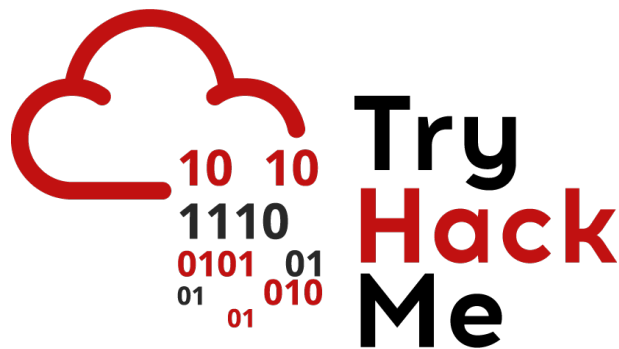


Writeup: Sala *Python Basics*

Autor: Ismaeldevs

Plataforma: TryHackMe

4 de julio de 2025



Índice

1. Introducción	2
2. Sala	2
2.1. Tarea 1 - Introducción a Python	2
2.2. Tarea 2 - Hello World	3
2.3. Tarea 3 - Operadores Matemáticos	3
2.4. Tarea 4 - Variables y Tipos de Datos	4
2.5. Tarea 5 - Operadores lógicos y Booleanos	5
2.6. Tarea 6 - Introducción a las declaraciones If	6
2.7. Tarea 7 - Bucles	8
2.8. Tarea 8 - Introducción a las Funciones	8
2.9. Tarea 9 - Archivos	10
2.10. Tarea 10 - Importaciones	11
3. Conclusión sobre la Sala	11

1. Introducción

En esta sala de Python Basics vamos a introducirnos en los fundamentos del lenguaje de programación **Python**, uno de los lenguajes más populares y versátiles en el mundo del desarrollo y la ciberseguridad. A lo largo de esta sala aprenderemos los conceptos esenciales como variables, tipos de datos, estructuras de control, funciones, listas, diccionarios y más.

2. Sala

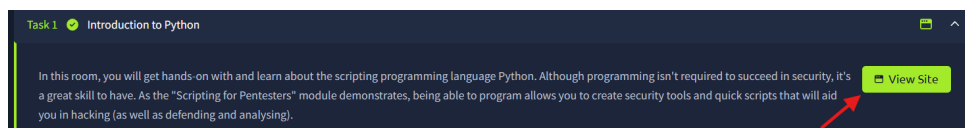
2.1. Tarea 1 - Introducción a Python

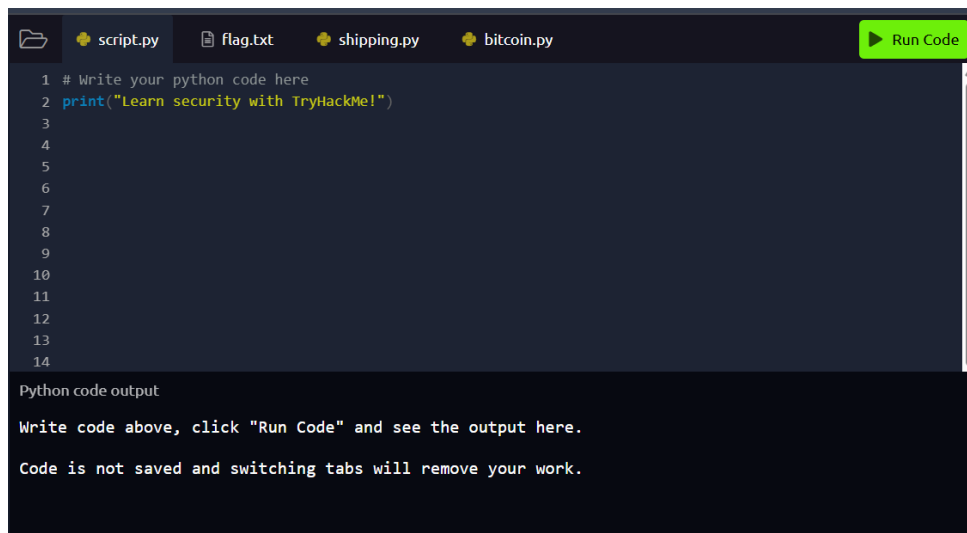
En esta primera tarea conoceremos al lenguaje de programación Python, destacando su importancia y utilidad en el ámbito de la ciberseguridad. Aunque no es obligatorio saber programar para trabajar en seguridad informática, tener conocimientos en scripting puede ser una herramienta poderosa para automatizar tareas, desarrollar herramientas personalizadas y analizar datos de manera eficiente.

Algunos conceptos que se aprenderán son:

- **Variables**
- **Bucles**
- **Funciones**
- **Estructura de Datos**
- **Condicionales**
- **Manejo de archivos**

Para empezar a usar Python debemos levantar nuestro entorno donde escribiremos y ejecutaremos código. Para ello, tenemos que hacer clic en **View Site** y se nos abra el entorno con diferentes pestañas donde escribiremos código.





```
1 # Write your python code here
2 print("Learn security with TryHackMe!")
3
4
5
6
7
8
9
10
11
12
13
14
```

Python code output

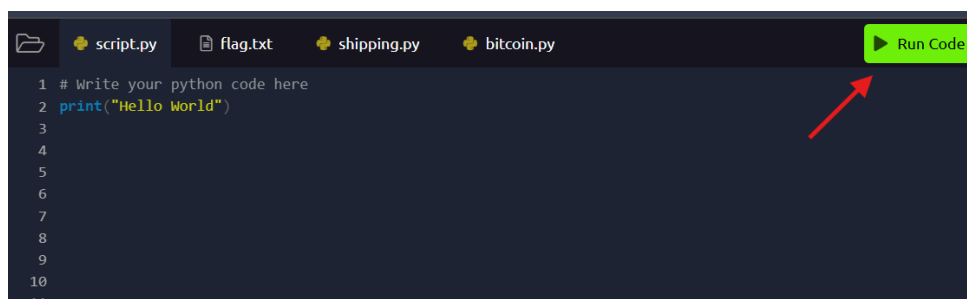
Write code above, click "Run Code" and see the output here.

Code is not saved and switching tabs will remove your work.

2.2. Tarea 2 - Hello World

Esta tarea tiene como objetivo principal aprender la función **print()**, que permite mostrar mensajes en la consola.

Para completar esta tarea debemos escribir el siguiente código en nuestro entorno en script.py: **print("Hello World")** y después le daremos a **Run Code**.



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

Procederá a ejecutarse y mostrara el mensaje junto con la flag de respuesta que necesitamos para completar la tarea.

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

Respuesta: **THM{PRINT_STATEMENTS}**

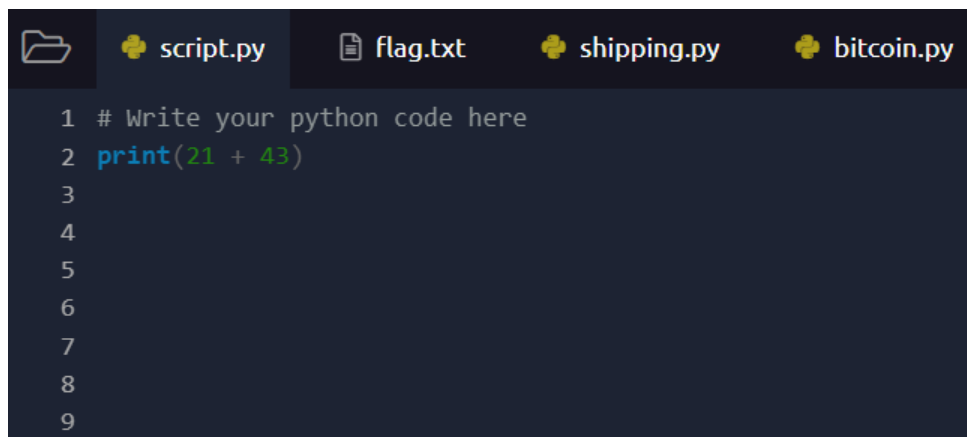
2.3. Tara 3 - Operadores Matemáticos

Ahora aprenderemos sobre los operadores matemáticos fundamentales en Python, permitiendo realizar cálculos básicos directamente en el código. Los operadores que veremos son:

- Suma (+).
- Resta (-).

- Multiplicación (*).
- División (/).
- Módulo o Porcentaje (%).
- Potencia (**).

Procederemos a hacer uso de estos operadores para obtener las diferentes flags para completar la tarea. Vamos a ir escribiendo los diferentes ejercicios en la función **print()** y ejecutando el código en nuestro entorno.

A screenshot of a code editor interface. At the top, there are five tabs: a folder icon, 'script.py', 'flag.txt', 'shipping.py', and 'bitcoin.py'. The 'script.py' tab is active. The editor shows a Python script with line numbers 1 through 9 on the left. The code is: 1 # Write your python code here, 2 print(21 + 43), 3, 4, 5, 6, 7, 8, 9.

```
1 # Write your python code here
2 print(21 + 43)
3
4
5
6
7
8
9
```

Una vez que escribimos el código, le daremos a **Run Code** en la parte superior y nos mostrara la flag de respuesta.

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

Respuesta: **THM{ADDITION}**

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

Respuesta: **THM{SUBTRACT}**

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

Respuesta: **THM{MULTIPLICATION_PYTHON}**

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

Respuesta: **THM{EXPONENT_POWER}**

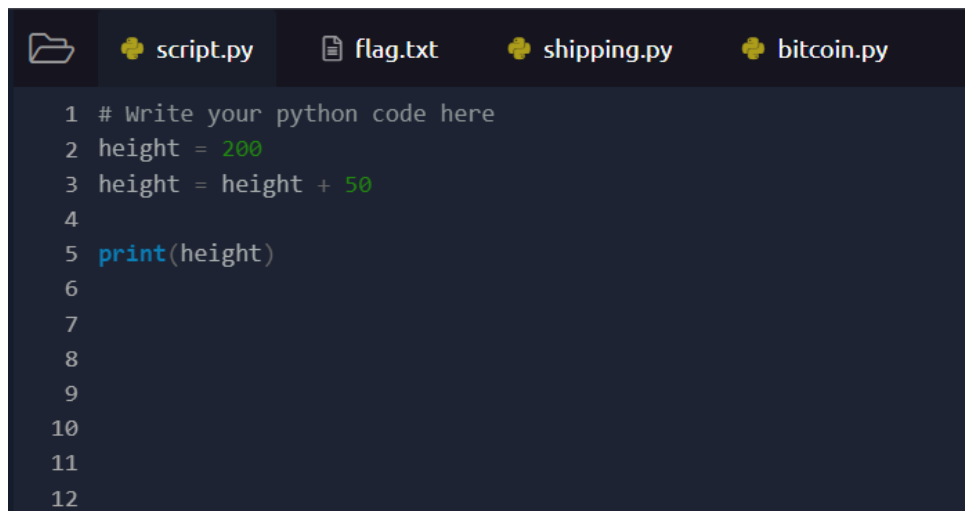
2.4. Tarea 4 - Variables y Tipos de Datos

Aquí aprenderemos dos conceptos fundamentales en Python que son las **variables** y los **tipos de datos**. Los diferentes tipos de datos que python logra admitir son:

- **String:** Secuencia de caracteres

- **Integer:** Números enteros
- **Float:** Números decimales
- **Boolean:** Valores de verdad
- **List:** Colección ordenada de elementos

Para completar esta tarea debemos realizar un ejercicio práctico que al resolverlo se nos proporcionara una flag de respuesta. Primero debemos crear una variable llamada **height** y asignarle el valor 200 para después sumarle 50 de valor, por último, imprimir la variable por medio de la función **print()** y procedemos a darle a **Run Code**.

A screenshot of a code editor interface. At the top, there are five tabs: a folder icon, 'script.py', 'flag.txt', 'shipping.py', and 'bitcoin.py'. The 'script.py' tab is active. The code editor shows a Python script with the following lines:

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

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

Respuesta: **No requiere respuesta** (Hacemos clic en **Submit**).

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

Respuesta: **No requiere respuesta** (Hacemos clic en **Submit**).

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

Respuesta: **THM{VARIABLES}**

2.5. Tarea 5 - Operadores lógicos y Booleanos

Procedemos a introducirnos en los operadores lógicos y booleanos en Python, fundamentales para la toma de decisiones dentro de los programas mediante estructuras condicionales.

Los operadores lógicos permiten realizar comparaciones entre valores y son esenciales para evaluar condiciones. Los principales operadores lógicos en Python son:

- **>** : Mayor que
- **<** : Menor que
- **>=** : Mayor o igual que
- **<=** : Menor o igual que
- **==** : Igual a
- **!=** : Diferente que

Después están los operadores booleanos que se emplean para combinar múltiples condiciones lógicas:

- **and** : Devuelve **True** si ambas condiciones son verdad.
- **or** : Devuelve **True** si al menos una de las condiciones es verdadera
- **not** : Invierte el valor de verdad de una condición.

Pregunta: Read the above section.

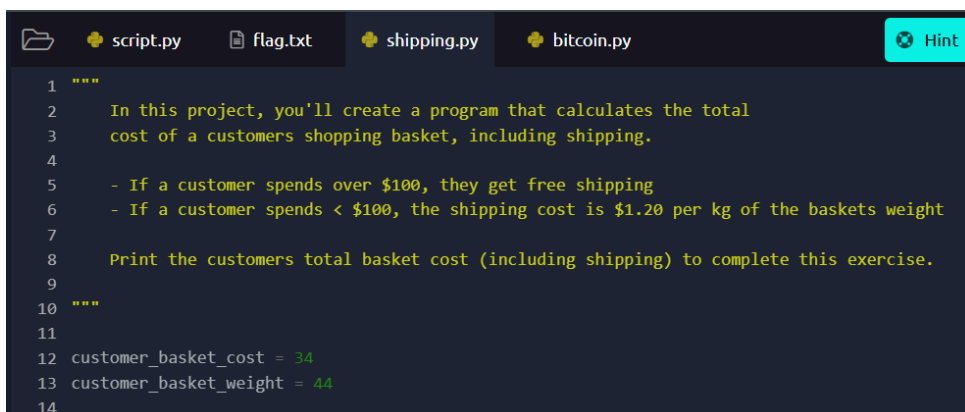
Respuesta: **No requiere respuesta** (Hacemos clic en **Submit**).

2.6. Tarea 6 - Introducción a las declaraciones If

Entramos en el concepto de las declaraciones condicionales if en Python, que son fundamentales para permitir que un programa tome decisiones basadas en condiciones específicas. Algunos conceptos clave para entender las declaraciones:

- **if** : Evalúa una condición que si es verdadera, ejecuta el bloque de código.
- **else** : Se ejecuta si la condición del **if** es falsa.
- **else if** : Permite verificar múltiples condiciones adicionales si las anteriores no se cumplen.

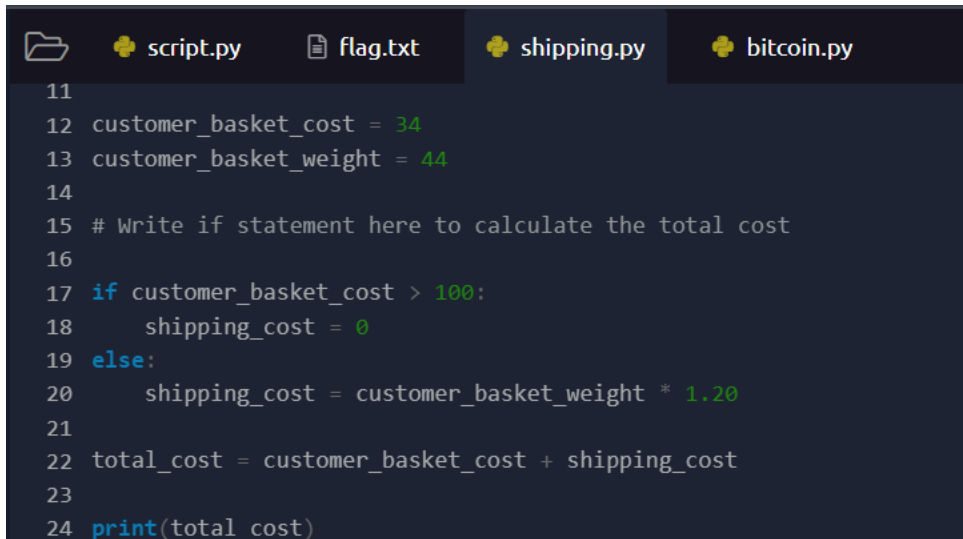
Una vez que aprendimos estos conceptos debemos resolver un ejercicio práctico en el archivo **shipping.py** para obtener la flag de respuesta.



```
1 """
2     In this project, you'll create a program that calculates the total
3     cost of a customers shopping basket, including shipping.
4
5     - If a customer spends over $100, they get free shipping
6     - If a customer spends < $100, the shipping cost is $1.20 per kg of the baskets weight
7
8     Print the customers total basket cost (including shipping) to complete this exercise.
9
10 """
11
12 customer_basket_cost = 34
13 customer_basket_weight = 44
14
```

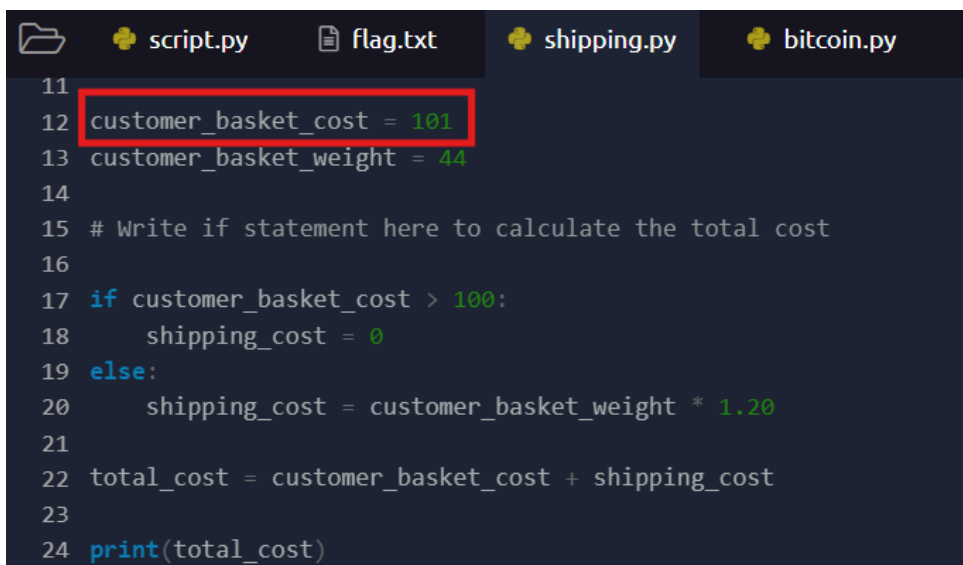
Para lograr completar este ejercicio debemos realizar lo siguiente:

1. Utilizar una declaración **if** para verificar si el costo de la cesta supera los 100
2. Si la condición se cumple, asignar en un variable el costo 0 de la cesta.
3. Si no se cumple, calcular el costo de envío multiplicando el peso por 1.20 y sumar este valor al costo de la cesta.
4. Imprimir el costo.



```
11
12 customer_basket_cost = 34
13 customer_basket_weight = 44
14
15 # Write if statement here to calculate the total cost
16
17 if customer_basket_cost > 100:
18     shipping_cost = 0
19 else:
20     shipping_cost = customer_basket_weight * 1.20
21
22 total_cost = customer_basket_cost + shipping_cost
23
24 print(total_cost)
```

Una vez que ejecutamos el código nos mostrara una de las flag de respuesta, para obtener la otra flag debemos cambiar el valor de la cesta a **101** y volver a correr nuestro código.



```
11
12 customer_basket_cost = 101
13 customer_basket_weight = 44
14
15 # Write if statement here to calculate the total cost
16
17 if customer_basket_cost > 100:
18     shipping_cost = 0
19 else:
20     shipping_cost = customer_basket_weight * 1.20
21
22 total_cost = customer_basket_cost + shipping_cost
23
24 print(total_cost)
```

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

Respuesta: **No requiere respuesta** (Hacemos clic en **Submit**).

Pregunta: 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.

Respuesta: **THM{IF_STATEMENT_SHOPPING}**

Pregunta: In shipping.py, on line 15, change the customer_basket_cost variable to 101 and re-run your code. You will get a flag; the flag is the answer to this question.

Respuesta: **THM{MY_FIRST_APP}**

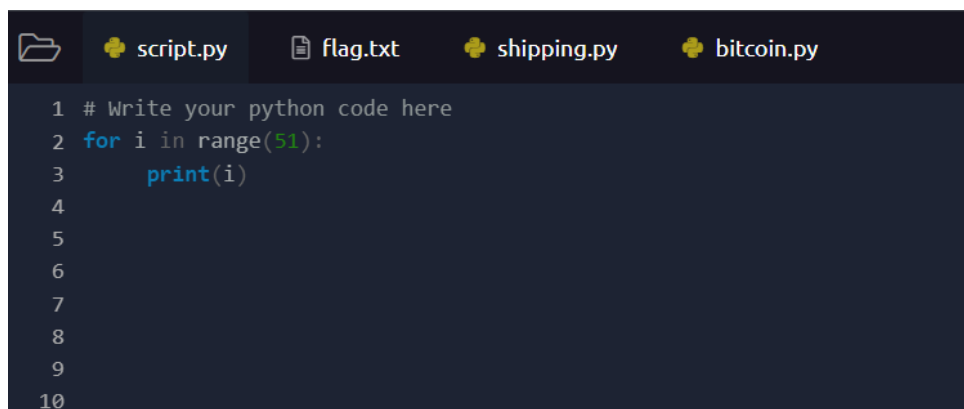
2.7. Tarea 7 - Bucles

En esta tarea exploraremos los bucles, estructuras fundamentales que permiten ejecutar un bloque de código repetidamente bajo ciertas condiciones. Se enfocan en dos tipos principales de bucles que son **while** y **for**.

Una vez que aprendemos como funcionan los bucles realizaremos un ejercicio para conseguir la flag de respuesta.

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

Para resolverlo debemos ir al archivo **script.py** y crear un bucle **for** que recorra los números del 0 hasta el 50, después ejecutamos nuestro código y nos proporcionara la flag.



```
1 # Write your python code here
2 for i in range(51):
3     print(i)
4
5
6
7
8
9
10
```

Respuesta: **THM{LOOPS_WHILE_FOR}**

2.8. Tarea 8 - Introducción a las Funciones

Aprenderemos en esta tarea sobre funciones en Python, que son bloques de código diseñados para realizar una tarea específica. Las funciones permiten estructurar el

código de manera más organizada y eficiente, facilitando su mantenimiento y reutilización.

Después de entender como se utilizan las funciones vamos a realizar el siguiente ejercicio para obtener la flag para completar la tarea.

Pregunta: 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.

Para ello necesitamos dirigirnos al archivo **bitcoin.py** y leer el ejercicio que nos proporcionan y lo que debemos hacer para completarlo. La solución al problema es la siguiente:

1. Creamos una función llamada **bitcoinToUSD** donde reciba por parametros dos valores: cantidad de bitcon y el valor del bitcoin en dolares USD.
2. En la función vamos a multiplicar la cantidad por el valor y el resultado asignarla en una variable que tendrá el valor en dolares usd y retonarla
3. Después fuera de la función debemos almacenar en una variable el resultado de la función haciendo un llamado a la misma y enviandole las variables con los valores por parámetro.
4. Luego declararemos un if para comprobar si el valor en dolares USD es menor a 30k.
5. Por último, imprimimos un mensaje en caso que se cumpla la condición.



```
13 investment_in_bitcoin = 1.2
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     usd_value = bitcoin_amount * bitcoin_value_usd
19     return usd_value
20
21 # 2) use function to calculate if the investment is below $30,000
22
23 usd_value = bitcoinToUSD(investment_in_bitcoin, bitcoin_to_usd)
24
25 if usd_value < 30000:
26     print("El valor del Bitcoin ha caído por debajo de $30,000")
```

Respuesta: THM{BITCOIN_INVESTOR}

Pregunta: 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.

Respuesta: **No requiere respuesta** (Hacemos clic en **Submit**).

2.9. Tarea 9 - Archivos

Ahora aprenderemos en esta tarea sobre el manejo de archivos, una habilidad esencial para leer y escribir datos desde y hacia archivos, lo cual es común en la automatización de tareas y en la manipulación de datos en ciberseguridad.

Esta tarea también cuenta con un ejercicio el cual tenemos que solucionar para conseguir la flag de respuesta.

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

Para completar este ejercicio debemos dirigirnos al archivo **script.py** y realizar lo siguiente:

1. Asignar en una variable el resultado de la función **open('flag.txt' 'r')**
2. Después imprimir la variable dentro de una función **print()** con la propiedad **.read()** para leer y mostrar los datos del archivo.

A screenshot of a code editor interface. At the top, there are four tabs: 'script.py' (active), 'flag.txt', 'shipping.py', and 'bitcoin.py'. The 'script.py' tab is selected, showing a Python script with the following code:

```
1 # Write your python code here
2 f = open("flag.txt", "r")
3 print(f.read())
4
5
6
7
8
9
10
11
12
13
14
```

Below the code editor, there is a section labeled 'Python code output' which displays the text 'THM{FILE_R3AD}'.

Respuesta: **THM{FILE_R3AD}**

2.10. Tarea 10 - Importaciones

Esta última tarea aprenderemos los concepto de importar módulos, una práctica esencial que permite reutilizar código existente y aprovechar funcionalidades adicionales sin necesidad de escribirlas desde cero.

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

Respuesta: **No requiere respuesta** (Hacemos clic en **Submit**).

3. Conclusión sobre la Sala

En esta sala se aprendieron conceptos claves como la impresión de texto, operadores matemáticos y lógicos, manejo de variables y tipos de datos, estructuras condicionales, bucles, definición de funciones, manejo de archivos e importación de módulos.

Al finalizar, no solo obtenemos el conocimiento de la sintaxis y lógica de Python, sino que también empezamos a aplicarla con un enfoque práctico y orientado a la seguridad informática.