

CAPÍTULO 3. INTRODUCCIÓN DE DATOS Y CONVERSIÓN DE TIPOS

1. ¿Por qué necesitamos introducir datos?

En los capítulos anteriores hemos aprendido a mostrar información en pantalla y a almacenarla en variables.

Por ejemplo:

```
nombre = "Ana"
```

```
print(nombre)
```

Este programa funciona correctamente, pero tiene una limitación importante: siempre utiliza el mismo nombre.

En muchas ocasiones queremos que sea el usuario quien introduzca la información.

Por ejemplo:

- Su nombre.
- Su edad.
- Su altura.
- Su ciudad.
- Su nota media.

Para conseguirlo utilizaremos la función **input()**.

2. La función input()

La función **input()** permite que el usuario introduzca información mediante el teclado.

Su sintaxis básica es:

```
variable = input("Mensaje para el usuario")
```

Ejemplo:

```
nombre = input("Introduce tu nombre: ")
```

Al ejecutar el programa aparecerá:

Introduce tu nombre:

El programa quedará esperando hasta que el usuario escriba algo y pulse la tecla Enter.

Si el usuario escribe:

Ana

la variable nombre almacenará el valor:

"Ana"

3. Nuestro primer programa interactivo

Hasta ahora los programas mostraban información.

A partir de este capítulo los programas también podrán recibir información.

Problema

Pedir el nombre de una persona y mostrarlo posteriormente.

Algoritmo

1. Solicitar el nombre.
 2. Guardarlo en una variable.
 3. Mostrar el contenido de la variable.
-

Seudocódigo

Leer nombre
Escribir nombre

Python

```
nombre = input("Introduce tu nombre: ")  
  
print(nombre)
```

4. Representación visual

Supongamos que el usuario introduce:

Ana

Después de ejecutar la instrucción:

```
nombre = input("Introduce tu nombre: ")
```

la memoria podría representarse de la siguiente manera:

nombre → "Ana"

o bien:

Variable: nombre

Ana

5. Mostrar mensajes personalizados

Una vez almacenado un dato podemos utilizarlo dentro del programa.

Ejemplo:

```
nombre = input("Introduce tu nombre: ")
```

```
print("Hola", nombre)
```

Si el usuario introduce:

Ana

el resultado será:

Hola Ana

Ahora el programa genera una respuesta diferente para cada usuario.

6. Un detalle muy importante

Muchos alumnos creen que si escriben un número Python lo almacenará automáticamente como un número.

Sin embargo, esto no ocurre.

Observa el siguiente programa:

```
edad = input("Introduce tu edad: ")  
  
print(type(edad))
```

Si escribimos:

17

el resultado será:

```
<class 'str'>
```

Aunque el usuario haya escrito un número, Python lo almacena inicialmente como texto.

Esta es una de las ideas más importantes de todo el capítulo.

7. Conversión de tipos

Para trabajar correctamente con los datos necesitamos convertirlos al tipo adecuado.

7.1 Conversión a entero: int()

La función **int()** convierte texto en números enteros.

Ejemplo:

```
edad = int(input("Introduce tu edad: "))
```

Si el usuario escribe:

17

la variable almacenará:

17

como entero.

Comprobación:

```
edad = int(input("Introduce tu edad: "))  
  
print(type(edad))
```

Resultado:

```
<class 'int'>
```

7.2 Conversión a real: float()

La función **float()** convierte texto en números decimales.

Ejemplo:

```
altura = float(input("Introduce tu altura: "))
```

Si el usuario introduce:

1.72

la variable almacenará un número real.

Comprobación:

```
altura = float(input("Introduce tu altura: "))  
  
print(type(altura))
```

Resultado:

```
<class 'float'>
```

7.3 Conversión a texto: str()

La función **str()** convierte cualquier dato en una cadena de texto.

Ejemplo:

```
edad = 17  
  
texto = str(edad)  
  
print(type(texto))
```

Resultado:

```
<class 'str'>
```

8. Ejemplos resueltos

Ejemplo 1. Pedir un nombre

Algoritmo

1. Pedir el nombre.
2. Mostrar el nombre.

Seudocódigo

Leer nombre

Escribir nombre

Python

```
nombre = input("Introduce tu nombre: ")  
  
print(nombre)
```

Ejemplo 2. Pedir edad

Algoritmo

1. Pedir la edad.
2. Mostrar la edad.

Python

```
edad = int(input("Introduce tu edad: "))  
  
print("Tu edad es", edad)
```

Ejemplo 3. Pedir altura

Algoritmo

1. Pedir la altura.
2. Mostrar la altura.

Python

```
altura = float(input("Introduce tu altura: "))  
  
print("Tu altura es", altura)
```

Ejemplo 4. Sumar dos números

Problema

Calcular la suma de dos números introducidos por el usuario.

Algoritmo

1. Pedir el primer número.
2. Pedir el segundo número.
3. Calcular la suma.
4. Mostrar el resultado.

Seudocódigo

```
Leer num1
Leer num2
suma ← num1 + num2
Escribir suma
```

Python

```
num1 = int(input("Primer número: "))
num2 = int(input("Segundo número: "))

suma = num1 + num2

print("Resultado:", suma)
```

9. Errores frecuentes

Error 1. Olvidar convertir a entero

Código:

```
num1 = input("Primer número: ")
num2 = input("Segundo número: ")

print(num1 + num2)
```

Si el usuario introduce:

2
3

obtendrá:

23

Python está uniendo dos cadenas de texto.

La solución es:

```
num1 = int(input("Primer número: "))
num2 = int(input("Segundo número: "))

print(num1 + num2)
```

Error 2. Introducir texto cuando se espera un número

```
edad = int(input("Edad: "))
```

Si el usuario escribe:

diecisiete

Python mostrará un error porque no puede convertir esa palabra a entero.

Error 3. Utilizar la coma decimal

Incorrecto:

1,72

Correcto:

1.72

Python utiliza el punto como separador decimal.

Error 4. Olvidar almacenar el resultado

Incorrecto:

```
input("Introduce tu nombre: ")
```

Correcto:

```
nombre = input("Introduce tu nombre: ")
```

10. Actividades propuestas

Actividades básicas

1. Pide el nombre del usuario y muéstralo.
 2. Pide la edad del usuario y muéstrala.
 3. Pide la ciudad del usuario y muéstrala.
 4. Pide el instituto del usuario y muéstralo.
-

Actividades de práctica

5. Pide dos números enteros y muestra su suma.
 6. Pide dos números enteros y muestra su producto.
 7. Pide una altura y muéstrala.
 8. Pide una temperatura y muéstrala.
-

Actividades de algoritmia

9. Diseña el algoritmo para pedir tres notas y mostrarlas.
 10. Escribe el pseudocódigo correspondiente.
 11. Implementa el programa en Python.
 12. Diseña el algoritmo para pedir la base y la altura de un rectángulo y mostrar ambos valores.
-

Actividades avanzadas

13. Pide dos números y calcula su suma, resta y producto.
 14. Pide el nombre y la edad de una persona y muestra un mensaje personalizado.
 15. Pide el lado de un cuadrado y muestra su perímetro.
 16. Pide la longitud y la anchura de un rectángulo y muestra sus datos.
-

Reto final

Diseña un programa que solicite al usuario:

- Nombre.
- Edad.

- Ciudad.
- Altura.
- Nota media.

Antes de programar:

1. Diseña el algoritmo.
2. Escribe el pseudocódigo.

Finalmente, implementa el programa en Python y muestra una ficha personal completa utilizando toda la información introducida.

Resumen

En este capítulo hemos aprendido que:

- La función `input()` permite introducir datos mediante el teclado.
- Todo dato introducido mediante `input()` se almacena inicialmente como texto.
- Podemos convertir datos mediante `int()`, `float()` y `str()`.
- Los programas pueden interactuar con el usuario.
- La conversión de tipos es fundamental para realizar cálculos correctamente.
- La secuencia Problema → Algoritmo → Pseudocódigo → Python sigue siendo la base para resolver cualquier problema de programación.

En el próximo capítulo aprenderemos a realizar operaciones matemáticas y a construir programas capaces de resolver problemas numéricos más complejos.