

# Python Scripting (Basic)

## Introduction

For the learner to be equipped well in security analysis role, they need to have a hands-on Python experience and in this module we shall cover several tasks including variables, Loops, functions, Data Structures, if statements, files and how to install Python – for beginners.

## Activities

### Task 1: Introduction to Python

The learner immersed themselves in hands-on with and learning about the scripting programming language Python. It is a great knowledge to have so and allows the learner to create security tools and create quick scripts that will aid in hacking (as well as defending and analysing).

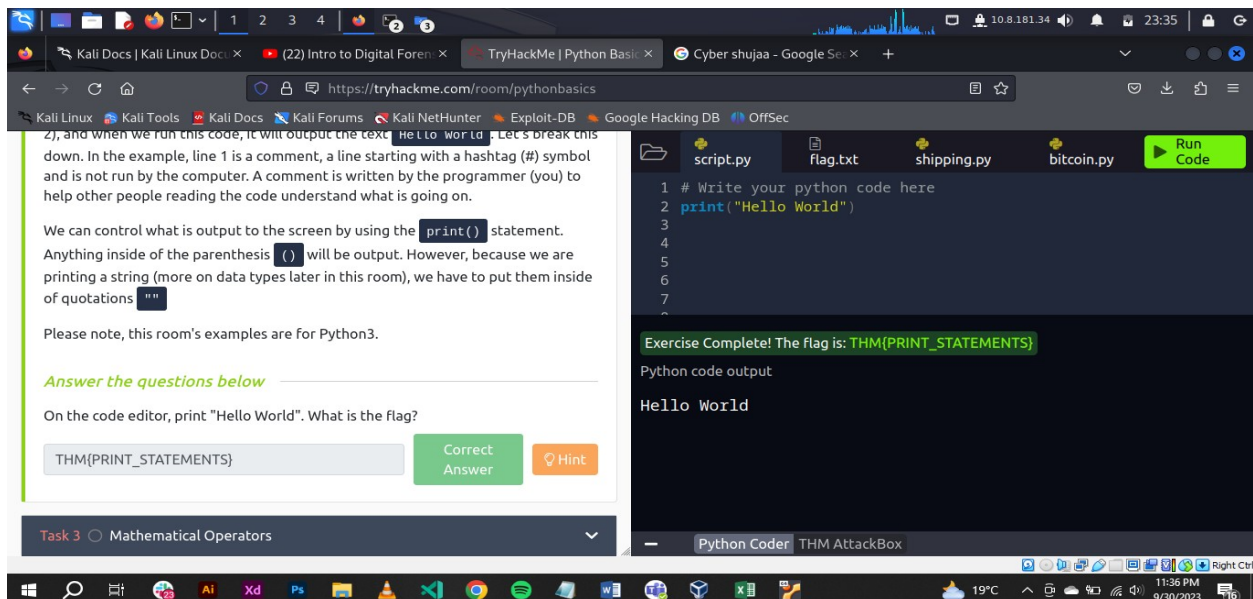
The screenshot displays a web browser window with multiple tabs open, including 'Kali Linux', 'Kali Docs', 'Kali Forums', 'Kali NetHunter', 'Exploit-DB', 'Google Hacking DB', and 'OffSec'. The active tab is 'TryHackMe | Python Basics', showing a tutorial page. The page content includes text explaining the `print()` statement and its use for outputting text. A code editor on the right side of the page shows a Python script: 

```
1 # Write your python code here
2 print("Hello World")
3
4
5
6
7
```

. Below the code editor, a green banner reads 'Exercise Complete! The flag is: THM{PRINT\_STATEMENTS}'. The 'Python code output' section shows 'Hello World'. At the bottom of the page, a task list is visible, with 'Task 3: Mathematical Operators' selected. The browser's address bar shows the URL 'https://tryhackme.com/room/pythonbasics'. The system tray at the bottom of the screen shows the date and time as '11:36 PM 9/30/2023'.

### Task 2: Hello World

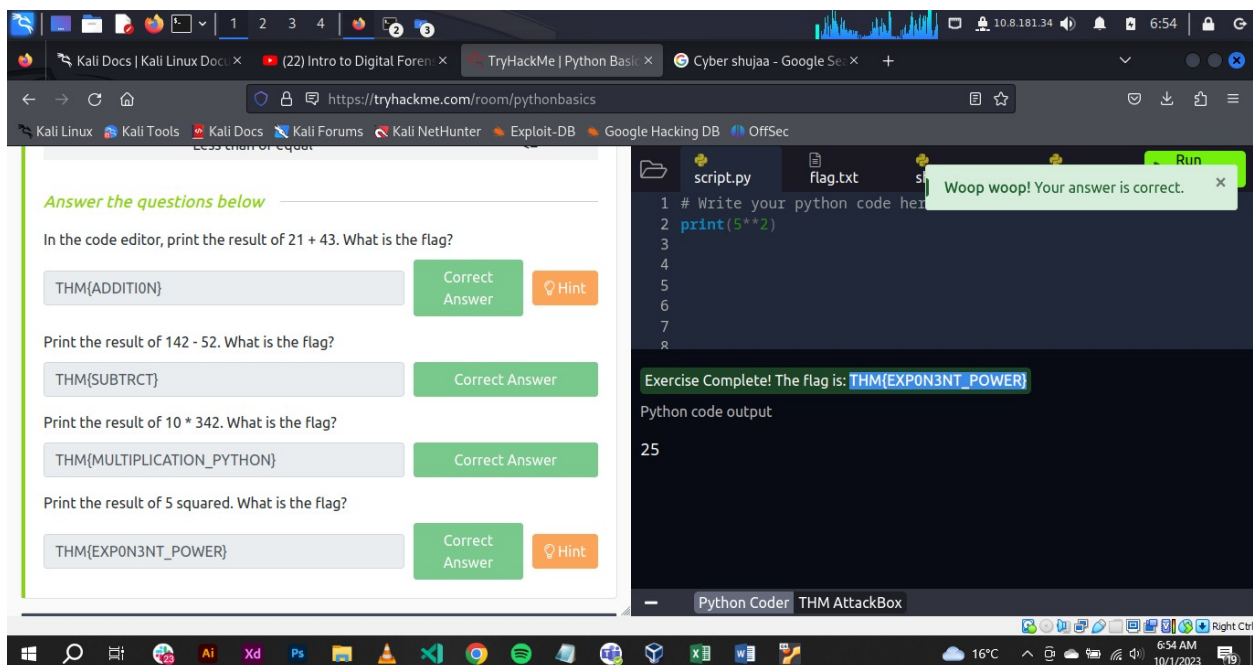
The learner gets started by creating a program that prints out (an output) the statement “Hello world” as shown below.

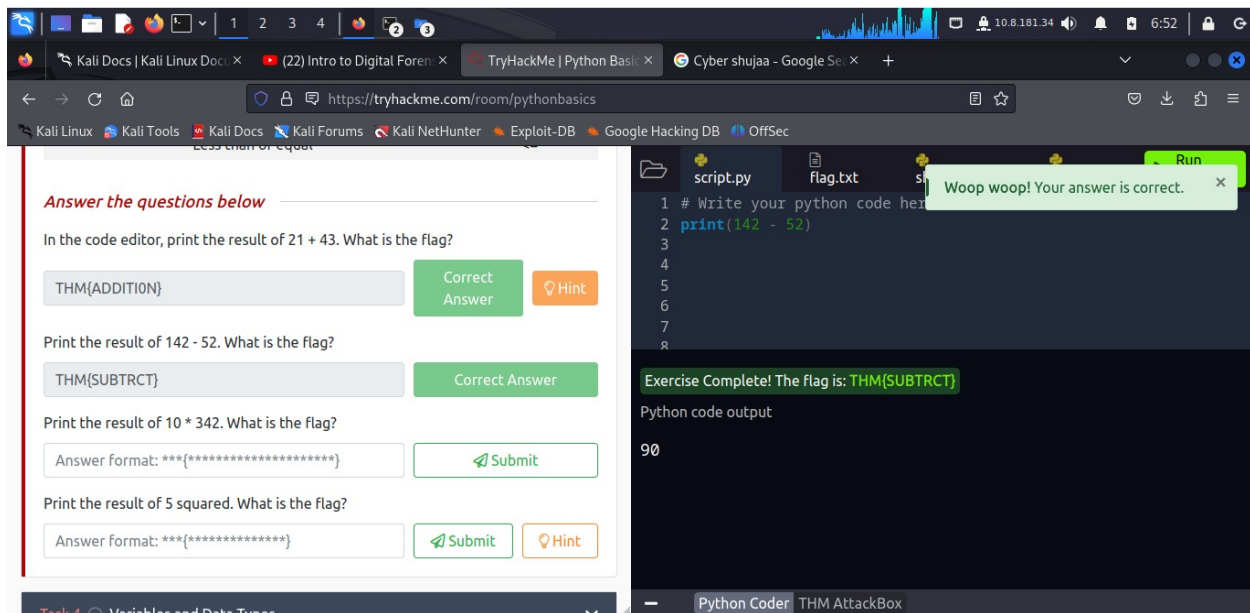


### Task 3: Mathematical Operators

In this task the learner went through understanding Mathematics operation from Addition, multiplication, subtraction, exponent, modulus and division.

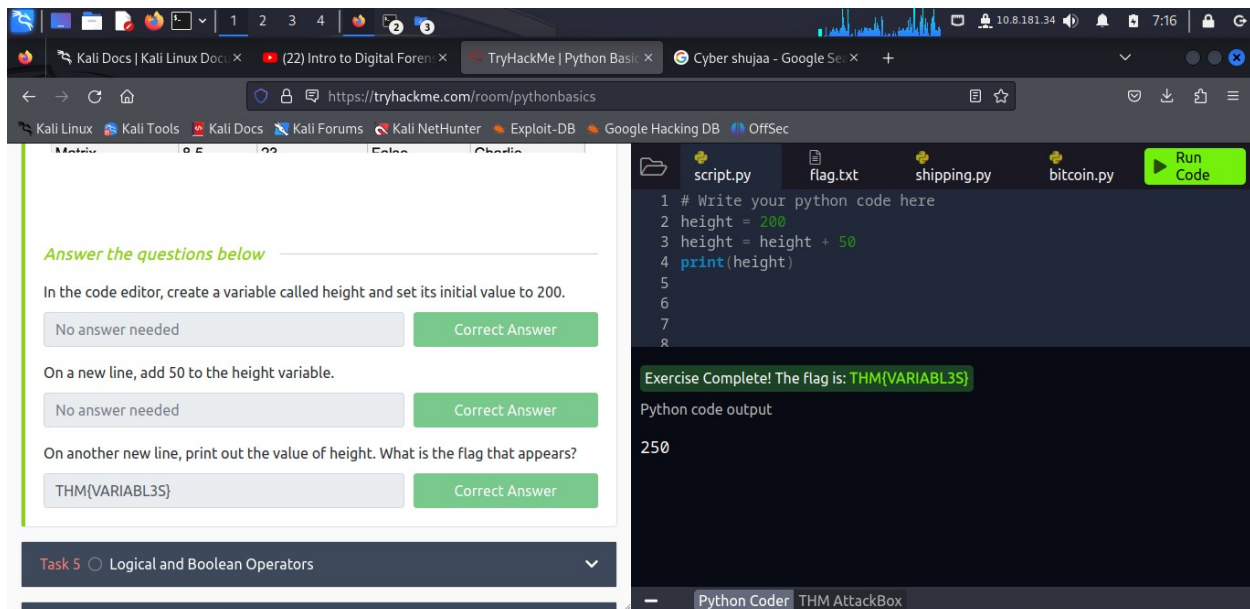
Also learnt is comparison operators: greater than or equal to, less than or equal to, equal to or not equal to.





#### Task 4: Variables and Data Types

Variables allow to store and update data in a computer program. We have a variable name and store data to that name. Data type refers to the type of data being stored in a variable. Text, or numbers, and many other types can be stored. Example of data types are String, Integers, Boolean, Float and List.



## Task 5: Logical and Boolean Operators

Logical operators allow assignment and comparisons to be made and are used in conditional testing (such as if statements). Example: if  $x == 5$  tests equivalence, if  $x < 5$  tests less than operation, if  $x <= 5$  tests less than or equal to operations.

Boolean operators are used to connect and compare relationships between statements. Like an if statement, conditions can be true or false. They are AND, OR and NOT.

The screenshot shows a web browser window with the URL <https://tryhackme.com/room/pythonbasics>. The browser's address bar and tabs are visible at the top. The main content area is split into two panes. The left pane contains two code blocks. The first block shows a simple if statement: 

```
a = 1
if a == 1 or a > 10:
    print("a is either 1 or above 10")
```

. The second block shows an if-elif-else statement: 

```
name = "bob" hungry = True
if name == "bob" and hungry == True:
    print("bob is hungry")
elif name == "bob" and not hungry:
    print("Bob is not hungry")
else: # If all other if conditions are not met
    print("Not sure who this is or if they are hungry")
```

. Below the code blocks, there is a section titled "Answer the questions below" with a prompt "Read the above section." and two buttons: "No answer needed" and "Correct Answer". The right pane is a code editor with a dark background. It shows a Python script: 

```
1 # Write your python code here
2 height = 200
3 height = height + 50
4 print(height)
5
6
7
8
```

. A green banner at the bottom of the code editor says "Exercise Complete! The flag is: THM{VARIABLES}". Below the code editor, the output is shown: "Python code output" followed by "250". The bottom of the browser window shows the task name "Python Basics" and the THM AttackBox logo.

## Task 6: Shipping Project Introduction to If Statements

**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.

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.

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.

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

```
script.py  flag.txt  Run
19 print("Free shipping: ")
20 else:
21     shipping_cost = customer_basket_weight * shipping_cost_per_kg
22     customer_basket_cost += shipping_cost
23
24 print("Total basket cost including shipping is " +
25       str(customer_basket_cost))
```

Exercise Complete! The flag is: **THM{IF\_STATEMENT\_SHOPPING}**

Python code output

Total basket cost including shipping is 86.8

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.

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.

**Task 7** ☐ Loops

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

**Task 9** ☐ Files

```
script.py  flag.txt  Run
10 ==> You've redeemed a h
11 complete this exercise.
12 """
13
14 shipping_cost_per_kg = 1.20
15 customer_basket_cost = 101
16 customer_basket_weight = 44
```

Exercise Complete! The flag is: **THM{MY\_FIRST\_APP}**

Python code output

Free shipping!  
Total basket cost including shipping is 101



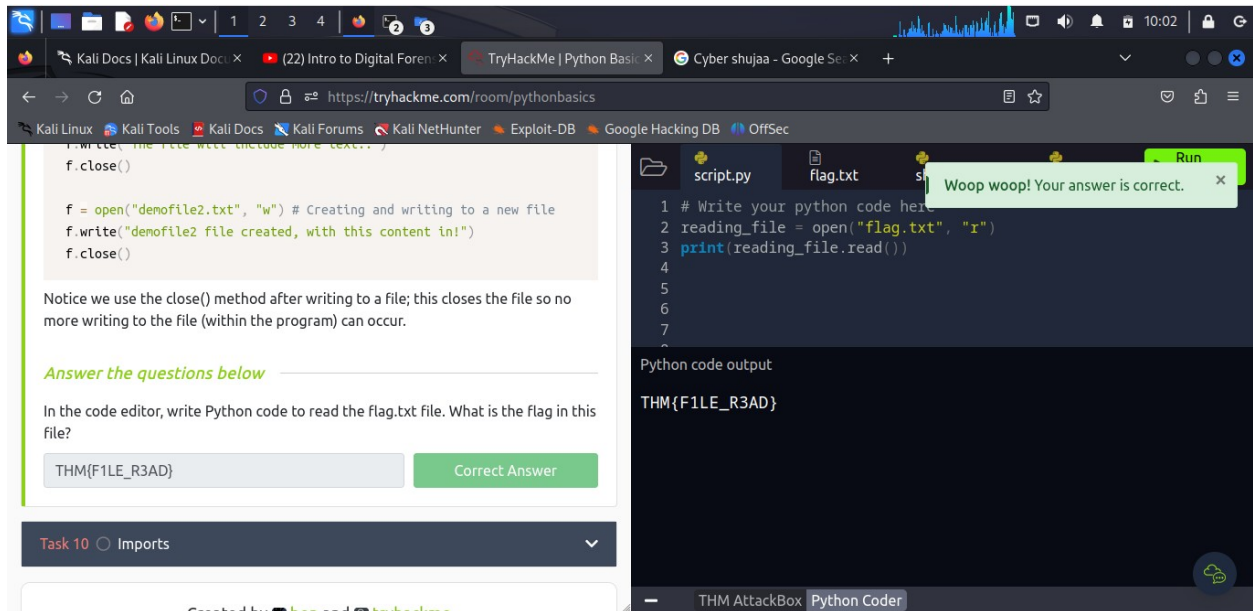
## Task 7: Loops

Loops allow programs to iterate and perform actions a number of times. There are two types of loops, for A for loop is used to iterate over a sequence such as a list. Lists are used to store multiple items in a single variable, and are created using square brackets - example shown:

```
websites = ["facebook.com", "google.com", "amazon.com"]
```

for site in websites:

```
    print(site)
```



## Task 8: Bitcoin Project Introduction to Functions

A function is a block of code that can be called at different places in your program. And, in this we have applied the concept in bitcoin technology to calculate bitcoin converted to USD and created a function to check if amount is below or above the requirement.

The screenshot shows the TryHackMe interface for the Bitcoin project. On the left, the task instructions are displayed, asking the user to write a function called `bitcoinToUSD` that takes `bitcoin_amount` and `bitcoin_value_usd` as parameters. The function should return `usd_value`. Below the instructions, there is a code editor with the following code:

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

Below the code editor, there is a text input field containing `THM{BITCOIN_INVESTOR}` and a "Correct Answer" button. On the right, the code editor shows the implementation of the `bitcoinToUSD` function and an if statement to check if the investment is below or above \$30,000. The code is as follows:

```
def bitcoinToUSD(bitcoin_amount, bitcoin_value_usd):  
    usd_value = bitcoin_amount * bitcoin_value_usd  
    investment_in_usd = bitcoinToUSD(bitcoin_amount, bitcoin_value_usd)  
    if investment_in_usd <= 3000:  
        print("Investment below $30,000! SELL!")  
    else:  
        print("Investment above $30,000")
```

The code editor also shows the output of the code, which is "Investment above \$30,000". A notification bubble says "Woop woop! Your answer is correct." and another says "Exercise Complete! The flag is: THM{BITCOIN\_INVESTOR}". A badge indicates "You've earned the 7 day streak badge! 30 next?"

## Task 9: Files

Python, enables reading and writing from files. In cyber security, it's common to write a script and import or export it from a file.

The screenshot shows the TryHackMe interface for the Files project. On the left, the task instructions are displayed, asking the user to write Python code to read the `flag.txt` file. Below the instructions, there is a code editor with the following code:

```
f = open("demoFile1.txt", "a") # Append to an existing file  
f.write("The file will include more text..")  
f.close()  
  
f = open("demoFile2.txt", "w") # Creating and writing to a new file  
f.write("demoFile2 file created, with this content in!")  
f.close()
```

Below the code editor, there is a text input field containing `THM{FILE_R3AD}` and a "Correct Answer" button. On the right, the code editor shows the implementation of the file operations. The code is as follows:

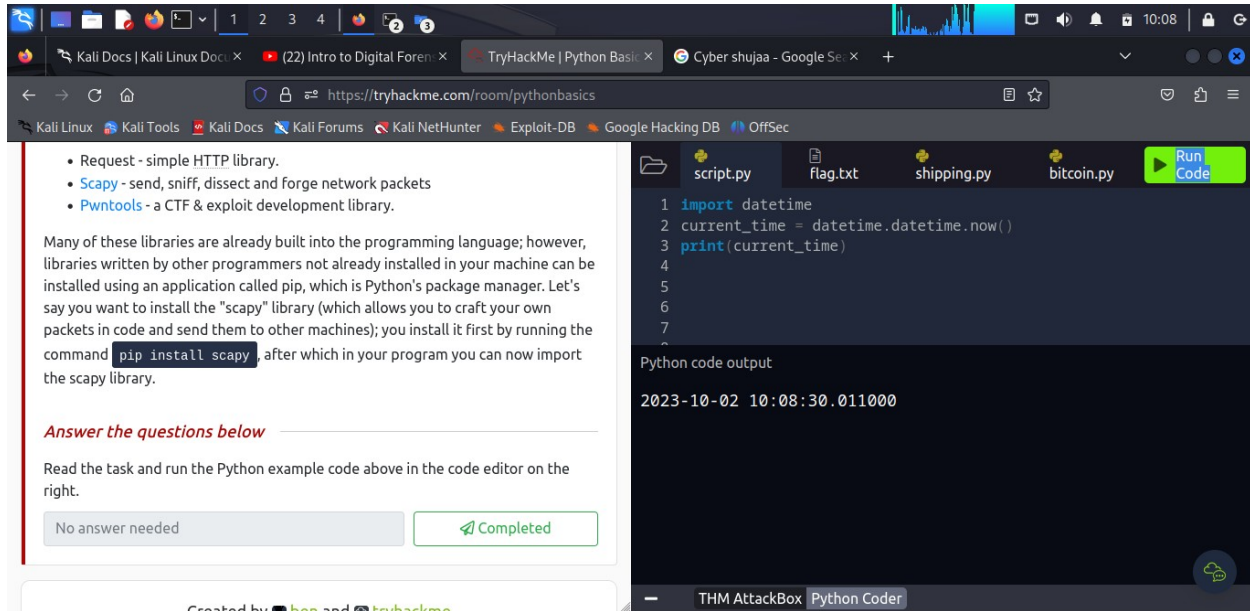
```
# Write your python code here  
reading_file = open("flag.txt", "r")  
print(reading_file.read())
```

The code editor also shows the output of the code, which is the flag `THM{FILE_R3AD}`. A "Run Code" button is visible in the top right corner of the code editor.

## Task 10: Imports

Python is so versatile that it enables the learner to import libraries – libraries can be thought of functions already created in a file for a specific purpose for example importing Pandas and numpy files in analysis operations.

Example: import datetime – depicted in the below description.



The screenshot shows a web browser window with the URL <https://tryhackme.com/room/pythonbasics>. The page content includes:

- Request - simple HTTP library.
- Scapy - send, sniff, dissect and forge network packets
- Pwntools - a CTF & exploit development library.

Many of these libraries are already built into the programming language; however, libraries written by other programmers not already installed in your machine can be installed using an application called pip, which is Python's package manager. Let's say you want to install the "scapy" library (which allows you to craft your own packets in code and send them to other machines); you install it first by running the command `pip install scapy`, after which in your program you can now import the scapy library.

**Answer the questions below**

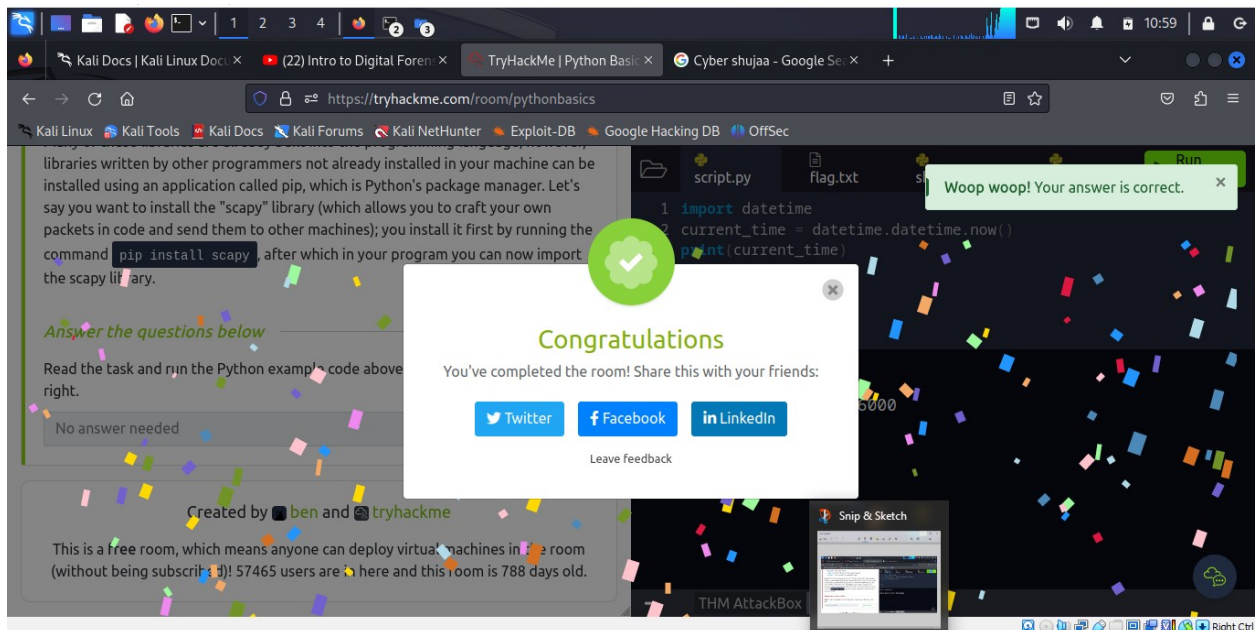
Read the task and run the Python example code above in the code editor on the right.

No answer needed Completed

Python code output

```
1 import datetime
2 current_time = datetime.datetime.now()
3 print(current_time)
4
5
6
7
```

2023-10-02 10:08:30.011000



The screenshot shows the same web browser window, but with a completion modal overlay. The modal text reads:

**Congratulations**

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

[Twitter](#) [Facebook](#) [LinkedIn](#)

[Leave feedback](#)

The background shows the same Python code and output as the previous screenshot, along with a green checkmark icon and a 'Woop woop! Your answer is correct.' message.