

PL106 Python-Programmierung Advanced (Unix/Linux)

Kurzbeschreibung:

Mit Python ist es möglich, auch komplexere Programme übersichtlich und effizient zu verwirklichen. Dieser Kurs für Fortgeschrittene vermittelt das nötige Hintergrundwissen und die dazugehörige Programmierpraxis.

Zielgruppe:

- System-Administratoren
- Datenbank-Administratoren
- Applikations-Administratoren
- Netzwerk-Administratoren

Voraussetzungen:

Kenntnisse auf Administrationsebene von Unix/Linux-Systemen
Python-Kenntnisse vergleichbar dem Basis-Kurs [PL105 Python-Programmierung Basics \(Unix/Linux\)](#)

Sonstiges:

Dauer: 5 Tage

Preis: 2790 Euro plus Mwst.

Ziele:

Sie lernen, fortgeschrittene Programmier Techniken von Python zu verstehen, einzusetzen und zu optimieren:

- Funktionen,
- hierarchische Klassen,
- Performance-Tuning,
- Debugging

Zahlreiche Beispiele überführen das Verständnis in praxisnahe Codes. Der Kurs basiert auf Python 3.x.

Inhalte/Agenda:

- ◆ Python-Interna
 - ◇ Mutable vs. Immutable und damit verbundene Irrtümer
 - ◇ Magic Methods und ihre Zuordnung zu Operatoren und Funktionen
- ◆ Performance-Optimierung
 - ◇ Generatoren und Iteratoren statt vollständiger Objekt-Listen
 - ◇ Messung von Ausführungszeit, Memory- und CPU-Last
 - ◇ Paralleles Processing und Multithreading
 - ◇ Effiziente im Unterschied zu belastenden Code-Beispielen
- ◆ Knapper Code
 - ◇ Comprehensions für Listen, Dictionaries, Generatoren und Sets
 - ◇ Lambda-Funktionen, z.B. bei fortgeschrittenem Sortieren
 - ◇ Collections statt der Basis-Datentypen
 - ◇ Modularisierung über eigene Module und Pakete
- ◆ Fortgeschrittene Erstellung eigener Funktionen
 - ◇ Funktionen mit Default-Parametern und flexibler Parameter-Liste
 - ◇ Annotations in Funktions-Parametern
 - ◇ Geltungsbereich von Objekten innerhalb und außerhalb einer Funktion
 - ◇ Funktions-Dekoratoren
 - ◇ "yield" statt "return"
- ◆ Objektorientierung
 - ◇ Klassen, Instanzen, Metaklassen und Vererbung
 - ◇ Attribute und Methoden zu Klassen
 - ◇ Setter, Getter und Property-Attribute
 - ◇ Erstellung eigener Klassen
- ◆ Diverses
 - ◇ Advanced Regular Expressions: Look-Arounds, greedy vs. non-greedy, compile
 - ◇ Eigene Exceptions definieren und auslösen
 - ◇ Neue Features ab Python 3.6
- ◆ Debugging