

# INTRODUCTION TO PYTHON PROGRAMMING FOR SECURITY ANALYSTS & PROFESSIONALS TRAINING



## Course Snapshot

**Course:** Python Security | Introduction to Python Programming for Security Analysts & Professionals

**Duration:** 4 days

**Audience & Skill Level:** This is an introductory-level course geared for Security professionals whose responsibilities include application security relative to Python applications.

**Hands-on Learning:** This hands-on course combines engaging expert lessons, demos, and group discussions with real-world, skills-focused machine-based labs and exercises. Student machines are required.

**Delivery Options:** This course is available for onsite private classroom presentations, live online virtual presentations, or can be presented in a blended learning format.

**Public Schedule:** This course has active dates on our live-online open enrollment Public Schedule.

**Customizable:** This course agenda, topics, and labs can be further adjusted to target your specific training skills objectives, tools, and learning goals. Please ask for details.

## Overview

Microtek Learning's Introduction to Python Programming for Security Analysts training program offers a boost to the career of all security professionals. It is an introductory class that mainly focuses on python training. In this course, professionals will know more advanced features of Python script. It includes file operations, regular expressions, working with binary data, and implementation of comprehensive functionality of python modules. The course also emphasizes Python's unique features like tuples, array slices, output formatting, and many more. So, reading this comprehensive course will offer in-depth knowledge of working with a programming language.

## What you will learn

- Create working Python scripts following best practices
- Use python data types appropriately
- Read and write files with both text and binary data
- Search and replace text with regular expressions
- Get familiar with the standard library and its work-saving modules
- Use lesser-known but powerful Python data types
- Create "real-world", professional Python applications
- Work with dates, times, and calendars
- Know when to use collections such as lists, dictionaries, and sets
- Understand Pythonic features such as comprehensions and iterators

## Who should attend this course?

This course is appropriate for advanced users, system administrators, and website administrators who want to use Python to support their server installations, as well as anyone else who wants to automate or simplify common tasks with the use of Python scripts.

## Prerequisites for this training

Students are required to have some basic programming experience and exposure prior to attending this course. Students should have basic development experience in any programming language, along with a working, user-level knowledge of Unix/Linux, Mac, or Windows.

## Curriculum

### Module 1: An Overview of Python

- What is python?
- Python Timeline
- Advantages/Disadvantages of Python
- Getting help with pydoc

### Module 2: The Python Environment

- Starting Python
- Using the interpreter
- Running a Python script
- Python scripts on Unix/Windows
- Editors and IDEs

### Module 3: Getting Started

- Using variables
- Builtin functions
- Strings
- Numbers
- Converting among types
- Writing to the screen
- Command line parameters

### Module 4: Flow Control

- About flow control
- White space
- Conditional expressions
- Relational and Boolean operators
- While loops
- Alternate loop exits

### Module 5: Sequences

- About sequences
- Lists and list methods
- Tuples

- Indexing and slicing
- Iterating through a sequence
- Sequence functions, keywords, and operators
- List comprehensions
- Generator Expressions
- Nested sequences

#### **Module 6: Working with files**

- File overview
- Opening a text file
- Reading a text file
- Writing to a text file
- Reading and writing raw (binary) data
- Converting binary data with struct

#### **Module 7: Dictionaries and Sets**

- About dictionaries
- Creating dictionaries
- Iterating through a dictionary
- About sets
- Creating sets
- Working with sets

#### **Module 8: Functions**

- Defining functions
- Parameters
- Global and local scope
- Nested functions
- Returning values

#### **Module 9: Sorting**

- The sorted() function
- Alternate keys
- Lambda functions
- Sorting collections

#### **Module 10: Errors and Exception Handling**

- Syntax errors
- Exceptions
- Using try/catch/else/finally
- Handling multiple exceptions
- Ignoring exceptions

#### **Module 11: Modules and Packages**

- The import statement
- Module search path
- Creating modules and Using packages
- Function and Module aliases

#### **Module 12: Working with Classes**

- About o-o programming
- Defining classes

- Constructors
- Methods
- Instance data
- Properties
- Class methods and data

#### **Module 13: Regular Expressions**

- RE syntax overview
- RE Objects
- Searching and matching
- Compilation flags
- Groups and special groups
- Replacing text
- Splitting strings

#### **Module 14: The standard library**

- The sys module
- Launching external programs
- The string module
- Reading CSV data

#### **Module 15: Dates and times**

- Working with dates and times
- Translating timestamps
- Parsing dates from text

#### **Module 16: Working with the file system**

- Paths, directories, and filenames
- Checking for existence
- Permissions and other file attributes
- Walking directory trees
- Creating filters with fileinput
- Security and File Access

#### **Module 17: Network services**

- Grabbing web content
- Detecting Malformed Input

#### **Module 18: Writing secure Python applications**

- Parsing command-line options
- Getting help with pydoc
- Safely handling untrusted data
- Managing eval() permissions
- Potential insecure packages
- Embedding code snippets in Python
- Embedding authentication data in Python
- Potentially dangerous operations:
  - File access
  - Operating system access
  - Calls to external services
  - Called to external data sources

- Static analysis tools such as Bandit

#### **Module 19: Log File Analysis**

- Raw log file manipulation
- Fail2Ban
- Customizing Fail2Ban with Python

#### **Module 20: Security Filters**

- SQL-Injection Detection
- ModSecurity CRS filtering

#### **Module 21: Packet Analysis**

- Packet Sniffing in Python

#### **Module 22: Analytics**

- Security Logging and Analytics
- Attack Detection and Defense
- Python and Spark High-Level Overview

#### ***Bonus Content / Time Permitting***

#### **Module 23: RESTful Web Services**

- What is Flask?
- Developing a Flask Web service
- Mapping resources using URLs
- Mapping resources using HTTP
- Negotiating data content

#### **Module 24: Python application security**

- OWASP 2021 Top Ten Overview
- Python Code Access Control
- Options for Protecting Data
- Injection and Python
- Python and Data Validation
- Python and XML Processing
- Python and Known Vulnerable Components
- Python and Serialization/Deserialization

---

*For any query [Contact Us – Microtek Learning](#)*

---