

PYTHON ESSENTIALS

Syllabus



WHO SHOULD TAKE THIS COURSE?

- This course assumes no previous experience with Python. However, the student should have some prior experience with another programming language, such as JavaScript, PHP, Java, Perl or C++.
- Students should be already comfortable with basic programming concepts, such as variables, boolean expressions, flow control, loops, functions, arrays and common data structures.
- Students must already know how to work with files, directories, simple text editors, command shells, environment variables, internet connections and other essential aspects of using a computer for software development.
- This course is ideal for anyone with some programming experience who wants to quickly gain a working knowledge of the Python 3.x language.

SYLLABUS

Working With Data

- Learn the basics of the Swift programming language including control flow, numbers and types, data structures, and enumerations.
- Work with advanced Swift features including protocol extensions and generics
- Understand how memory is managed for reference and value types with ARC

Working With More Data

- Learn how to organize larger programs.
- Work with functions.
- Understand exceptions and error handling in Python.
- Learn about modules, namespaces and the difference between global and local definitions.
- Get an overview of Python's most useful modules.

Classes and the Python Object Model

- Discover what Object-Oriented Programming (OOP) is and how it works in Python.
- Learn how to define classes and methods, and how to manage instance data.
- Master single inheritance, multiple inheritance and method overriding.
- Understand how the object model works in python.
- Discover how dictionaries are at the core of how objects work in Python.
- Understand how class and instance attribute lookups work.

Schedule Your Onsite Training: hello@bignerdranch.com

Iterators and Generator

- Learn about the iterator protocol and how to make your own objects respond to iteration.
- Discover generators and generator expressions.
- Understand the similarities and differences between list comprehensions and generator expressions.
- Assemble Generator pipelines to process data in a flexible and modular way.

Advanced Topics: Testing, Debugging and Packages

- Learn about closures and use them to define function decorators.
- Understand static and class methods.
- Discover the powerful flexibility of Python's function arguments.
- Learn about Python's facilities for testing and debugging code.
- Organize your projects into importable packages.
- Learn how to install third-party packages.