

Python Programming

Módulo 3

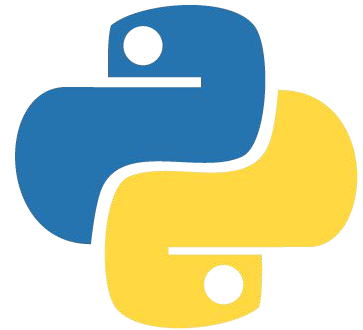
Argumentos de programa

Argumentos de programa

Cuando invocamos un programa a través de la terminal (o línea de comandos o símbolo del sistema) podemos pasarle argumentos como si se tratase de una función, que por lo general determinan alguna acción o configuración inicial del *script*. Dichos argumentos se especifican luego del nombre del programa y separados por espacios.

Supongamos que tenemos un archivo llamado `saludar.py` y que lo ejecutamos desde la terminal del siguiente modo:

```
> python saludar.py Lautaro
```



Ya vimos que, Python es un lenguaje interpretado, los *scripts* que escribimos son ejecutados por un programa al cual llamamos intérprete. En lo que acabamos de escribir, estamos haciendo referencia al intérprete con la palabra `python`. Podemos comprobar esto escribiendo lo siguiente (en Windows):

```
> where python  
C:\python37\python.exe
```

O en alguna distribución de Linux:

```
$which python  
/usr/bin/python
```

Entonces, si `python` es el programa ejecutable, los argumentos que recibirá en el caso anterior son los siguientes:

- `saludar.py`
- `Lautaro`

Para acceder a estos argumentos desde nuestro código, la librería estándar nos provee la lista `sys.argv` en donde cada uno de los argumentos será una cadena según el orden en que fueron especificados.

De modo que si nuestro archivo `saludar.py` contiene el siguiente código...

```
import sys  
print("Argumentos:", sys.argv)
```

Al ejecutarlo desde la consola con `python saludar.py Lautaro` veremos impreso en pantalla:

```
Argumentos: ['saludar.py', 'Lautaro']
```

La lista `sys.argv` siempre contendrá, al menos, un elemento. El nombre de nuestro archivo de Python (en este caso, `saludar.py`).

Todos los elementos subsiguientes habrán sido pasados a través de la terminal tal como acabamos de hacerlo. Entonces, si queremos que nuestro programa muestre un saludo al nombre que recibió como argumento a través de la consola, haremos:

```
import sys  
nombre = sys.argv[1]  
print("¡Hola" + nombre + "!")
```

Puede ocurrir que el usuario se haya olvidado de proveer su nombre. En dicho caso, por tratarse de una lista convencional, intentaremos capturar la excepción `IndexError`:

```
import sys

try:
    nombre = sys.argv[1]
    print("¡Hola" + nombre + "!")
except IndexError:
    print("No me has dicho tu nombre.")
```

Ahora bien, ¿qué tal si queremos que el programa imprima un saludo para cada uno de los nombres que se hayan pasado como argumentos —es decir, separados por espacios—? Bien podríamos hacer:

```
import sys

# Usamos la propiedad de «slicing» para
# ignorar el primer
# argumento, i. e. el nombre del programa.
nombres = sys.argv[1:]
# Chequeamos que la lista no esté vacía.
if nombres:
    for nombre in nombres:
        print(f"¡Hola, {nombre}!")
else:
    print("No has indicado ningún nombre.")
```

¿Cómo deberíamos invocar este programa? Así:

```
> python saludar.py Lautaro José Sofía
```

Y el resultado en pantalla es:

```
¡Hola, Lautaro!  
¡Hola, José!  
¡Hola, Sofía!
```



¡Muchas gracias!

¡Sigamos trabajando!