

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.

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("Me alegro de conocerle,", nombre)
```

Salida:

```
¿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("Me alegro de conocerle,", nombre)
```

Salida:

```
¿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("Me alegro de conocerle,", nombre)
```

Salida:

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

Conversión de tipos

De forma predeterminada, la función `input()` convierte la entrada en una cadena. 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(cantidad, "pesetas son", cantidad / 166.386, "euros")
```

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

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

Salida:

```
Dígame una cantidad en euros: 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 real, la función `int()` producirá un error:

```
edad = int(input("Dígame su edad: "))
print("Su edad son", edad, "años")
```

Salida:

```
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("Su peso es", peso, "kg")
```

Salida:

```
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. Esto puede causar problemas como en el ejemplo siguiente, en el que se quiere mostrar la respuesta de la primera instrucción en la segunda pregunta:

```
nombre = input("Dígame su nombre: ")
apellido = input("Dígame su apellido, ", nombre, ": ")
print("Me alegre de conocerle,", nombre, apellido)
```

Salida:

```
Dígame su nombre: Pepito
Traceback
(most recent call last):
  File
"ejemplo.py", line 2, in
<module>
    apellido = input("Dígame su apellido, ", nombre, ": ")
TypeError:
input expected at most 1
arguments, got 3
```

Una solución consiste en separar la pregunta de la recogida de la respuesta, escribiendo dos instrucciones:

```
nombre = input("Dígame su nombre: ")
print("Dígame su apellido,", nombre, ": ", end="")
apellido = input()
print("Me alegre de conocerle,", nombre, apellido)
```

Salida:

```
Dígame su nombre: Pepito
Dígame su apellido, Pepito : Conejo
Me alegre de conocerle, PepitoConejo
```

Si se quiere que los dos puntos (:) salgan pegados al nombre habría que utilizar la **concatenación** (operador +):

```
nombre = input("Dígame su nombre: ")
print("Dígame su apellido,", nombre + ": ", end="")
apellido = input()
print("Me alegre de conocerle,", nombre, apellido)
```

Salida:

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

Pero también se puede incluir la pregunta en la recogida de la respuesta, utilizando la concatenación (operador +):

```
nombre = input("Dígame su nombre: ")
apellido = input("Dígame su apellido, " + nombre + ": ")
print("Me alegro de conocerle,", nombre, apellido)
```

Salida:

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

Si se opta por esta solución, hay que tener en cuenta que, en caso de querer concatenar cadenas y números, se debe utilizar la función `str()`, como se ha comentado en el apartado anterior:

```
numero1 = int(input("Dígame un número: "))
numero2 = int(input("Dígame un número mayor que " + numero1 + ":"))
print("La diferencia entre ellos es", numero2 - numero1)
```

Salida:

```
Dígame un número: 5
Traceback (most recent call last):
  File
"ejemplo.py", line 2, in <module>
    numero2 = int(input("Dígame un número mayor que" + numero1 + ": "))
TypeError: Can't convert 'int' object to str implicitly
```

Lo arreglamos con la función `str()`

```
numero1 = int(input("Dígame un número: "))
numero2 = int(input("Dígame un número mayor que " + str(numero1) + ": "))
print("La diferencia entre ellos es", numero2 - numero1)
```

Salida:

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

Referencias

Apuntes generados a partir del curso [Introducción a la programación con Python](#) que se distribuye bajo una [Licencia Creative Commons Reconocimiento-CompartirIgual 4.0 Internacional \(CC BY-SA 4.0\)](#).



