Document Generated: 07/05/2024

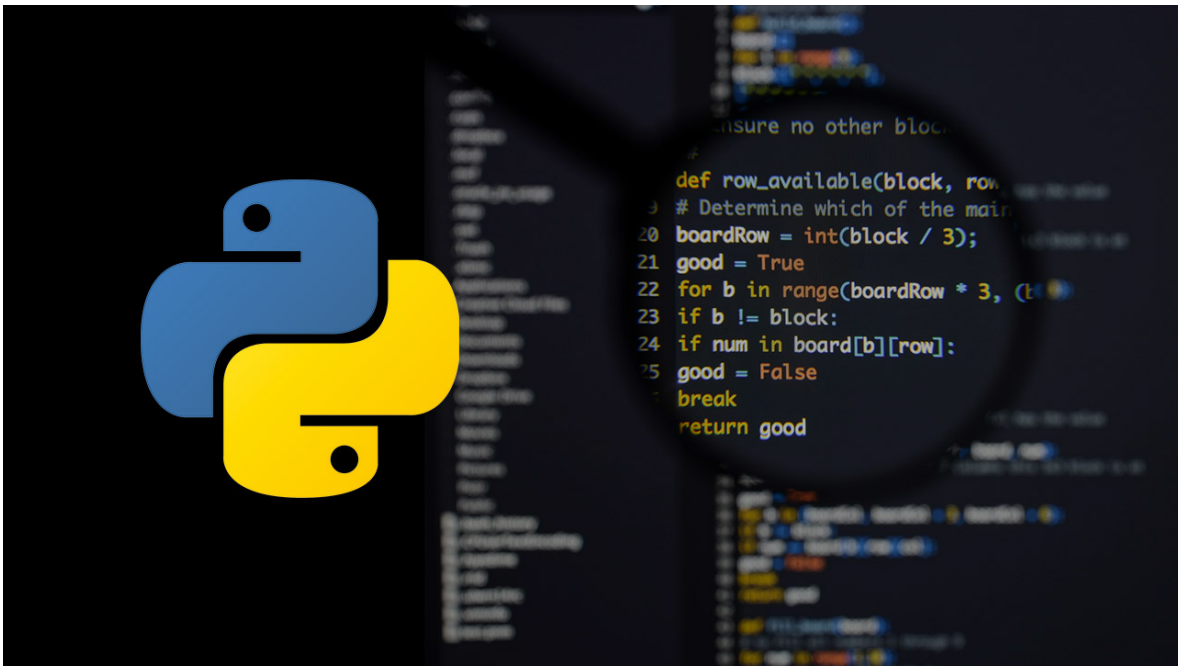
Learning Style: Virtual Classroom

Provider:

Difficulty: Beginner

Course Duration: 5 Days

Python Programming for Security Professionals (TTPS4890)



About this course:

Geared for experienced security professionals, this class is an **introductory,** practical, hands-on Python training course that leads the student from the basics of writing and running Python scripts to more advanced features such as file operations, regular expressions, working with binary data, and using the extensive functionality of Python modules. Extra emphasis is placed on features unique to Python, such as tuples, array slices, and output formatting. This comprehensive, practical course provides an in-depth exploration of working with the programming language, not an academic overview of syntax and grammar.

This course is tailored specifically for Security Analysts and others who wish to use Python functionality for security-related tasks such as log manipulation or forensics. This course, which addresses secure coding practices, is essential for security professionals that are performing security reviews and audits of Python applications or are supporting development teams in implementing better defenses in Python.

The average salary of a Python programmer is **$116,379** per year.

Course Objective:

Throughout the course students will be led through a series of progressively advanced topics, where each topic consists of lecture, group discussion, comprehensive hands-on lab exercises, and lab review. This course is “skills-centric”, designed to train attendees in core Python and web development skills beyond an intermediate level, coupling the most current, effective techniques with best practices.

Working within in an engaging, hands-on learning environment, guided by our expert Python practitioner, students will learn to:

- 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
- Write robust code using exception handling

Audience:

- This course is appropriate for advanced users, system administrators and web site 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.

Prerequisite:

- 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.

Course Outline:

Module 1: An Overview of Python

- What is python?
- 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: 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: Essential, Safe DB Access

- DB-API
- ORM with SQLAlchemy
- Preventing SQL Injection in DB Access
- Parameterization in DB Interactions

Module 20: Log File Analysis

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

Module 21: Security FIlters

- SQL-Injection Detection
- ModSecurity CRS filtering

Module 22: Packet Analysis

- Packet Sniffing in Python

Module 23: Analytics

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

Credly Badge:

Display your Completion Badge And Get The Recognition You Deserve.

Add a completion and readiness badge to your Linkedin profile, Facebook page, or Twitter account to validate your professional and technical expertise. With badges issued and validated by Credly, you can:

- Let anyone verify your completion and

achievement by clicking on the badge
- Display your hard work and validate your expertise
- Display each badge's details about specific skills you developed.

Badges are issued by QuickStart and verified through Credly.

Find Out More or See List Of Badges