



Advanced Python

Course 55285: 2 days; Instructor-Led

Introduction

This Advanced Python training course picks up where our Introduction to Python course leaves off. This course is for students who have taken the introductory course and are ready to learn more and for students who have some experience programming with Python and are ready to go to the next level. The course was written using Python 3.8, but is relevant for all students using Python 3.

At Course Completion

After completing this course, students will be able to:

- Work with Lambda functions.
- Work with advanced list comprehensions.
- Work with the collections module.
- Use mapping and filtering.
- Sort sequences.
- Unpack sequences in function calls.
- Work with modules and packages.
- Understand regular expressions.
- Use Python's re module.
- Store data in a relational database.
- Store data in a CSV file.
- Work with data from a web page.
- Use HTML, XML, and JSON.
- Access an API.
- Test performance with timers and the timeit module.
- Work with the unittest module.
- Work with classes and objects in Python.
- Work with instance methods, class methods, and static methods.
- Work with properties.
- Work with decorators.
- Work with subclasses and inheritance.

Audience profile

This course is for students who have taken the introductory course and are ready to learn more and for students who have some experience programming with Python and are ready to go to the next level. The course was written using Python 3.8, but is relevant for all students using Python 3.

Prerequisites

- Experience in the following is required for this Python class:
- Basic Python programming experience. In particular, you should be very comfortable with:
- Working with strings.
- Working with lists, tuples and dictionaries.
- Loops and conditionals.
- Writing your own functions.
- Experience in the following would be useful for this Python class:
- Some exposure to HTML, XML, JSON, and SQL.

Course Outline

Module 1: Advanced Python Concepts

In this lesson, you will learn about some Python functionality and techniques that are commonly used but require a solid foundation in Python to understand.

Lessons

- Lambda Functions
- Advanced List Comprehensions
- Collections Module
- Mapping and Filtering
- Mutable and Immutable Built-in Objects
- Sorting
- Unpacking Sequences in Function Calls

Lab 1: Exercises in this Lesson

- Rolling Five Dice
- Creating a defaultdict
- Creating an OrderedDict
- Creating a Counter
- Working with a deque
- Converting list.sort() to sorted(iterable)
- Converting a String to a datetime.date Object

After completing this module, students will be able to:

- Work with lambda functions.
- Write more advanced list comprehensions.
- Work with the collections module to create named tuples, defaultdicts, ordereddicts, counters, and deques.
- Use mapping and filtering.
- Sort sequences.
- Unpack sequences in function calls.
- Create modules and packages.

Module 2: Regular Expressions

Regular expressions are used to do pattern matching in many programming languages, including Java, PHP, JavaScript, C, C++, and Perl. We will provide a brief introduction to regular expressions and then we'll show you how to work with them in Python.

Lessons

- Regular Expression Tester
- Regular Expression Syntax
- Python's Handling of Regular Expressions

Lab 1: Exercises in this Lesson

- Green Glass Door

After completing this module, students will be able to:

- Understand regular expressions.
- Be familiar with Python's module.

Module 3: Working with Data

Data is stored in many different places and in many different ways. In this lesson, you'll learn about the Python modules that help you access data.

Lessons

- Virtual Environment
- Relational Databases
- Passing Parameters
- SQLite
- SQLite Database in Memory
- Drivers for Other Databases
- CSV
- Creating a New CSV File
- Getting Data from the Web
- XML
- JSON

Lab 1: Exercises in this Lesson

After completing this module, students will be able to:

- Store data in a relational database.
- Store data in a CSV file.

- Work with data from a web page.
- Work with HTML, XML, and JSON.
- Access an API.

Module 4: Testing and Debugging

In this lesson, you will learn to test the performance and the functionality of your Python code.

Lessons

- Testing for Performance
- The unittest Module
- Special unittest.TestCase Methods

Lab 1: Exercises in this Lesson

- Comparing Times to Execute
- Fixing Functions

After completing this module, students will be able to:

- Learn to test the performance of different pieces of code and to create unit tests to test your Python code.

MODULE 5. Classes and Objects

An object is something that has attributes and/or behaviors, meaning it is certain ways and does certain things. In the real world, everything could be considered an object. Some objects are tangible, like rocks, trees, tennis racquets, and tennis players. And some objects are intangible, like words, colors, tennis swings, and tennis matches. In this lesson, you will learn how to write object-oriented Python code.

- Attributes
- Behaviors
- Classes vs. Objects
- Attributes and Methods
- Private Attributes
- Properties
- Objects that Track their Own History
- Documenting Classes
- Inheritance
- Extending a Class Method
- Static Methods
- Class Attributes and Methods
- Abstract Classes and Methods
- Understanding Decorators

Labs/Exercises in this module:

- Exercise: Adding a roll() Method to Die
- Exercise: Properties
- Exercise: Documenting the Die Class
- Exercise: Extending the Die Class
- Exercise: Extending the roll() Method
- How to create Python classes and write object-oriented code.