

Woche 1: Einführung in die Programmierung mit Hilfe von Python

Einleitung

In der ersten Woche unseres Kurses tauchen wir in die Welt der Programmierung ein, wobei Python unser Werkzeug der Wahl ist. Python ist bekannt für seine einfache und lesbare Syntax, was es zu einer hervorragenden Sprache für Anfänger macht. In diesem Kurs lernen Sie Grundlagen, die auch auf andere Programmiersprachen übertragbar sind.

Voraussetzung ist, dass Sie Python und eine Entwicklungsumgebung mit Hilfe Ihres Trainers installiert haben. Falls dies noch nicht geschehen ist, nutzen Sie bitte die Ressourcen auf folgenden Seiten:

- www.online-python.com
- www.programiz.com/python-programming/online-compiler
- www.programiz.pro/ide/python
- www.pythontutor.com

Ziel dieser Woche ist es, ein grundlegendes Verständnis dafür zu entwickeln, was ein Programm ist, wie Daten gespeichert und verarbeitet werden, und wie Ein- und Ausgaben in einem Programm gehandhabt werden. Wir werden uns auch mit der grundlegenden Syntax von Python befassen, einschließlich der Definition und Verwendung von Variablen, Datentypen und grundlegenden Operatoren.

Die praktischen Aufgaben zielen darauf ab, die neu erlernten Kenntnisse zu vertiefen und zu festigen. Dabei wird von einfachen Berechnungen bis hin zu Ein- und Ausgaben eine breite Palette abgedeckt. Betrachten Sie die Übungsaufgaben als Ideensammlung. Wenn Sie eigene Ideen realisieren wollen, steht dem nichts im Wege. Sollten Sie nicht alle Aufgaben schaffen, ist das vollkommen in Ordnung, lernen Sie so schnell wie es Ihrem Vorgehen entspricht.

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, einen guten Einstieg in die Programmierung, mit Hilfe von Python, zu schaffen, der in den folgenden Wochen ausgebaut wird.

Gesamtüberblick

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

- Selbststudium:

- Programmiergrundlagen
- Algorithmen und Programme
- Variablen
- Datentypen
- Berechnungen
- Einfache Eingabe
- Einfache Ausgabe
- Verzweigung
- Schleife

- Aufgaben:

- Einfache Berechnungen
- Benutzereingabe und -ausgabe

- Übungstag 1:

- Wiederholung
- Vertiefung: Variablen und Datentypen
- Vertiefung: Berechnungen
- Vertiefung: Ein- und Ausgabe
- Ausblick: Verzweigungen und Schleifen

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. Programme und Algorithmen

- Was ist ein Programm?
- Was ist ein Algorithmus?
- Wie baut man ein Programm?
- Wie wandelt man eine Aufgabe in ein Programm um?
- Wozu gibt es Programmiersprachen?

2. Python-Syntax (Grundlagen)

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

3. Variablen

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

4. Datentypen

- Zahlen (Ganzzahlen, Fließkommazahlen)
- Zeichenketten (Erstellen, Formatieren, Methoden)
- Wahrheitswerte (Booleans)
- Typumwandlung und ihre Grenzen

5. Einfache Ein-/Ausgabe

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

6. 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. Erarbeiten Sie sich die Grundbegriffe der Programmierung. Verstehen Sie, was ein Algorithmus ist und was ein Programm macht.
3. Machen Sie sich mit Variablen vertraut und mit Datentypen, verstehen Sie was man mit Variablen macht, wozu Datentypen dienen und was das mit Operatoren zu tun hat.
4. Machen Sie sich mit der Python-Syntax vertraut: Lernen Sie, wie man Kommentare schreibt, wie die Einrückung funktioniert, und was Anweisungen und Ausdrücke sind
5. 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.
6. Erkunden Sie die verschiedenen Datentypen in Python: Zahlen, Zeichenketten und Booleans. Lernen Sie, wie man den Typ einer Variablen ändert.
7. Lernen Sie, wie man in Python Ein- und Ausgaben handhabt. Üben Sie das Verwenden der `print()`- und `input()`-Funktionen.
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. 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.
- Erstellen eines Programms, das den Benutzer nach seinem Namen und Alter fragt und dann eine Nachricht ausgibt, die beides enthält.
- Erstellen eines Programms, welches vom Benutzer Daten entgegennimmt und den BMI errechnet und ausgibt.
- Berechnen der Summe, Differenz, Multiplikation und Division von zwei Zahlen, die vom Benutzer eingegeben wurden („Taschenrechner“, achten Sie auf die nötige Typumwandlung der Eingabe!)

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. Diese und das nächste Abschlussquiz sind etwas umfangreicher, als die dann später im Kurs folgenden. Der Grund dafür ist, dass die hier zu legenden Grundlagen so wichtig sind, um Ihnen einen guten Erfolg und viel Freude an der Programmierung zu sichern.

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.

Teil 1: Grundfragen des Programmierens

1. Was ist eine Variable in der Programmierung?
 - a) Ein Befehl, der vom Computer ausgeführt wird
 - b) Ein Speicherplatz, der einen Wert hält
 - c) Eine Fehlermeldung im Code
 - d) Eine Art von Programmiersprache
2. Wozu dient ein Algorithmus in der Programmierung?
 - a) Zur Speicherung von Daten
 - b) Zur Behebung von Fehlern im Code
 - c) Zur Darstellung von Informationen
 - d) Zur Lösung eines Problems oder zur Ausführung einer Aufgabe
3. Was versteht man unter "Syntax" in einer Programmiersprache?
 - a) Die Geschwindigkeit, mit der ein Programm ausgeführt wird
 - b) Die Regeln, die festlegen, wie Code geschrieben werden muss
 - c) Die Art, wie Fehler im Code identifiziert werden
 - d) Die Dokumentation, die ein Programm begleitet
4. Was ist ein Datentyp in der Programmierung?
 - a) Ein Tool zur Fehlerbehebung
 - b) Eine Kategorie von Daten, die ähnliche Eigenschaften aufweisen
 - c) Ein spezieller Programmierbefehl
 - d) Eine Art von Algorithmus
5. Was bedeutet "Debugging" in der Programmierung?
 - a) Ein Programm schneller machen
 - b) Ein Programm in einer anderen Sprache schreiben

- c) Fehler im Code finden und beheben
- d) Daten von einem Format in ein anderes konvertieren

6. Wofür steht IDE in der Programmierung?

- a) Integrated Development Environment
- b) Internal Data Exchange
- c) Interactive Debugging Experience
- d) Intelligent Design Engine

7. Was ist ein Kommentar im Code?

- a) Ein Teil des Codes, der vom Computer ausgeführt wird
- b) Ein Fehler im Code
- c) Ein Abschnitt, der Informationen für andere Programmierer enthält, aber vom Computer ignoriert wird
- d) Eine Methode, um Variablen zu deklarieren

8. Was versteht man unter Quellcode?

- a) Die Daten, die ein Programm verarbeitet
- b) Der von Programmierern geschriebene Code
- c) Die Ausgabe, die ein Programm erzeugt
- d) Ein Tool zur Fehlerbehebung im Programm

9. Was ist eine Funktion in der Programmierung?

- a) Eine Art von Variable
- b) Ein Fehler im Code
- c) Ein wiederverwendbarer Codeblock, der eine bestimmte Aufgabe ausführt
- d) Eine spezielle Art von Kommentar im Code

10. Was bedeutet "Kompilieren" in der Programmierung?

- a) Ein Programm in einer höheren Programmiersprache schreiben
- b) Den Quellcode in maschinenlesbaren Code umwandeln
- c) Fehler im Code finden und beheben
- d) Daten von einem Format in ein anderes konvertieren

Teil 2: Fragen zu Python

1. Was ist Python?
 - a) Ein Betriebssystem
 - b) Eine Programmiersprache
 - c) Ein Texteditor
 - d) Ein Webbrowser
2. Wofür wird Python häufig verwendet?
 - a) Nur für Webentwicklung
 - b) Nur für Datenanalyse
 - c) Für eine Vielzahl von Anwendungen wie Webentwicklung, Datenanalyse, KI und mehr
 - d) Ausschließlich für künstliche Intelligenz
3. Was stellt in Python die print() Funktion dar?
 - a) Eine Schleife
 - b) Eine Bedingung
 - c) Eine Funktion zur Ausgabe von Daten
 - d) Eine Variable
4. Was bedeutet es, dass Python eine interpretierte Sprache ist?
 - a) Dass Python-Code in maschinenlesbaren Code kompiliert werden muss
 - b) Dass Python-Code direkt ausgeführt werden kann, ohne zuvor kompiliert werden zu müssen
 - c) Dass Python eine künstliche Sprache ist
 - d) Dass Python-Code nicht auf verschiedenen Plattformen ausgeführt werden kann
5. Wie beginnt ein typischer Variablenname in Python?
 - a) Mit einem Zahlenwert
 - b) Mit einem Unterstrich oder einem Buchstaben
 - c) Mit einem Sonderzeichen wie @ oder #
 - d) Mit einem Großbuchstaben
6. Welches der folgenden Elemente ist ein gültiger Datentyp in Python?
 - a) user-defined
 - b) string
 - c) paragraph
 - d) element

7. Wie weist man den Wert 5 einer Variablen namens x in Python zu?

- a) `x == 5`
- b) `x = 5`
- c) `x : 5`
- d) `x -> 5`

8. Welcher der folgenden ist ein Kommentar in Python?

- a) `//` Dies ist ein Kommentar
- b) `/*` Dies ist ein Kommentar `*/`
- c) `#` Dies ist ein Kommentar
- d) `--` Dies ist ein Kommentar

9. Wie wird in Python ein String deklariert?

- a) Mit eckigen Klammern: `[string]`
- b) Mit einfachen oder doppelten Anführungszeichen: `'string'` oder `"string"`
- c) Mit geschweiften Klammern: `{string}`
- d) Mit Klammern: `(string)`

10. Welche der folgenden Optionen erstellt eine Liste in Python?

- a) `{1, 2, 3}`
- b) `[1, 2, 3]`
- c) `(1, 2, 3)`
- d) `<1, 2, 3>`

Teil 3 – Programmiergrundlagen in Python

1. Wie weist man in Python einer Variablen "x" den Wert 10 zu?
 - a) `x = 10`
 - b) `x == 10`
 - c) `x: 10`
 - d) `x -> 10`
2. Welcher Datentyp wird in Python für ganze Zahlen verwendet?
 - a) float
 - b) string
 - c) int
 - d) bool
3. Welcher Operator wird für die Addition in Python verwendet?
 - a) `+`
 - b) `-`
 - c) `*`
 - d) `/`
4. Wie ändert man den Typ einer Variablen von Integer zu String in Python?
 - a) `int(x)`
 - b) `str(x)`
 - c) `float(x)`
 - d) `bool(x)`
5. Welche Funktion wird verwendet, um Text auf dem Bildschirm auszugeben?
 - a) `output()`
 - b) `echo()`
 - c) `print()`
 - d) `show()`
6. Welche Funktion liest Benutzereingaben als String?
 - a) `read()`
 - b) `get()`
 - c) `input()`
 - d) `fetch()`

7. Welcher Ausdruck erzeugt eine Zeichenkette (String) aus der Zahl 123?

- a) "123"
- b) str(123)
- c) int("123")
- d) "123".int()

8. Wie werden in Python Variablen mit dem Wert False des Typs Boolean erstellt?

- a) x = "False"
- b) x = 0
- c) x = False
- d) x = none

9. Welcher Operator wird für die Multiplikation in Python verwendet?

- a) +
- b) -
- c) *
- d) /

10. Wie gibt man den Typ einer Variablen in Python aus?

- a) print(type(var))
- b) print(typeof var)
- c) echo(var.type)
- d) type(var).print()

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 – sollten Sie Probleme haben, wenden Sie sich an Ihren Trainer.

- **Buch:** Get Programming: Learn to Code with Python

Grundlagen der Programmierung mit Hilfe von Python erklärt. Dieses Buch passt sehr gut zu +den ersten beiden Wochen des Kurses, zeigt aber auch darüber hinausgehende Konzepte in verständliche Form.

- **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.

- **Video:** CodeAcademy

Codecademy: Python for Programmers – Ein erster Einstieg

Codecademy: Learn Python 3 – Ein umfassender Kurs mit 27h Dauer

- **Lab:**

Python Novice – Einsteiger

Python for Developers – Grundlegendes Python inklusive Datenstrukturen