

# Manual de Python para Bachillerato (v.1)



# Generalidades. Uso del intérprete

### Conceptos Básicos:

**Programa:** Secuencia de instrucciones para un ordenador

**Lenguaje de bajo nivel o código máquina:** secuencia de unos y ceros, muy difícil de manejar y muy fácil cometer errores si lo manejamos.

**Lenguaje de alto nivel:** Es un lenguaje de programación más parecido al humano(inglés), el cual es traducido a código máquina por diversas herramientas.

**Algoritmo:** Secuencia de pasos (finita y sin ambigüedades) necesaria para resolver un problema.

**Programa:** Resultado de implementar un algoritmo, empleando un sistema informático.

Para obtener el código máquina que "entiende" el ordenador a partir de nuestro programa escrito en alto nivel necesitamos unas herramientas que "traducen" y son de dos tipos:

1. **Compiladores:** Convierten el programa que hemos escrito en alto nivel (Programa Fuente) en un programa escrito en código máquina al que se denomina Ejecutable (porque este programa es entendido ya por el ordenador). Si en el proceso hay un error, se interrumpe el proceso y no se crea el ejecutable.
2. **Intérpretes:** No se crea ningún programa ejecutable capaz de funcionar por sí solo. Se van traduciendo una a una las órdenes del programa y se van ejecutando y se pasa a la siguiente...Si se encuentra un error, se detiene en este punto, pero las anteriores órdenes ya se han ejecutado

**Un programa interpretado empezará a funcionar antes que uno compilado, pero será más lento en los programas de cálculo intensivo**

Python es un lenguaje de programación ***interpretado*** que por defecto ya se encuentra instalado en muchos sistemas operativos. De todas formas, se trata de un lenguaje desarrollado bajo la filosofía de software libre y está disponible de manera gratuita para todos los sistemas operativos. (<http://www.python.org>).

Para escribir programa en Python, sólo necesitamos un editor de textos (Bloc de Notas, gEdit, etc...) y el intérprete que será el que haga que las instrucciones escritas en el editor de textos se ejecuten.

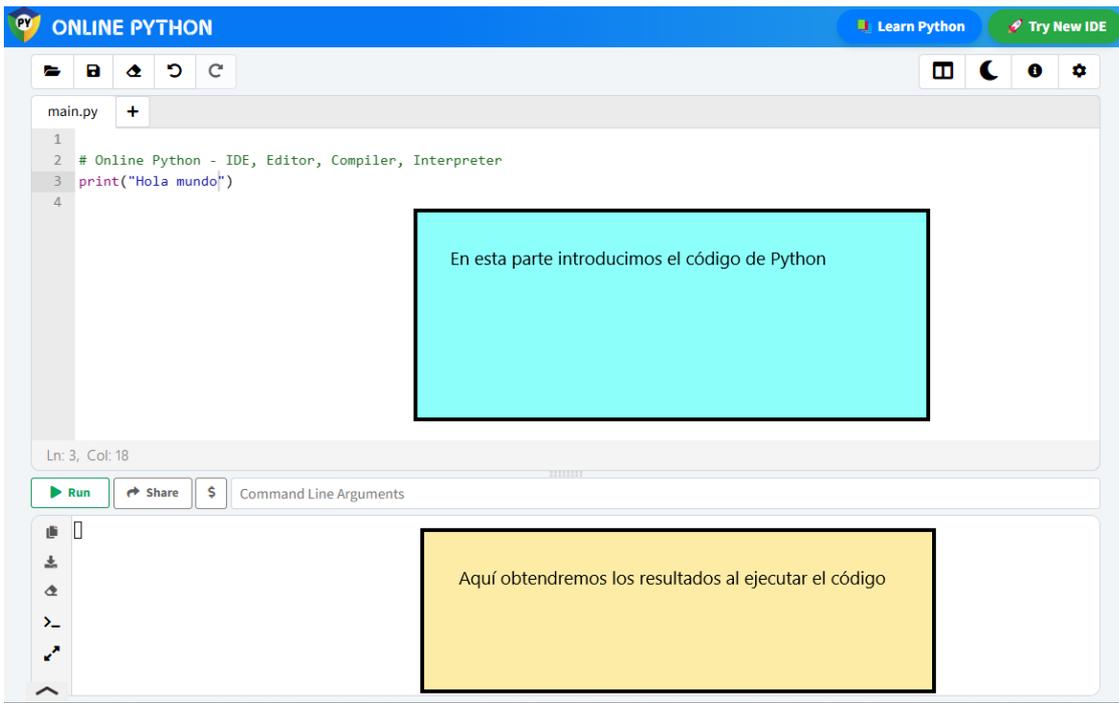
Hay muchos (programas) entornos de desarrollo integrados (IDE's) de gran calidad y con muchísimas posibilidades para desarrollar aplicaciones en Python de manera profesional. Sin embargo, estos entornos tienen tantas características y opciones, que suelen abrumar a las personas se están iniciando en el mundo de la programación y resultan más bien una barrera, si de lo que se trata es de dar los primeros pasos en programación.

Nosotros usaremos online Python, que es un entorno online que nos permite compartir los códigos generados a partir de un enlace.

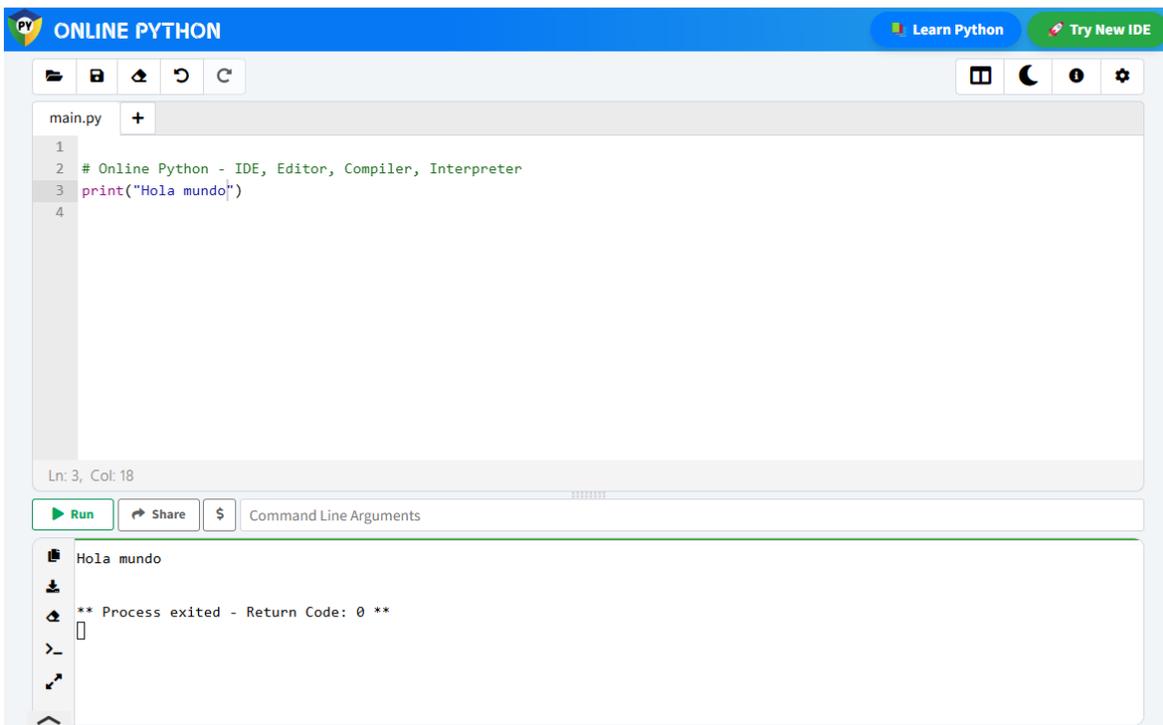
Sin embargo, todo lo recogido en este manual se puede llevar a cabo con cualquiera de los restantes IDE's existentes en la actualidad para Python.

Como hemos dicho, utilizaremos un entorno de programación online de fácil uso, que está en Lo encontramos en la dirección <https://www.online-python.com>

Al acceder a esta URL, nos encontramos lo siguiente:

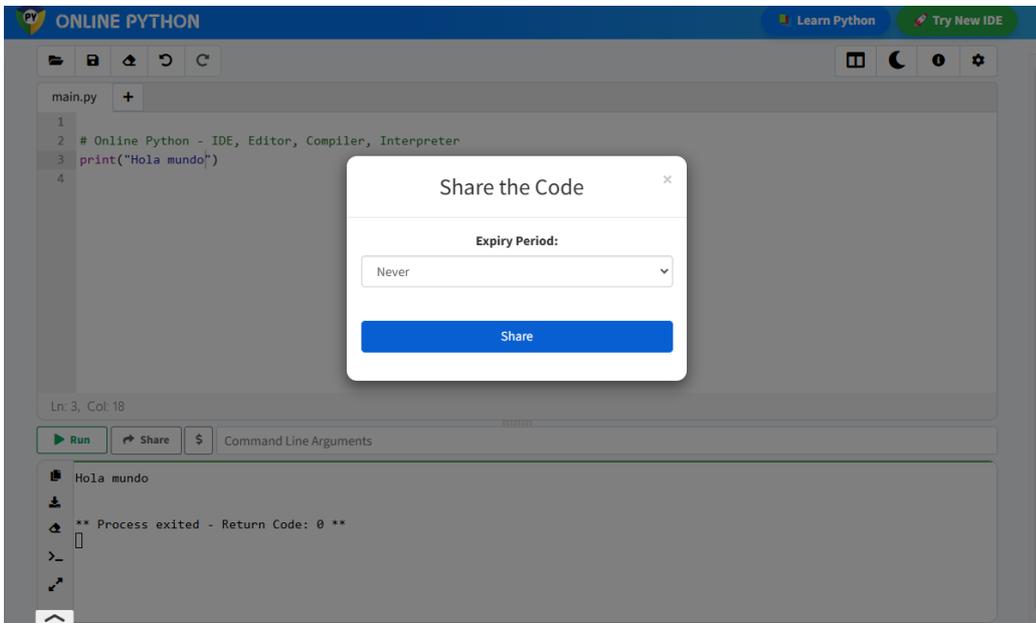


En la ventana de arriba, tenemos el código que nosotros hemos creado, que en este caso consta de una sola instrucción (`print("Hola mundo")`). Si hacemos clic en el botón "RUN", obtendremos el resultado de ejecución del programa:

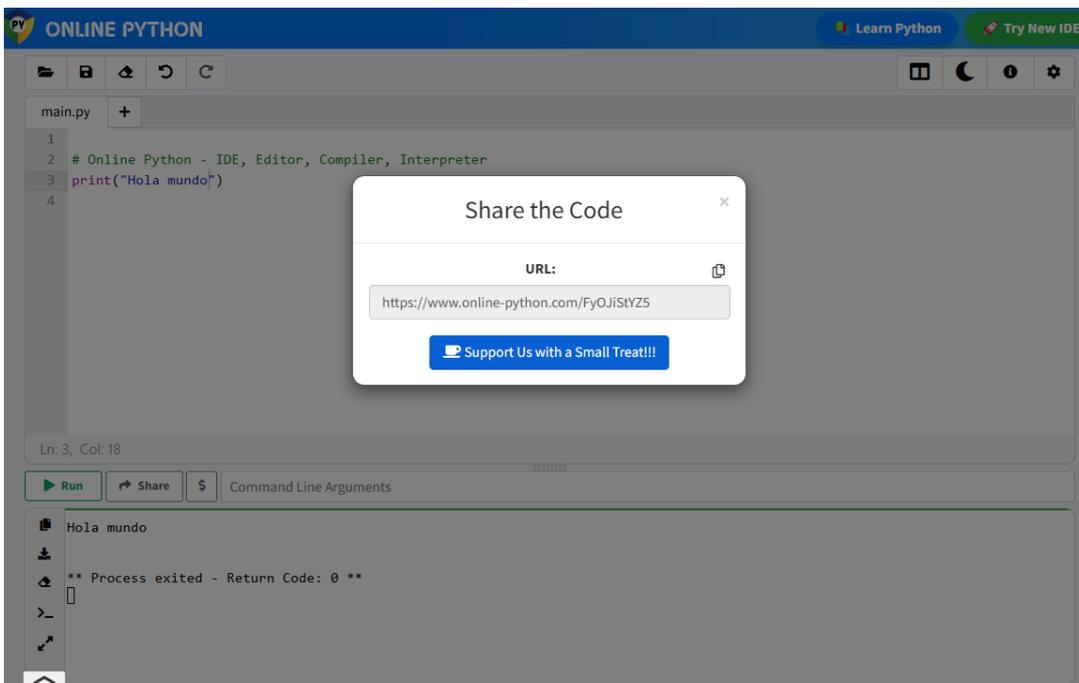


Aquí podemos ver el resultado de la ejecución del programa en la ventana inferior, que no es más que la impresión (por pantalla) de la frase "Hola mundo". En cuanto a la frase de abajo que aparece en inglés, nos dice que todo ha ido bien, sin errores...

Por último, podremos guardar el código generado en el ordenador, simplemente haciendo clic en el icono del disco, o compartirlo con una URL; para hacer esto último, basta con pulsar el botón SHARE:



Elegimos la opción de que no expire (NEVER), a continuación obtendremos la dirección (URL) que podremos enviar para compartir nuestro código, tal y como aparece en la siguiente imagen:



**Palabras reservadas de Python (palabras claves):**

and	as	assert	break	class
continue	def	del	elif	else
except	finally	for	from	global
if	import	in	is	lambda
nonlocal	not	or	pass	raise
return	try	while	with	yield
False	None	True		

**Generalidades**

Para finalizar este primer capítulo introductorio, vamos a exponer algunos aspectos generales sobre Python que resultan de vital importancia para poder escribir código en este lenguaje.

**Sensible a mayúsculas y minúsculas**, Python es un lenguaje que diferencia entre mayúsculas y minúsculas, esto significa: For es diferente de for, nombre es diferente de Nombre, etc.

**Bloques de código**. Hay determinadas instrucciones (for, while, id, elif, else, def, try, except) que tienen por objetivo ejecutar un conjunto de instrucciones que están relacionadas entre sí, a estas instrucciones se le llaman bloque de código.

Las instrucciones que definen el comienzo del bloque de código, deben terminar en el carácter de los 2 puntos (:).

**Indentación**, es la forma que tiene Python de realizar bloques de código. Se llama así una especie de sangría antes del texto que se establece para clarificar la escritura del código.

Veámoslo mejor con un ejemplo:

```

1     for n in (1,2,3,4,5,6):
2         print(n, n*10, n*100)
3         print('-----', n, '----- ')
4         print()

```

En la línea 1 se ha usado una instrucción que lleva implícita la ejecución de un bloque de código, por ese motivo esa instrucción finaliza en :

Las líneas 2, 3, y 4 forman el bloque de código asociado a la instrucción de la línea 1. Estas líneas están asociadas porque están desplazadas unos espacios hacia la derecha con respecto a la línea 1.

Aunque no es obligatorio, se ha acordado que la indentación estándar sea de 4 espacios.

La indentación es una práctica habitual en la mayoría de lenguajes de programación porque mejora la lectura del código, es una buena forma de visualizar las líneas que están relacionadas entre sí y las que están incluidas, unas dentro de otras. Sin embargo, en Python es una regla para escribir el código, es decir, no existen otros caracteres para agrupar código, necesariamente hay que hacerlo a través de la indentación.

## Comentarios

Un comentario se llama a un texto incluido dentro del código pero que no tiene ningún efecto sobre la ejecución del código. Un comentario es una nota aclaratoria sobre la funcionalidad del programa, el uso de una variable, o cualquier otra aclaración que se considere necesaria para comprender mejor el código que se está escribiendo.

En Python tenemos 2 formas de escribir comentarios:

Con la almohadilla #, podemos escribir comentarios de una sola línea: # Comentario

Con las triples comillas podemos escribir código de varias líneas. Son válidas las comillas dobles y las comillas simples:

```
''' Este texto aunque sea de varias líneas contiene un comentario '''
```

**iii En todos los programas pondremos, al inicio, el nombre del programa, el autor y lo que hace!!!**