



PYTHON BOOTCAMP

OVERVIEW:

This course is customized to make the learner understand the basics of python to advanced level applications in python. As an extension to it, we added machine learning and artificial intelligence concepts as well

CONTENTS

Module 1:

- Introduction to Python
- Future of Python
- Features, Reasons to learn and Interesting Python facts
- Python best practices
- Python syntax, setup and applications

Module 2:

- Python variable and data types
- Advantages over Java
- Variable scope and Identifiers
- Python Operators and exercises on the same
- Data structures, Lists and tuples
- Dictionaries

Module 3:

- Python Decision making
- Looping
- Functions
- Range(), Zip(), eval() and exec() functions hands-on

Module 4:

- Classes
- Methods
- Methods vs functions



Constructors in python
Objects
Inheritance
Complete object-oriented programming on python
Python os module
Python I/O

Module 5:

Python calendar module
Python date time
Exception handling
Python tools
XML processing
Sending email with python 3
GUI programming
Scipy and Numpy Modules

Detail Course contents

An Introduction to Python

- 1) Introductory Remarks about Python
- 2) Strengths and Weaknesses
- 3) A Brief History of Python
- 4) Python Versions
- 5) Installing Python
- 6) Environment Variables
- 7) Executing Python from the Command Line
- 8) IDLE
- 9) Editing Python Files
- 10) Getting Help
- 11) Dynamic Types
- 12) Python Reserved Words
- 13) Naming Conventions

Basic Python Syntax

- 1) Basic Syntax
- 2) Comments
- 3) String Values
- 4) String Operations
- 5) The format Method
- 6) String Slices



- 7) String Operators
- 8) Numeric Data Types
- 9) Conversions
- 10) Simple Input and Output
- 11) The print Function

Language Components

- 1) Control Flow and Syntax
- 2) Indenting
- 3) The if Statement
- 4) Arithmetic operators
- 5) Assignment operators
- 6) Relational Operators
- 7) Logical Operators
- 8) True or False
- 9) The while Loop
- 10) Break, continue and pass statement
- 11) For loop

Collections

- 1) Introduction
- 2) Lists
- 3) Tuples
- 4) Sets
- 5) Dictionaries
- 6) Sorting Dictionaries
- 7) List Comprehension
- 8) Nested List Comprehension
- 9) Dict Comprehension
- 10) Iterators
- 11) Generators

Functions

- 1) Defining Your Own Functions
- 2) Parameters
- 3) Function Documentation
- 4) Keyword and Optional Parameters
- 5) Default Parameters
- 6) Passing Collections to a Function
- 7) Variable Number of Arguments
- 8) Scope
- 9) Passing Functions to a Function - Decorators
- 10) Lambda

Modules



-
- 1) Standard Modules - sys
 - 2) Standard Modules - math
 - 3) Standard Modules – time
 - 4) Standard Modules - os
 - 5) The dir Function

Exceptions

- 1) Errors
- 2) Run Time Errors
- 3) The Exception Model
- 4) Exception Hierarchy
- 5) Handling Multiple Exceptions
- 6) raise
- 7) assert
- 8) Writing Your Own Exception Classes

Input output

- 1) Data Streams
- 2) Creating Your Own Data Streams
- 3) Access Modes
- 4) Writing Data to a File
- 5) Reading Data From a File
- 6) Additional File Methods
- 7) Handling IO Exceptions
- 8) Working with Directories
- 9) The pickle module

Classes in Python

- 1) Classes in Python
- 2) Principles of Object Orientation
- 3) Creating Classes
- 4) Instance Methods
- 5) File Organization
- 6) Special Methods
- 7) Class Variables
- 8) Inheritance
- 9) Polymorphism
- 10) Type Identification
- 11) Custom Exception Classes
- 12) MRO Technique

Regular Expression



- 1) Simple Character Matches
- 2) Special Characters
- 3) Character Classes
- 4) Quantifiers
- 5) The Dot Character
- 6) Greedy Matches
- 7) Grouping
- 8) Matching at Beginning or End
- 9) Match Objects
- 10) Compiling Regular Expressions

Writing GUI's in Python

- 1) Introduction
- 2) Components and Events
- 3) An Example GUI
- 4) The root component
- 5) Adding a button
- 6) Entry Widget
- 7) Text Widget
- 8) Check button
- 9) Radio button
- 10) Listboxes
- 11) Menus

Note:

1. Every module will have an exercise (project) on completion
2. 80% of the concepts are dealt with hands-on coding rest 20% is dedicated to conceptual understanding of ML, Python and AI
3. Modules 5 to 7 will have two capstone exercises covering the same concepts
4. Capstone exercises are real-time project cases.