Dynamic web training
The Ultimate Training Experience

Phone: 1300 888 724
Email: info@dynamicwebtraining.com.au

Python Intermediate

Python

About The Course

This 3-day Python intermediate course extends the skills you have learned at the Beginner level. Highlights of this course include learning how to organise your code with packages; write and use lambda expressions; and use decorators and closures to extend your code. Students will also learn how to define properties, static and class methods, and override them. Further into this Python course, you will learn about formatting and managing strings and representations and dealing with all the different types of numbers, including complex numbers. Other focus areas of this course include managing inheritance and subtypes; understanding collections and protocols; handling exceptions and assertions; using context managers using 'with,' and introspection.

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 know the basic principles of Python but would like to delve deeper into writing code and better understanding not just what to do but why we do it. The course will enable you to develop the skill required to be a professional well-rounded Python programmer.

Prerequisites

The course assumes you have completed the beginners' course or have equivalent knowledge of Python Programming, including using strings, bytes, bool, and the other datatypes. You will also need to understand control flow, collections, and generators. It is also assumed you know how to create your own classes, functions, and exceptions and understand the concepts of polymorphism, inheritance. You will also need to be able to use Python in modules as well as scripting.

Content

Unit 1: Organizing Larger Programs

- Introducing Packages
- Implementing Packages
- Using Relative Imports
- Using __All__
- Understanding Namespace Packages
- Defining Executable Directories
- Defining Executable Packages
- Investigating Recommended Layouts
- Understanding Modules as Singletons

Unit 2: Beyond Basic Functions

- Reviewing Functions
- Understanding Functions as Callable Objects
- Understanding Callable Instances and Using the __Call__() Method
- Understanding Classes are Callable
- Leveraging Callable Classes
- Using Lambdas
- Detecting Callable Objects
- Extending Formal Parameter Syntax
- Extending Call Syntax
- Transposing Tables

Phone: 1300 888 724
Email: info@dynamicwebtraining.com.au

Unit 3: Closures and Decorators

- Understanding Local Functions
- Understanding Closures and Nested Scopes
- Using Function Decorators
- Validating Arguments

Unit 4: Properties and Class Methods

- Using Class Attributes
- Using Static Methods
- Using Class Methods
- Developing Named Constructors
- Overriding Static- and Class-Methods
- Understanding Properties
- Overriding Properties
- Understanding the Template Method Pattern

Unit 5: Strings and Representations

- Understanding the Two String Representations
- Using Repr()
- Using Str()
- Understanding When to Use Representation
- Using Format()
- Leveraging Reprlib for Large Strings
- Using the Ascii(), Ord() and Chr() Built-in Functions

Unit 6: Numeric and Scalar Types

- Using Python's Basic Numeric Types
- Understanding the Limits of Floats
- Using the Decimal Module
- Using the Fractions Module
- Using Complex Numbers
- Using Built-in Functions Relating to Numbers
- Using Dates and Times with the Datetime Module

Unit 7: Iterables and Iteration

- Reviewing Comprehensions
- Understanding Multi-Input Comprehensions
- Understanding Functional-Style Tools
- Using Map()
- Using Filter()
- Using Functools.Reduce()
- Combining Functional Concepts: Map-Reduce
- Investigating the Iteration Protocols

Unit 8: Inheritance and Subtype Polymorphism

- Reviewing Single Inheritance
- Using Type Inspection
- Understanding Multiple Inheritance
- Investigating Method Resolution Order
- Using Super()
- Understanding Object
- Using Inheritance for Implementation Sharing

Unit 9: Implementing Collections with Protocols

- Understanding Collection Protocols
- Using Test First
- Using the Initializer
- Investigating the Container Protocol
- Investigating the Sized Protocol
- Investigating the Iterable Protocol
- Investigating the Sequence Protocol
- Investigating the Set Protocol

Unit 10: Errors and Exceptions in Depth

- Understanding Exception Dangers
- Using Exceptions Hierarchies
- Utilising Exception Payloads
- Creating User-Defined Exceptions
- Using Exception Chaining
- Using Tracebacks
- Using Assertions
- Using Assertions to Enforce Preconditions, Postconditions

Unit 11: Defining Context Managers

- Understanding the Context Manager
- Implementing a Context Manager
- Using Contextlib.Contextmanager
- Handling Multiple Context-Managers in a with-Statement
- Using Context Managers for Transactions

Unit 12: Introspection

- Introspecting Types
- Introspecting Objects
- Introspecting Scopes
- Using the Inspect Module
- Building an Object Introspection Tool