Python for Data Analysts (Certification Course)

Please read our Terms and Conditions and our Privacy Policy.

Duration: 32 hours

Learning Overview:

This course will give you a solid understanding of the Python building blocks and an introduction to the pandas module for data analytics.

Completion of this course will help you attain PCEP professional certification from the Python Institute.

There are mini quizzes and hands-on exercises throughout to help assess and reinforce your learning with access to FREE bonus material.

Learning Outcomes:

By the end of this course, you will be able to:

1. Explain the concept of Python's Object Orientated Programming (OOP) framework.
2. Describe the basic concepts of Python syntax.
3. Describe what Python data objects are and how they relate to a Class.
4. Demonstrate how to implement common programming functionality such as data storage, functions, loops and conditional logic etc, within Python.
5. Use the Python pandas package to access, manage, analyse and report on data tables.

Delivery Approach:

This course is delivered using an approach of self-paced eLearning to be taken in your own time and at your own pace.

eLearning Modules x 24:	approx 16 hours eLearning plus 16 hours for hand-on exercises

Pre-requisites:

This course is aimed at people that are new to programming in Python.

For the hands-on practice activities in the course, you will need access to an environment that runs Python. On our course we signpost you to Anaconda, a free open source platform, to run Python via Jupyter Notebook.

Anaconda must have a suitable operating platform installed:

- Windows 8 or newer
- 64-bit macOS 10.13+
- or Linux, including Ubuntu, RedHat, CentOS 6+, and others.

Full system requirements can be found here.

Learning Modules:

Introduction to Python

Learning Objective: Explain the history of Python and its common uses today

- Explain Python's common uses and benefit
- Describe how Python was founded and grew into the language we know today

Jupyter Notebooks

Learning Objective: Explain what an Integrated Development Environment is and describe how use Jupyter Notebook.

- Identify the usefulness of integrated development environments (IDE)
- Explain why the Jupyter Notebook IDE is a great tool for getting started in Python
- Discuss the main functionality of Jupyter Notebook

Python Syntax

Learning Objective: Describe the basic concepts of Python syntax.

- Outline Python's syntax rules with regards to lines, indentation and tokens
- Discuss the print() function

Python Variables

Learning Objective: Describe variable naming conventions, storage concepts and data types used in Python.

- Identify the rules with regards to variable naming conventions
- Name the five native data types of Python
- Describe the key elements of a variable assignment statement
- Outline how variables are stored and the importance of knowing a memory location

Everything is Data

Learning Objective: Explain what data objects are and how they relate to a class.

- Identify what a class is and why it's important
- Describe what objects are and their relation to properties and methods
- Explain how to call the properties and methods within your code
- Discuss what is meant by the term 'Everything is Data'
- Explain what is meant by the term 'Object-orientated programming language'

Help Functions

Learning Objective: Know where and how to access help in Python

- Locate help topics within python, using the help() function and/or interactive menu
- Identify the attributes available for any object, using the dir() function
- Outline other useful help options in Jupyter Notebook.

Lists, Tuples & Dictionaries

Learning Objectives: How to initialise and use lists, tuples and dictionaries

- Describe the differences between list, tuples and dictionaries and when to use each
- List the governing classes for these data structures as well as some useful methods
- Expand on the intricacies of memory storage for mutable and immutable data types.

Functions

Learning Objectives: Explain how Python user-defined functions can help you to write more efficient and dynamic code

- Use Python to create your functions that help you perform common task more efficiently
- Demonstrate how to apply a function to an iterable object
- Discuss the pros and cons of lambda functions

Loops

Learning Objective: Explain how Python loops can help you to write more efficient code for performing repetitive tasks

- Use iterative and conditional loops within the Python environment
- Describe the functions and methods available for use that let you manipulate data structures such as lists and dictionaries, whilst iterating through them
- Differentiate when to use iterative for loops versus conditional while loops

Branches

Learning Objective: Using conditional logic within your Python code

- Expand on basic conditional logic by using logical operators to link multiple conditions
- Use conditional logic with iterative loops to perform operations on iterative objects such as lists and dictionaries
- Use the ternary operator to simplify basic if-else conditional logic

Importing Modules

Learning Objectives: Explain the benefits of Python modules and how to use them

- Use modules to perform a multitude of programming tasks not available to native Python
- Solve the complexities of importing modules by using good practice techniques such as aliasing or importing only the functionality that you require

Dates

Learning Objectives: Explain Python date and time objects

- Describe how dates and datetimes are handled in Python
- Use the datetime and dateutil modules to work with and manipulates dates and times
- Describe the most useful function and methods available for date and datetime objects

Pandas Introduction

Learning Objectives: Explain how the pandas module allows for data analytics in Python

- Describe the pandas Series object
- Describe the pandas DataFrame object
- Describe the difference between the Series and DataFrame objects

Creating Data

Learning Objectives: Using pandas to create DataFrames

- Use pandas to manually create DataFrames from Python dictionaries
- Use pandas to create DataFrames from an existing DataFrame
- Explain the possible memory storage problems you are likely to run into when using pandas DataFrames
- Solve possible memory storage implications of using pandas DataFrames

Data Input/Output

Learning Objective: How to importing and export pandas DataFrames

- Loading external data files into pandas DataFrames
- Exporting DataFrames from pandas into external files
- Supported file types
- Managing data types

Metadata

Learning Objective: How to investigate the metadata of a pandas DataFrame

- Use properties and methods of the DataFrame class to return metadata
- Outline why DataFrame properties and methods are useful for getting to know your data

Understanding Data

Learning Objective: How to investigate the data portion of a pandas DataFrame

- Use properties and methods of the DataFrame class to give you a quick look at the data within
- Demonstrate how to implement indexing syntax to subset rows and columns of a DataFrame
- Use the describe() method to return summary statistics for numeric columns in a DataFrame

Querying & Updating Data

Learning Objective: Using the loc[I,C] property to query and update a pandas DataFrame

- Describe how the loc[I,C] property allows you to subset rows and columns of a DataFrame object
- Use the loc[I,C] property to subset DataFrames in place or create and store new DataFrames
- Demonstrate how to implement boolean indexers to return subsets of data based on conditional logic.

Iteration

Learning Objective: Using iteration techniques to update a pandas DataFrame

- Discuss when iterating through a DataFrame is useful
- Demonstrate an understanding of how to use the iterrows() and apply() methods for iterating through a DataFrame
- Demonstrate an understanding of how to use vectorisation for iterating through a DataFrame
- Identify the pros and cons of each iteration technique

Branches in a Pandas Context

Learning Objective: Using suitable techniques to update a pandas DataFrame cell based on conditional logic

- Describe when to use the pandas Series where() method
- Use iterative methods, together with conditional logic, to update full columns in a DataFrame.

Cleaning Data

Learning Objective: How to identify and dealing with missing data

- Identify missing data, including the rows and columns they are within
- Compare the different methods available for dealing with missing data
- Use the rename() method to update row and column indexes
- Outline the different options available for cleaning your data

Sorting Data

Learning Objective: How to sort pandas DataFrames

- Identify methods for sorting DataFrames based in data columns
- Identify methods for sorting or resetting DataFrames based on their index
- Use the drop_duplicates() method to remove duplicate rows of data, based on key columns

Combining Data Vertically

Learning Objective: How to combine pandas DataFrames vertically

- Use the pandas concat() function and DataFrame append() method to join DataFrames vertically
- Discuss the different options available within the concat() function and append() method

Combining Data Horizontally

Learning Objective: How to combine pandas DataFrames horizontally

- Use the pandas concat() function and DataFrame merge() method to join DataFrames horizontally
- Explain the special situation in which the pandas concat() function is applicable for merging pandas DataFrames
- Discuss the different options available within the merge() method and how they can be used to perform a host of different merge types