

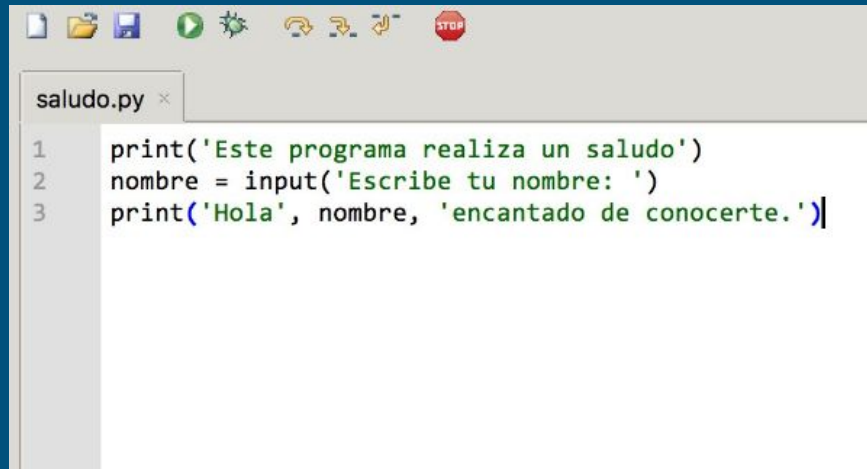


# Python para principiantes



## Capítulo 3: Primeros Programas





The image shows a screenshot of a Python IDE window titled "saludo.py". The window contains three lines of Python code:

```
1 print('Este programa realiza un saludo')
2 nombre = input('Escribe tu nombre: ')
3 print('Hola', nombre, 'encantado de conocerte.')
```

The code is displayed in a monospaced font with line numbers on the left. The IDE interface includes a toolbar at the top with icons for file operations, running, and debugging.



# ¡Hola Mundo!

---

En el mundo de la programación, normalmente se comienza escribiendo un programa que visualice un mensaje por pantalla.

A este programa se le suele llamar **¡Hola Mundo!**

A partir de este momento, nuestras instrucciones las escribiremos en el editor de código,

```
1 print('¡Hola Mundo!')
```

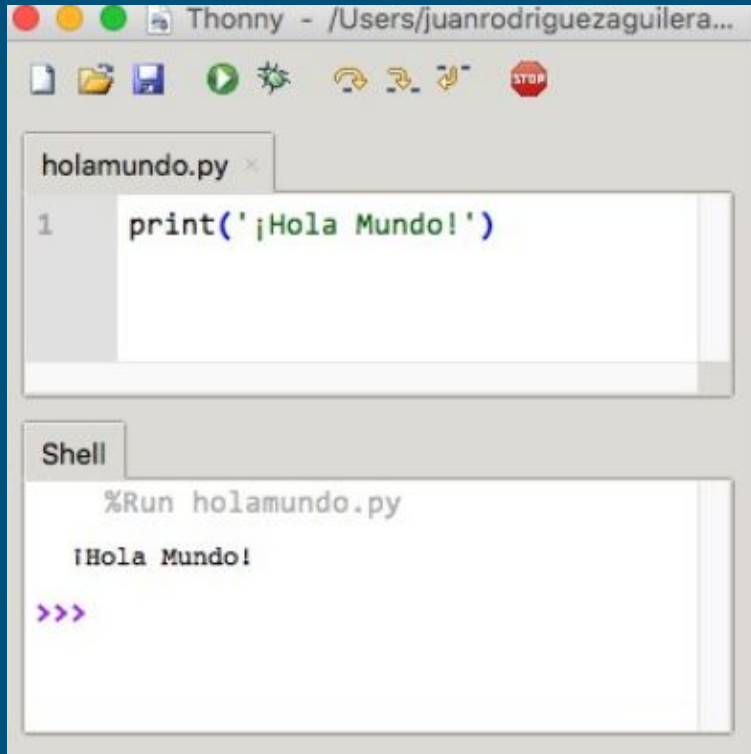
En este programa hemos usado la función `print` para mostrar la cadena de texto `'¡Hola Mundo!'`, recuerda que también se podría haber escrito con comillas dobles `print("¡HolaMundo!")`

# ¡Hola Mundo!

---

- Antes de ejecutar el programa vamos a guardar nuestro código un archivo al cual le pondremos el nombre: holamundo.py
- Thonny antes de ejecutar cualquier código escrito en el editor, nos pedirá que lo guardemos en un archivo
- Recuerda que en Python se recomienda que los archivos de código tengan la extensión .py
- Para ejecutar nuestro programa sólo hay que pulsar sobre el botón Run o la tecla F5

# ¡Hola Mundo!



The screenshot shows the Thonny Python IDE interface. At the top, the title bar reads "Thonny - /Users/juanrodriguezaguilera...". Below the title bar is a toolbar with various icons. The main editor window displays a file named "holamundo.py" with the following code:

```
1 print('¡Hola Mundo!')
```

Below the editor is a "Shell" window. It shows the command "%Run holamundo.py" and the output "¡Hola Mundo!". The prompt ">>>" is visible at the bottom of the shell window.

- Haz algunas pruebas para familiarizarte con Thonny
- Utiliza el modo depuración (Debug) pulsando el botón que se encuentra justo a la derecha de Run (o Ctrl+F5)
- Haz pruebas con los botones siguientes o sus correspondientes acciones con el teclado (F6, F7 y F8).
- En ventana de nuestro intérprete, que se denomina **Shell**, estamos viendo el resultado de la ejecución de nuestro programa
- Es normal, que esta ventana se llene pronto de contenido, mensajes de error, pruebas, etc.
  - botón derecho del ratón sobre ella y eligiendo la opción **Clear Shell** o desde **menú Edit**

# Programa: saludo.py

---

- 1º Mostrará un mensaje indicando que se trata de un programa para elaborar un saludo.
- 2º Solicitará el nombre de la persona que está usando el programa.
- 3º Mostrará un nuevo mensaje saludando a la persona con el nombre que haya escrito.

# Programa: saludo.py

---

```
1 print('Este programa realiza un saludo')
2 nombre = input('Escribe tu nombre: ')
3 print('Hola', nombre, 'encantado de conocerte.')
```

- **La línea 1** muestra un mensaje de texto, en este caso el uso de la función print no tiene mayor explicación
- **La línea 2** muestra un mensaje mediante la función input y queda a la espera de que el usuario introduzca alguna información y finalice con la pulsación de la tecla Intro (o Enter) Una vez que el usuario ha escrito su nombre, en este caso, la información introducida queda guardada en la variable nombre en formato cadena de texto.
- **La línea 3** vuelve a mostrar un mensaje, pero en esta ocasión a la función print se le han pasado varios parámetros separados por comas.

# Programa: saludo.py

---

Nuestro programa generará una salida similar a la siguiente imagen:

```
>>> %Run saludo.py
```

```
Este programa realiza un saludo  
Escribe tu nombre: Arturo  
Hola Arturo encantado de conocerte.
```

```
>>> |
```

Es importante resaltar que la información introducida quedará guardada en la variable **nombre** en **formato texto**, incluso si se introduce un valor numérico.

¿Qué ocurre si quiero tratar la información introducida como un valor numérico?

Disponemos de varias funciones para convertir texto en su correspondiente valor numérico:

- **int** Interpreta el texto introducido como un valor numérico entero.
- **float** Interpreta el texto introducido como un valor numérico decimal.
- **eval** Evalúa matemáticamente el texto introducido. Es la forma más versátil.



# Programa: sumar2numeros.py

---

- 1º El programa mostrará un mensaje indicando para que sirve.
- 2º Solicitará el primero de los números.
- 3º Solicitará el segundo de los números.
- 4º Mostrará los 2 números introducidos y el resultado de la suma.

# Programa: sumar2numeros.py

---

```
1 print('Programa para calcular la suma de 2 números')
2 n1 = input('Escribe el primero de los números: ')
3 n2 = input('Escribe el segundo de los números: ')
4 suma=eval(n1)+eval(n2)
5 print('La suma de', n1, 'y', n2, 'es', suma)
```

En la líneas 2 y 3, hemos solicitado al usuario que introduzca 2 números y se han almacenado en las **variables n1** y **n2**, pero se han almacenado como cadenas de texto. Si en nuestro caso '32' y '47', concatenamos dichas cadenas, obtendríamos '3247'.

# Programa: sumar2numeros.py

---

```
>>> %Run suma2numeros.py

Programa para calcular la suma de 2 números
Escribe el primero de los números: 32
Escribe el segundo de los números: 47
La suma de 32 y 47 es 79

>>> |
```

En la línea 4:

Evalúa como un número la cadena de texto que contenga n1, evalúa también la cadena que contenga n2 y suma el resultado que te hayan dado. Además, asigna el resultado de la suma a la variable suma

En la línea 5 únicamente hemos construido una forma de mostrar el resultado, podríamos haberlo hecho también así: `print(n1,'+',n2,'=',suma)`

# Programa: sumar2numeros.py

---

Otras posibles maneras de resolverlo:

```
2     n1 = eval(input('Escribe el primero de los números: '))
3     n2 = eval(input('Escribe el segundo de los números: '))
```

Estamos diciendo que al mismo tiempo de solicitar los números, el texto introducido sea evaluado matemáticamente y asignado a las correspondientes variables.

Así, las línea 4 tendría que haber sido:

```
4     suma = n1 + n2
```

# Programa: sumar2numeros.py

---

Última forma de resolverlo:

```
1     print('Programa para calcular la suma de 2 números')
2     n1 = input('Escribe el primero de los números: ')
3     n1 = eval(n1)
4     n2 = input('Escribe el segundo de los números: ')
5     n2 = eval(n2)
6     suma = n1 + n2
7     print(n1, '+', n2, '=', suma)
```

# Programa: sumar2numeros.py

---

Los valores no tienen por qué ser números enteros, pueden ser valores decimales siempre que **usemos el punto (.) como separador decimal**

```
>>> %Run suma2numeros.py

Programa para calcular la suma de 2 números
Escribe el primero de los números: 3.5
Escribe el segundo de los números: 2.8
La suma de 3.5 y 2.8 es 6.3

>>> |
```