

The lab is for python basics, a significant in penetration testing.

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

Answer: **THM{PRINT_STATEMENTS}**

The screenshot shows the TryHackMe Python Basics room. On the left, the 'Room progress (11%)' section lists tasks: Task 3 (Mathematical Operators), Task 4 (Variables and Data Types), Task 5 (Logical and Boolean Operators), Task 6 (Shipping Project), and Task 7 (Loops). The 'Loops' task is highlighted with a green circle. The main area contains a question: 'On the code editor, print "Hello World". What is the flag?'. Below the question, the answer 'THM{PRINT_STATEMENTS}' is entered in a text box, and a green 'Correct Answer' button is visible. On the right, the 'script.py' file in the code editor contains the following Python code:

```
1 # Write your python code here
2 print("Learn security with TryHackMe!")
3 print("Hello World!")
4
5
6
7
8
9
10
11
12
```

 A red box highlights the second line of code. A red callout bubble points to it with the text 'Hello World! is the code I run to print the flag'. Below the code editor, the output shows 'Learn security with TryHackMe!' and 'Hello World!'. A green banner at the bottom of the code editor states 'Exercise Complete! The flag is: THM{PRINT_STATEMENTS}'.

Print (21+43)

The screenshot shows the TryHackMe Python Basics room. On the left, the 'Room progress (16%)' section lists tasks: Task 3 (Mathematical Operators), Task 4 (Variables and Data Types), Task 5 (Logical and Boolean Operators), Task 6 (Shipping Project), and Task 7 (Loops). The 'Loops' task is highlighted with a green circle. The main area contains a question: 'In the code editor, print the result of 21 + 43. What is the flag?'. Below the question, the answer 'THM{ADDITION}' is entered in a text box, and a green 'Correct Answer' button is visible. On the right, the 'script.py' file in the code editor contains the following Python code:

```
1 # Write your python code here
2 print("Learn security with TryHackMe!")
3 print(21 + 43)
4
5
6
7
8
9
10
11
12
```

 A red box highlights the third line of code. Below the code editor, the output shows 'Learn security with TryHackMe!' and '64'. A green banner at the bottom of the code editor states 'Exercise Complete! The flag is: THM{ADDITION}'.

Performing subtraction to get the flag: print(142-52)

The screenshot shows the TryHackMe Python Basics room interface. On the left, a list of tasks is visible, with 'Variables and Data Types' selected. The main area displays a series of exercises. The first exercise, 'Less than or equal', is completed. The second exercise, 'In the code editor, print the result of 21 + 43. What is the flag?', is completed with the answer 'THM[ADDITION]'. The third exercise, 'Print the result of 142 - 52. What is the flag?', is completed with the answer 'THM[SUBTRACT]'. The fourth exercise, 'Print the result of 10 * 342. What is the flag?', is pending submission. The fifth exercise, 'Print the result of 5 squared. What is the flag?', is pending submission. On the right, a code editor shows a Python script with the following code:

```
1 # Write your python code here
2 print("Learn security with TryHackMe!")
3 print(142-52)
4
5
6
7
8
9
10
11
```

The output of the code is displayed below the editor:

```
Learn security with TryHackMe!
90
```

The bottom status bar shows the system clock as 9:50 PM on 8/17/2025.

Performing multiplication

The screenshot shows the TryHackMe Python Basics room interface. On the left, a list of tasks is visible, with 'Variables and Data Types' selected. The main area displays a series of exercises. The first exercise, 'Less than or equal', is completed. The second exercise, 'In the code editor, print the result of 21 + 43. What is the flag?', is completed with the answer 'THM[ADDITION]'. The third exercise, 'Print the result of 142 - 52. What is the flag?', is completed with the answer 'THM[SUBTRACT]'. The fourth exercise, 'Print the result of 10 * 342. What is the flag?', is completed with the answer 'THM[MULTIPLICATION_PYTHON]'. The fifth exercise, 'Print the result of 5 squared. What is the flag?', is pending submission. On the right, a code editor shows a Python script with the following code:

```
1 # Write your python code here
2 print("Learn security with TryHackMe!")
3 print(10*342)
4
5
6
7
8
9
10
11
```

The output of the code is displayed below the editor:

```
Learn security with TryHackMe!
3420
```

The bottom status bar shows the system clock as 9:55 PM on 8/17/2025.

Performing exponential

The screenshot shows the TryHackMe Python Basics room interface. On the left, a list of tasks is visible, with 'Task 4: Variables and Data Types' selected. The main area displays three questions with their answers: 'THM[SUBTRACT]' (Correct Answer), 'THM[MULTIPLICATION_PYTHON]' (Correct Answer), and 'THM[EXPONENT_POWER]' (Correct Answer). The Python code editor on the right shows the following code:

```
1 # Write your python code here
2 print("Learn security with TryHackMe!")
3 print(5**2)
4
5
6
7
8
9
10
11
12
```

A red box highlights the code `print(5**2)` with the annotation "I performed exponential to get the flag below". The output of the code is shown as:

```
Learn security with TryHackMe!
25
```

The flag is displayed as `THM[EXPONENT_POWER]`.

Creating variables in Python

The screenshot shows the TryHackMe Python Basics room interface. On the left, a list of tasks is visible, with 'Task 5: Logical and Boolean Operators' selected. The main area displays a table with the following data:

Matrix	8.5	23	False	Charlie
Indiana Jones	6.1	3	False	Daniel, Evie

Below the table, the instructions are: "Answer the questions below", "In the code editor, create a variable called height and set its initial value to 200.", "On a new line, add 50 to the height variable.", and "On another new line, print out the value of height. What is the flag that appears?". The Python code editor on the right shows the following code:

```
1 # Write your python code here
2 print("Learn security with TryHackMe!")
3
4 height = 200
5 height += 50
6 print(height)
7
8
9
10
11
12
```

A red box highlights the code `height = 200` and `height += 50` with the annotation "command output". The output of the code is shown as:

```
Learn security with TryHackMe!
250
```

The flag is displayed as `THM[VARIABLES]`.

I used for loop to get the flag as shown in the diagram below.

The screenshot shows the TryHackMe Python Basics room. On the left, the instructions explain using the range function to iterate through a range of numbers. A code block shows:

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

 Below this, a text input field contains the flag `THM[LOOPS_WHILE_FOR]`, which is marked as the correct answer. On the right, the code editor shows a script.py file with the following code:

```
1 # Write your python code here  
2 print("Learn security with TryHackMe!")  
3  
4 for i in range(51) # print poitive interger 1 to 50 use for loop or while  
5   print(i)  
6  
7  
8  
9  
10
```

 A red box highlights the for loop, and a red arrow points from a text box saying "Below is the command output and the subsequent flag" to the output area. The output shows the text "Learn security with TryHackMe!" followed by numbers 0 through 10. The exercise is marked as complete with the flag `THM[LOOPS_WHILE_FOR]`.

I written function in python to calculate Bitcoin in USD. See the figure below

The screenshot shows the TryHackMe Python Basics room. On the left, the instructions explain writing a function called `bitcoinToUSD` with two parameters: `bitcoin_amount` and `bitcoin_value_usd`. A code block shows the function definition:

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

 Below this, a text input field contains the flag `THM[BITCOIN_INVESTOR]`, which is marked as the correct answer. On the right, the code editor shows a script.py file with the following code:

```
14 bitcoin_to_usd = 40000  
15  
16 # 1) write a function to calculate bitcoin to usd  
17 def bitcoinToUSD(bitcoin_amount, bitcoin_value_usd):  
18     total = bitcoin_amount * bitcoin_value_usd  
19     return total  
20 # 2) use function to calculate if the investment is below $30,000  
21 amount = bitcoinToUSD(1.2, 40000)  
22 if(amount < 30000):  
23     print("Price fall!")  
24 elif (total == 30000):  
25     print("A better price")
```

 A red box highlights the function definition, and a red arrow points from a text box saying "The code include function and if and else, elif loops" to the if/elif block. The exercise is marked as complete with the flag `THM[BITCOIN_INVESTOR]`. The output shows "Price fall!".

Reading the flat.txt file using Python code

The screenshot displays a web browser window with the URL `tryhackme.com/room/pythonbasics`. The page is titled "Room progress (94%)" and shows a list of files: `script.py`, `flag.txt`, `shipping.py`, and `bitcoin.py`. A "Run Code" button is visible.

On the left, a code editor shows the following Python 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, a note states: "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."

A section titled "Answer the questions below" contains a question: "In the code editor, write Python code to read the flag.txt file. What is the flag in this file?". The answer input field contains `THM{FILE_R3AD}`, and a "Correct Answer" button is visible.

On the right, a code editor shows the following Python code:

```
2 print("Learn security with TryHackMe!")
3
4 f = open("flag.txt", "r") # read the file
5 print(f.read()) # print the flag on the screen
6 #f.close()
7
8
9
10
11
12
13
```

Below the code, the "Python code output" section displays:

```
Learn security with TryHackMe!
THM{FILE_R3AD}
```

A red box highlights the output `THM{FILE_R3AD}`, and a label "command output" points to it.

At the bottom, a "Task 10" section is visible, along with a "Created by" section listing users: `ben`, `tryhackme`, and `str3g4tt4`. The "Room Type" is "Python Basics", and the "Users in Room" are listed as `ben`, `tryhackme`, and `str3g4tt4`.