

Tipos de datos en Python

En cualquier lenguaje de programación, los tipos de datos son fundamentales para definir cómo se almacenará y manipulará la información. En Python, existen varios tipos de datos básicos que nos permiten trabajar con diferentes formas de información.

Python maneja diferentes tipos de datos que nos permiten almacenar y manipular información:

Cadenas de texto (str)

```
nombre = "Ana"  
mensaje = 'Hola mundo'
```

Las cadenas pueden definirse con comillas dobles o simples y sirven para representar texto. Se representan con `str` debido a que viene del inglés "string", de su nombre completo "character string".

Números enteros (int)

```
edad = 25  
año = 2024
```

Los enteros son números sin parte decimal, pueden ser positivos o negativos. Se representan con `int` debido a que viene del inglés "integer", que significa "entero".

Números decimales (float)

```
altura = 1.75  
temperatura = 36.5
```

Los flotantes representan números con parte decimal. Se representan con `float` debido a que viene del inglés "floating point", que significa "punto flotante".

Booleanos (bool)

```
activo = True  
esAdulto = False
```

Los booleanos solo pueden tener dos valores: `True` o `False`, y se utilizan para representar condiciones lógicas. Se representan con `bool` debido a que viene del inglés "boolean", en honor al matemático George Boole.

Fun fact: George Boole fue un matemático y lógico inglés que desarrolló la lógica booleana, base de la computación moderna. Esta es fundamental para la programación ya que permite comprender si un hecho es verdadero o falso, lo que es esencial para la toma de decisiones en los programas que veremos más adelante.

Salida de datos

En la programación, la salida de datos se refiere a mostrar información al usuario. A través de una **terminal** o **consola**, en Windows se conoce como **cmd** o **PowerShell** y podemos interactuar con nuestro programa a través de comandos. En Python, utilizamos la función **print()** para mostrar información en la consola.

La función **print()** nos permite mostrar información en la consola:

```
print("Hola, bienvenido a Python")
print(2024)
print(3.14)
print(True)
```

También podemos imprimir múltiples elementos separados por comas, lo que los mostrará con un espacio entre ellos:

```
nombre = "María"
edad = 30
print("Hola", nombre, "tienes", edad, "años")
```

Es necesario recordar que la función **print()** necesita que sus parentesis estén correctamente cerrados y que los elementos a imprimir estén separados por comas. Si olvidamos alguno de estos detalles, el programa mostrará un error.

Operadores matemáticos en Python

Python nos permite realizar operaciones matemáticas fácilmente:

Operador	Descripción	Ejemplo	Resultado
+	Suma	5 + 3	8
-	Resta	10 - 4	6
*	Multiplicación	4 * 2	8
**	Potencia	2 ** 3	8
/	División	7 / 2	3.5
//	División entera	7 // 2	3
%	Módulo (Restante de una división)	7 % 2	1

Entrada de datos

En diversos programas, es común que necesitemos recibir información del usuario. En Python, esto se logra utilizando la función `input()`, que permite capturar datos ingresados por el usuario a través del teclado. Para recibir información del usuario utilizamos la función `input()`:

```
nombre = input("Ingresa tu nombre: ")
print("Hola", nombre)
```

La función `input()` siempre devuelve una cadena de texto, por lo que si necesitamos otro tipo de dato, debemos convertirlo con funciones como `int()` para enteros o `float()` para números decimales. Por ejemplo, si queremos recibir un número entero o decimal del usuario, podemos hacer lo siguiente:

```
edad = int(input("Ingresa tu edad: "))
altura = float(input("Ingresa tu altura en metros: "))
```

Nota: Existe una función llamada `type()` que nos permite conocer el tipo de dato de una variable. Por ejemplo, si queremos saber el tipo de dato de la variable `edad`, podemos hacer lo siguiente:

```
edad = 30
print(type(edad))
```

Ejercicios prácticos

Ejercicio 1: Calculadora básica

Un ejercicio común para practicar los conceptos básicos es crear una calculadora que realice operaciones matemáticas simples. A continuación, se muestra un ejemplo de cómo podría implementarse:

```
# Programa que suma dos números ingresados por el usuario
numero1 = float(input("Ingresa el primer número: "))
numero2 = float(input("Ingresa el segundo número: "))

suma = numero1 + numero2
resta = numero1 - numero2
multiplicacion = numero1 * numero2
division = numero1 / numero2

print("Suma:", suma)
print("Resta:", resta)
print("Multiplicación:", multiplicacion)
print("División:", division)
```

Ejercicio 2: Conversor de temperatura

En diversas partes del mundo, es común utilizar diferentes escalas de temperatura. Un ejercicio interesante es crear un conversor de temperatura entre Celsius y Fahrenheit. Aquí tienes un ejemplo:

```
# Convertir de Celsius a Fahrenheit
celsius = float(input("Ingresa la temperatura en grados Celsius: "))
fahrenheit = (celsius * 9/5) + 32
print(celsius, "grados Celsius equivalen a", fahrenheit, "grados Fahrenheit")
```

Ejercicio 3: Calculadora de área de un rectángulo

Cómo podemos calcular en Python, podemos crear un programa que calcule el área de un rectángulo ya que existe una fórmula matemática para ello. El área de un rectángulo se calcula multiplicando la base por la altura. Aquí tienes un ejemplo:

```
# Calcular el área de un rectángulo
base = float(input("Ingresa la base del rectángulo: "))
altura = float(input("Ingresa la altura del rectángulo: "))
area = base * altura
print("El área del rectángulo es:", area, "unidades cuadradas")
```

Conclusión

En este capítulo hemos aprendido los fundamentos básicos de Python, incluyendo los tipos de datos más comunes, cómo imprimir información en la consola, realizar operaciones matemáticas y recibir datos del usuario. Estos conceptos son esenciales para comenzar a programar en Python y nos servirán como base para desarrollar programas más complejos en el futuro.

Ahora, puedes continuar con el siguiente capítulo, donde exploraremos las estructuras de decisión en Python, que nos permitirán tomar decisiones en nuestros programas basadas en condiciones.

Continuar ➡