

Python Essentials Training (5 Days)

Course Outline

Learn Python programming quickly! This hands-on class gives you practical experience building applications using Python. You will learn to use Python's built-in modules, structure your code efficiently, manage files, handle exceptions, create reusable code, test and debug code, access data, and much more. Whether you are a new programmer or an existing programmer who is new to Python, this class gives you a solid foundation in Python programming.

Course Benefits

- Learn to create video demos.
- Learn to work with styles.
- Learn to create question pools and random question slides that make it difficult for learners to share quiz content.
- Learn Section 508 best practices.
- Learn to work with widgets and learn how to add them to a project.
- Learn to add multiple and conditional actions to your buttons and click boxes.
- Learn to leverage Captivate's collaboration tools.
- Learn to create a consistent look and feel across your projects using themes.
- Learn how to track quiz results.

Course Outline

Python Basics

Getting Familiar with the Terminal
Running Python
Running a Python File
Exercise: Hello, world!
Literals
Exercise: Exploring Types
Variables

Exercise: A Simple Python Script
Constants and Deleting Variables
Writing a Python Module
print() Function
Collecting User Input
Exercise: Hello, You!
Reading from and Writing to Files
Exercise: Working with Files

Functions and Modules

- Defining Functions
- Variable Scope
- Global Variables
- Function Parameters
- Exercise: A Function with Parameters
- Returning Values
- Exercise: Parameters with Default Values
- Returning Values
- Importing Modules
- Methods vs. Functions

Math

- Arithmetic Operators
- Exercise: Floor and Modulus
- Assignment Operators
- Precedence of Operations
- Built-in Math Functions
- The math Module
- The random Module
- Exercise: How Many Pizzas Do We Need?
- Exercise: Dice Rolling

Python Strings

- Quotation Marks and Special Characters
- String Indexing
- Exercise: Indexing Strings
- Slicing Strings
- Exercise: Slicing Strings
- Concatenation and Repetition
- Exercise: Repetition
- Combining Concatenation and Repetition
- Python Strings are Immutable
- Common String Methods
- String Formatting
- Exercise: Playing with Formatting
- Formatted String Literals (f-strings)
(introduced in Python 3.6)
- Built-in String Functions
- Exercise: Outputting Tab-delimited Text

Iterables: Sequences, Dictionaries, and Sets

- Definitions
- Sequences

- Lists
- Sequences and Random
- Exercise: Remove and Return Random Element
- Tuples
- Ranges
- Converting Sequences to Lists
- Indexing
- Exercise: Simple Rock, Paper, Scissors Game
- Slicing
- Exercise: Slicing Sequences
- min(), max(), and sum()
- Converting between Sequences and Strings
- Unpacking Sequences
- Dictionaries
- The len() Function
- Exercise: Creating a Dictionary from User Input
- Sets
- *args and **kwargs

Virtual Environments, Packages, and pip

- Exercise: Creating, Activating, Deactivating, and Deleting a Virtual Environment
- Packages with pip
- Exercise: Working with a Virtual Environment

Flow Control

- Conditional Statements
- Compound Conditions
- The is and is not Operators
- all() and any() and the Ternary Operator
- In Between
- Loops in Python
- Exercise: All True and Any True
- break and continue
- Looping through Lines in a File
- Exercise: Word Guessing Game
- The else Clause in Loops
- Exercise: for...else
- The enumerate() Function
- Generators
- List Comprehensions

Exception Handling

- Exception Basics
- Generic Exceptions
- Exercise: Raising Exceptions
- The else and finally Clauses
- Using Exceptions for Flow Control
- Exercise: Running Sum
- Raising Your Own Exceptions

Python Dates and Times

- Understanding Time
- The time Module
- Time Structures
- Times as Strings
- Time and Formatted Strings
- Pausing Execution with time.sleep()
- The datetime Module
- datetime.datetime Objects
- Exercise: What Color Pants Should I Wear?
- datetime.timedelta Objects
- Exercise: Report on Departure Times

File Processing

- Opening Files
- Exercise: Finding Text in a File
- Writing to Files
- Exercise: Writing to Files
- Exercise: List Creator
- The os Module
- os.walk()
- The os.path Module
- A Better Way to Open Files
- Exercise: Comparing Lists

PEP8 and Pylint

- PEP8
- Pylint

Advanced Python Concepts

- Lambda Functions
- Advanced List Comprehensions
- Exercise: Rolling Five Dice
- Collections Module
- Exercise: Creating a defaultdict

Counters

- Exercise: Creating a Counter
- Mapping and Filtering
- Mutable and Immutable Built-in Objects
- Sorting
- Exercise: Converting list.sort() to sorted(iterable)
- Sorting Sequences of Sequences
- Creating a Dictionary from Two Sequences
- Unpacking Sequences in Function Calls
- Exercise: Converting a String to a datetime.date Object
- Modules and Packages

Regular Expressions

- Regular Expression Tester
- Regular Expression Syntax
- Python's Handling of Regular Expressions
- Exercise: Green Glass Door

Working with Data

- Virtual Environment
- Relational Databases
- Passing Parameters
- SQLite
- Exercise: Querying a SQLite Database
- SQLite Database in Memory
- Exercise: Inserting File Data into a Database
- Drivers for Other Databases
- CSV
- Exercise: Finding Data in a CSV File
- Creating a New CSV File
- Exercise: Creating a CSV with DictWriter
- Getting Data from the Web
- Exercise: HTML Scraping
- XML
- JSON
- Exercise: JSON Home Runs

Testing and Debugging

- Testing for Performance
- Exercise: Comparing Times to Execute
- The unittest Module
- Exercise: Fixing Functions
- Special unittest.TestCase Methods

Classes and Objects

- Attributes
- Behaviors
- Classes vs. Objects
- Attributes and Methods
- Exercise: Adding a roll() Method to Die
- Private Attributes
- Properties
- Exercise: Properties
- Objects that Track their Own History
- Documenting Classes
- Exercise: Documenting the Die Class
- Inheritance
- Exercise: Extending the Die Class
- Extending a Class Method
- Exercise: Extending the roll() Method
- Static Methods
- Class Attributes and Methods
- Abstract Classes and Methods
- Understanding Decorators