PL105 Python-Programmierung Basics (Unix/Linux)

Kurzbeschreibung:

Einführung in die Grundlagen und die Programmierung mit Python (2.x/3.x)

Zielgruppe:

Das Training **PL105 Python-Programmierung Basics (Unix/Linux)** richtet sich an:

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

Voraussetzungen:

Um dem Lerntempo und den Lerninhalten des Kurses P**L105 Python-Programmierung Basics (Unix/Linux)** gut folgen zu können, benötigen Sie Kenntnisse auf Administrationsebene von Unix/Linux-Systemen.

Sonstiges:

Dauer: 5 Tage

Preis: 2790 Euro plus Mwst.

Ziele:

Python ist nicht nur flexibel einsetzbar und rasch programmiert, sondern durch die Unterstützung mehrerer Programmier-Paradigmen, speziell durch die Objekt-Orientierung mächtig und im Code gut lesbar.

Im Workshop **PL105 Python-Programmierung Basics (Unix/Linux)** lernen Sie, Python überall dort einzusetzen, wo die klassischen, OS-spezifischen Programmier-Tools wenig geeignet sind: effizientes Management selbst komplexer Datenstrukturen durch die leistungsfähige Objektstruktur, hohe Geschwindigkeit in der Verarbeitung der Daten, mächtiges Pattern Matching und OS-Unabhängigkeit des Codes. Der Kurs berücksichtigt zwar auch Python 2.x, fokussiert aber auf Python 3.x.

qSk!lls
innovation through education

Inhalte/Agenda:

- ♦ **Einführung in Programmier-Paradigmen: objekt-orientiert, prozedural, funktional, imperativ, deklarativ**
  ♦ **Grundlegendes zu Python, Unterschiede zwischen den Versionen 2.x und 3.x<**
  ♦ **Builtin-Objekttypen in Python, Unterscheidung von „mutable“ und „immutable“**
  ♦ **Mapping von Operatoren und globalen Funktionen zu „magic methods“**
  ♦ **NoneType und die Typen Boolean, Integer, Float und Complex**
  ♦ **Die Sequence-Typen String, Bytes, Bytearray, Tuple, List, Memoryview**
  ♦ **Die Set-Typen Set und Frozenset und der Mapping-Typ Dictionary**
  ♦ **Konvertierung von Datentypen**
  ♦ **Bedingungen, Match-Case, Short Circuit und Ternärer Operator**
  ♦ **Schleifen und Schleifenkontrolle**
  ♦ **Auffangen und Aufrufen von Exceptions**
  ♦ **Überblick über und Beispiele zur Python Standard-Library, Import**
  ♦ **OS-Interaktion: Dateien, System-Aufrufe, Environment**
  ♦ **Path Wildcards und Regular Expressions**
  ♦ **Eigene Funktionen, Parameter-Übergabe, Return- und Yield-Werte**
  ♦ **Erstellung eigener Packages**
  ♦ **Debugging in Grundzügen**
  ♦