Aprendiendo Python

Escritura de texto:

En un script de Python, el elemento más básico que se puede escribir es texto. Simplemente un conjunto de palabras que quieren expresar algo, sin ninguna característica más.

Para ello, se deben de seguir estos pasos:

* Escribir la palabra “print”
* Abrir un paréntesis
* Introducir unas comillas
* Escribir el texto
* Cerrar las comillas
* Cerrar el paréntesis

Después el resultado será algo así:

print(“Hello, World!”)

Cuando el usuario corra el script, Python devolverá “Hello, World!

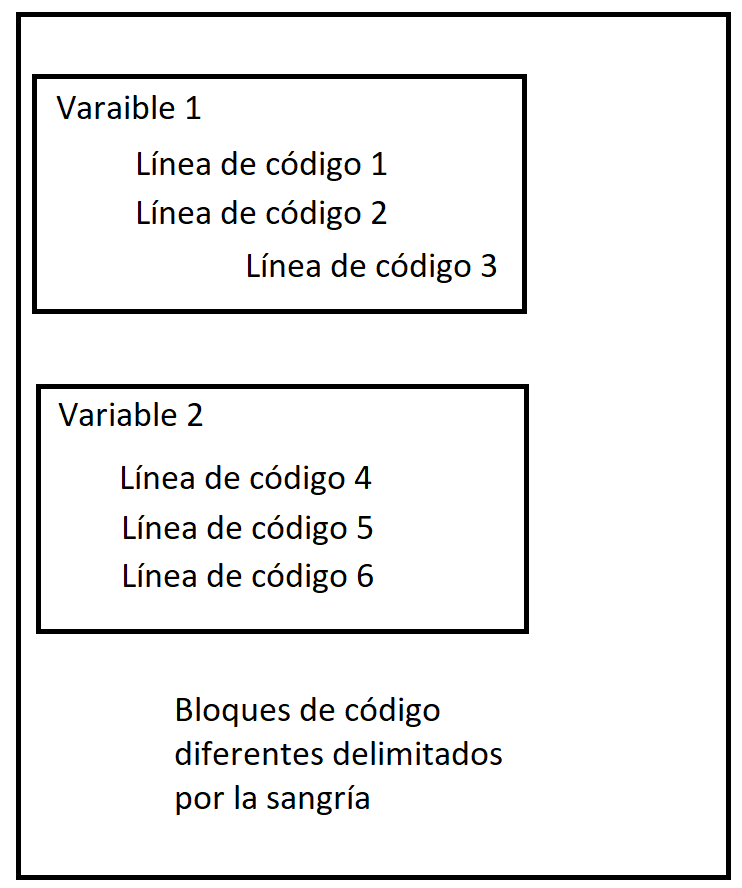
Sangría del texto:

Sin embargo, en este script se añadirán más líneas, más texto y otros elementos, y la manera de separarlos individualmente del resto es mediante el uso de espacios y tabulaciones.

Si nos fijamos en otros lenguajes como HTML, los espacios y tabulaciones se tratan de una manera de ordenar el código y facilitar la lectura, pero nada más. En este lenguaje de marcas, estos espacios tienen una función completamente estética y un usuario es capaz de crear una página web sin haber hecho uso de un solo espacio o tabulación.

Volviendo a Python, aquí es todo lo contrario. Los espacios y tabulaciones son recursos imprescindibles para que el código funcione, pues no solo se trata de la estética, también abarca la correcta escritura y sintaxis del código.

De esta manera, gracias a la sangría se pueden separar bloques de código distintos que expresan cosas distintas:



En un ejemplo, sería algo así:

if 5 > 2:

print(“5 es mayor que 2)

Sin la sangría, este código no hubiese funcionado. La cantidad de espacios y tabulaciones la elige el desarrollador, y si se necesitan poner más elementos como varios “prints”, se hace de la siguiente manera:

if 5 > 2:

print(“5 es mayor que 2)

print(“5 es mucho mayor que 2)

Comentarios

En Python, los comentarios son líneas de código que ayudan a comprender mejor el código, se podrían definir como anotaciones que hace el creador para que alguien que lo lea (o para él mismo) entienda mejor el código. Los comentarios suelen aparecer de color verde.

Para hacer un comentario, se escribe una almohadilla: #

#Esto es un comentario.

También se puede escribir en distintas líneas, aunque para ello habría que escribirlo de la siguiente manera:

#Esto

#es

#un comentario

Otra manera es poniendo dobles comillas, aunque se usa más la primera manera:

“””

Esto es un comentario

“””

Variables

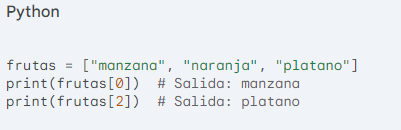
Una variable es un contenedor de datos que cambian, es decir, que no siempre son los mismos. Pueden guardar muchos tipos de datos como:

Números enteros: Pueden ser positivos y negativos, y se separan con comas cada uno de otro.

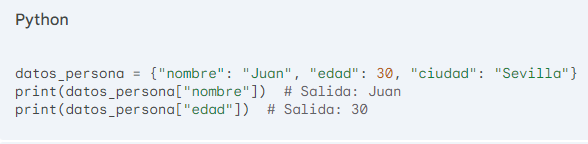
Decimales: También se separan con comas y para escribir una coma, se pone un punto.

Cadenas de texto: Sirven para representar texto. El texto debe de ser escrito entre comillas.

Listas: Listan elementos de forma ordenada. Para abrir una cadena se escribe “[“ y para cerrarla, “]”.



Diccionarios: Lo mismo que el anterior, pero el orden es menos importante y la manera de acceder a los datos es mediante palabras clave.



Como se puede comprobar, las variables guardan distintos tipos de datos, por esta razón es por la que existen los siguientes elementos:

X = str(hola): Este “str” significa que el tipo de dato que guarda la variable es un texto y Python debe de interpretarlo y procesarlo como texto. El script devolverá “hola”.

X = int(3): Lo mismo que el anterior pero indica que el valor es un número. El script devolverá “3”.

X = float(3): Lo mismo que int, pero para expresar números decimales. El script devolverá “3.0”.

Si se desea consultar el tipo de dato que contiene una variable, se usa el siguiente método:

X = 5

Y = “Python”

Print(type(x))

Print(type(y))

El código devolverá datos como:

<class ‘int’> o <class ‘str’>

Cuando se define una variable, existen varios factores a tener en cuenta como:

Si se escribe texto, este va entre comillas (ya sean dobles o unas comillas normales, no importa). Ejemplo:

X = “Python”

Y = ‘HTML’

Las dos son válidas.

En el caso de poner texto y no escribir comillas, Python da el error “NameError: name ‘palabra’ is not defined”.

Además, hay que tener en cuenta que las minúsculas y mayúsculas importan:

X = 4

x = 2

Python las detecta como variables distintas.

Por cosas como estas, antes de crear un nombre para una variable es recomendable ver estos puntos:

* El nombre de una variable debe comenzar con una letra o un carácter de subrayado.
* El nombre de una variable no puede comenzar con un número
* El nombre de una variable solo puede contener caracteres alfanuméricos y guiones bajos (Az, 0-9 y \_).
* Los nombres de las variables distinguen entre mayúsculas y minúsculas (edad, Edad y EDAD son tres variables diferentes)
* Un nombre de variable no puede ser ninguna de las [palabras clave de Python](https://www.w3schools.com/python/python_ref_keywords.asp) .

En el caso de poner un número, da igual si se ponen comillas o no, no afecta en nada. Ejemplo:

X = “4”

Y = 6

Ambas son válidas.

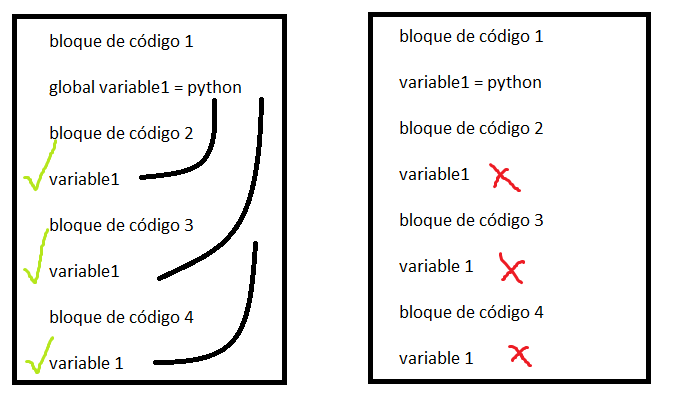
Para utilizar las variables es necesario activarlas, sin embargo, se activan automáticamente cuando se les asigna un valor. Este es un ejemplo de activación de una variable cuando se le acaba de vincular un dato a ella:

X = “Python”

Print(X)

Cuando esta variable se ejecute, el código devuelve la palabra “Python”. Sin embargo, no hay que olvidar que existen 2 tipos de variables: globales y locales.

La diferencia ente estos tipos de variables es que las globales son accesibles desde cualquier parte del código y las locales son solo accesibles desde ese mismo bloque de código, un ejemplo en la siguiente imagen:



Como se puede apreciar, el primer recuadro contiene una variable global, y todos los bloques de código distintos pueden ver y usar esa variable, por muy lejos que esté. En el segundo caso, la variable no puede ser vista ni usada en los demás bloques de código, y solo puede ser usada en el mismo bloque de código donde fue creada.

Además, La mutabilidad de una variable determina si su valor puede ser modificado después de su creación. Las variables en Python son mutables, lo que significa que sus valores pueden ser cambiados a lo largo del programa.

Cuando se crea una variable, esta es una variable local, pero se puede cambiar simplemente añadiéndole “global”:

global variable1 = “X”

print(variable1)

El código devolverá la letra X.

Funciones