Lesson Guide

Cybersecurity Professional Program

Introduction to Python for Security

Loops

Course Path

- PY-01: Introduction to Programming
- PY-02: Data Types & Conditions
- **PY-03: Loops**
- PY-04: File System & Error Handling
- PY-05: Functions
- PY-06: Network Communication
- PY-07: Python for Security

Lesson Objectives

This lesson is an introduction to Python ***for*** and ***while*** loops, as well as loops and conditions. During the lesson, management methods will be pointed out to emphasize organized and efficient code development.

Lesson Overview

The module focuses on working with loops and where in the code loops can be used for more effective programming.

Learner Level

Learners have knowledge of Python basics, including data types and conditions.

Lesson Notes

During the lesson, when explaining about loops, let learners practice them on their laptops whenever possible.

Environment & Tools

- Windows, macOS, Linux
  - Python3
  - PyCharm

For & While Loops

Explain the ***for*** loop and demonstrate how it works. Make sure learners understand how range is used in a loop instead of lists.

What Are Loops?

Explain the main idea behind loops, why they are used, and how they can be used effectively. Briefly review ***for*** and ***while*** loops. The differences between them will be explained later.

Explain the following:

- ***For*** loops can repeat statements an ***x*** number of times.
- ***For*** loops can repeat statements the number of times specified in a variable.
- ***While*** loops repeat statements continuously until something in the code block stops them.

***For* Loops**

Demonstrate how to use ***for*** loops in different contexts (like in the example). Explain that in the example, the variables ***i*** and ***n*** can be used in any name. Point out that an iteration over a sequence is known as *traversal*. Describe the sequences (list, tuple, string).

Range vs. List in Loops

Demonstrate how to use ranges in lists and how ***for*** loops work (as in the example). Explain why this method is effective. A more detailed explanation of range will be provided later in the presentation.

Calculate Loop Size

Demonstrate the ***len()*** and ***range()*** functions. Point out that ***len()*** can be performed on a list, string, or any other variable, while ***range()*** can be used to generate a sequence of numbers over time. Describe when they are used in the code and how they can be effective.

Lab PY-03-L1: Range Loop

This lab should be completed in class.

Lab PY-03-L2: Loops in Nested Lists

This lab should be completed in class.

While Loops

Demonstrate how to use ***while*** loops and point out the differences between ***while*** and ***for*** loops. Explain that a condition usually depends on something that happens within the code block. Point out that if a statement is false, the program will move on to the next code block. Mention how ***break*** exits a loop if a specific condition is met.

Break Command

Demonstrate how to use ***break*** in both loops and conditions. Explain the code in the example. Discuss when ***break*** should be used and how it can be effective in loops.

Continue Command

Demonstrate how to use ***continue*** in loops. Verify that learners understand the difference between ***continue*** and ***break***. Explain the code in the example. Point out when ***continue*** is used in a loop and how it can be effective.

Pass Command

Briefly describe the code in the example. Demonstrate how to use ***pass*** in Python and point out common uses.

Pass vs. *Continue*

Explain the difference between ***pass*** and ***continue*** and where you would use them in the code. Explain that ***continue*** will jump back to the top of the loop, while ***pass*** will continue processing the code.

Loops & Conditions

Explain the benefits of working with loops and conditions and demonstrate how the combination operates. Discuss how infinite loops can be avoided and how loop sizes are calculated. Make sure learners understand how nested loops work and when they are effective.

Mixing Conditions and Loops

Briefly explain the code in the example. Demonstrate how to use conditions in loops and how they can be effective. Mention that the difference will be at the end, where ***break*** will be used in ***while*** loops. Point out that by mixing conditions and loops, different checks can be performed on iterating objects and outputs can be displayed accordingly.

Infinite Loops

Briefly explain the code in the example. An infinite loop will occur if a ***while*** loop's condition is always true. Demonstrate what an infinite loop looks like and how to avoid it. Mention that infinite loops will typically occur in ***while*** loops. Point out that the ***print*** function in the example will never be executed.

Nested Loops and Lists

Explain nested lists. Briefly explain the code in the example. Demonstrate how to perform a nested loop and verify that learners understand how it is done. Discuss the ***append()*** function and how it should be used.

Lab PY-03-L3: Loops with Conditions

This lab should be completed in class.

Lab PY-03-L4: User Dictionary

This lab should be completed in class.

Lab PY-03-L5: While and Conditions

This lab should be completed in class.