



Python Beginner

Python

About The Course

Starting from Python programming basics, this 3-day course will build up your Python skills so that you will write code with confidence and clarity in the end. This course will teach you how to use Python as a scripting and modular programming language.

Starting from understanding the syntax, formatting, indenting, and documenting code with comments and docstrings, you will learn Python programming basics such as using data types, operators, and control flow. Furthermore, you will learn how to define and use your own functions and classes; create and use collection types such as tuples, strings, ranges, lists, dictionaries, and sets; handle problems using exceptions; manipulate and generate collections using comprehensions, iterators, and generators. Students will also learn how to investigate classes, instances, polymorphism, and inheritance in Python, managing files and other resources, and finally, unit testing and debugging.

Duration: 3 days

Class size: 10 students max

Times: 9:00am - 5:00pm

Price: Refer to our website for current course and package pricing

After the course?

Each student will receive:

- Certificate of completion
- Training manual
- 12 months FREE email support
- FREE class re-sit (if necessary)

Who Should Do This Course?

This course is ideal for those who have never used Python before or would like to consolidate their understanding of the fundamentals of Python, so they have a firm foundation to build on.

Prerequisites

Although students would undoubtedly benefit from having some basic programming skills in other languages, especially object-oriented languages, it is not assumed. This course will cover everything you need to know to get started with Python programming.

Content

Unit 1: Getting Started

- Obtaining and Installing Python
- Starting Python Command Line Repl
- Indenting and Structuring Code
- Understanding the Culture of Python
- Importing Standard Library Modules
- Getting Help
- Understanding Scalar Data Types: Integers, Floats, None and Bool

- Using Relational Operators
- Introducing Control Flow: If-Statements and While-Loops

Unit 2: Strings and Collections

- Understanding Strings
- Understanding Bytes
- Introducing Lists

- Introducing Dictionaries
- Iterating Over Series of Items with For-Loops

Unit 3: Modularity

- Organizing Code in a .Py File
- Defining Functions
- Organizing Our Module into Functions
- Investigating the Python Execution Model
- Setting Up a Main Function with Command Line Argument
- Including Docstrings in Code
- Adding Comments to Code

Unit 4: Built-in Types and the Object Model

- Understanding Python Object References
- Argument Passing Semantics – Pass By Object-Reference
- Using Python's Return Semantics
- Investigating Function Arguments in Detail
- Understanding the Python Type System
- Investigating Variable Declaration and Scoping

Unit 5: Exploring Built in Collection Types

- Understanding Tuples
- Using the Tuple Constructor
- Investigating the Use of Strings
- Using Range
- Using Lists
- Using Dictionaries
- Using Sets
- Understanding Collection Protocols

Unit 6: Exceptions

- Introducing Exceptions and Control Flow
- Handling Exceptions
- Handling Multiple Exceptions
- Investigating Programmer Errors
- Defining Empty Blocks Using the Pass Statement
- Understanding Exception Objects
- Avoiding Return Codes
- Re-Raising Exceptions
- Understanding the Importance of Exceptions
- Validating Data
- Understanding Exceptions, Apis, and Protocols
- Choosing Not To Guard Against Typeerror
- Using Pythonic Style – Deciding between Eafp and Lbyl
- Clean-Up Actions
- Handling Platform-Specific Code

Unit 7: Comprehensions, Iterables and Generators

- Introducing Comprehensions
- Utilising Iteration Protocols
- Creating Generator Functions
- Creating Generator Expressions
- Utilising the “Batteries Included” Iteration Tools
- Pulling It All Together

Unit 8: Defining New Types with Classes

- Defining Classes
- Defining Instance Methods
- Defining Instance Initializers
- Adding Validation and Invariants
- Adding a Second Class
- Understanding Collaborating Classes
- Naming Methods for Implementation Details
- Deciding Between a Function and an Object
- Understanding Polymorphism and Duck-Typing
- Utilising Inheritance and Implementation Sharing

Unit 9: Files and Resource Management

- Understanding Files in Python
- Introducing Context Managers
- Using Binary Files
- Developing Code for Other File-Like Objects
- Using Other Resources

Unit 10: Unit Testing With the Python Standard Library

- Creating Test Cases
- Adding Fixtures
- Defining Assertions
- Developing Unit Testing Through an Example.
- Using Fixtures to Create Temporary Files
- Using the New Fixtures
- Using Assertions to Test Behaviour
- Testing For Exceptions
- Testing For File Existence

Unit 11: Debugging with Pdb

- Introducing the Debugging Commands
- Using the Command to Debug Code