



Python Grundlagen und Einführung in objektorientierte Programmierung

Schulung in Python für Anfänger



Python Grundlagen

Schulung in Python für Anfänger

Über den Kurs



Dauer: 3 Tage



Gruppengröße: 3-10



Level: Anfänger mit Programmiererfahrung*



Anteil Coding: 60%



Sprache: Python



Bibliotheken: os, numpy, sys, pandas

Auf einen Blick

- ✓ Wichtige Datenstrukturen
- ✓ Umgang mit Text / Strings
- ✓ Python als Schnittstelle zum Betriebssystem nutzen
- ✓ Eine eigene Funktion schreiben / Control Flows
- ✓ Ausnahmebehandlung mit Exceptions
- ✓ Ein- und Ausgabe von Daten
- ✓ Grundlage von NumPy
- ✓ Objektorientierte Programmierung
- ✓ Debugging, Testen, Aufbau eines Projekts

Überblick über das Seminar

Der moderne dreitägige Einführungskurs in Python erklärt umfassend die Grundlagen, um Python als Skriptsprache kennen zu lernen und Python als objektorientierte Programmiersprache benutzen zu können. Der Schwerpunkt wird hierbei auf die neuesten Entwicklungen in Python gelegt. Python Code überzeugt durch seine Übersichtlichkeit und leichte Wartbarkeit. Der Fokus liegt auf die praktische Umsetzung mit der Entwicklungsumgebung PyCharm, d.h. das Erklärte wird in vielen Übungen von den Teilnehmern vertieft. Am Ende des Seminars können Sie Programme in Python erstellen, verstehen die objektorientierte Programmierung und beherrschen eine der besten Programmiersprachen.

Inhalte des Seminars

Das Seminar gliedert sich in drei Bereiche:

- 1) Grundbegriffe, um Python in der Entwicklungsumgebung (IDE) PyCharm effizient benutzen zu können und die wichtigsten Daten-Strukturen werden behandelt. Des weiteren werden Grundlagen im Umgang mit Text erläutert. Überdies wird eine Einführung gegeben, um Python als Schnittstelle zum Betriebssystem benutzen zu können.
- 2) Die wichtigsten Kontrollstrukturen (If-Else, While, For-Schleife) werden erklärt. Erste eigene Funktionen mit einer eigener Dokumentation werden geschrieben. Fehler- und Ausnahmebehandlung mit exceptions wird erläutert. Überdies die Ein- und Ausgabe von Daten und die Grundlagen von numpy zur Berechnung bei arrays.
- 3) Der letzte Bereich erklärt die Grundlagen und Ideen der Objektorientierten Programmierung (OOP). Eigene Klassen zur OOP in Python werden definiert. Um Fehler im Code zu erkennen und robusten Code zu erstellen wird das Debugging von eigenem Code und Unittests erläutert.

Nach dem Seminar haben Sie einen umfassenden Einblick in die Grundlagen von Python, um Python als Skriptsprache zu verwenden und kennen die Grundlagen der objektorientierten Programmierung.

* Programmierkenntnisse in Python sind nicht notwendig. Erste Erfahrung mit einer anderen Sprache (z.B. R, VBA, C) ist ausreichend.

Die Einstiegshürde für die Benutzung von Python ist genommen, so dass Sie eigenständig Ihr Wissen nach dem Seminar erweitern können.

Ein wichtiger Bestandteil des Seminars ist das eigenständige Arbeiten und Lösen von Übungsaufgaben, so dass Sie mit direkter Hilfe des Trainers das Besprochene direkt in der Praxis umsetzen und anwenden können.

Wer sollte teilnehmen?

Dieser Kurs mit Python richtet sich an Programmierer mit Grundkenntnissen einer anderen Programmiersprache. Ihre Zielsetzung ist leicht wartbare Programme in Python erstellen zu können, denn Sie verstehen nach dem Seminar das Konzept der objektorientierten Programmierung in Python und der Python-Klassen. Sie können nun in einer der besten Programmiersprache erste Skripte programmieren.

Voraussetzungen

Dieser Python Kurs setzt keine Kenntnisse in Python voraus. Notwendig ist jedoch Vorerfahrung mit einer Programmiersprache, damit Konzepte einer Variablen, Zuweisung von Werten zu einer Variablen, der Hintergrund einer eine Funktion bzw. eine for-Schleife bekannt sind.

Das Seminar wird auf Deutsch gehalten. Englischkenntnisse (lediglich im Verstehen von englischen Texten) sind sehr empfehlenswert, da die Programmiersprache, Fachbegriffe und die Dokumentationen im Internet auf Englisch sind. Aus diesem Grund sind auch die erstellten Folien in der Schulung auf Englisch.

Methode des Seminars



Dieses Seminar ist sehr praxisorientiert. Die Teilnehmer arbeiten direkt und selbstständig mit der Programmiersprache Python in der Entwicklungsumgebung PyCharm, so dass das Erlernte direkt geübt und vertieft werden kann. Der Trainer moderiert dabei verschiedene Aufgaben und begleitet die Teilnehmer durch die einzelnen Lehreinheiten.

Technische Voraussetzungen der Teilnehmer (Laptop, etc.)

- ❑ Die Teilnehmer benötigen für die Übungsaufgaben Laptops. Wir empfehlen, Ihren eigenen Laptop mit der vorab installierten Software mitzubringen. Eine genaue Installationsanleitung für die Software wird Ihnen vor dem Seminar per E-mail zugesandt. Auf Anfrage stellen wir auch Schulungslaptops zur Verfügung.
- ❑ Bitte prüfen Sie, ob Ihr Firmenlaptop Zugangsbeschränkungen im Internet hat. Die digitalen Unterlagen (Skript, Code, Dateien) werden im Seminar online zum Download zur Verfügung gestellt. Sie erhalten vor dem Seminar per E-Mail den Link zu einer Testdatei zum Download, um dies überprüfen zu können.
- ❑ Sie sollten sich in firmenfremde WLAN-Netze registrieren können.
- ❑ Als Backup Lösung ist es möglich, dass der USB Port bei Ihrem Laptop freigeschaltet ist, um damit verwendete Dateien oder sonstige Unterlagen übertragen zu können.
- ❑ Im Seminar wird das Betriebssystem Windows verwendet. Der Umgang mit Ihrem verwendeten Betriebssystem und Laptop sollte bekannt sein. Insbesondere sollten Sie ohne Schwierigkeiten Sonderzeichen auf der Tastatur finden (insbesondere bei Apple Geräten werden auf manchen Tastaturen nicht immer runde, eckige bzw. geschweifte Klammern dargestellt).

10.00-10.15

Begrüßung und Organisatorisches

- Vorstellungsrunde
- Erwartungen der Teilnehmer

10.15-11.45

Grundlagen von Python

- Neue Pakete installieren
- PyCharm als Entwicklungsumgebung/IDE
- Funktionen und Methoden
- Wichtige Aspekte von Python im Vergleich zu anderen Programmiersprachen

11.45-12.00

Kaffeepause

12.00-13.30

Grundlegende Datenstrukturen

- Überblick über die basic data types (string, integer, float, NaN)
- Erläuterung der wichtigsten Datenstrukturen: list, tuple, dictionary
- List comprehension
- Slicing, um Daten zu extrahieren
- Konvertierungen zwischen Datentypen

13.30-14.30

Mittagspause

14.30-16.00

Umgang mit Text / Strings

- Übersicht über häufige Methoden für die Verarbeitung von Text, u.a.
- Häufigkeit eines Charakters zählen (count)
- Die Stelle eines Substrings finden (index, find)
- Strings bearbeiten (replace, split)
- Reguläre Ausdrücke (RegEx) in Python
- String Ausgabe mit der format-Methode

16.00-16.15

Kaffeepause

16.15-18.00

Python als Schnittstelle zum Betriebssystem

- Daten finden
- Daten und Ordner kopieren bzw. verschieben
- Daten und Ordner löschen
- Ordner anlegen
- Die Module os, sys, shutil

18.00

Ende

09.00-09.15

Rückblick und offene Fragen von Tag 1

09.15-10.45

Control Flows (Kontrollstrukturen)

- Eine eigene Funktion schreiben
- Default Parameter in einer Funktion setzen: position arguments und keyword arguments
- For Schleifen
- continue, break, if und else in Schleifen
- pass-Anweisung
- Globale und lokale Parameter
- While-Schleife
- List comprehension mit if-else

10.45-11.00

Kaffeepause

11.00-12.30

Einführung in die Ausnahmebehandlung mit Exceptions

- Fehler in Syntax und Semantik
- Ausnahmebehandlung
- Erzeugen von Ausnahmen
- Assertions
- try – except

12.30-13.30

Mittagspause

13.30-15.15

Ein- und Ausgabe (I/O von Daten)

- Dateien lesen und schreiben klassisch mit open()
- Weitere Datenformate (csv, excel,...) effizient einlesen mit pandas
- Python pickle

15.15-15.30

Kaffeepause

15.30-17.00

Grundlagen von numpy zur Berechnung mit Matrizen und Arrays

- Ein numpy array und dessen Attribute
- Arrays erstellen und mit Daten befüllen (bzw. mit Zufallszahlen)
- Mathematische Operationen mit Numpy
- Funktionen der Statistik umsetzen

17.00

Ende

09.00-09.15

Rückblick und offene Fragen von Tag 2

09.15-10.45

Einführung in die Objektorientierte Programmierung (OOP)

- Grundidee der objektorientierten Programmierung
- Beispiele zur Vertiefung des Verständnis
- Erste einfache Umsetzung einer Klasse in Python
- Alles in Python sind Objekte

10.45-11.00

Kaffeepause

11.00-12.30

Fortführung der Objektorientierten Programmierung

- Klassen, Instanzen und Objekte
- Instanz- und Klassenattribute, Properties
- privat, geschützt und öffentlich
- Operator-Überladung
- Vererbung
- Mehrfachvererbung

12.30-13.30

Mittagspause

13.30-15.15

Weiterführende Themen

- Iteratoren und Generatoren
- args* und kwargs** bei Funktionen
- Rekursive Funktionen und deren Komplexität
- Lambda-Operator, map, filter und reduce
- Itertools
- Sortierung mit 'key'-Funktionen

15.15-15.30

Kaffeepause

15.30-17.00

Testen und Debugging & Aufbau eines typischen Projekts

- Debugging
- Schreiben einfacher Unit-Tests
- Das Doctest-Modul
- Wie UnitTests und docstrings zusammenhängen
- Exemplarischer Aufbau eines Projekts

17.00

Ende

Ihre Dozenten

Unser folgender Experte leitet das Seminar



Dr. Rolf Köhler

Nach dem Studium der Mathematik und BWL promovierte er im Cyber Valley am Max-Planck Institut Tübingen. Sein Forschungsschwerpunkt war im Bereich Machine Learning und Bildverarbeitung. Seit 2015 arbeitet er bei der Robert Bosch GmbH im Bereich Deep Learning und implementiert und adaptiert verschiedene Algorithmen für industrielle Anwendungsfälle. Daraus sind mehrere Patentanmeldungen entstanden. Seit 7 Jahren verwendet er die Programmiersprache Python.

Zusammenfassung

€ Preise

1420 € zzgl. MwSt.

📅 Termin und Ort

Termine und Orte finden Sie unter <https://enable-ai.de>

Haben Sie Fragen? Wir helfen Ihnen. Versprochen.

📍 Enable AI, Stuttgart

☎ 0711 96881553

✉ info@enable-ai.de