

# Programmieren mit Python: Kursgliederung

## Tag 1 (Montag) Vormittag (9:00 - 12:30 Uhr)

### Vorstellung & Organisatorisches (15-20 min)

- Vorstellung SoftEd, Trainer, Seminarbegleitung
- Ausblick auf die Seminarinhalte
- Vorstellungsrunde Teilnehmer, Präsenz und Online
- Ablauf, Pausen, Mittagessen etc.

### Einführung in die Programmiersprache Python (45-60 min)

- Einführung in die Arbeitsumgebung für den Kurs
- Verwendung des Interpreters (Interactive Modus)
  - Aufruf
  - Interaktives Testen von Code
  - Python als Taschenrechner
  - Erweiterte interaktive Umgebungen (IPython, Jupyter)
- Genereller Aufbau eines Programms
  - Hallo, Welt!
  - Ein kurzes, aber komplettes Kommandozeilenprogramm
  - Syntaxelemente von Python (Ausdrücke, Anweisungen, Operatoren, Trennzeichen, Einrückung usw.)
- Grundlegende Datentypen und Operationen
  - Integer und mathematische Operatoren
  - Fließkommazahlen (Floats)
  - Strings
- Kontrollstrukturen und Vergleichsoperatoren
  - Branching (Verzweigung): `if / elif / else` und `match`
  - Loops (Schleifen): `for` und `while`
  - Funktionen: `def` und `return`

### Warum Python? (15 min)

Was macht Python besonders und unterscheidet es von anderen Programmiersprachen?

### Anwendungsbereiche von Python

- als "Skriptsprache"
- für Desktop- und Webanwendungen
- für Scientific Computing und Data Science
- für Machine Learning
- als Bindeglied zwischen Komponenten und Bibliotheken in anderen Sprachen
- Für Embedded Programming (MicroPython)

## **Kaffee/Teepause (20 min)**

### **Erweiterte Grundlagen in Python**

#### **Erweiterte Datentypen: Container (45 min)**

- Listen
- Tupel
- Dictionary
- Sets

#### **Übungsaufgaben**

#### **String-Operation und String-Formatierung und Ausgabe (45 min)**

- Escape-Sequenzen, Single-, Double- und Triple-Quotes und Klammerung
- Stringvergleiche und Testen auf Substrings
- Strings zusammenfügen, auftrennen und slicen
- Uppercase, Lowercase, Capitalization, Stripping, usw.
- String-Formatierung und Formatstrings

#### **Übungsaufgaben**

## **Mittagspause (12:30 - 13:30 Uhr)**

### **Tag 1 (Montag) Nachmittag (13:30 Uhr - 17:00 Uhr)**

#### **Zugriff auf Dateisystem (Teil 1) (20 min)**

- Öffnen und Lesen von Dateien
- Ganzen Dateiinhalt einlesen
- Textdateien zeilenweise einlesen
- Textzeilen in Datenfelder zerlegen

#### **Fehlerbehandlung in Python (45-60 min)**

- Was sind Exceptions?
- Abfangen von Fehlern
- Eigene Fehler erzeugen
- Eigene Exceptions definieren und benutzen

#### **Übungsaufgaben**

#### **Objektorientierung in Python (45-60 min)**

- Allgemeiner Einblick in die Objektstruktur in Python
- Definieren von Klassen
- Erzeugen von Objekten
- Vererbung
- Spezialmethoden von Objekten ("Dunder methods")

**Kaffee-/Teepause (20 min)**

**Übungsaufgaben**

**Tipps und Tricks im täglichen Umgang mit Python (10-20 min)**

**Zusammenfassung, Fragen, Lösungen usw.**

**Tag 2 (Dienstag) Vormittag (9:00 - 12:30 Uhr)**

**Rekapitulierung der Inhalte vom Vortag und Fragen (20-30 min)**

**Module in Python (45-60 min)**

- Importieren, verwenden von Modulen
- Allgemeiner Umgang mit dem Packet Manager PIP
- Eigene Module definieren
- Module und Pakete ('Packages')
- Der Standardmodulsuchpfad (`sys.path`)
- Relative Importe

**Kaffee-/Teepause (20 min)**

**Zugriff auf das Dateisystem (Teil 2)**

- Öffnen, Lesen und Schreiben von Dateien
- Text- und Binärdaten
- Text-Encodings
- Lesen und Schreiben von Daten im JSON Format
- Fehlerbehandlung und Context Manager (`with`-Statement)
- Dateiordner auflisten und durchsuchen

**Übungsaufgaben**

**Datenbankzugriff mit Sqlite3**

**Übungsaufgaben**

**Mittagspause (12:30 - 13:30 Uhr)**

**Tag 2 (Dienstag) Nachmittag (13:30 Uhr - 17:00 Uhr)**

**Entwicklung eines HTML-Crawlers (60-90 min)**

**Kaffee-/Teepause (20 min)**

**Zusammenfassung, Fragen, Lösungen usw. (30 min)**

**Best Practices (30 min)**

- Code Style
- Verwendung von Dekoratoren (falls Zeit)