

Entrada por teclado: la función `input()`

En Informática, la "entrada" de un programa son los datos que llegan al programa desde el exterior. Actualmente, el origen más habitual es el teclado.



Los ejemplos de esta lección se deben ejecutar en un programa, no en el prompt de IDLE (a menos que se vayan ejecutando línea a línea).

La función `input()`

La función `input()` permite obtener texto escrito por teclado. Al llegar a la función, el programa se detiene esperando que se escriba algo y se pulse la tecla **Intro**, como muestra el siguiente ejemplo:

```
print("¿Cómo se llama?")
nombre = input()
print(f"Me alegro de conocerle,
{nombre}")
```

```
¿Cómo se llama?
Pepe
Me alegro de conocerle, Pepe
```

En el ejemplo anterior, el usuario escribe su respuesta en una línea distinta a la pregunta porque Python añade un salto de línea al final de cada `print()`.

Si se prefiere que el usuario escriba su respuesta a continuación de la pregunta, se podría utilizar el argumento opcional `end` en la función `print()`, que indica el carácter o caracteres a utilizar en vez del salto de línea. Para separar la respuesta de la pregunta se ha añadido un espacio al final de la pregunta.

```
print("¿Cómo se llama? ", end=" ")
nombre = input()
print(f"Me alegro de conocerle,
{nombre}")
```

```
¿Cómo se llama? Pepe
Me alegro de conocerle, Pepe
```

Otra solución, más compacta, es aprovechar que a la función `input()` se le puede enviar un argumento que se escribe en la pantalla (sin añadir un salto de línea):

```
nombre = input("¿Cómo se llama? ")
print(f"Me alegro de conocerle,
{nombre}")
```

```
¿Cómo se llama? Pepe
Me alegro de conocerle, Pepe
```

Conversión de tipos

De forma predeterminada, la función `input()` convierte la entrada en una cadena, aunque escribamos un número. Si intentamos hacer operaciones, se producirá un error.

```
cantidad = input("Dígame una cantidad en pesetas: ")
print(f"{cantidad} pesetas son {round(cantidad / 166.386, 2)} euros")
```

```
Dígame una cantidad en pesetas: 500
Traceback (most recent call last):
  File "ejemplo.py", line 2, in <module>
    print(f"{cantidad} pesetas son {round(cantidad / 166.386)} euros")
TypeError: unsupported operand type(s) for /: 'str' and 'float'
```

Si se quiere que Python interprete la entrada como un número entero, se debe utilizar la función `int()` de la siguiente manera:

```
cantidad = int(input("Dígame una cantidad en pesetas: "))
print(f"{cantidad} pesetas son {round(cantidad / 166.386, 2)} euros")
```

```
Dígame una cantidad en pesetas: 500
500 pesetas son 3.01 euros
```

De la misma manera, para que Python interprete la entrada como un número decimal, se debe utilizar la función `float()` de la siguiente manera:

```
cantidad = float(input("Dígame una cantidad en euros (hasta con 2 decimales): "))
print(f"{cantidad} euros son {round(cantidad * 166.386)} pesetas")
```

```
Dígame una cantidad en euros (hasta con 2 decimales): 9.99
9.99 euros son 1662 pesetas
```

Pero si el usuario no escribe un número, las funciones `int()` o `float()` producirán un error:

```
cantidad = float(input("Dígame una cantidad en euros (hasta con 2 decimales): "))
print(f"{cantidad} euros son {round(cantidad * 166.386)} pesetas")
```

```
Dígame una cantidad en euros (hasta con 2 decimales): Pepito
Traceback (most recent call last):
  File "ejemplo.py", line 1, in <module>
    cantidad = float(input("Dígame una cantidad en euros: "))
ValueError: could not convert string to float: 'Pepito'
```

De la misma manera, si el usuario escribe un número decimal, la función `int()` producirá un error:

```
edad = int(input("Dígame su edad: "))
print(f"Su edad son {edad} años")
```

```
Dígame su edad: 15.5
Traceback (most recent call last):
  File "ejemplo.py", line 1, in <module>
    edad = int(input("Dígame su edad: "))
ValueError: invalid literal for int() with base 10: '15.5'
```

Pero si el usuario escribe un número entero, la función `float()` **no** producirá un error, aunque el número se escribirá con parte decimal (.0):

```
peso = float(input("Dígame su peso en kg: "))
print(f"Su peso es {peso} kg")
```

```
Dígame su peso en kg: 84
Su peso es 84.0 kg
```

Variables como argumento de la función `input()`

La función `input()` sólo puede tener un argumento.

En versiones de Python anteriores a la versión 3.6 esto causaba problemas cuando se querían incorporar variables en el argumento de la función `input()`, como se comenta en la lección de [Temas obsoletos](#), pero las cadenas "f" permiten hacerlo fácilmente:

```
nombre = input("Dígame su nombre: ")
apellido = input(f"Dígame su apellido,
{nombre}: ")
print(f"Me alegro de conocerle, {nombre}
{apellido}.")
```

```
Dígame su nombre: Pepito
Dígame su apellido, Pepito: Conejo
Me alegro de conocerle, Pepito Conejo.
```

```
numero1 = int(input("Dígame un número:
"))
numero2 = int(input(f"Dígame un número
mayor que {numero1}: "))
print(f"La diferencia entre ellos es
{numero2 - numero1}.")
```

```
Dígame un número: 5
Dígame un número mayor que 5: 8
La diferencia entre ellos es 3.
```

Última modificación de esta página: 16 de febrero de 2018



Esta página forma parte del curso [Introducción a la programación con Python](#) por [Bartolomé Sintes Marco](#) que se distribuye bajo una [Licencia Creative Commons Reconocimiento-CompartirIgual 4.0 Internacional \(CC BY-SA 4.0\)](#).