



Lab 110: Trabajo con el tipo de dato de cadena

Información general sobre el laboratorio

En Python, un conjunto de letras y símbolos se denomina *cadena*. Las cadenas se suelen utilizar en Python para la entrada y la salida.

En este laboratorio, deberá realizar lo siguiente:

- escribir código Python que utilice el tipo de dato de *cadena*
- concatenar cadenas
- utilizar un string para solicitar datos de entrada
- dar formato a cadenas para la salida

Ejercicio 1: Presentar el tipo de dato de cadena

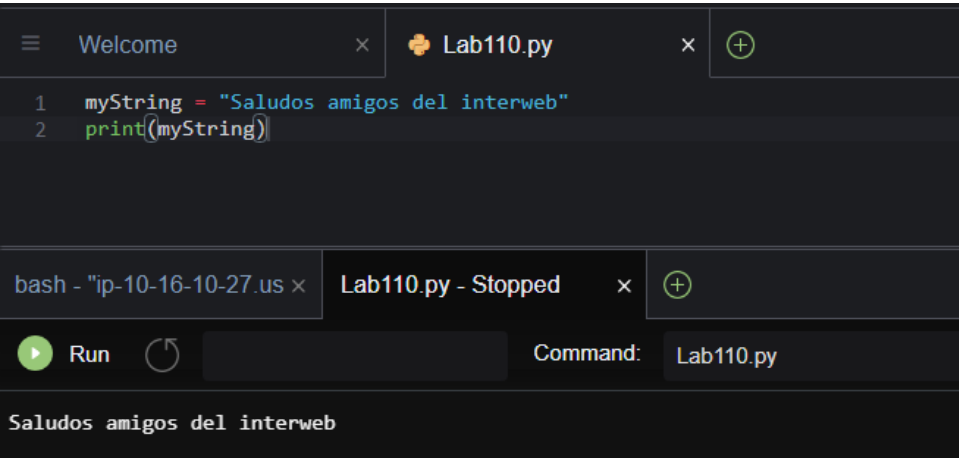
Se denomina script a un archivo de texto que contiene una secuencia lógica de comandos.

1. En el panel de navegación del IDE, elija el archivo **.py** que creó en la sección *Creación del archivo de ejercicios de Python* anterior.
2. En el archivo, escriba el siguiente código:

```
myString = "This is a string."
print(myString)
```

3. Guarde el archivo.
4. Ejecute el archivo.
5. Confirme que el script se ejecuta de forma correcta y que la salida se muestra según lo previsto.

```
This is a string.
```



6. Amplíe el script de Python con la función integrada `type()` para obtener el tipo de dato de la variable. Escriba el siguiente código:

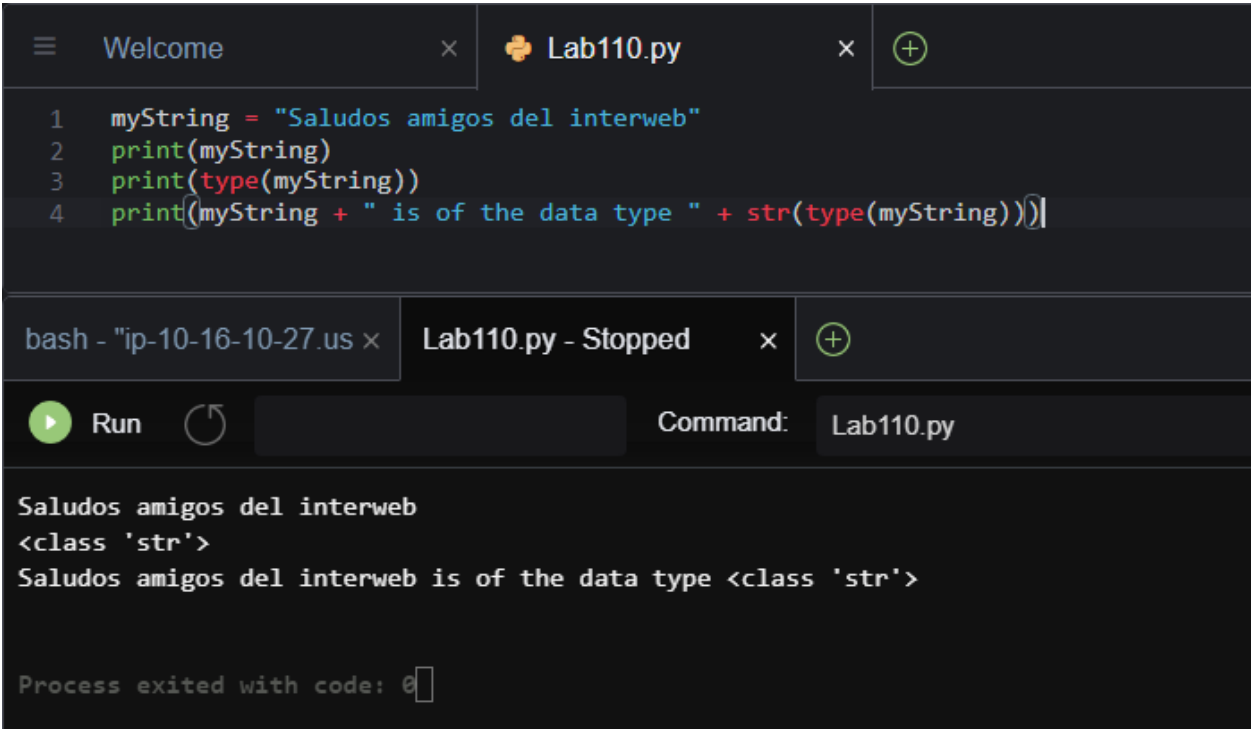
```
print(type(myString))
```


7. Para convertir el valor de retorno del tipo en una cadena, utilice la función integrada `str()`:

```
print(myString + " is of the data type " + str(type(myString)))
```

- Guarde el archivo.
- Ejecute el archivo.
- Confirme que el script se ejecuta de forma correcta y que la salida se muestra según lo previsto.




```
This is a string.
<class 'str'>
This is a string. is of the data type <class 'str'>
```





La segunda línea en la salida del código muestra `<class 'str'>`. Los símbolos `<` y `>` se utilizan para encerrar el nombre de la clase de objeto al que pertenece `myString`. En este caso, `myString` es una cadena de caracteres (`str`), por lo que `<class 'str'>` significa que `myString` es una instancia de la clase `str`.

Es una convención en Python que los nombres de las clases se escriban entre comillas simples dentro de los símbolos `<` y `>`. Esto ayuda a identificar el tipo de datos de un objeto de manera rápida y sencilla.



Ejercicio 2: Trabajar con concatenación de cadenas

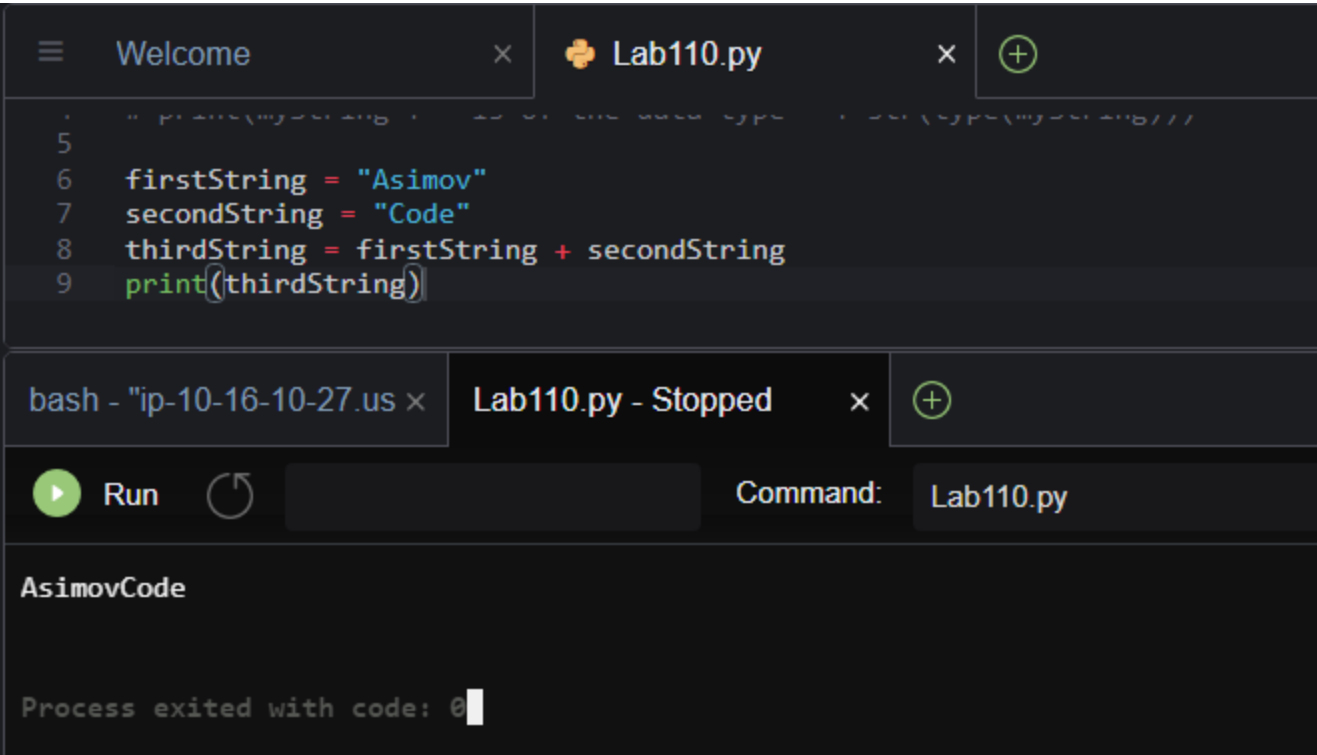
La concatenación de cadenas es el proceso por el cual se combinan dos cadenas para formar una sola. De hecho, ha estado realizando concatenaciones de cadenas desde el laboratorio 1. Sin embargo, simplemente no se refería a este proceso con ese término. El símbolo “+” se emplea para concatenar cadenas. Cuando se utiliza el símbolo “+” con cadenas, funciona de una forma distinta a cuando se utiliza con números. En el laboratorio 1, utilizó el signo más “+” para sumar números. Ahora, utilizará el signo más “+” para combinar o concatenar cadenas.

- Regrese al script en Python.
- Cree dos cadenas y, luego, concaténelas escribiendo el siguiente código:

```
firstString = "water"
secondString = "fall"
thirdString = firstString + secondString
print(thirdString)
```

- Guarde el archivo.
- Ejecute el archivo.
- Confirme que el script se ejecuta de forma correcta y que la salida se muestra según lo previsto.

```
firstString = "Asimov"
secondString = "Code"
thirdString = firstString + secondString
print(thirdString)
```



```
1 # print(myString + " is of the data type " + str(type(myString)))
5
6 firstString = "Asimov"
7 secondString = "Code"
8 thirdString = firstString + secondString
9 print(thirdString)
```

bash - "ip-10-16-10-27.us x Lab110.py - Stopped x (+)

Run Command: Lab110.py

AsimovCode

Process exited with code: 0

Ejercicio 3: Trabajar con cadenas de entrada

En la codificación, la información que escribe un usuario se conoce como *entrada*. Utilizará una función integrada denominada `input()` para obtener información del usuario. La función `input()` detendrá el código hasta que un usuario escriba una cadena y presione ENTER (Intro). Regrese al script en Python:

- 1. Escriba el siguiente código:

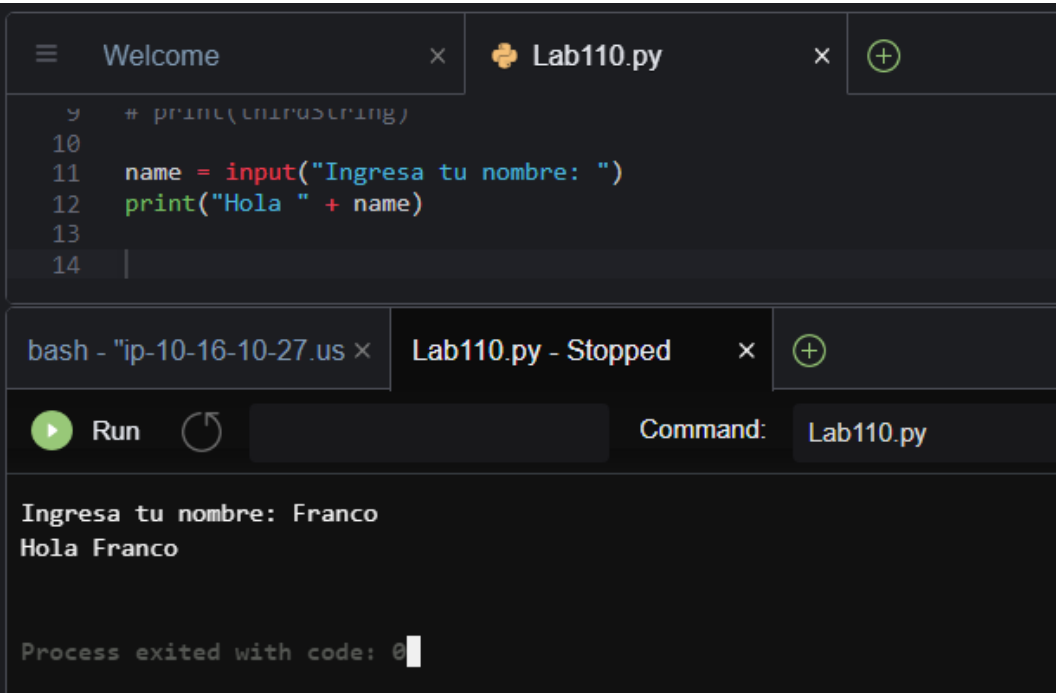
```
name = input("Wha is your name? ")
```

- 2. Utilice la función `print()` para escribir el valor de la variable en el shell:

```
print(name)
```

- 3. Guarde el archivo.
- 4. Ejecute el archivo.
- 5. Confirme que el script se ejecuta de forma correcta y que la salida se muestra según lo previsto.

```
name = input("Ingresa tu nombre: ")
print("Hola " + name)
```



```
9 # print(thirdString)
10
11 name = input("Ingresa tu nombre: ")
12 print("Hola " + name)
13
14 |
```

bash - "ip-10-16-10-27.us x Lab110.py - Stopped x (+)

Run Command: Lab110.py

Ingresa tu nombre: Franco
Hola Franco

Process exited with code: 0

Ejercicio 4: Dar formato a las cadenas de salida

Cuando un script busca comunicar información al usuario, se denomina *salida*. Ha estado utilizando la función `print()` para escribir la salida en el shell. Creará una encuesta y enviará la información recopilada de vuelta al usuario.

1. Regrese al script en Python y escriba el siguiente código:

```
color = input("What is your favorite color? ")
animal = input("What is your favorite animal? ")
```

2. Ha estado utilizando la función `print()` con una sola variable, pero también se puede usar con múltiples variables para dar formato a una cadena. Escriba el siguiente código:

```
print("{} , you like a {} {}!".format(name,color,animal))
```

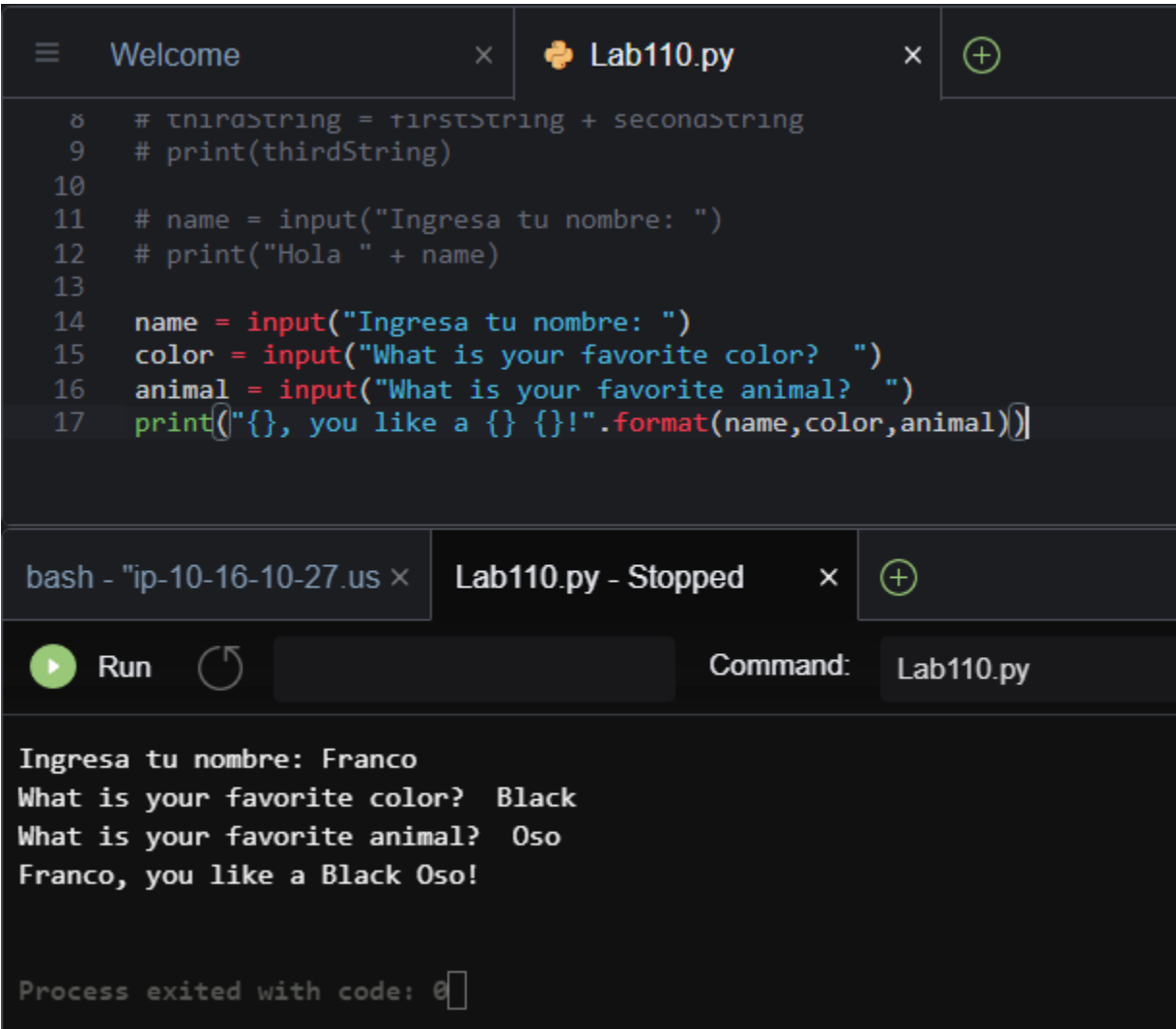
3. Guarde el archivo.
4. Ejecute el archivo.

El shell de Python está en pausa a la espera de su entrada.

5. Escriba un nombre y presione ENTER (Intro).
6. A continuación, se le pedirá que indique cuál es su color favorito. Escriba un color y presione ENTER (Intro).
7. A continuación, se le pedirá que indique cuál es su animal favorito. Escriba un animal y presione ENTER (Intro).
8. Por último, el script devuelve una cadena con formato al usuario utilizando los tres datos proporcionados. Confirme que la salida en el shell sea similar a la siguiente salida.

```
name = input("Ingresa tu nombre: ")
color = input("What is your favorite color? ")
animal = input("What is your favorite animal? ")
print("{} , you like a {} {}!".format(name,color,animal))

#Salida
Franco, you like a Black Oso!
```



Nota: La instrucción final `print()` utiliza la función `format()`. En la función `format()`, las llaves de apertura y cierre “{}” actúan como marcadores de posición para las variables que se transmitirán, es decir, se ubicarán entre los paréntesis de la función.

