Online Python Developer Course

155 Hours / 6 Months

Course Description

Whether you're new to programming or just want to learn a new language, this in-depth course will teach you the ins and outs of Python programming. In this comprehensive course you will cover introductory through advanced methods of Python and begin coding quickly.

Start by learning the basics of programming in Python including how it works and what it's good for. You will also gain an understanding of Python's place in the wider programming world. Then move on to more advanced methods where you'll learn how to work with iPhone Notebook, the Collections Module, regular expressions, databases, CSV files, JSON, and XML. You will also learn advanced sorting, how to write object-oriented code, and how to test and debug your code. Finally, you'll get a rapid introduction to NumPy, pandas, and matplotlib, which are Python libraries.

Hardware Requirements

- This course can be taken on either a PC or Mac.
- Dual monitors are helpful but NOT required

Software Requirements

- PC: Windows XP (Service Pack 2) and later operating systems.
- Mac: OS X Snow Leopard 10.6 or later
- Browser: The latest version of Google Chrome or Mozilla Firefox are preferred. Opera and Safari are also compatible.
- Our Introduction to Python course uses IDLE, a simple Python IDE/editor that comes bundled with Python. All you need for class is to have Python 3.3 or later installed and to download the class files.
- Adobe Flash Player. Click here to download the Flash Player.
- Adobe Acrobat Reader. Click here to download the Acrobat Reader.
- Software must be installed and fully operational before the course begins.
- Email capabilities and access to a personal email.

Prerequisites

There are no mandatory prerequisites prior to taking Python Developer. However, prior knowledge of any programming language is helpful.

Course Outline

MODULE	TOPICS COVERED
Introduction to Python	

TU TOWSON UNIVERSITY

1: Python Basics	• Running Python • Hello, World! • Literals • Python Comments • Data Types • Variables	• Writing a Python Module • print( ) Function • Named Arguments • Collecting User Input • Getting Help
2: Functions and Modules	• Defining Functions • Variable Scope • Global Variables	• Function Parameters • Returning Values • Importing Modules
3: Math	• Arithmetic Operators • Modulus and Floor Division • Assignment Operators • Built-in Math Functions	• The math Module • The random Module • Seeding
4: Python Strings	• Quotation Marks and Special Characters • String Indexing • Slicing Strings • Concatenation and Repetition	• Common String Methods • String Formatting • Built-in String Functions
5: Iterables: Sequences, Dictionaries, and Sets	• Definitions • Sequences • Unpacking Sequences • Dictionaries	• The len( ) Function • Sets • *args and **kwargs
6: Flow Control	• Conditional Statements • The is and is not Operators • Python's Ternary Operator • Loops in Python	• The enumerate( ) Function • Generators • List Comprehensions
7: File Processing	• Opening Files	• The os and os.path Modules
8: Exception Handling	• Wildcard except Clauses • Getting Information on Exceptions • The else Clause	• The finally Clause • Using Exceptions for Flow Control • Exception Hierarchy
9: Dates and Times	• Understanding Time • The time Module	• The datetime Module
10: Running Python Scripts from the Command Line	• sys.argv	
11: Introduction to Python Final Exam	• Introduction to Python Final Exam	

Advanced Python		
1: IPython Notebook	• Getting Started with IPython Notebook • Creating Your First IPython Notebook • IPython Notebook Modes • Useful Shortcut Keys	• Markdown • Magic Commands • Getting Help
2: Advanced Python Concepts	• Advanced List Comprehensions • Collections Module • Mapping and Filtering • Lambda Functions	• Advanced Sorting • Unpacking Sequences in Function Calls • Modules and Packages
3: Regular Expressions	• Regular Expression Syntax	• Python's Handling of Regular Expressions
4: Working with Data	• Databases • CSV • Getting Data from the Web	• HTML • XML • JSON
5: Classes and Objects	• Creating Classes • Attributes, Methods and Properties • Extending Classes	• Documenting Classes • Static, Class, Abstract Methods • Decorators
6: Testing and Debugging	• Creating Simulations • Testing for Performance	• The unittest Module
7: Unicode and Encoding	• Encoding and Decoding Files in Python	• Converting a File from cp1252 to UTF-8
8: Advanced Python Final Exam	• Advanced Python Final Exam	

Python Data Analysis with NumPy and pandas		
1: NumPy	• One-dimensional Arrays • Multi-dimensional Arrays • Getting Basic Information about an Array • NumPy Arrays Compared to Python Lists	• Universal Functions • Modifying Parts of an Array • Adding a Row Vector to All Rows • Random Sampling
2: Pandas	• Series and DataFrames • Accessing Elements from a Series • Series Alignment • Comparing One Series with Another • Element-wise Operations • Creating a DataFrame from NumPy Array • Creating a DataFrame from Series • Creating a DataFrame from a CSVI	• Getting Columns and Rows • Cleaning Data • Combining Row and Column Selection • Scalar Data: at[ ] and iat[ ] • Boolean Selection • Plotting with matplotlib
3: Python Data Analysis with NumPy and pandas Final Exam	• Python Data Analysis with NumPy and pandas Final Exam	

Python Programmer Final Exam	
1: Python Programmer Final Exam	• Python Programmer Final Exam

Python Programmer Final Project	
1: Python Programmer Final Project	• Python Programmer Final Project