Python 3.7

Language Features, OO Constructs, Framework, Pip Installers, IPython, Jupyter Notebooks, Testing

Python is the world's most versatile language. Unlike other languages that tend to be really good at one area but not so good at other areas, Python is really good at many areas. We see Python being successfully used for AI and machine learning, general application development, rich shell scripting, configuration, build systems, interactive notebooks, and lots more. Some of the largest technology companies in the world (e.g. Google) heavily use Python for their engineering and production systems. Some of the latest hot technologies (e.g. Keras and TensorFlow 2) use it extensively; so now is time for your team to use Python.

This fast-paced course covers all important aspects of Python programming., It is aimed at multi-disciplinary software engineers already experienced with object oriented programming using other languages. They will find much of their hard-earned knowledge easily transfers to Python programming – albeit delivered via a significantly simpler and more compact syntax. Invariably a given algorithm written in a different language when re-written in Python will result in smaller amounts of code, which is an excellent result (after all, the best developer writes the least amount of code).

Target Audience

Developers wishing to create modern apps using the very latest version of Python.

Prerequisites

Software developers with practical programming experience of an object-oriented languages such as C++, C#, or Java.

No prior Python experience needed.

Contents of One-Day Training Course

Python Tour

What Python offers
Feature tour
What make Python different from competing languages
Emphasis on clean syntax

Language Constructs

Common data types
Control flow
Loops
Functions
RegEx

OO Programming in Python

Classes: layout, methods and attributes
The __init__() method
Inheritance
Typing

Runtime Features

Memory management
Generators
Modules
Multithreading & locks (threading.py)

Error Handling

Raising and catching exceptions (try, raise, except, ..)
Designing with error handling in mind

Framework

The Python Standard Library offers:
* Collections
* File I/O
* Data access
* Network programming
* User interface

Pip

Standard installer
Pip usage
Python Packaging Index
Virtual environments
Python modules

(Interactive) IPython

Interactive shell that supports a wide range of Python features, from visualization to threading to data access
Also useful for and other languages

Jupyter Notebook

Jupyter Notebook mixes code, execution results, visualizations and markdown content in a single deliverable

Embedding Python

Many apps could benefit from a built-in macro language and Python is optimum
We explore how to easily embed Python runtime in your custom application

Testing

Exploring Python's testing infrastructure
Unit testing – what's similar & different
Mocking
Debugging

Interacting with C

Most OS APIs are written in C
How Python and C code can interact – threading, memory, lifecycles, exceptions
Handling common data types & constructs

Project

Using Python in a larger project to highlight its real-world capabilities