodeAcademy
  - Codecademy: Python for Programmers – Ein erster Einstieg
  - Codecademy: Learn Python 3 – Ein umfassender Kurs mit 27h Dauer
- Lab:
  - Python Novice – Einsteiger
  - Lab: Python for Developers – Grundlegendes Python inklusive Datenstrukturen