

PYTHON PROGRAMMING COURSE

The Python Programming Course is a concentrated, hands-on course that arms students with the skills and knowledge to leverage the Python programming language in everyday computer network operations. Students will start at the beginning with Python, no assumptions are made on prior skill level, and work towards becoming proficient in the language both in reading source code and designing and developing their own applications. The course uses real-world techniques within each of the modules, demonstrating how Python can be leveraged in each scenario to help improve success and efficiency. Topics range from creating your own password cracking tool to setting up client-server applications, all within 5 days. On the final day of the course, students will be challenged with a multi-level culmination exercise, helping to reinforce the skills gained during the week and acquire new skills as well. This culmination exercise will not only test the students understanding of the Python language, but will also help demonstrate the different areas within network operations that Python can be useful. Students will be required to think creatively to get through this challenge.

CHIRON METHODOLOGY DOMAIN

CYBER DEVELOPMENT PROFESSIONAL™ (CDP)™



DURATION
5 DAYS

RECOMMENDED PRE-REQUISITES

- ↗ Familiarity with VMware Player or Workstation
- ↗ Exposure to Linux or UNIX-based Operating System
- ↗ An interest in learning to program in Python

PYTHON PROGRAMMING COURSE

COURSE SCHEDULE		
DAY 1:	LEARNING OBJECTIVES	OUTLINE
	<ul style="list-style-type: none"> ➤ Understand the basics of programming in any language ➤ Ability to write basic Python scripts independently ➤ Understanding the use of Classes in object-oriented programming 	<ul style="list-style-type: none"> ➤ Orientation & Introductions ➤ First Program ➤ Basics of learning a new programming language ➤ Importing modules ➤ Command-line arguments ➤ Classes
DAY 2:	LEARNING OBJECTIVES	OUTLINE
	<ul style="list-style-type: none"> ➤ Understanding how to interact with files through Python ➤ Understanding how to interact with the Operating System through Python 	<ul style="list-style-type: none"> ➤ File input ➤ File output ➤ Replicating built-in programs with Python <ul style="list-style-type: none"> » cap » cp » Logfile generator » grep » sed » ls » find ➤ String manipulation ➤ Creating a math program ➤ Parsing logfiles ➤ Using the 'os' module ➤ Understanding the built-in help systems ➤ Defensive scripts <ul style="list-style-type: none"> » Change detection » Killing malware
DAY 3:	LEARNING OBJECTIVES	OUTLINE
	<ul style="list-style-type: none"> ➤ Ability to use Python to perform different types of password attacks ➤ Understanding the history of Python development, including multiple versions of Python, and how to port Python code to other Operating Systems ➤ Understanding the importance of programming style 	<ul style="list-style-type: none"> ➤ Using the 'crypt' module ➤ Password attacks vs. /etc/shadow ➤ Time calculations for password guessing ➤ Generating password lists ➤ Manipulating password lists ➤ History of Python development ➤ Python3 ➤ Python for Windows ➤ Programming style <ul style="list-style-type: none"> » Why style matters » PEP8 – Python style guide

PYTHON PROGRAMMING COURSE

COURSE SCHEDULE		
DAY 4:	LEARNING OBJECTIVES	OUTLINE
	<ul style="list-style-type: none">↗ Understanding network programming↗ Ability to create custom applications for network clients and servers↗ Ability to apply knowledge from previous lessons to solving new problems in an exercise	<ul style="list-style-type: none">➤ Building a client socket application<ul style="list-style-type: none">» Web client» Custom clients» Password guessing attacks against remote servers➤ Building a server socket application<ul style="list-style-type: none">» Using the 'socket' module» Using the 'SocketServer' module» Web server» SimpleHTTPServer➤ CULEX Part 1<ul style="list-style-type: none">» CTF (Capture The Flag) exercise where every challenge must be solved with Python» In Part 1, the students use Python to replace-command-line tools in order to break out of a restricted shell
DAY 5:	LEARNING OBJECTIVES	OUTLINE
	<ul style="list-style-type: none">↗ Understanding threaded programming↗ Ability to generate custom network packets with Scapy↗ Ability to apply knowledge from previous lessons to solving new problems in an exercise	<ul style="list-style-type: none">➤ Threaded programming➤ Network programming with Scapy<ul style="list-style-type: none">» Overview» Replacing ping» Building an ICMP scanner» Building an ARP scanner» Sniffing network traffic» Replicating the TCP 3-way handshake➤ CULEX Part 2<ul style="list-style-type: none">» CTF (Capture The Flag) exercise where every challenge must be solved with Python» In Part 2, the challenges are all network-based. Students must develop custom Python applications to interact with remote servers and solve multiple challenges.