

## Python Basiswissen (S1226)

Diese moderne, hochgradig objektorientierte Programmiersprache ermöglicht einen glatten Übergang von kleinen, einfachen Skripten zu sehr komplexen Anwendungen nach den neuesten technischen Standards. Python verliert dabei nie die Übersichtlichkeit und Pflegbarkeit seines Codes. Das Seminar hat drei Schwerpunkte: Zuerst werden Grundbegriffe aus der Python-Einführung kurz wiederholt, dann erweitert und spezialisiert (Funktionsparameter, Iteratoren, Generatoren, List Comprehensions, etc.). Danach wird auf das Konzept der Python-Klassen und auf die objektorientierten Eigenschaften der Sprache ausführlich eingegangen. Zuletzt wird eine Reihe von kurzen Einführungen in mehrere verschiedene spezielle Features der Sprache vorgenommen, einschließlich Threads, GUI und SQL-Datenbanken.

### Offene Termine

| Termin  | Tage | Freie Plätze | Ort  | Preis        |
|---|------|--------------|------|--------------|
| 08.04.-10.04.2019 ✓ <span style="border: 1px solid black; padding: 0 2px;">3=2</span> | 3    | >3           | Köln | € 1.330,00 * |
| 03.06.-05.06.2019 ✓ <span style="border: 1px solid black; padding: 0 2px;">3=2</span> | 3    | >3           | Köln | € 1.330,00 * |
| 05.08.-07.08.2019 ✓ <span style="border: 1px solid black; padding: 0 2px;">3=2</span> | 3    | >3           | Köln | € 1.330,00 * |
| 07.10.-09.10.2019 ✓ <span style="border: 1px solid black; padding: 0 2px;">3=2</span> | 3    | >3           | Köln | € 1.330,00 * |
| 02.12.-04.12.2019 ✓ <span style="border: 1px solid black; padding: 0 2px;">3=2</span> | 3    | >3           | Köln | € 1.330,00 * |
| 10.02.-12.02.2020 ✓ <span style="border: 1px solid black; padding: 0 2px;">3=2</span> | 3    | >3           | Köln | € 1.330,00 * |

#### \* Buchen ohne Risiko

- › Keine Vorkasse
- › Kostenloses Storno bis zum Vortag des Seminars
- › Rechnung nach erfolgreichem Seminar

#### ✓ Garantierter Termin und Veranstaltungsort

- € Preise zzgl. Mehrwertsteuer
- 3=2 Der dritte Mitarbeiter nimmt kostenlos teil

### Weitere Buchungsmöglichkeiten

|                           |  |
|---------------------------|--|
| <b>Firmenschulung</b>     | Schulung für Ihre Mitarbeiter mit individuellen Inhalten zum Wunschtermin im GFU-Schulungszentrum.                                 |
| <b>Inhouse-Schulung</b>   | Schulung für Ihre Mitarbeiter mit individuellen Inhalten zum Wunschtermin in Ihrem Hause.  |
| <b>Individualschulung</b> | Schulung für eine Einzelperson mit individuellen Inhalten zum Wunschtermin, wahlweise in Ihrem Hause oder im GFU-Schulungszentrum. |

## Schulungs-Ziel

Sie kennen den Weg, um mit Python unter Einbehaltung der neuesten Standards der Code-Qualität (Dokumentation, UnitTests, Objektorientierung und XML) zu entwickeln. Sie haben eine im Laufe des Seminars getestete Bibliothek von Utilities, die Sie sofort erweitern und miteinander kombinieren oder auch als Ansatzpunkt für komplexere Projekte nutzen können. Sie verstehen das Konzept der Python-Klassen von einem höheren Standpunkt aus und können dadurch knappen und leicht pflegbaren Python-Code schreiben.

## Wer sollte teilnehmen

Programmierer mit Grundkenntnissen der Python-Sprachelemente: Interaktiver Interpreter, Module, Packages, numerische Datentypen, Strings, Operatoren, Listen, Tupeln und Dictionaries. Etwas Erfahrung mit einer anderen objektorientierten Programmiersprache wie C# oder Java ist vorteilhaft.

## Organisation

### Teilnehmerzahl

min. 1, max. 8 Personen

### Seminarzeiten

3 Tage, 1. Tag 10:00 - 17:00 Uhr, Folgetage 09:00 - 16:00 Uhr

### Ort der Schulung

GFU-Schulungszentrum Köln oder bei Ihnen als Inhouse-Schulung

## Enthaltene Leistungen

### Im Preis enthalten:

- > Voll ausgestatteter Arbeitsplatz pro Teilnehmer
- > Fachbuch zum Seminar
- > Teilnahmezertifikat
- > Kostenloser persönlicher Parkplatz
- > Kostenloser Shuttle-Service
- > Frühstück, Snacks und Getränke ganztägig

## Inhalt

Diese Agenda ist auch Bestandteil (1.-3. Tag) des Kurses [Python Komplett](#)

- > **Organisation von Python-Code bei umfangreichen Projekten**
  - > Packages
  - > import
- > **list, tuple, dict (Zusammenfassung aus "Python-Einführung")**
- > **list, tuple dict, set (advanced features):**
  - > List Comprehensions
  - > Generator-Ausdrücke
  - > Iteratoren
  - > Mengen (sets)
  - > dict: update, get, setdefault, pop, items
- > **Funktionsparameter**
  - > Parameter-Defaults
  - > Funktionen mit beliebig langen Parameterlisten
  - > Packen und Entpacken von Listen, Tupeln und Dictionaries
  - > als Sequenzen von Funktionsparametern
- > **Typumwandlungen der Grunddatentypen**
  - > int, float, str, dict, list, tuple, set
- > **Verschachtelte Collections**
  - > Result sets, list of dictionaries
- > **Dateien, Objektserialisierung und Objektbanken**
  - > file
  - > shelf als Beispiel von pickle-Anwendung
- > **Python-Klassen (Python-Typen)**
  - > object methods
  - > static methods
  - > class methods
  - > get/set methods (Vergleich mit JavaBeans)
  - > properties
  - > filigrane Zugriffsrechte auf Objektattribute (Vergleich mit Java-Zugriffsrechten)
  - > Operatorenüberladung
  - > list-like objects
  - > dictionary-like objects
  - > callable objects
  - > Vererbung
- > **Exceptions**
  - > Eigene Fehlerklassen modellieren
  - > Eigene Fehlerobjekte in Methoden einbauen
- > **Das Dokumentationssystem von Python**
  - > das Objektattribut `__doc__` (docstring)
  - > reStructuredText als Syntax von docstring
  - > automatische Generierung von Dokumentation (epydoc)
- > **Python UnitTests (eine kurze Einführung)**

- › Mittagessen im eigenen Restaurant, täglich 6 Menüs, auch vegetarisch

## Haben Sie Fragen?

---

Gerne beraten wir Sie persönlich per [Mail](#) oder Telefon.

- › [info@gfu.net](mailto:info@gfu.net)
- › Infoline 0221 82 80 90

- › der Zusammenhang von UnitTests und docstrings
- › der Python-Interpreter als Betriebssystem
- › die Module os, sys, shutils
- › **Graphische Benutzeroberfläche (GUI) (eine kurze Einführung)**
- › **SQL-Datenbankanbindung mit Python (kurze Beispiele)**
- › **Threads (eine kurze Einführung)**

**Aufbaukurs im Anschluss:** [Python Aufbaukurs](#)