

Francisco Javier Vargas Ogando

2021-0160

Python es un lenguaje interpretado de tipado dinámico. Los programas (.py) pueden escribirse en un editor ligero como Geany, que es capaz de ejecutar el código y compilarlo (.pyc). Además, tiene autocompletado, soporte multidocumento, gestión de proyectos, aplica colores de sintaxis y cuenta con un emulador de Terminal integrado. Aunque hay muchas distribuciones de Linux, como Ubuntu, que instalan Python de manera predeterminada, podemos encontrarnos que la versión instalada del intérprete no es 3.x. Hay que tener en cuenta que existen dos líneas de desarrollo de intérpretes Python (2.x y 3.x.) y que es posible tener instaladas versiones de ambas líneas. Dependiendo de la situación que tengamos el intérprete 3.x se podrá invocar desde la línea de comandos con: **python** y/o **python3**. Normalmente, la primera opción no se podrá utilizar si tenemos instalado algún intérprete 2.x porque se reserva para éste.

En GNU/Linux, para comprobar si tenemos instalado **Python3** ejecutar desde la línea de comandos.

La longitud máxima de línea recomendada es de **79 caracteres**.

El punto y coma “;” se puede usar para separar varias sentencias en una misma línea, pero no se aconseja su uso:

```
edad = 15; print(edad)
```

El código Python se escribe en cada línea desde la primera posición excepto cuando es necesario dejar el código sangrado. El **sangrado** en Python es obligatorio, se hace con **espacios en blanco** o **saltos de tabulación** y sirve para agrupar bloques de líneas de código que tienen relación a distintos niveles. Se utiliza con estructuras de control (if-else, while, for), con funciones y clases. Además, permite que la lectura del código sea cómoda y agradable.

El sangrado se puede hacer con espacios y tabulaciones pero ambos tipos no se pueden mezclar.

Es posible sangrar con un único espacio en blanco pero **lo normal es utilizar un número de espacios múltiplo de cuatro** en cada nivel de sangrado (cuatro, ocho, doce espacios, etc.), o bien, distinto número de saltos de tabulación.

Por defecto, **Geany** inserta tabuladores aunque se puede cambiar este modo de sangrar en la opción “**Preferencias**” del menú “**Editar**”, apartado “**Sección**”, pestaña “**Sangría**”.

¿Cómo funciona el sangrado en Geany? Cuando se presiona **[Enter]** si la línea actual finaliza con dos puntos “:” la siguiente y sucesivas comenzarán sangradas. Para desangrar presionar la tecla de borrado [←].

La barra invertida “\” al final de línea se emplea para dividir una línea muy larga en dos o más líneas.

Las expresiones entre paréntesis “()”, llaves “{}” y corchetes “[]” separadas por comas “,” se pueden escribir ocupando varias líneas.

No se pueden declarar variables, objetos, funciones y clases con estos términos. El siguiente código muestra la lista de palabras reservadas:

```
importar
impresión de palabra clave (keyword.kwlist)
```

El signo igual “=” se utiliza para asignar números, booleanos, cadenas y expresiones a las variables de un programa. El tipo de la variable será el tipo del dato asignado: al declarar una variable no es necesario especificar el tipo de dato porque mientras se asigna el área de memoria necesaria el intérprete Python elige automáticamente el tipo más apropiado.

El nombre de una variable tiene que empezar por una letra del alfabeto o un guión bajo. En las siguientes posiciones pueden aparecer también números. Python distingue entre mayúsculas y minúsculas

Los operadores aritméticos en Python son:

+	Suma
-	Resta
*	Multiplicación
/	División
**	Potenciación
//	División, se obtiene parte entera del cociente
%	División, se obtiene el resto

Operadores binarios

Los operadores binarios emplean en sus operaciones la representación binaria de los datos. Los operadores binarios son:

&	AND
	OR
^	XOR
~	NOT
<<	Desplazamiento a la izquierda
>>	Desplazamiento a la derecha

Operadores de comparación o relacionales

Los operadores de comparación en Python son:

<	Menor que
>	Mayor que
<=	Menor o igual que
>=	Mayor o igual que
==	Igual a
!=	Distinto de

Operadores lógicos

Los operadores lógicos permiten encadenar varias condiciones.

and	Y
or	O
not	NO

Visual Studio Code es uno de los editores de código gratuitos más conocidos en todo el mundo. Su sencillez de uso y su interfaz gráfica son algunos de sus muchos puntos fuertes. Con su instalación se incorporan los lenguajes de JavaScript (junto con TypeScript y Node.js), HTML y CSS, aunque con las extensiones es posible utilizar lenguaje Java, C++, PHP o Python.

A continuación, se explican los 5 pasos para instalar y empezar a usar Visual Studio Code:

1. Descargar el instalador del programa. Podemos elegir los sistemas operativos de Windows, Linux y Mac.
2. Ejecutar el fichero que se nos ha descargado. La instalación es muy sencilla. Simplemente debemos ir aceptando y avanzando en el proceso de instalación.
3. Abrir Visual Studio Code. Nos aparecerá una ventana como la siguiente.
4. Crear un nuevo fichero (el cual podemos guardar con extensión .js, .html, .py, etc), abrir uno existente o, por ejemplo, abrir una carpeta donde tengamos nuestros ficheros.
5. Arrastrar desde el explorador de la parte izquierda los ficheros que queramos editar hacia el panel de edición. Podremos navegar fácilmente entre ellos utilizando la barra superior. En la parte derecha nos aparece un “minimapa” para localizarnos dentro del código que estamos editando.

Con estos 5 sencillos pasos ya estamos listos para empezar a desarrollar nuestras aplicaciones utilizando Visual Studio Code.