



## Python Basiswissen (S1226)

### Überblick

#### Beschreibung

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.

Termin	freie Plätze	Tage	Preis in Euro zzgl. MwSt.
08.10.-10.10.2012	>3	3	1.180,00 <span style="border: 1px solid black; padding: 0 2px;">3=2</span>

Bitte beachten Sie die aktuellen Termine und Preise auf unserer Webseite.

### Details

#### Seminar-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.

#### 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

### Buchungsalternativen

#### Firmenschulung

Inhouse-Schulung mit individuellen Inhalten zum Wunschtermin. Preis auf Anfrage.

#### Individualschulung

Schulung für eine Einzelperson mit individuellen Inhalten zum Wunschtermin. Preis auf Anfrage.

### Enthaltene Leistungen

Schulungsunterlagen  
Teilnahmezertifikat  
Frühstück und Mittagessen  
Getränke, Obst und Snacks

### Organisation

#### Teilnehmerzahl

min. 1, max. 8 Personen \*\*\* Der Kurs findet garantiert statt! \*\*\*

#### Seminarzeiten

1. Tag 10:00-17:00h, Folgetag(e)  
09:00-16:00h

#### Ort der Schulung

GFU-Schulungszentrum Köln oder bei Ihnen als Inhouse-Schulung

### Haben Sie Fragen?

Gerne beraten wir Sie persönlich per Mail oder Telefon.

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



- 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)**
  - 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](#)