

Fundamentos de Programación

INTRODUCCIÓN A LOS FUNDAMENTOS DE PROGRAMACIÓN I

Semana 01

OBJETIVO DEL LABORATORIO

Explica los conceptos básicos de la programación y como se representa la información.

MARCO TEÓRICO

Python es un lenguaje de programación de alto nivel, multipropósito, multiplataforma, interpretado, de código abierto, con soporte de primer nivel para programación funcional y orientada a objetos, y muchas otras felices características.

Y prueba de ello es la popularidad de la que se ha valido en los últimos años. Es utilizado por las empresas más grandes del rubro de la tecnología, en áreas tales como desarrollo de aplicaciones web, aplicaciones de escritorio, scripting, networking, inteligencia artificial y automatización.

Su sintaxis simple y clara lo hace ideal para aquellos que se acercan por vez primera al mundo de la programación. Además, como dijimos, es de código abierto, libre y gratuito. Mantenido por una comunidad de desarrolladores de todo el mundo. ¡Cualquiera que esté interesado en colaborar puede hacerlo!

No quiero extenderme demasiado ni es mi intención hacer aquí una apología del lenguaje; probablemente, si has llegado a este lugar, ya estarás lo suficientemente interesado como para pasar directo a la práctica. Pero antes me interesaría comentarte algunas cuestiones acerca de este tutorial.

RECURSOS

a. Hardware

- Pc Pentium IV a superior
- Conexión de red

b. Software

- Sistema Operativo Windows XP a superior
- Navegador Chrome o Firefox
- Edube Sandbox de Python desde Netacad

PROCEDIMIENTO

1. INSTALACIÓN DE PYTHON

Python permite tener instaladas distintas versiones de Python (Python 2.7 y Python 3.6, por ejemplo, o Python 3.5 y Python 3.6).

Se recomienda tener instalada únicamente una versión de Python, la más reciente. Por ello, si va a instalar una versión de Python 3.6:

- si ya tiene instalada una versión anterior (Python 3.5 o anteriores), se recomienda desinstalar primero las versiones anteriores.
- si ya tiene instalada una versión de Python 3.6, no es necesario desinstalar nada.

2. CONSEGUIR PYTHON

La página oficial de Python es <https://www.python.org/>.



La última versión de Python 3.6 disponible actualmente (junio de 2018) es Python 3.6.6 (del 27 de junio de 2018).

Enlaces: página de [descarga de Python 3.6.6](#), instaladores de Python 3.6.6 para [Windows de 64 bits \(30,4 MB\)](#) y [Windows de 32 bits \(29,4 MB\)](#).

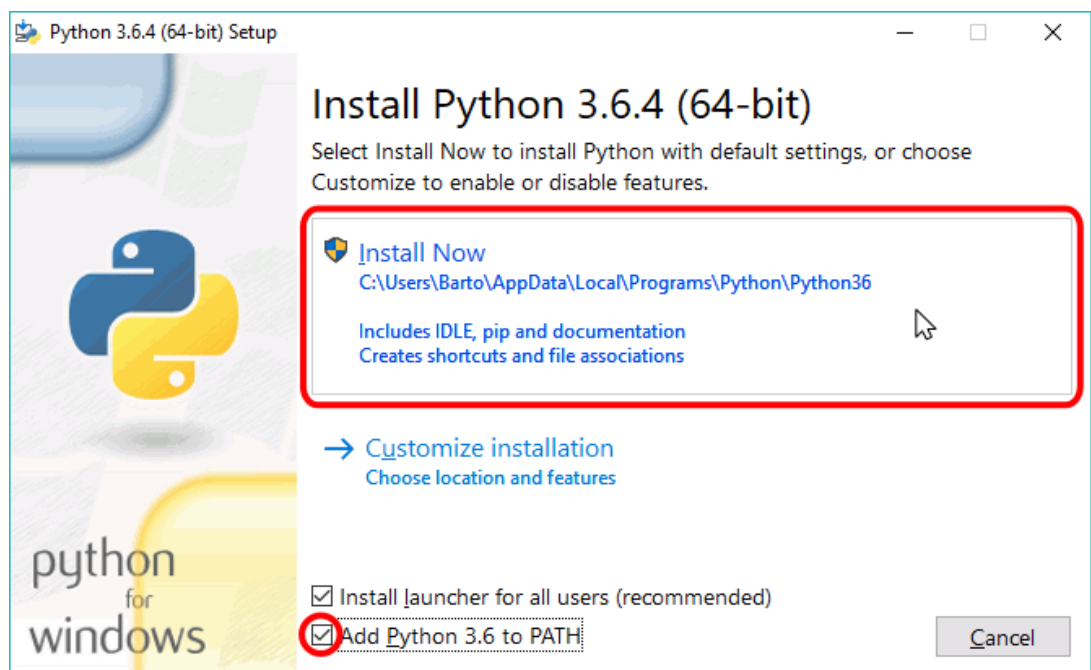
En [cdlibre.org](#) hay una sección dedicada al [intérprete de Python](#), con información detallada sobre las últimas versiones publicadas para Windows.

3. INSTALAR PYTHON

Una vez descargado el instalador, haga doble clic en él para iniciar la instalación.

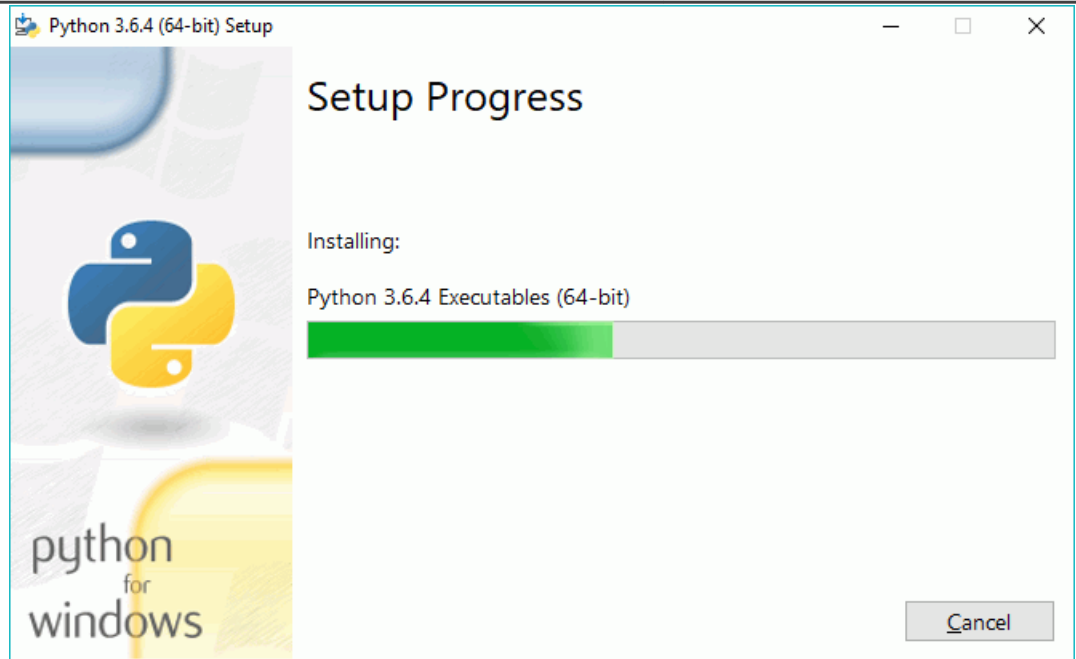
Nota: Las imágenes siguientes corresponden a la instalación de Python 3.6.4 (64 bits), pero son similares en cualquier versión 3.6.X.

- La primera pantalla permite seleccionar las opciones de instalación haciendo clic en "Customize installation", aunque en principio, no es necesario modificarlas. Tan sólo se aconseja marcar la casilla "Add Python to PATH" para poder ejecutar programas desde la línea de comandos. Para continuar, haga clic en "Install Now".

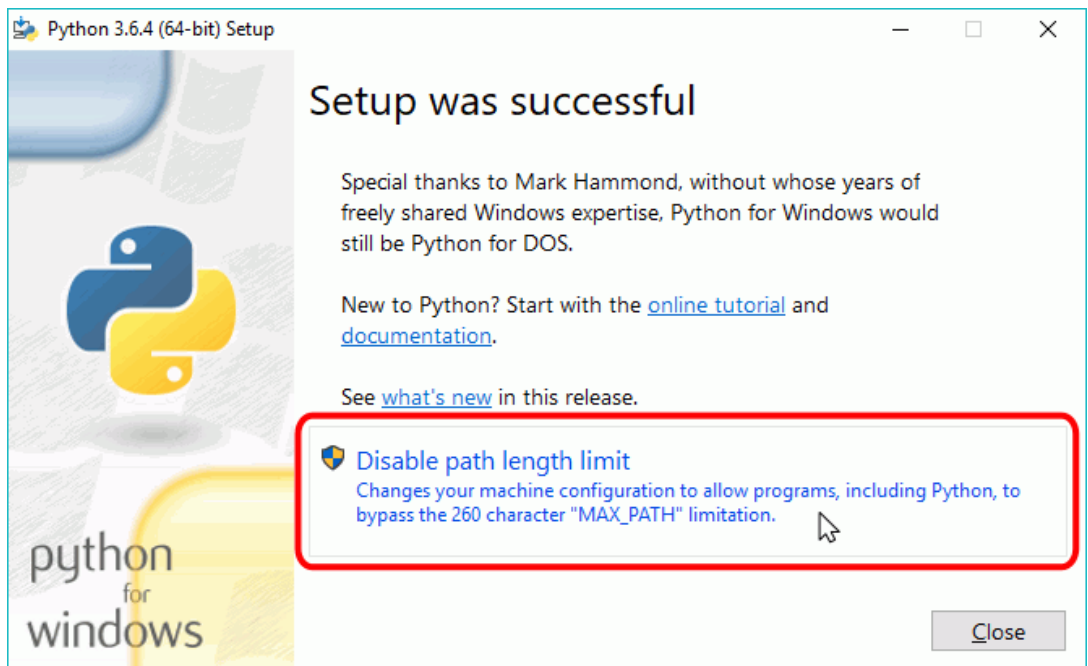


- A continuación, se iniciará el proceso de instalación. La instalación de Python puede durar varios minutos.



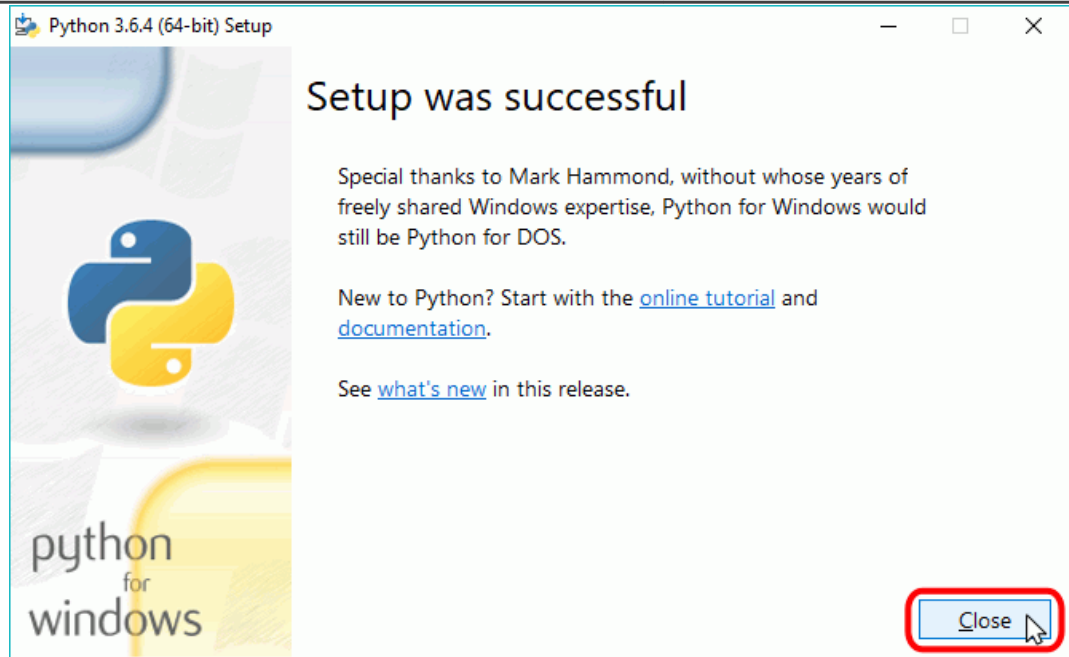


- Una vez completada la instalación, se mostrará la pantalla final. Haga clic en el botón "Disable path length limit".



- Finalmente, haga clic en el botón "Close" para cerrar el programa de instalación. ¡Ya puede empezar a programar en Python!



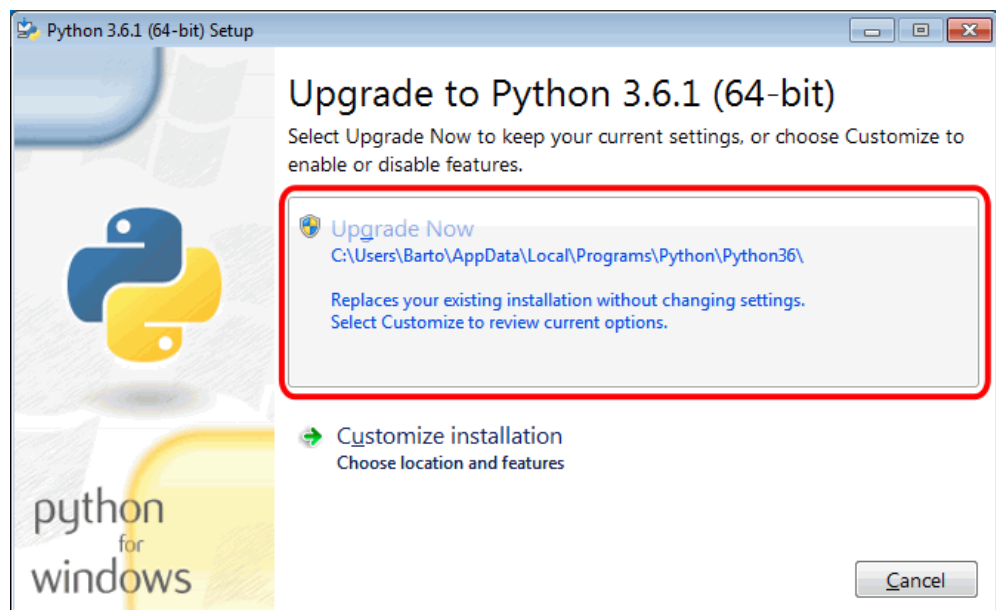


4. ACTUALIZAR PYTHON

Una vez descargado el instalador, haga doble clic en él para iniciar la actualización. Si el instalador detecta una subversión anterior de la misma versión de Python (por ejemplo, si el instalador de Python 3.6.1 detecta que ya está instalado Python 3.6.0), ofrecerá la opción de actualizar.

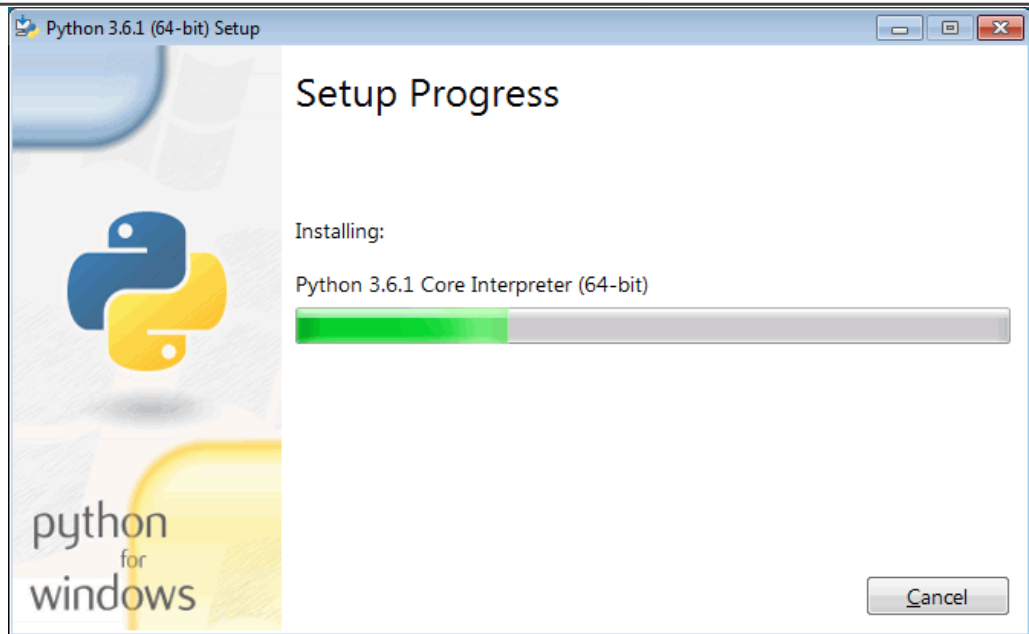
Nota: Las imágenes siguientes corresponden a la instalación de Python 3.6.1 (64 bits), pero son similares en cualquier versión 3.6.X.

- La primera pantalla permite seleccionar las opciones de instalación haciendo clic en "Customize installation", aunque en principio, no es necesario modificarlas. Para continuar, haga clic en "Upgrade Now".

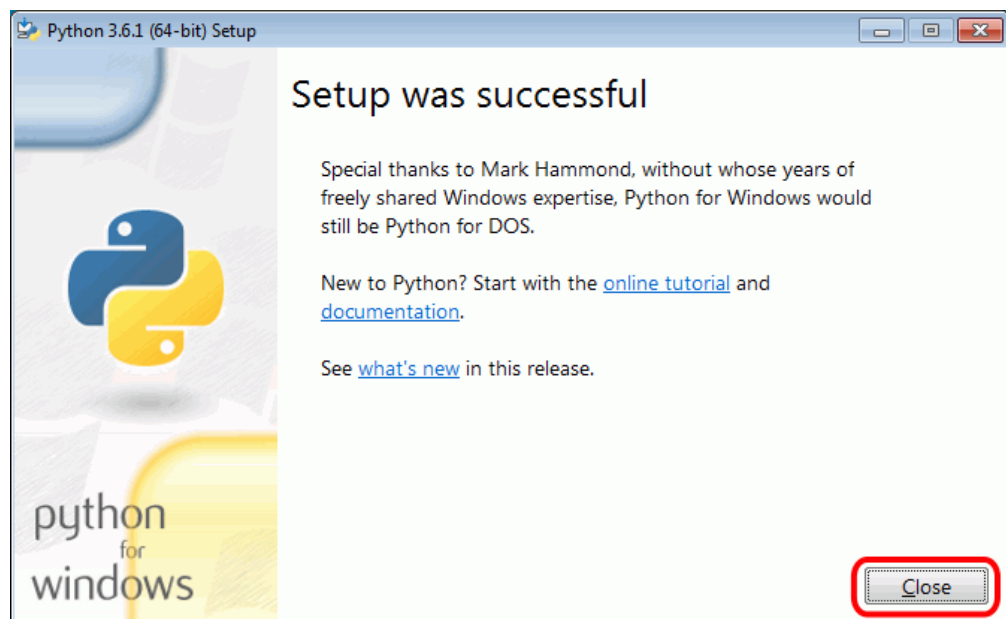


- A continuación, se iniciará el proceso de actualización. La actualización de Python puede durar varios minutos.





- Una vez completada la actualización, se mostrará la pantalla final. Haciendo clic en el botón "Close" se cerrará el programa de instalación.

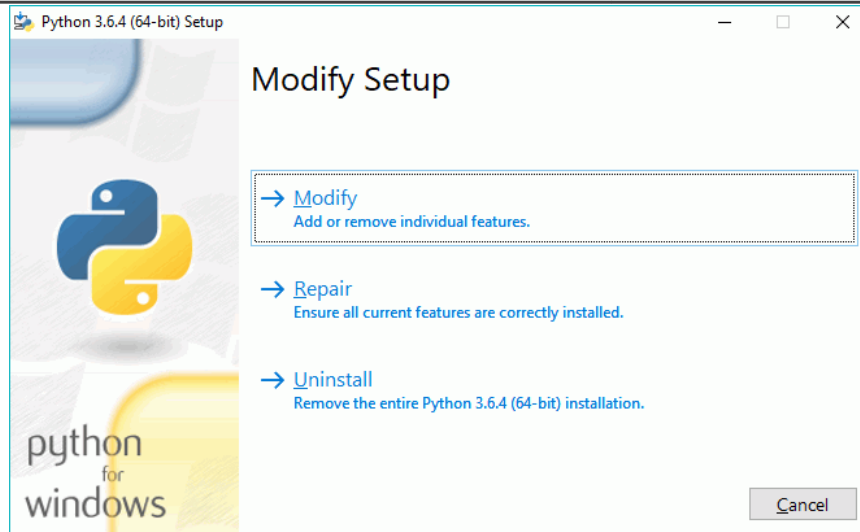


5. MODIFICAR O REPARAR PYTHON

Si el instalador detecta que ya está instalada la misma versión de Python (por ejemplo, si el instalador de Python 3.6.4 detecta que ya está instalado Python 3.6.4), ofrecerá la opción de modificar, reparar o desinstalar Python.

Nota: Las imágenes siguientes corresponden a la instalación de Python 3.6.4 (64 bits), pero son similares en cualquier versión 3.6.X.
Python





6. PYTHON E IDLE

Python es el nombre del lenguaje de programación.

IDLE (Integrated DeveLopment Environment for Python) es el entorno de desarrollo que permite editar y ejecutar los programas. Se pueden crear y ejecutar programas en Python sin utilizar IDLE, pero IDLE hace mucho más fáciles esas tareas (de hecho, IDLE en inglés significa holgazán).

Al abrir IDLE con el acceso directo **Inicio > Python 3.6 > IDLE (Python 3.6)**, se abrirá la ventana principal de IDLE, como muestra la imagen siguiente.



Esta ventana indica la versión de Python instalada.

Nota: La ventana de la imagen tiene un tamaño de 80x20 caracteres en vez del tamaño predeterminado de 80x40 caracteres

En la ventana de IDLE se pueden escribir órdenes de Python después del símbolo de petición >>> (en inglés, prompt). Al pulsar Intro, IDLE ejecutará la orden inmediatamente. Si la orden produce algún resultado, éste se mostrará en color **azul** y sin el símbolo de petición. Por ejemplo, se puede comprobar que Python sabe sumar dos más dos:

```
>>> 2 + 2
```



4

```
>>>
```

Al terminar de ejecutar la orden, IDLE vuelve a mostrar el símbolo de petición, en espera de una nueva orden.

Se pueden recuperar órdenes anteriores mediante los atajos de teclado:

- **Alt + p** para ver la instrucción anterior (supongo que p es por previous, en español, anterior)
- **Alt + n** para ver la instrucción siguiente (supongo que n es por next, en español, siguiente)

7. COLORES EN LOS PROGRAMAS

Al escribir órdenes en IDLE, algunas palabras cambian de color. Los colores ayudan a identificar los distintos tipos de elementos y a localizar errores:

- Las palabras reservadas de Python (las que forman parte del lenguaje) se muestran en color **naranja**.
- Las cadenas de texto se muestran en **verde**.
- Los resultados de las órdenes se escriben en **azul**.
- Los mensajes de error se muestran en **rojo**.
- Las funciones se muestran en **púrpura**.

Las cadenas de texto deben delimitarse con comillas dobles (") o con comillas simples ('):

```
>>> print('hola')
hola
>>> print("hola")
hola
>>>
```

Si la orden contiene algún error, Python escribe un mensaje de error en inglés que indica dónde se ha producido el error y por qué.

```
>>> print(hola)
Traceback (most recent call last):
  File "<pyshell#3>", line 1, in ?
    print hola
NameError: name 'hola' is not defined
>>>
```

En este ejemplo, el error se debe a que **hola** no está entre comillas y Python entiende que se hace referencia a una variable que se llama hola; al no estar definida previamente ninguna variable que se llame así, se produce un error.

Los argumentos de las funciones deben ir rodeados de paréntesis (en los ejemplos anteriores, la función es la función print() y el argumento de la función, es decir, lo que se le da a la función para que haga su trabajo, es la cadena de texto). Si no se escriben los paréntesis, Python indica que se ha cometido un error de sintaxis:

```
>>> print "hola"
SyntaxError: invalid syntax
>>>
```



8. ¡Hola, IDAT!

Si ya tienes tu editor de código favorito también puedes usarlo.

Para crear nuestro primer programa vamos a abrir IDLE y seleccionar el menú **File > New File** para crear un nuevo documento. Luego, escribiremos lo siguiente.

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-

print("¡Hola, IDAT!")
```

Detengámonos un momento en estas tres líneas. En Python todo lo que comience con un numeral (#) es considerado un comentario; por ende, las primeras dos líneas no constituyen un código de Python pero sí tienen injerencia sobre nuestro programa. Aún más, ¡todo programa de Python debe comenzar con ellas!

La primera se denomina **Shebang** e indica la ruta del intérprete. Si bien es propio de sistemas operativos basados en Unix, por convención también se incluye cuando desarrollamos en Windows.

La segunda determina la **codificación de caracteres** de nuestro archivo. Nuevamente, por convención, se utiliza UTF-8.

La tercera línea, ahora sí, es un auténtico código de Python. Allí llamamos a la función incorporada `print()` y le pasamos una cadena de caracteres como argumento para que imprima en la pantalla. Se dice que es incorporada ya que es una herramienta que el lenguaje nos pone siempre a disposición en nuestros programas. Existen muchas otras que iremos conociendo en el camino.

Para poder ejecutar este pequeño código primero debemos guardarlo. Para ello, en IDLE vamos a ir al menú **File > Save** y lo guardaremos en el escritorio como `hola.py`.

Ahora bien, recordemos que Python es un lenguaje interpretado. Esto quiere decir que no hay un programa compilador que transforme nuestro código fuente (`hola.py`) y lo convierta en un archivo ejecutable (`hola.exe`, por ejemplo); más bien, hay un programa llamado intérprete al cual le indicamos que queremos ejecutar un archivo determinado. Todos los editores de código pueden hacer esto automáticamente (por ejemplo, en IDLE, presionando F5), sin embargo, lo haremos de forma manual, esto es, invocando al intérprete desde la terminal. Esto nos dará un panorama más amplio sobre cómo funciona todo en el mundo de Python.

Entonces, como decíamos, vamos a abrir la terminal. Todo sistema operativo tiene algún atajo para esto. En Windows, puedes presionar **CTRL + R** y escribir cmd, o bien buscar el programa de nombre "Símbolo del sistema". El primer paso es ubicarnos en la ruta en donde hemos guardado nuestro archivo (el escritorio) vía el comando cd. Hecho esto, ejecutamos nuestro script de Python escribiendo `python` seguido del nombre del archivo.

```
> cd Desktop
> python hola.py
¡Hola, IDAT!
```



¡Felicitaciones! Has ejecutado tu primer código de Python.

Recuerda que los programas que escribimos en Python son por defecto aplicaciones de consola. Haciendo doble clic sobre `hola.py` hará que el intérprete ejecute nuestro archivo, pero una vez impreso el mensaje se cerrará automáticamente (pues es lo que ocurre con todo programa cuando alcanza la última línea de código).

9. LAUNCH IN EDUBE

EJERCICIO 1:

- familiarizarse con la función `print()` y sus capacidades de formato;
- experimentando con el código de Python.

Modifique la primera línea de código en el editor, usando las palabras clave `sep` y `end`, para que coincida con el resultado esperado.

Recuerde, utilice dos funciones `print()` y no cambie nada en el código existente.

Sample Solution

```
print("Programming", "Essentials", "in", sep="***", end="...")  
print("Python")
```

RESULTADO:

Programación *** Essentials *** en ... Python

CONCLUSIONES Y RECOMENDACIONES DE LA EXPERIENCIA

- Sabiendo qué salida mostrará nuestro programa, podremos adecuar la forma que solicitará nuestro cliente.
- Se recomienda usar el Edube SandBox para realizar nuestros ejercicios sin necesidad de instalar algún software adicional.

ACTIVIDAD VIRTUAL

1. Observa y analiza el siguiente video: <https://www.youtube.com/watch?v=U3CGMyjzlvM>, y responde las siguientes preguntas:
 - ¿Qué es un algoritmo? – ¿Cuáles son las partes de un algoritmo?
 - Menciona 2 algoritmos que aparecen en el video
 - De tu vida cotidiana, menciona un algoritmo
 - Según el video, ¿Qué es la programación?

