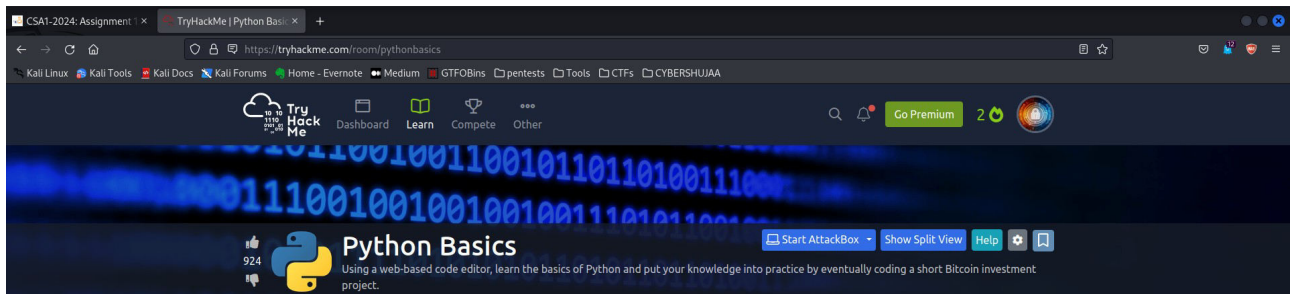




Eric Mwenda

Python Scripting

<https://tryhackme.com/p/Ericm>



Introduction to Python.

First I began by understanding what is python; here is one of the explanations I gathered in one of the blogs “Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together”.

(<https://www.python.org/doc/essays/blurb/>)

In TRYHACKME they begin by explaining that even tho programming isn't required to succeed in security, it's a great skill to have.

In this room am to cover:-

- Variables
- Loops
- Functions
- Data Structures
- If statements
- Files

Hello world

In this section we begin by creating just a simple program called hello world.

Code:-

```
# This is an example of hello world program
```

```
print("Hello World")
```

Answer the questions below

On the code editor, print "Hello World". What is the flag? ANS: THM{PRINT STATEMENTS}

The screenshot shows a web browser with two tabs: 'TryHackMe | Python Basics' and 'What is Python? Executive'. The 'TryHackMe' tab is active, displaying a page titled 'pythonbasics'. The page content includes a paragraph about the `print()` statement, a task instruction to print 'Hello World', and a list of tasks. The 'script.py' file in the code editor contains the line `print("Hello World")`. The output shows 'Hello World' and the flag 'THM{PRINT STATEMENTS}'.

the code understand what is going on.

We can control what is output to the screen by using the `print()` statement. Anything inside of the parenthesis `()` will be output. However, because we are printing a string (more on data types later in this room), we have to put them inside of quotations `" "`.

Please note, this room's examples are for Python3.

Answer the questions below

On the code editor, print "Hello World". What is the flag?

Answer format: `THM{*****}`

Task 3 ☐ Mathematical Operators

Task 4 ☐ Variables and Data Types

Task 5 ☐ Logical and Boolean Operators

Task 6 ☐ **Shipping Project** Introduction to IF Statements

Task 7 ☐ Loops

Task 8 ☐ **Bitcoin Project** Introduction to Functions

Task 9 ☐ Files

```
1 # Write your python code here
2 print("Hello World")
3
4
5
6
7
8
9
10
11
12
13
14
```

Exercise Complete! The flag is **THM{PRINT STATEMENTS}**

Python code output

Hello World

Mathematical Operators.

In this section I got to learn a few mathematical operators and how they can be applied in python.

Operator	Syntax	Example
Addition	+	1 + 1 = 2
Subtraction	-	5 - 1 = 4
Multiplication	*	10 * 10 = 100
Division	/	10 / 2 = 5
Modulus	%	10 % 2 = 0
Exponent	**	5**2 = 25 (5 ²)

Next was to learn about comparison operators which are used to elevate a programs condition at a particular state.

Symbol	Syntax
Greater than	>
Less than	<
Equal to	==
Not Equal to	!=
Greater than or equal to	>=
Less than or equal	<=

Answer the questions below

In the code editor, print the result of 21 + 43. What is the flag?

Code: `print(21+43)`

ANS: THM{ADDITION}

The screenshot displays a web browser window with the URL `https://tryhackme.com/room/pythonbasics`. The left sidebar shows a table of comparison operators and a list of tasks. The main area contains a code editor with the code `print(21+43)` and a submission form where the flag `THM{ADDITION}` has been entered. A green notification bubble says "Woop woop! Your answer is correct." and the output shows "64".

Symbol	Syntax
Greater than	>
Less than	<
Equal to	==
Not Equal to	!=
Greater than or equal to	>=
Less than or equal	<=

Answer the questions below

In the code editor, print the result of 21 + 43. What is the flag?

THM{ADDITION} Correct Answer Hint

Print the result of 142 - 52. What is the flag?

Answer format: `***{*****}` Submit

Print the result of 10 * 342. What is the flag?

Answer format: `***{*****}` Submit

Print the result of 5 squared. What is the flag?

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

Task 4 Variables and Data Types

Task 5 Logical and Boolean Operators

Task 6 Shipping Project: Introduction to IF Statements

Exercise Complete! The flag is: `THM{ADDITION}`

Python code output

64

Print the result of 142 - 52. What is the flag?

Code: `print(142-52)`

ANS: THM{SUBTRACT}

The screenshot shows the TryHackMe Python Basics room interface. On the left, a list of comparison operators is displayed: Equal to (==), Not Equal to (!=), Greater than or equal to (>=), and Less than or equal to (<=). Below this, a section titled "Answer the questions below" contains four tasks. The second task asks for the flag for the result of 142 - 52, with the correct answer THM{SUBTRACT} shown. The third task asks for the flag for the result of 10 * 342, with the correct answer THM{MULTIPLICATION_PYTHON} shown. The right side of the image shows a code editor with a Python script:

```
1 # Write your python code here
2 print(142-52)
```

 The output of the code is 90. A notification at the bottom right says "Exercise Complete! The flag is: THM{SUBTRACT}".

Print the result of 10 * 342. What is the flag?

Code: `print(10*342)`

ANS: THM{MULTIPLICATION PYTHON}

The screenshot shows the TryHackMe Python Basics room interface. On the left, a list of comparison operators is displayed: Equal to (==), Not Equal to (!=), Greater than or equal to (>=), and Less than or equal to (<=). Below this, a section titled "Answer the questions below" contains four tasks. The second task asks for the flag for the result of 142 - 52, with the correct answer THM{SUBTRACT} shown. The third task asks for the flag for the result of 10 * 342, with the correct answer THM{MULTIPLICATION_PYTHON} shown. The right side of the image shows a code editor with a Python script:

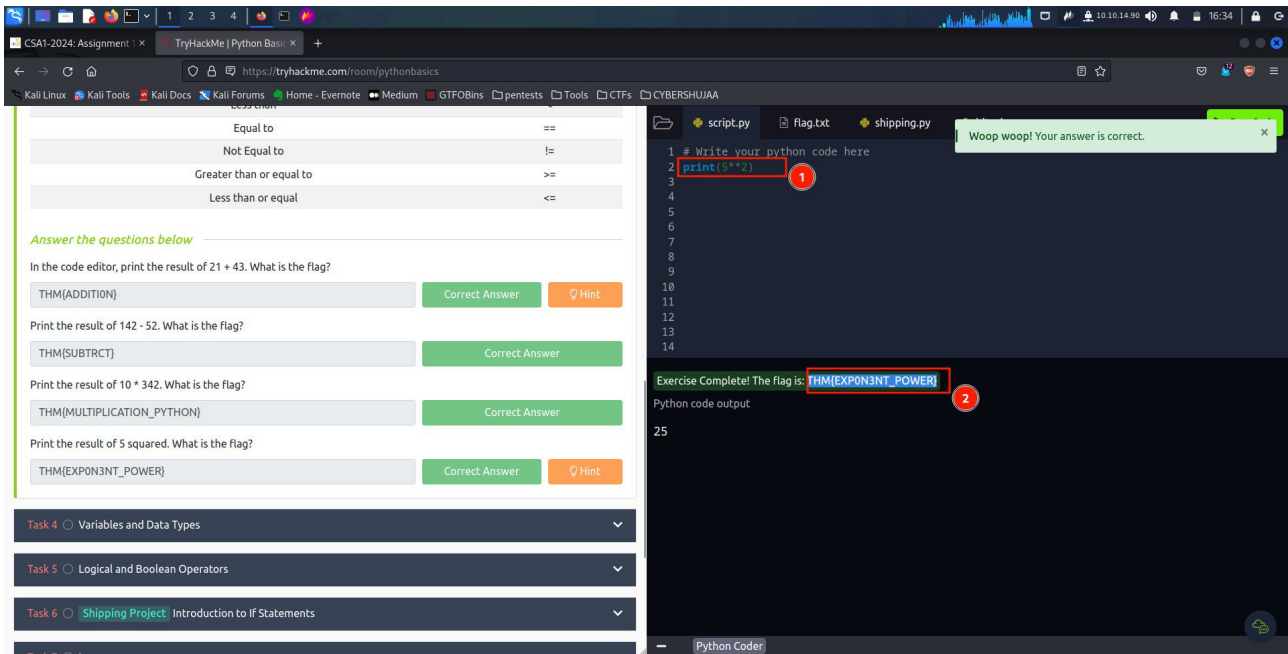
```
1 # Write your python code here
2 print(10*342)
```

 The output of the code is 3420. A notification at the bottom right says "Exercise Complete! The flag is: THM{MULTIPLICATION_PYTHON}" and a green message box says "Woop woop! Your answer is correct."

Print the result of 5 squared. What is the flag?

Code: `print(5**2)`

ANS: THM{EXP0N3NT POWER}



Variables and Data Types

Variables allow you to store and update data in a computer program.

For variables you can change them throughout the program.

Another explanation in python for a variable is that; it is a reserved memory location to store values. In other words, a variable in a python program gives data to the computer for processing. Every value in Python has a datatype.

Below is an example of a variable whereby:-

food is the variable name storing a string ice cream.

The other example, variable name is called money stores a number 2000.

```
food = "ice cream"
money = 2000
```

Questions

In the code editor, create a variable called height and set its initial value to 200.

Code: height = 200

On a new line, add 50 to the height variable.

Code: height+=50

On another new line, print out the value of height. What is the flag that appears?

Code: print(height)

ANS: THM{VARIABLES}

The screenshot displays the TryHackMe Python Basics room interface. On the left, a task list includes 'Task 5: Logical and Boolean Operators', 'Task 6: Shipping Project: Introduction to If Statements', 'Task 7: Loops', 'Task 8: Bitcoin Project: Introduction to Functions', 'Task 9: Files', and 'Task 10: Imports'. The main area features a code editor with the following Python code:

```
1 # Write your python code here
2 height = 200
3 height+=50
4 print(height)
5
```

Below the code editor, the output shows '250'. A notification box states 'Exercise Complete! The flag is: THM{VARIABLES}'. On the right, a sidebar shows the task list with 'Task 5' selected. A green banner at the top right says 'Woop woop! Your answer is correct.'

Logical and Boolean Operators

Logical operators allow assignment and comparisons to be made and are used in conditional testing (such as if statements).

Logical Operation	Operator	Example
Equivalence	==	if x == 5
Less than	<	if x < 5
Less than or equal to	<=	if x <= 5
Greater than	>	if x > 5
Greater than or equal to	>=	if x >= 5

Boolean operators are used to connect and compare relationships between statements. Like an if statement, conditions can be true or false.

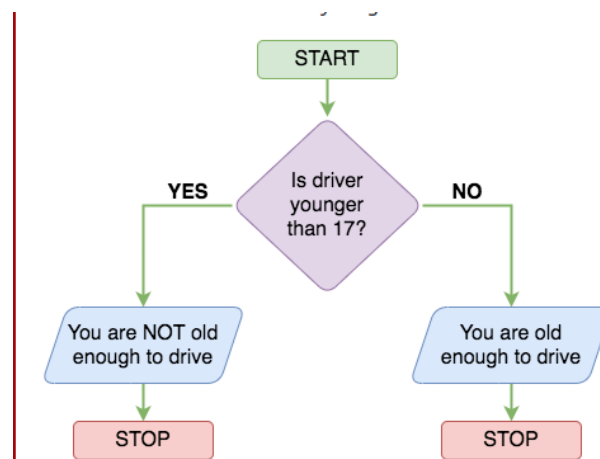
Boolean Operation	Operator	Example
Both conditions must be true for the statement to be true	AND	if x >= 5 AND x <= 100
		Returns TRUE if x is a number between 5 and 100
Only one condition of the statement needs to be true	OR	if x == 1 OR x == 10
		Returns TRUE if X is 1 or 10
If a condition is the opposite of an argument	NOT	if NOT y
		Returns TRUE if the y value is False

Shipping Project Introduction to If Statements

Using the if statements will allow programs to make decisions based on conditions programmed to meet.

```
if age < 17:
    print('You are NOT old enough to drive')
else:
    print('You are old enough to drive')
```

Using the above example I learn that this program should print 'You are NOT old enough to drive' in the case the subject is 17 tears old, in other cases the output should always be 'You are old enough to drive'. **A colon :** marks the end of the if statement.



Using the above flow chat I am able to understand more about this program.

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.

DONE

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.

ANS: THM{IF STATEMENT SHOPPING}

The screenshot shows the TryHackMe Python Basics room interface. On the left, a flowchart illustrates an if-statement logic: "You are NOT old enough to drive" leads to "STOP", and "You are old enough to drive" leads to "STOP". Below the flowchart, the task instructions are displayed. The code editor on the right shows the shipping.py file with the following code:

```
1 # Write a program that calculates the shipping cost for a customer based on how much they've spent.
2
3 Print the customers total basket cost (including shipping) to complete this exercise.
4
5 """
6
7
8
9
10
11
12 customer_basket_cost = 34
13 customer_basket_weight = 44
14
15 # Write if statement here to calculate the total cost
16
17 shipping_cost = 0
18 if customer_basket_cost >= 100:
19     shipping_cost = 10
20 else:
```

The code editor shows the output of the Python code as 86.8. A green notification bubble at the top right says "Woop woop! Your answer is correct." The task completion message at the bottom of the code editor states: "Exercise Complete! The flag is: THM{IF STATEMENT SHOPPING}".

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 the total cost is correct based on your code); the flag is the answer to this question.

ANS: THM{MY FIRST APP}

The screenshot shows the TryHackMe Python Basics room interface. On the left, a flowchart illustrates an if-statement logic: "You are NOT old enough to drive" leads to "STOP", and "You are old enough to drive" leads to "STOP". Below the flowchart, the task instructions are displayed. The code editor on the right shows the shipping.py file with the following code:

```
1 # Write a program that calculates the shipping cost for a customer based on how much they've spent.
2
3 Print the customers total basket cost (including shipping) to complete this exercise.
4
5 """
6
7
8
9
10
11
12 customer_basket_cost = 101
13 customer_basket_weight = 44
14
15 # Write if statement here to calculate the total cost
16
17 shipping_cost = 0
18 if customer_basket_cost >= 100:
19     shipping_cost = 10
20 else:
```

The code editor shows the output of the Python code as 101. A green notification bubble at the top right says "Woop woop! Your answer is correct." The task completion message at the bottom of the code editor states: "Exercise Complete! The flag is: THM{MY FIRST APP}".

Loops

In programming, loops allow programs to iterate and perform actions a number of times. There are two types of loops, for and while loops.

While Loops - A while loop is a control flow statement which repeatedly executes a block of code until the condition is satisfied

Below is an example for While Loop

```
i = 1
while i <= 10:
    print(i)
    i = i + 1
```

This while loop will run 10 times, outputting the value of the i variable each time it iterates (loops). Explanation provided in TRYHACKME.

- The i variable is set to 1
- The while statement specifies where the start of the loop should begin
- Every time it loops, it will start at the top (outputting the value of i)
- Then it goes to the next line in the loop, which increases the value of i by 1
- Then (as there is no more code for the program to execute), it goes to the top of the loop, starting the process over again
- The program will keep on looping until the value of the i variable is greater than 10

For Loops

A for loop in Python is a control flow statement that is used to repeatedly execute a group of statements as long as the condition is satisfied.

```
websites = ["facebook.com", "google.com", "amazon.com"]
for site in websites:
    print(site)
```

This for loop shown in the code block above, will run 3 times, outputting each website in the list. Let's break this down. Explanation given in TRYHACKME.

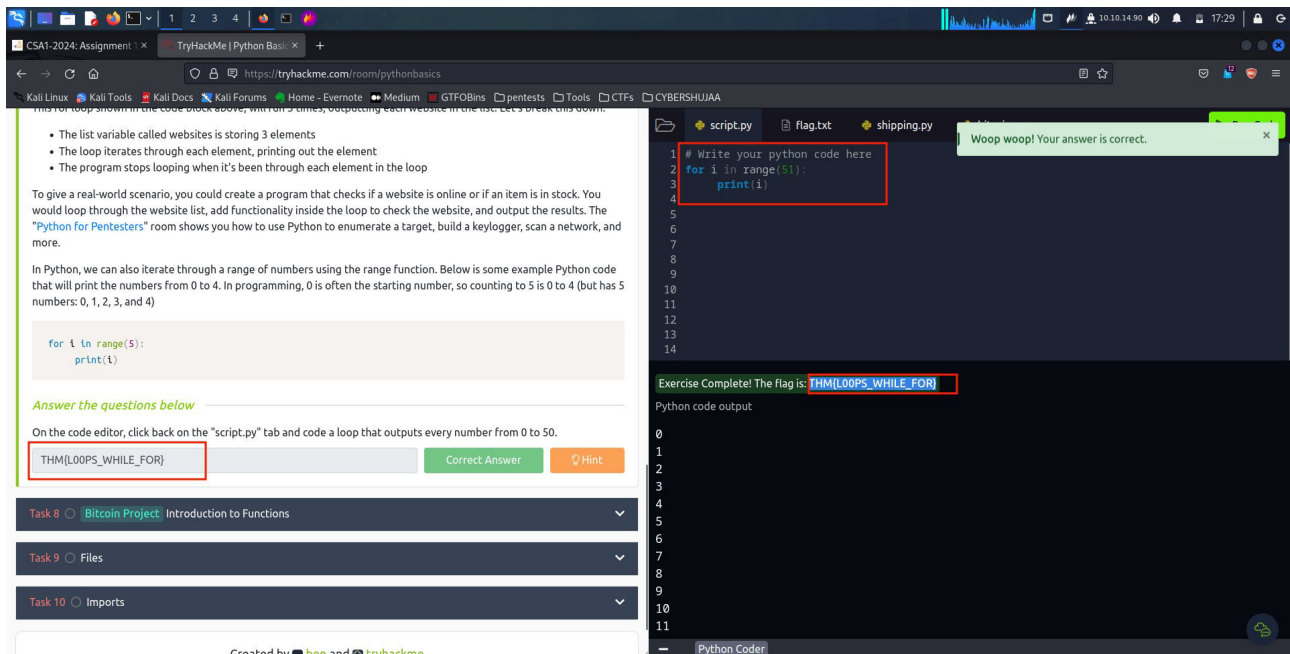
- The list variable called websites is storing 3 elements
- The loop iterates through each element, printing out the element
- The program stops looping when it's been through each element in the loop

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.

Code:

```
for i in range(51):  
    print(i)
```



Bitcoin Project Introduction to Functions

First I start by understanding what is a Function:

A function is a block of code that can be called at different places in your program. As we continue writing a program especially on more complex programs, some lines of codes are repeated to avoid repetition this is where functions come in.

Answer the questions below

You've invested in Bitcoin and want to write a program that tells you when the value of Bitcoin falls below a particular value in dollars.

In the code editor, click on the bitcoin.py tab. Write a function called `bitcoinToUSD` with two parameters: `bitcoin_amount`, the amount of Bitcoin you own, and `bitcoin_value_usd`, the value of bitcoin in USD. The function should return `usd_value`, which is your bitcoin value in USD (to calculate this, in the function, you times `bitcoin_amount` variable by `bitcoin_value_usd` variable and return the value). The start of the function should look like this:

ANS: THM{BITCOIN_INVESTOR}

If we call the `calcCost` function and pass in "sweets" as the item parameter, the function will return a decimal number (float). In the code above, we take a variable called `spent` and add the cost of "sweets" through the `calcCost` function; when we call `calcCost`, it will return the number 3.99.

Answer the questions below

You've invested in Bitcoin and want to write a program that tells you when the value of Bitcoin falls below a particular value in dollars.

In the code editor, click on the `bitcoin.py` tab. Write a function called `bitcoinToUSD` with two parameters: `bitcoin_amount`, the amount of Bitcoin you own, and `bitcoin_value_usd`, the value of bitcoin in USD. The function should return `usd_value`, which is your bitcoin value in USD (to calculate this, in the function, you times `bitcoin_amount` variable by `bitcoin_value_usd` variable and return the value). The start of the function should look like this:

```
def bitcoinToUSD(bitcoin_amount, bitcoin_value_usd):
```

Once you've written the `bitcoinToUSD` function, use it to calculate the value of your Bitcoin in USD, and then create an if statement to determine if the value falls below \$30,000; if it does, output a message to alert you (via a print statement).

THM{BITCOIN_INVESTOR} Correct Answer Hint

1 Bitcoin is now worth \$24,000. In the code editor on line 14, update the `bitcoin_to_usd` variable value to 24000 and see if your Python program recognises that your investment is below the \$30,000 threshold.

No answer needed Completed

Task 9 Files Task 10 Imports

script.py flag.txt shipping.py

```
1 # If your Bitcoin falls below $30,000, print a message.
2
3 You can assume that 1 Bitcoin is worth $40,000
4
5
6
7
8
9
10
11 """
12 investment_in_bitcoin = 1.2
13 bitcoin_to_usd = 40000
14 # 1) write a function to calculate bitcoin to usd
15 def bitcoinToUSD(bitcoin_amount, bitcoin_value_usd):
16     usd_value= bitcoin_amount * bitcoin_value_usd
17     return usd_value
18 answer = bitcoinToUSD(1.2,40000)
19 if answer <= 30000:
20     print(alert)
```

Exercise Complete! The flag is: THM{BITCOIN_INVESTOR}

Python code output

1 Bitcoin is now worth \$24,000. In the code editor on line 14, update the `bitcoin_to_usd` variable value to 24000 and see if your Python program recognises that your investment is below the \$30,000 threshold.

Answer the questions below

You've invested in Bitcoin and want to write a program that tells you when the value of Bitcoin falls below a particular value in dollars.

In the code editor, click on the `bitcoin.py` tab. Write a function called `bitcoinToUSD` with two parameters: `bitcoin_amount`, the amount of Bitcoin you own, and `bitcoin_value_usd`, the value of bitcoin in USD. The function should return `usd_value`, which is your bitcoin value in USD (to calculate this, in the function, you times `bitcoin_amount` variable by `bitcoin_value_usd` variable and return the value). The start of the function should look like this:

```
def bitcoinToUSD(bitcoin_amount, bitcoin_value_usd):
```

Once you've written the `bitcoinToUSD` function, use it to calculate the value of your Bitcoin in USD, and then create an if statement to determine if the value falls below \$30,000; if it does, output a message to alert you (via a print statement).

THM{BITCOIN_INVESTOR} Correct Answer Hint

1 Bitcoin is now worth \$24,000. In the code editor on line 14, update the `bitcoin_to_usd` variable value to 24000 and see if your Python program recognises that your investment is below the \$30,000 threshold.

No answer needed Correct Answer

script.py flag.txt shipping.py

```
1 # If your Bitcoin falls below $30,000, print a message.
2
3 You can assume that 1 Bitcoin is worth $40,000
4
5
6
7
8
9
10
11 """
12 investment_in_bitcoin = 1.2
13 bitcoin_to_usd = 24000
14 # 1) write a function to calculate bitcoin to usd
15 def bitcoinToUSD(bitcoin_amount, bitcoin_value_usd):
16     usd_value= bitcoin_amount * bitcoin_value_usd
17     return usd_value
18 answer = bitcoinToUSD(1.2,40000)
19 if answer <= 30000:
20     print(alert)
```

Exercise Complete! The flag is: THM{BITCOIN_INVESTOR}

Python code output

Files

It is also possible to read and write from files in python

it's common to write a script and import or export it from a file; whether that be as a way to store the output of your script or to import a list of 100's of websites from a file to enumerate.

```
f = open("file_name", "r")
print(f.read())
```

NOTES

- To open the file, we use the built-in open() function.
- "r" parameter stands for "read" which is used as when reading file contents.
- read() method is used for reading the contents of the file

We can append an existing file or create and write to a new file in python.

Appending to an existing file

```
f = open("demofile1.txt", "a") # Append to an existing file
f.write("The file will include more text..")
f.close()
```

Creating and writing to a new file

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

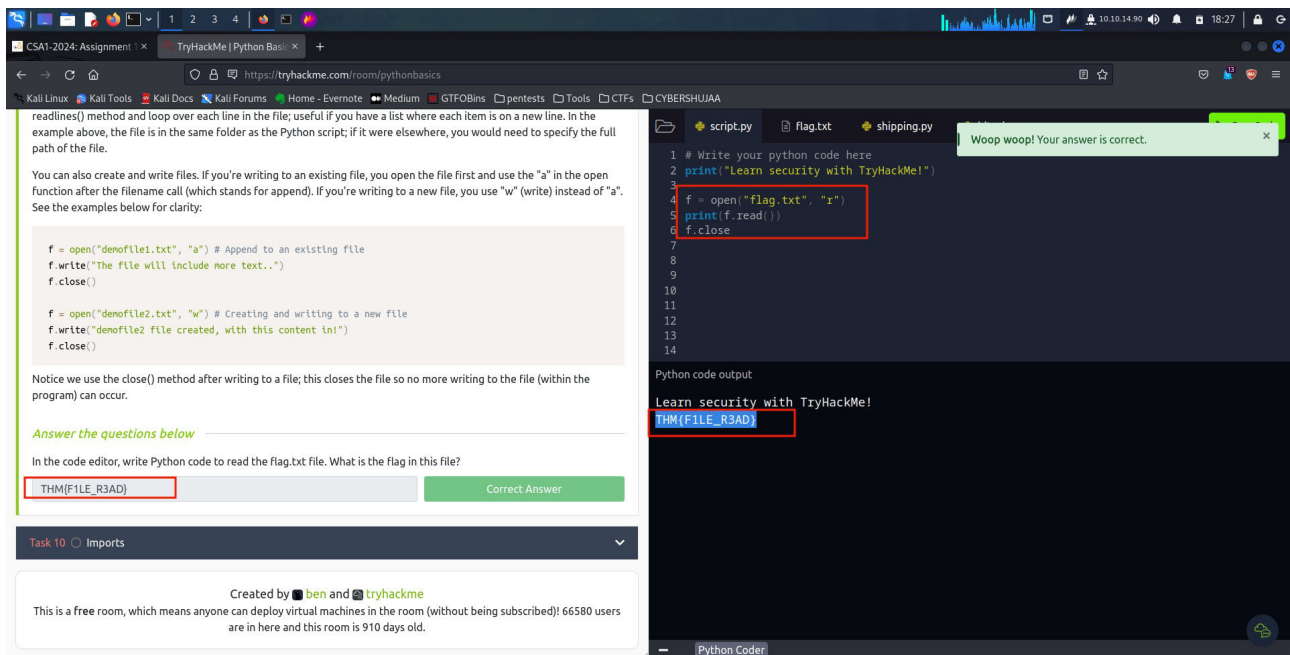
Answer the questions below

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

Code:

```
f = open("flag.txt", "r")
print(f.read())
f.close
```

ANS: THM{F1LE_R3AD}



Imports

In python we can import libraries, which are a collection of files that contain functions.

Importing libraries uses keyword **import**.

Example:

```
import datetime
current_time = datetime.datetime.now()
print(current_time)
```

In the example above we are importing datetime, then access the .now() method by calling library_name.method_name().

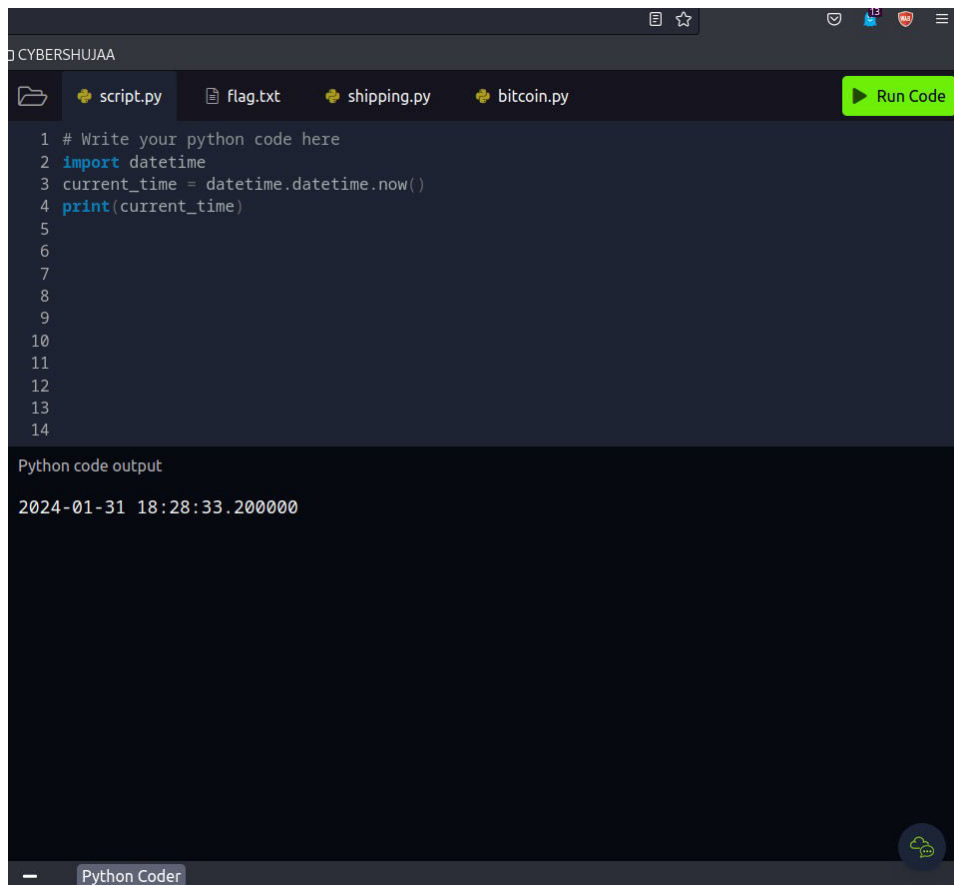
.now() means the current, so this program is supposed to return the exact time and date.

Code:

import datetime

current_time = datetime.datetime.now()

print(current_time)



The screenshot shows a web-based Python code editor. At the top, there's a header with the username 'CYBERSHUIJAA' and a navigation bar with tabs for 'script.py', 'flag.txt', 'shipping.py', and 'bitcoin.py'. A green 'Run Code' button is on the right. The main area contains a Python script with the following code:

```
1 # Write your python code here
2 import datetime
3 current_time = datetime.datetime.now()
4 print(current_time)
5
6
7
8
9
10
11
12
13
14
```

Below the code editor, the output is displayed under the heading 'Python code output':

```
2024-01-31 18:28:33.200000
```

The bottom of the interface shows a 'Python Coder' label and a small chat icon.

conclusion:

In conclusion, the Python Basics module on TryHackMe has given me a solid foundation on how to apply Python programming in the field of cybersecurity and ethical hacking.

I have also been introduced to fundamental concepts such as variables, data types, loops and conditionals, gradually building up to more advanced topics like functions and file handling. This module serves as an excellent starting point in Python for scripting and automation in the cybersecurity field. This module lays the groundwork for further exploration to more specialized areas such as script writing.

Thank You.

