🧠 Ejemplo de Tipos de Datos en Python - Guía Paso a Paso

✨ Introducción

En esta guía aprenderás cómo trabajar con los tipos de datos más básicos en Python: enteros, flotantes, cadenas, booleanos y el valor especial `None`. Veremos cómo se almacenan en memoria, cómo se accede a ellos mediante variables y cómo seguir buenas prácticas al nombrarlas.

Vamos a crear un archivo en nuestro proyecto para experimentar con estos conceptos de forma práctica 🧪.

---

1 Crear el archivo del ejemplo

📄 **Archivo en los recursos de esta lección:** `02-09-00-EjemploTiposDatos-UP.py`

📁 Ruta: `Variables/tipos_datos.py`

📓 Descripción

Comenzamos creando un archivo nuevo con un nombre que siga las buenas prácticas: todo en minúsculas y separado por guion bajo.

✅ Nombre del archivo: `tipos_datos.py`
🧠 El nombre usa palabras en minúscula separadas por guion bajo, como si fuera una variable.

---

2 Tipo entero (`int`)

📓 Descripción

Definimos una variable que almacena un número entero.

💻 Código

```
# tipo entero
edad = 30
print(f"Edad: {edad}")
```

🧩 Explicación

Creamos la variable `edad` y le asignamos el valor `30`. Esta variable apunta a un objeto en memoria de tipo entero. Al imprimirla, se muestra el valor almacenado: `30`.

---

3 Tipo flotante (`float`)

📓 Descripción

Ahora vamos a trabajar con un número decimal, que en Python se representa como `float`.

💻 Código

```
# tipo float
altura = 1.75
print(f"Altura: {altura}")
```

🧩 Explicación

La variable `altura` almacena el valor `1.75`. Este valor es de tipo `float` y está almacenado como objeto en la memoria. Al acceder a esta variable, se imprime su valor.

---

4 Tipo cadena (`str`)

📓 Descripción

Definimos una variable que contiene texto, es decir, una cadena de caracteres.

💻 Código

```
# tipo str (cadena)
nombre = "Ana"
print(f"Nombre: {nombre}")
```

🧩 Explicación

La variable `nombre` almacena la cadena `"Ana"`. Al imprimirla, accedemos al objeto de tipo `str` que contiene ese valor.

---

5 Tipo booleano (`bool`)

📓 Descripción

Definimos una variable de tipo booleano que puede ser `True` o `False`.

💻 Código

```
# tipo bool
es_estudiante = False
print(f"¿Es estudiante?: {es_estudiante}")
```

🧩 Explicación

La variable `es_estudiante` almacena el valor `False`, que representa una condición falsa. Al ejecutarse, imprime el valor de esta condición.

---

6 Tipo especial (None)

📓 Descripción

Finalmente, usamos el valor especial `None` para representar ausencia de valor en una variable.

💻 Código

```
# tipo None
direccion = None
print(f"Dirección: {direccion}")
```

🧩 Explicación

Cuando no sabemos qué valor asignar a una variable, podemos usar `None` para indicar ausencia de valor. La variable `direccion` apunta a un objeto de tipo `NoneType`.

📌 Código final por archivo trabajado

```
# Ejemplo de tipos de datos en Python

# Entero
edad = 30
print('Edad:', edad)

# Numero con punto flotante
altura = 1.75
print('Altura:', altura)

# Cadena de texto
nombre = 'Ana'
print('Nombre:', nombre)

# Tipo boolean
es_estudiante = False
print('Es estudiante?', es_estudiante)

# None, ausencia de valor
direccion = None
print('Dirección:',direccion)
```

🎯 Conclusión

En esta guía aprendimos a utilizar los tipos de datos más importantes y fundamentales en Python: `int`, `float`, `str`, `bool` y `None`.
También vimos cómo se almacenan en memoria y cómo acceder a sus valores mediante variables.
Comprender estos conceptos es esencial para dominar el manejo de datos y estructuras más complejas en futuros proyectos 🧠💡.

Sigue adelante con tu aprendizaje 🚀, ¡el esfuerzo vale la pena!

¡Saludos! 👋

Ing. Marcela Gamiño e Ing. Ubaldo Acosta

Fundadores de GlobalMentoring.com.mx