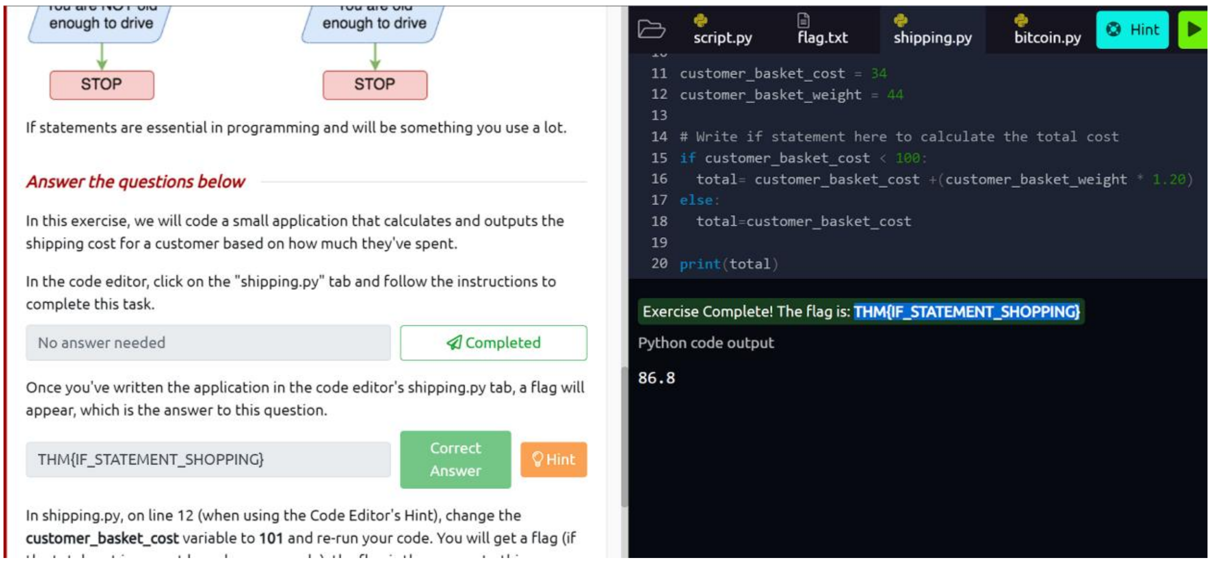
1.5 PYTHON BASICS

1. INTRODUCTION

Python scripting plays a crucial role in the field of cybersecurity, offering a versatile and powerful toolset for various security tasks. Python's extensive library ecosystem, simplicity, and readability make it an ideal choice for automating repetitive tasks, performing network scanning, vulnerability assessment, and penetration testing. It allows security professionals to develop custom tools and scripts for tasks like log analysis, malware analysis, and forensic analysis. Python's ability to interact with operating system APIs, network protocols, and security frameworks enables the creation of robust and scalable security solutions. Additionally, its integration with popular security tools and frameworks, such as Scapy, Metasploit, and Nmap, further extends its capabilities in areas like packet manipulation, exploit development, and security testing. Python's flexibility and ease of use make it a preferred language for cyber security professionals, enabling them to efficiently address emerging threats and protect systems and networks.



If statements are essential in programming and will be something you use a lot.

Answer the questions below

In this exercise, we will code a small application that calculates and outputs the shipping cost for a customer based on how much they've spent.

In the code editor, click on the "shipping.py" tab and follow the instructions to complete this task.

No answer needed — Completed

Once you've written the application in the code editor's shipping.py tab, a flag will appear, which is the answer to this question.

THM{IF_STATEMENT_SHOPPING} — Correct Answer — Hint

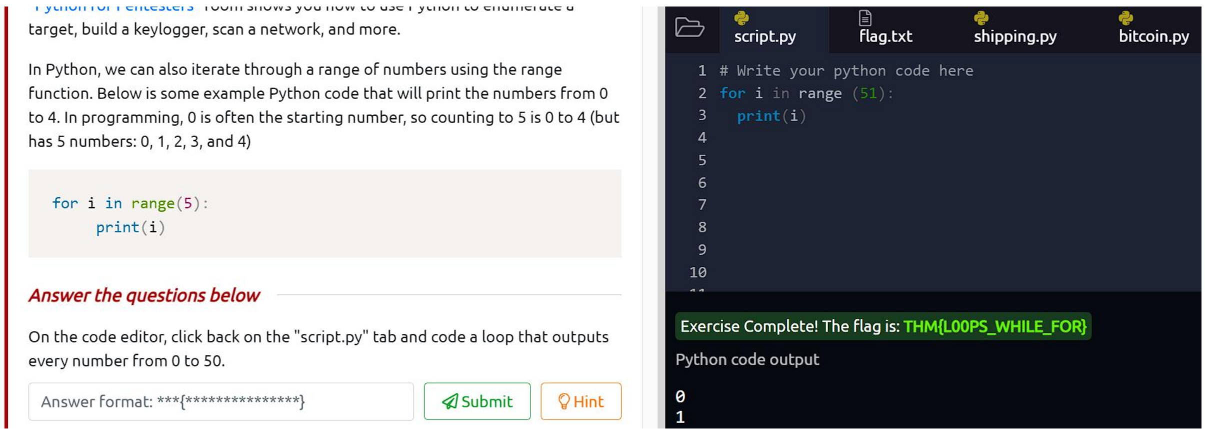
In shipping.py, on line 12 (when using the Code Editor's Hint), change the **customer_basket_cost** variable to **101** and re-run your code. You will get a flag (if

```
11  customer_basket_cost = 34
12  customer_basket_weight = 44
13
14  # Write if statement here to calculate the total cost
15  if customer_basket_cost < 100:
16    total= customer_basket_cost +(customer_basket_weight * 1.20)
17  else:
18    total=customer_basket_cost
19
20  print(total)
```

Exercise Complete! The flag is: THM{IF_STATEMENT_SHOPPING}

Python code output

```
86.8
```



"Python for Pentesters" room shows you how to use Python to enumerate a target, build a keylogger, scan a network, and more.

In Python, we can also iterate through a range of numbers using the range function. Below is some example Python code that will print the numbers from 0 to 4. In programming, 0 is often the starting number, so counting to 5 is 0 to 4 (but has 5 numbers: 0, 1, 2, 3, and 4)

```
for i in range(5):
    print(i)
```

Answer the questions below

On the code editor, click back on the "script.py" tab and code a loop that outputs every number from 0 to 50.

Answer format: ***{***************} Submit Hint

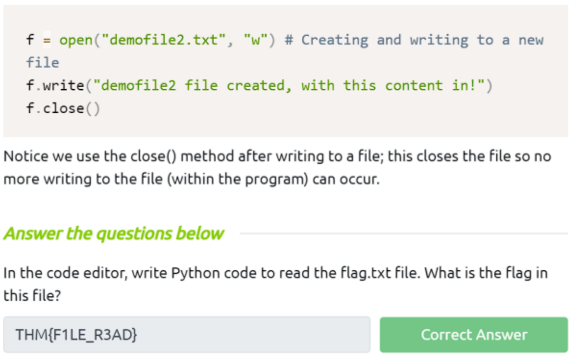
```
# Write your python code here
for i in range (51):
  print(i)
```

Exercise Complete! The flag is: THM{L00PS_WHILE_FOR}

Python code output

```
0
1
```

1.5 PYTHON BASICS



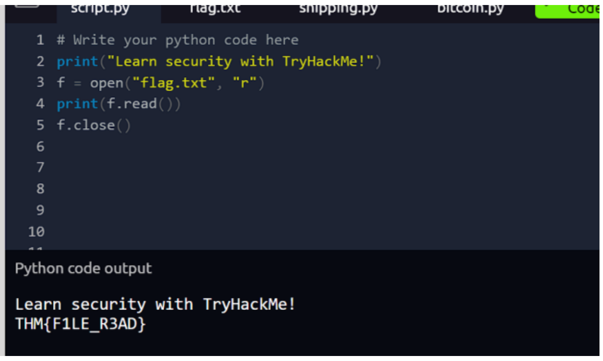
```
f = open("demofile2.txt", "w") # Creating and writing to a new
file
f.write("demofile2 file created, with this content in!")
f.close()
```

Notice we use the close() method after writing to a file; this closes the file so no more writing to the file (within the program) can occur.

Answer the questions below

In the code editor, write Python code to read the flag.txt file. What is the flag in this file?

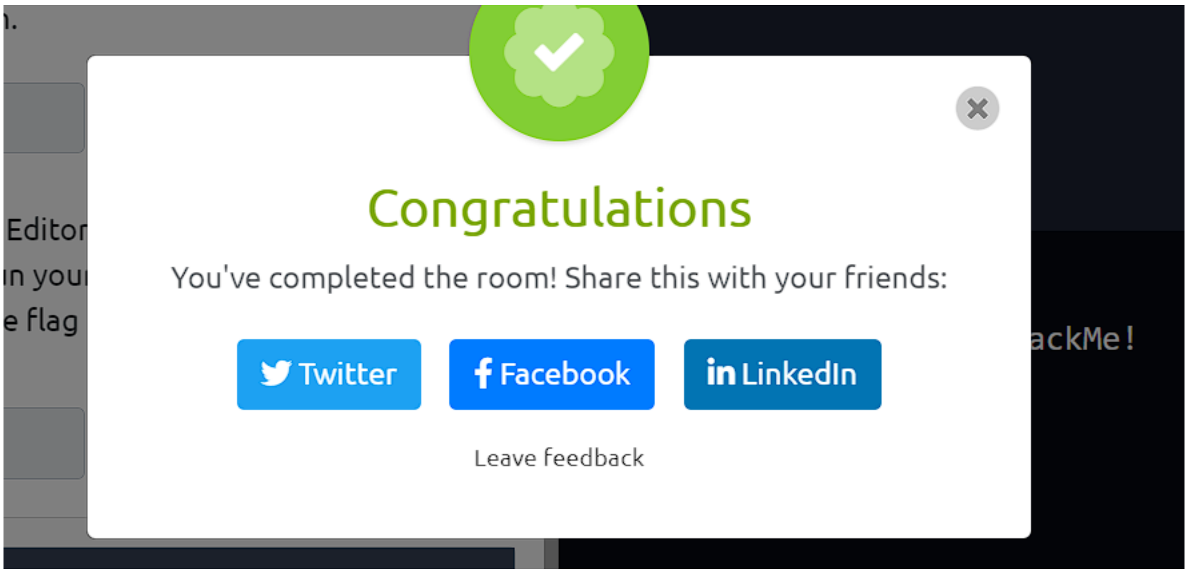
THM{F1LE_R3AD}



```
# Write your python code here
print("Learn security with TryHackMe!")
f = open("flag.txt", "r")
print(f.read())
f.close()
```

Python code output

```
Learn security with TryHackMe!
THM{F1LE_R3AD}
```



Congratulations

You've completed the room! Share this with your friends:

2. CONCLUSION

In conclusion, Python scripting is a valuable asset in the realm of cybersecurity, empowering professionals to automate tasks, perform security assessments, and develop custom solutions. Its extensive library ecosystem, ease of use, and integration with various security tools make it a preferred choice for tackling diverse security challenges. With Python, security professionals can streamline processes, enhance efficiency, and stay ahead of evolving threats, ultimately bolstering the protection of systems and networks.