Activity: Practice writing Python code

Introduction

In this lab, you will open a notebook environment to practice writing Python code. You'll be presented with a security scenario to explore throughout the lab. You'll become familiar with working in a notebook environment, writing code comments in Python, and displaying strings with the `print()` function.

What you'll do

You have multiple tasks in this lab:
- Learn about the capabilities of a notebook environment
- Write code comments in Python
- Display strings with the `print()` function

Activity: Practice writing Python code

Introduction

Python is a programming language that helps in automating instructions to the computer in a variety of contexts, including security contexts. Writing code in Python is a valuable skill that helps security analysts thrive in their technical work.

In this lab, you'll practice writing your first Python code while learning about a notebook environment. The hands-on practice you engage in throughout the labs will help you apply Python coding skills to your work as a security analyst. You'll benefit the most from the labs if you make sure to not only write in the cells that you're prompted to fill in, but also analyze all the cells thoroughly.

Tips for completing this lab

-
-
-
-
-

Scenario

As a security analyst, you'll often use notebook environments and notebooks to write and run code. This lab will help you get familiar with working in a notebook environment, writing code comments in Python, and displaying strings with the print() function.

In this lab, you'll complete a series of tasks that involve observing and running some pre-written cells of text and code, as well as filling in cells with your own text, Python code, and code comments.

Task 1

The lab environment you're working in is a notebook-based coding environment. Notebooks, such as this one, consist of two types of cells: (1) text cells, also known as markdown cells, and (2) code cells.

Markdown cells allow you to write plain text and format it in the markdown language. Markdown language is used for formatting plain text in text editors and code editors. For example, you can use markdown to make headers, bold or italicize words, format text as code, add hyperlinks, and more.

For this task, write something into the following markdown cell. Be sure to replace the "[Double-click to edit this markdown cell and write something here.]" with your own text. When you have finished editing, press the Shift and Enter keys (or on some keyboards, the Shift and Return keys) to display your text.
[Double-click to edit this markdown cell and write something here.]

Hint 1

Task 2

In Python notebooks, code cells allow you to write code comments and code in Python.

To run a code cell, first place your cursor on the cell. Then, you can either click on the play icon, or press the Shift and Enter keys (or on some keyboards, the Shift and Return keys).

For this task, run the following code cell as is and observe the output.

In [ ]:

```python
# This cell displays "Hello world!"

print("Hello world!")
```

Hint 1

Question 1

What do you observe about the output after you ran the code cell?
[Double-click to enter your responses here.]

Task 3

Writing code comments is a way to document the intention behind code. It's a standard that analysts commonly use in their workflow. Writing comments that accompany code allows you to keep track of the technical decisions you've made in your project. This makes it easier for you and your team to read and revisit your code in order to understand what it does and why you took certain approaches.

For this task, run the following code cell as is and observe the output.

In [ ]:

```
# In Python, comments do not get displayed
# This code cell contains only comments
```

Hint 1

Question 2

What do you observe about the output after you ran the cell above?
[Double-click to enter your responses here.]

Task 4

To type in a code cell, first click into the cell. Then you can write comments and code inside the cell.

For this task, add a comment at the beginning of the following code cell, describing what the code is doing. Write the comment to say `# This cell displays "I am using Python."`. Be sure to replace the `# YOUR COMMENT HERE` with your own comment before running the following cell.

```
# this code is a print statement

print("I am using Python.")
```

Hint 1

Question 3

What do you observe about the output after you ran the cell above?
[Double-click to enter your responses here.]

Task 5

In Python, `print()` helps you to display information to the screen.

For this task, use `print()` to display the message `"I am a security analyst."` by placing that message within the parentheses. Be sure to replace the `### YOUR CODE HERE ###` with your own code before running the following code cell.

```
# This cell displays "I am a security analyst."

print("I am a security analyst")
```

Hint 1

Question 4

What do you observe about the output after you ran the cell above?
[Double-click to enter your responses here.]

Task 6

For this task, write a `print()` statement to display the string `"Python is useful for security!"` Be sure to replace the `### YOUR CODE HERE ###` with your own code before running the following code cell.

```python
# This cell displays "Python is useful for security!"

print("Python is useful for security")
```

Hint 1

Hint 2

Question 5

What do you observe about the output after you ran the cell above?
[Double-click to enter your responses here.]

Task 7

For your final task, you'll combine all the `print()` statements you've encountered and written in this lab up to this point, into one code cell.

Complete the following code with the remaining messages. Be sure to replace each `### YOUR CODE HERE ###` with your own code before running the following cell.

```python
# This cell displays all the statements written so far

print("Hello world!")
print("I am using Python.")
print("I am a security analyst")
print("Python is useful for security")
```

Hint 1

Question 6

What do you observe about the output after you ran the cell above?
[Double-click to enter your responses here.]

Conclusion

What are your key takeaways from this lab?

[Double-click to enter your responses here.]

Exemplar: Practice writing Python code

Introduction

Python is a programming language that helps in automating instructions to the computer in a variety of contexts, including security contexts. Writing code in Python is a valuable skill that helps security analysts thrive in their technical work.

In this lab, you'll practice writing your first Python code while learning about a notebook environment. The hands-on practice you engage in throughout the labs will help you apply Python coding skills to your work as a security analyst. You'll benefit the most from the labs if you make sure to not only write in the cells that you're prompted to fill in, but also analyze all the cells thoroughly.

Tips for completing this lab

-
-
-
-
-

Scenario

As a security analyst, you'll often use notebook environments and notebooks to write and run code. This lab will help you get familiar with working in a notebook environment, writing code comments in Python, and displaying strings with the `print()` function.

In this lab, you'll complete a series of tasks that involve observing and running some pre-written cells of text and code, as well as filling in cells with your own text, Python code, and code comments.

Task 1

The lab environment you're working in is a notebook-based coding environment. Notebooks, such as this one, consist of two types of cells: (1) text cells, also known as markdown cells, and (2) code cells.

Markdown cells allow you to write plain text and format it in the markdown language. Markdown language is used for formatting plain text in text editors and code editors. For example, you can use markdown to make headers, bold or italicize words, format text as code, add hyperlinks, and more.

For this task, write something into the following markdown cell. Be sure to replace the "[Double-click to edit this markdown cell and write something here.]" with your own text. When you have finished editing, press the *Shift* and *Enter* keys (or on some keyboards, the *Shift* and *Return* keys) to display your text.
This is a markdown cell, and you can write text in here.

Hint 1

Task 2

In Python notebooks, code cells allow you to write code comments and code in Python.

To run a code cell, first place your cursor on the cell. Then, you can either click on the play icon, or press the *Shift* and *Enter* keys (or on some keyboards, the *Shift* and *Return* keys).

For this task, run the following code cell as is and observe the output.

In [ ]:

```python
# This cell displays "Hello world!"

print("Hello world!")
```

Hint 1

Question 1

What do you observe about the output after you ran the code cell?
The output shows the statement `Hello world!` on the screen.

Task 3

Writing code comments is a way to document the intention behind code. It's a standard that analysts commonly use in their workflow. Writing comments that accompany code allows you to keep track of the technical decisions you've made in your project. This makes it easier for you and your team to read and revisit your code in order to understand what it does and why you took certain approaches.

For this task, run the following code cell as is and observe the output.

In [ ]:

```python
# In Python, comments do not get displayed
# This code cell contains only comments
```

Hint 1

Question 2

What do you observe about the output after you ran the cell above?
There is no output. This is because the code cell above consists of two code comments, each comment starting with a hash symbol (`#`). When executing code, Python ignores comments. Code comments aren't interpreted by computers; they're interpreted by analysts.

Task 4

To type in a code cell, first click into the cell. Then you can write comments and code inside the cell.

For this task, add a comment at the beginning of the following code cell, describing what the code is doing. Write the comment to say `# This cell displays "I am using Python."`. Be sure to replace the `# YOUR COMMENT HERE` with your own comment before running the following cell.

```
# This cell displays "I am using Python."

print("I am using Python.")
```

Hint 1

Question 3

What do you observe about the output after you ran the cell above?
The output is `I am using Python.` Note that the quotes around a string do not appear in the output when the string is displayed.

Task 5

In Python, `print()` helps you to display information to the screen.

For this task, use `print()` to display the message `"I am a security analyst."` by placing that message within the parentheses. Be sure to replace the `### YOUR CODE HERE ###` with your own code before running the following code cell.

```
# This cell displays "I am a security analyst."

print("I am a security analyst.")
```

Hint 1

Question 4

What do you observe about the output after you ran the cell above?
The output is `I am a security analyst.` Note that the quotes around a string do not appear in the output when the string is displayed.

Task 6

For this task, write a `print()` statement to display the string `"Python is useful for security!"` Be sure to replace the `### YOUR CODE HERE ###` with your own code before running the following code cell.

```
# This cell displays "Python is useful for security!"

print("Python is useful for security!")
```

Hint 1

Hint 2

Question 5

What do you observe about the output after you ran the cell above?

The output is `Python is useful for security!` Note that the quotes around a string do not appear in the output when the string is displayed.

Task 7

For your final task, you'll combine all the `print()` statements you've encountered and written in this lab up to this point, into one code cell.

Complete the following code with the remaining messages. Be sure to replace each `### YOUR CODE HERE ###` with your own code before running the following cell.

In [ ]:

```python
# This cell displays all the statements written so far

print("Hello world!")
print("I am using Python.")
print("I am a security analyst.")
print("Python is useful for security!")
```

Hint 1

Question 6

What do you observe about the output after you ran the cell above?

Each line in the output resulted from a `print()` statement in the code. `Hello world!` appears in the first line of the output, `I am using Python.` appears on the next line of the output, `I am a security analyst.` appears next, and finally `Python is useful for security!` appears last. Note that Python interprets and executes code in order, from top to bottom and left to right.

Conclusion

What are your key takeaways from this lab?

- It's helpful to use code comments to document the decisions you make as you code.
- Code comments are ignored by computers; they're read by you and your team to understand the intentions behind the code.
- You can write comments in Python using the hash symbol (`#`).
- You can use `print()` in Python to display information to the screen.
    - When you use `print()` to display a string, the quotes around the string do not appear in the output on the screen.