educación IT
CODE YOUR FUTURE

Introducción a Python

Módulo 1

Expresiones

Expresiones

Avancemos un poco con el diseño del código. Todos los lenguajes de programación tienen la capacidad de ejecutar operaciones matemáticas simples (suma, resta, división, multiplicación), es decir, incorporan una suerte de calculadora virtual que podemos usar en nuestros programas para hacer este tipo de operaciones.

La sintaxis (por *sintaxis* entendemos el orden y la relación de los términos de un lenguaje necesarios para que una instrucción sea comprendida) para realizar una suma en Python es muy similar a la que usamos en matemática para expresar dicha operación.

Esto que acabamos de escribir es un código de Python perfectamente válido al igual que lo era **`print("Hola mundo")`**.

De hecho, modifiquemos nuestro programa **`hola.py`** para que ahora contenga el siguiente código (recuerda que debemos mantener siempre las primeras dos líneas que mencionamos en el apartado “Nuestro primer programa”):

```
7 + 5
print("Hola mundo")
```

Al ejecutarse, este código realiza dos operaciones en el orden en que las dispusimos: primero ejecuta la suma **`7 + 5`** y luego imprime en pantalla el mensaje **`“Hola mundo”`**.

¡No esperes ver el resultado de la suma en pantalla! No, Python no hace nada que no le hayamos indicado; y, en efecto, simplemente le hemos dicho “resuelve 7 + 5”, pero no hicimos mención sobre qué debe hacer con ese resultado.

Vayamos un poco más allá, entonces, y digámosle que lo imprima en la pantalla:

```
print(7 + 5)
print("Hola mundo")
```

Ahora sí, al ejecutar ese código veremos:

```
12
Hola mundo
```

Todo lo que pongamos dentro de paréntesis en una instrucción print será enviado a la consola.

Ahora bien, ¿por qué utilizamos comillas en el segundo caso y no en el primero? Por el momento basta con saber que **las comillas permiten que un conjunto de caracteres sea interpretado de forma literal y no como código de Python.**

Teniendo esto en cuenta, el siguiente código:

```
print("7 + 5")
print("Hola mundo")
```

imprime:

```
7 + 5
Hola mundo
```

En este caso, **"7 + 5"** es interpretado como un texto en lugar de una operación que Python deba ejecutar, puesto que está encerrado entre comillas. Precisaremos un poco más estas cuestiones en la próxima clase.

Volviendo al primer ejemplo, del código **7 + 5** decimos que es una expresión. Una **expresión** es un fragmento de código que retorna un resultado. Por “retornar” no entendemos más que lo siguiente: el lenguaje reemplazará ese fragmento de código por su resultado.

Así, en todas las partes del código en donde aparezca **7 + 5**, por cuanto se trata de una expresión, el lenguaje lo reemplazará por **12**.

> El término “reemplazar” es usado aquí con fines didácticos para comprender el concepto de expresión. En realidad, el proceso de evaluación de las expresiones que aparecen en un código es más complejo.

El código:

```
print(7 + 5)
```

es similar a:

```
print(12)
```

educación IT
CODE YOUR FUTURE

¡Sigamos trabajando!