



# PYTHON

PRIMEROS PASOS

# IDLE - Entorno de Desarrollo de Python

IDLE (Integrated DeveLopment Environment for Python) es un entorno gráfico de desarrollo elemental que permite editar y ejecutar programas en Python.

IDLE es también un entorno interactivo en el que se pueden ejecutar instrucciones sueltas de Python.

En Windows, IDLE se distribuye junto con el intérprete de Python, es decir, al instalar Python en Windows también se instala IDLE.



# IDLE como entorno interactivo

Al abrir IDLE con el acceso directo Inicio se abrirá la ventana principal de IDLE.

Además de indicarnos la versión de Python instalada en nuestra máquina, aquí podemos escribir órdenes después del símbolo de petición `>>>`

## Algunas recomendaciones:

1. Los nombres de los ficheros deben escribirse en minúscula.
2. Las llamadas a funciones se escriben en minúscula
3. Las variables se escriben en minúscula y, de estar formadas por varias palabras, éstas van unidas por guiones bajos.
4. Los tipos de dato se escriben en minúscula,
5. Se debe colocar un espacio después de cada coma.
6. Pon un espacio antes y después de cada operador.
7. Pon un espacio antes y después de cada operador.
8. Indenta con espacios.
9. Escribe abundantes comentarios en tu código, describiendo cada detalle, para hacer que sea lo más claro y legible posible



# Salida de información

Para mostrar texto en Python usamos la función print, cuya sintaxis es:

```
print('texto')
```

Recuerda: `print('el texto que quieres mostrar')`

¿Y si en lugar de texto quiero mostrar una operación?

Es muy fácil, solo debes tener en cuenta una regla: cuando vas a mostrar texto, el interior de los paréntesis de print va con comillas simples:

```
print('texto')
```

Pero cuando quieres mostrar números u operaciones, va sin comillas:

```
print(1 + 2)
```



¿podríamos mostrar texto y números al mismo tiempo en nuestro programa?

Para combinar texto con operaciones debes respetar la regla que mencionamos antes: el texto lleva comillas, los números no

```
print('10 * 3 =', 10 * 3)
```

# Comentarios

Los comentarios son anotaciones que hace el programador en su código fuente.

Sirven para que al leer el código, otros programadores puedan entenderlo fácilmente (o incluso el autor original, al volver a leerlo pasado un tiempo).

En Python los comentarios se hacen con un numeral (#).

```
print('Hola') # Te estoy saludando.
```



# Operaciones con Python

En Python, podemos operar con números enteros y números reales.

Ejemplo:

Número entero: 100

Número real: 20.30

Hay algo importante a tener en cuenta, y es que el resultado de una operación será entero o real dependiendo del tipo de números con los que se opere.

$20 + 30 = 50$

# El resultado es entero, porque todos los números lo son.

$5 * 2 = 10$

# El resultado también es entero.

$20.0 + 30 = 50.0$

# El resultado es real, porque hay un número real.

$5 * 2.5 = 12.5$

# El resultado también es real.

$2.2 - 0.2 = 2.0$

# El resultado también se muestra como real.

# Operaciones básicas en Python

OPERACIÓN	OPERADOR	EJEMPLO
SUMA	+	$2 + 4 = 6$
RESTA	-	$2 - 4 = -2$
MULTIPLICACIÓN	*	$3 * 3 = 9$
POTENCIA	**	$3 ** 3 = 27$
DIVISIÓN	/	$50 / 6 = 8.3$
COCIENTE	//	$50 // 6 = 8$
RESTO	%	$50 \% 6 = 2$



# Variables

Hasta aquí nuestra información era estática y no se guardaba en ningún lado.

Una variable es un elemento que permite almacenar información

```
nombre = 'Almudena' # Guardo el nombre de la chica.  
edad = 19 # Guardo su edad.  
# Muestro su nombre: print('Te llamas', nombre)  
# A continuación muestro cuantos años tiene: print('Tienes', edad, 'años.')
```

La función print también sirve para mostrar variables.  
Las variables también deben respetar la norma que vimos antes:  
van sin comillas y deben estar separadas del texto por coma

OPERACIÓN	OPERADOR	EJEMPLO
ASIGNACIÓN	$a = b$	$a = b$
SUMA	$a = a + b$	$a + = b$
RESTA	$a = a - b$	$a - = b$
MULTIPLICACIÓN	$a = a * b$	$a * = b$
POTENCIA	$a = a ** b$	$a ** = b$
DIVISIÓN	$a = a / b$	$a / = b$
COCIENTE	$a = a // b$	$a // = b$
RESTO	$a = a \% b$	$a \% = b$



# Entrada de información

Otra acción imprescindible que realiza un programa es la de tomar información.

Esto le permite interactuar con el usuario, adaptándose a las necesidades de éste. Si un programa no toma información los resultados siempre serán los mismos, por lo que resultará de poca utilidad.

## La función input()

Puede utilizarse para dos cosas: hacer pausas en nuestro programa y permitir que el usuario introduzca información.

```
input('Pulsa INTRO para continuar...')
```

Ahora vamos a pedirle información al usuario y guardarla en variables. Para ello necesitamos conocer algunos tipos de datos, los cuales se muestran en la siguiente tabla:

TIPO	CLASE	EJEMPLO
str	Texto	'Esto es un texto'
int	Número entero	23
float	Número Real	4.14

```
nombre = str(input('Introduce tu nombre'))  
print('Te llamas', nombre)
```



Usamos str cuando queremos que el usuario introduzca cualquier tipo de texto como palabras, frases, etc.

En Python también podemos operar con texto (sumarlo, restarlo, multiplicarlo...)

```
# Este programa une palabras:  
palabra_1 = str (input('Introduce la primera palabra'))  
palabra_2 = str (input('Introduce la segunda palabra'))  
compuesta = palabra_1 + palabra_2 # Hago la unión de ambos textos.  
print('Me he inventado esta palabra:', compuesta)
```

El tipo `int` se utiliza cuando queremos que el usuario introduzca números enteros.

Este tipo también es válido para números muy grandes

```
edad = int (input('Introduce tu edad'))  
print('Tienes', edad, 'años.')
```

Usaremos el tipo `float` para que el usuario introduzca números reales.

```
pi = float (input('¿Cuál es el número PI?'))  
print('El número PI es', pi)
```

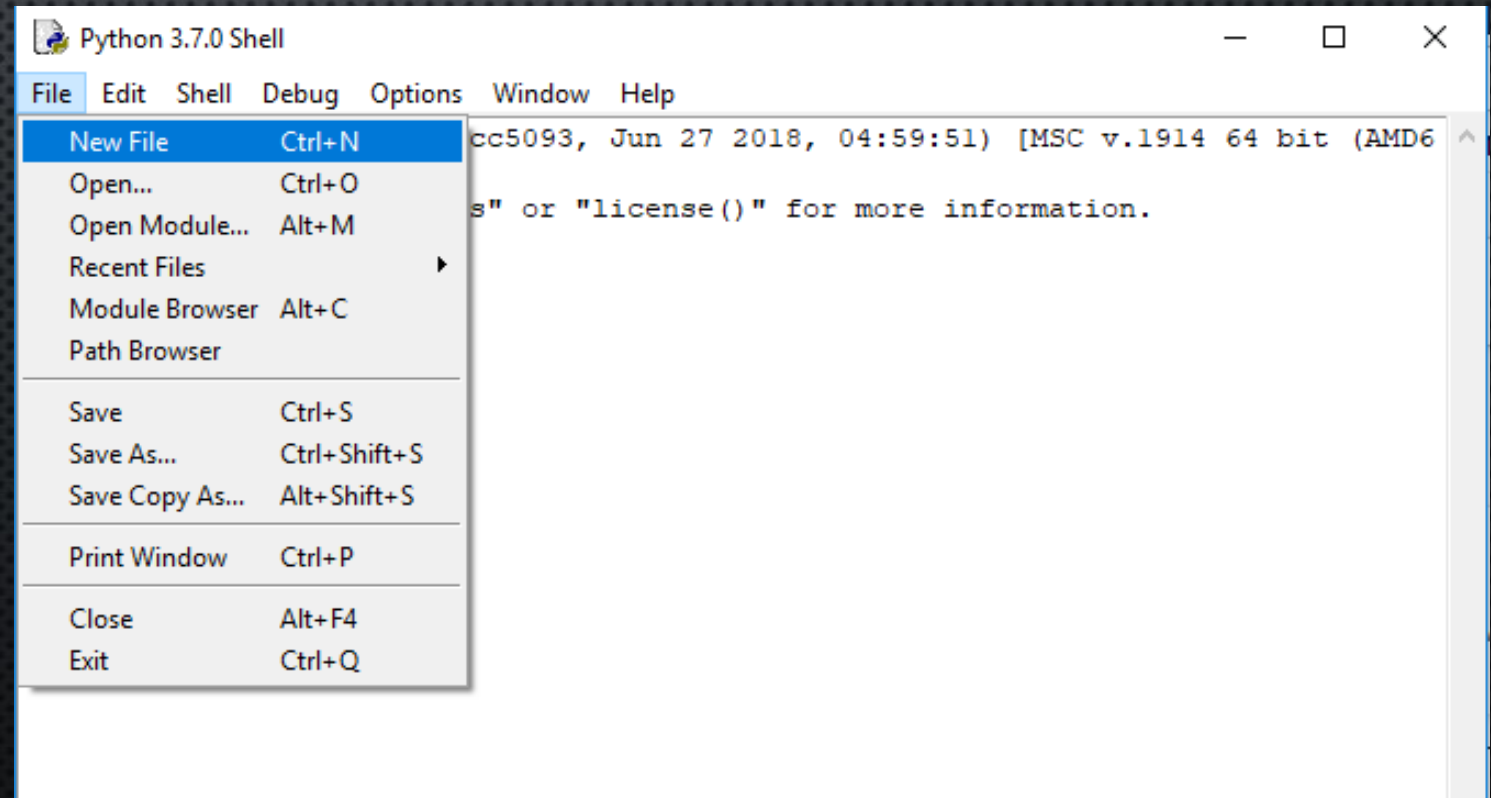


IDLE es también un editor de programas elemental, que permite redactar programas, guardarlos en archivos y ejecutarlos.

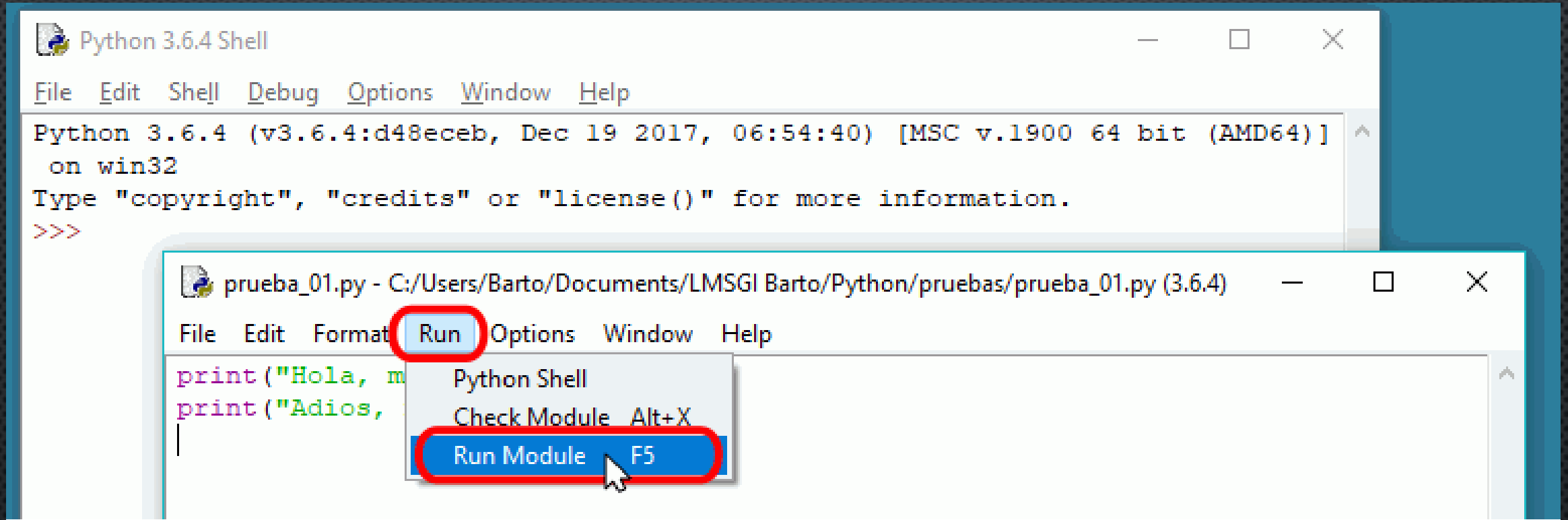
La ventana principal de IDLE es siempre el entorno interactivo, pero también es la ventana donde se ejecutarán los programas.

Para crear un archivo de programa con IDLE, abra una nueva ventana mediante el menú File > New File

La extensión habitual de los programas de Python es .py

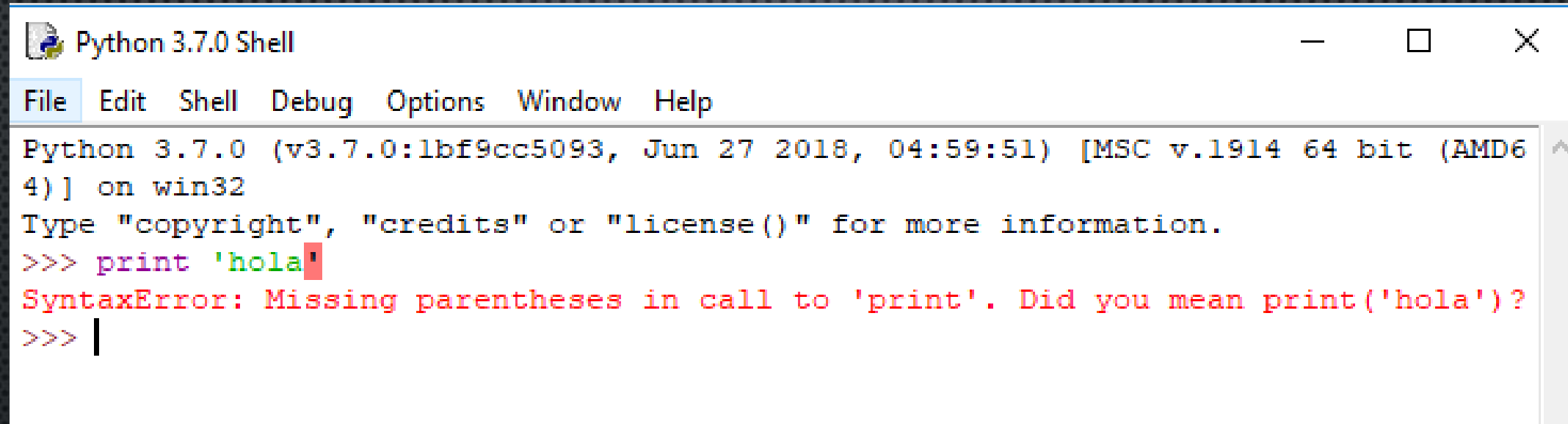


# Ejecutar programas de Python en IDLE





# Errores en los programas



The screenshot shows a window titled "Python 3.7.0 Shell" with a menu bar containing "File", "Edit", "Shell", "Debug", "Options", "Window", and "Help". The main text area displays the following content:

```
Python 3.7.0 (v3.7.0:1bf9cc5093, Jun 27 2018, 04:59:51) [MSC v.1914 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> print 'hola'
SyntaxError: Missing parentheses in call to 'print'. Did you mean print('hola')?
>>> |
```

The error message "SyntaxError: Missing parentheses in call to 'print'. Did you mean print('hola')?" is displayed in red text, indicating the type of error and a suggestion for the correct syntax.