

Primeros pasos con Python



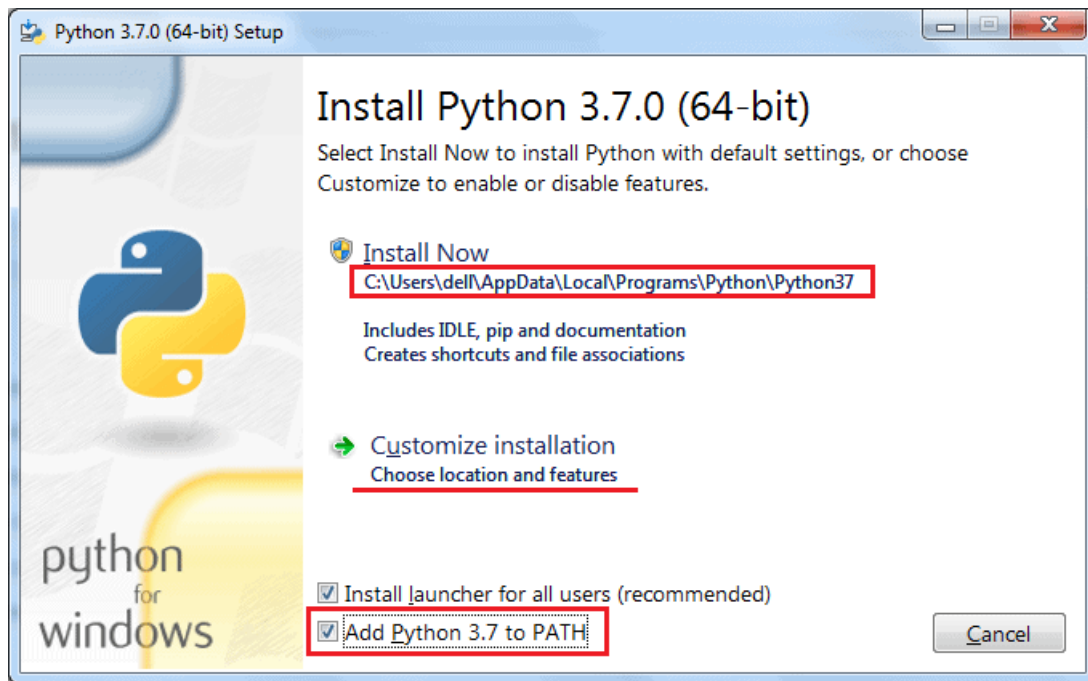
Python es un lenguaje de programación creado a principios de los años 90. Su nombre es un tributo al grupo de comedia *Monty Python*, y entre sus muchas virtudes, podemos destacar las siguientes:

- Se utiliza a menudo en etapas tempranas de aprendizaje de programación, porque es bastante sencillo de entender.
- Es un lenguaje multiplataforma, con el que podemos desarrollar aplicaciones de todo tipo (escritorio, web, etc) en diferentes sistemas (Windows, Mac, Linux).
- Es un lenguaje interpretado (no compilado), y puede utilizarse como un lenguaje de *script* en terminal, como ocurre con Perl, PowerShell u otros lenguajes de script.
- Utiliza tipado dinámico de datos, es decir, no existen tipos de datos implícitos, ni tenemos que indicarlos al utilizar las variables en el programa. A medida que asignamos valores a las variables, éstas toman el tipo de dato adecuado.
- Podemos utilizar Python tanto desde una perspectiva orientada a objetos (usando clases y objetos) como sin dicha perspectiva (sin necesidad de clases).

1. Instalación de Python

Para instalar Python en nuestro sistema, basta con ir a la [web oficial](#) y descargar la versión apropiada para nuestro sistema. El intérprete de Python es normalmente un comando llamado `python` o `python3`, dependiendo del sistema operativo en el que estemos. Por ejemplo, en Windows se utiliza el comando `python`, pero en Linux se emplea `python3`.

Cuando instalemos Python en Windows, debemos marcar en el asistente de instalación la casilla para añadir Python directamente al PATH del sistema. De lo contrario, deberemos editar manualmente nosotros la variable de entorno para añadir la carpeta de instalación de Python.



En lo que respecta a Linux, primero podemos comprobar si ya tenemos Python instalado en nuestra distribución, con este comando:

```
python3 --version
```

De lo contrario, podemos actualizar a Python 3 fácilmente con este comando:

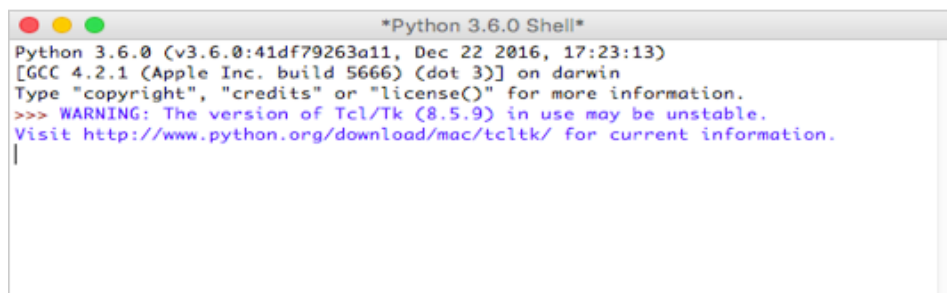
```
sudo apt-get install python3
```

2. Elegir un IDE apropiado

Una vez tengamos Python instalado en nuestro sistema, debemos elegir un IDE apropiado para desarrollar nuestras aplicaciones. Podemos encontrar diferentes alternativas en Internet, como por ejemplo IDLE (el entorno que viene integrado en la instalación de Python), Eclipse, Visual Studio Code... En este documento vamos a dar unas breves nociones sobre algunos de los que consideramos más interesantes.

2.1. Utilizando IDLE

IDLE es un entorno que se descarga en la propia instalación de Python. Podemos ejecutarlo escribiendo `idle` o `idle3` en el terminal (dependiendo de nuestro sistema operativo y versión de Python). También podemos elegir el correspondiente acceso directo en el menú de Windows, o en la sección de *Aplicaciones* de Mac. Finalmente, veremos una ventana como esta:



```
*Python 3.6.0 Shell*
Python 3.6.0 (v3.6.0:41df79263a11, Dec 22 2016, 17:23:13)
[GCC 4.2.1 (Apple Inc. build 5666) (dot 3)] on darwin
Type "copyright", "credits" or "license()" for more information.
>>> WARNING: The version of Tcl/Tk (8.5.9) in use may be unstable.
Visit http://www.python.org/download/mac/tcltk/ for current information.
>>>
```

Lo que vemos es un terminal de comandos Python. Podemos escribir comandos sueltos y se ejecutan uno a uno, pero esto no es lo habitual, sino que necesitaremos editar un archivo de código fuente. Para ello, desde el menú *File > New File* podemos crear nuevos archivos de código Python, y guardarlos con el nombre adecuado (los archivos deben tener extensión *.py*, como por ejemplo *prueba.py*). Después, editamos el código fuente y podemos ejecutarlo desde el menú *Run > Run Module* de la ventana de edición, o bien pulsando la tecla F5.

Sin embargo, este entorno se nos puede quedar algo corto o limitado si queremos gestionar varios archivos, y puede resultar algo engorroso estar cambiando entre el terminal y la ventana de edición.

2.3. Utilizando Geany

Si ya tenemos descargado e instalado Python, el intérprete `python` (o `python3`) estará ya correctamente configurado en nuestro sistema. En este caso, podemos emplear un IDE de propósito general como [Geany](#) para editar nuestros programas. Podemos instalarlo fácilmente desde su web oficial en Windows y en Mac. En el caso de Linux es mejor instalar el paquete *geany* junto con sus dependencias desde el gestor de dependencias del sistema operativo.

La mecánica es similar al caso anterior: con el menú *Archivo > Nuevo* creamos nuevos archivos fuente, que debemos guardar con el nombre adecuado (extensión *.py*) en la carpeta que elijamos. Después, editamos el código fuente y podemos ejecutarlos con el botón de *Ejecutar*. Geany se autoconfigura con la instalación de Python, y utiliza automáticamente el comando `python` o `python3` para lanzar los programas. Podemos comprobarlo abriendo el panel *Construir > Establecer comandos de construcción* desde el archivo Python que estemos editando.

2.3. Intérpretes online

Una opción alternativa para desarrollar programas en Python, especialmente cuando son relativamente simples y cortos, es utilizar algún editor online, como [éste](#). Simplemente debemos escribir el código en el panel correspondiente, y hacer clic en el botón para ejecutar la aplicación, viendo el resultado en la consola derecha.

2.4. Visual Studio Code

Como cuarta y última alternativa que propondremos en este documento, está utilizar Visual Studio Code. Ofrece unas funcionalidades similares a Geany, pero nos será de mayor utilidad cuando queramos gestionar un proyecto con varios archivos fuente. Podemos descargar VS Code de su [web oficial](#). En el caso de Windows

y Mac, simplemente ejecutamos el instalador. En el caso de Linux (Ubuntu), desde el terminal, ubicado en la carpeta donde hemos descargado el archivo, ejecutamos este comando:

```
sudo dpkg -i nombre_archivo.deb
```

donde *nombre_archivo.deb* será el archivo con extensión *.deb* que habremos descargado.

Para poder trabajar con Python desde Visual Studio Code, debemos abrir la carpeta donde vayamos a editar los archivos Python (menú *Archivo > Abrir carpeta*), y luego crear los archivos y carpetas que queramos desde el panel izquierdo del explorador. También es recomendable instalar la extensión *Code Runner* y configurarla para poder ejecutar los programas Python en el terminal integrado de Visual Studio Code.

Code-runner: Run In Terminal

☒ Whether to run code in Integrated Terminal.

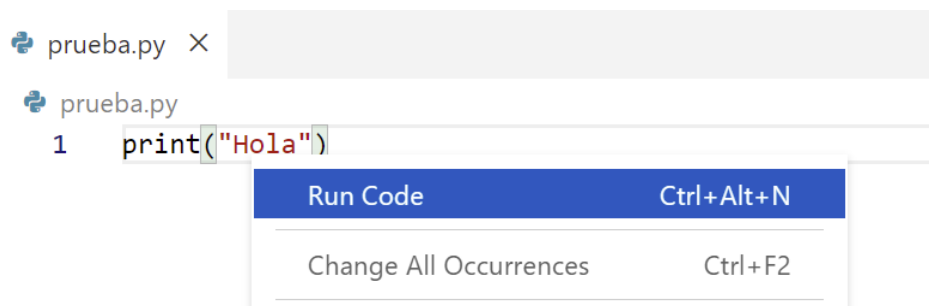
3. Nuestro primer programa Python

Vamos a crear nuestro primer programa Python que muestre un saludo por pantalla. Podemos emplear cualquiera de los IDEs comentados anteriormente. Creamos un archivo fuente Python llamado, por ejemplo, *prueba.py* en la carpeta donde queramos trabajar, y lo guardamos con este contenido:

```
print("Hola")
```

A la hora de ejecutarlo:

- Si usamos IDLE, pulsamos F5
- Si usamos Geany, pulsamos el botón de *Ejecutar* de la barra de herramientas
- Si usamos el editor online de *repllt*, pulsamos el botón de ejecutar o *Run*
- Si utilizamos Visual Studio Code y hemos instalado la extensión *Code Runner*, hacemos clic derecho sobre el código fuente y elegimos la opción *Run Code* que aparecerá en el menú contextual



El resultado en cualquier caso será el mismo, viendo por terminal el texto que hemos indicado en la instrucción `print`:

```
Hola
```