🐍 Clase 03 - Tipos de datos primitivos

🧠 1. Definición Conceptual

> 💡 En Python, **cada valor tiene un tipo de dato** que indica qué tipo de información representa y qué operaciones se pueden realizar con él. Cuando asignamos un valor a una variable, **Python automáticamente determina su tipo** según el valor asignado.

🔷 Principales tipos de datos básicos:

Tipo	Descripción	Ejemplo
`int`	Números enteros (positivos, negativos o cero)	`10` , `-3` , `0`
`float`	Números decimales (de punto flotante)	`3.14` , `-0.5` , `2.0`
`str`	Cadenas de texto (se escriben entre comillas)	`"Hola"` , `'Python'`
`bool`	Valores lógicos (booleanos)	`True` , `False`

🧩 2. Ejemplos básicos de comprensión

🧱 Ejemplo 1 — Identificar tipos de datos con `type()`

```
a = 10
b = 3.14
c = "Hola mundo"
d = True

print(type(a))
print(type(b))
print(type(c))
print(type(d))
```

> La función `type()` devuelve el tipo de dato de cada variable.

📦 Ejemplo 2 — Comprobación con `isinstance()`

```
x ="Python"
print(isinstance(x,str))# True
print(isinstance(x,int))# False
```

> `isinstance(variable, tipo)` devuelve `True` si la variable pertenece al tipo indicado.

📦 Ejemplo 3 — Inferencia automática de tipo

```
edad = 25
precio = 19.99
nombre = "Ana"
activo = False

print(edad,type(edad))
print(precio,type(precio))
print(nombre,type(nombre))
print(activo,type(activo))
```

> Python **infiere automáticamente** el tipo de dato al asignar un valor a la variable.

⚙️ 3. Ejercicios prácticos

🔶 Ejercicio 1 — Evaluar expresiones mixtas

```
a = 10
b = 2.5
c = a + b
print(c,type(c))
```

> Al sumar un `int` y un `float`, Python convierte automáticamente el resultado en `float`.

🔶 Ejercicio 2 — Conversión entre tipos

```
x = 3.99
y = int(x)
print(y,type(y))
```

> La función `int()` convierte un número decimal en entero **truncando la parte decimal**.

🔶 Ejercicio 3 — Evaluar cadenas y números

```
texto = "123"
numero = int(texto)
print(numero + 7)
```

> Se puede convertir una cadena numérica a entero con `int()`, siempre que contenga solo dígitos.

🔶 Ejercicio 4 — Evaluar booleanos en expresiones

```
a = True
b = False
print(a + b)
print(a * 10)
```

> En operaciones numéricas, `True` se interpreta como `1` y `False` como `0`.

Parte 2 Conversión de tipos

🔷 Tipos de datos comunes en Python

Tipo	Ejemplo	Descripción
`int`	`10`	Número entero
`float`	`3.14`	Número decimal
`str`	`"Hola"`	Cadena de texto
`bool`	`True`, `False`	Valor lógico
`list`	`[1, 2, 3]`	Colección ordenada y modificable

Podemos cambiar un tipo de dato a otro mediante funciones integradas:

```
int("45")     # Convierte cadena a entero → 45
float("3.14") # Convierte cadena a flotante → 3.14
str(100)      # Convierte número a texto → "100"
list("Hola")  # Convierte texto a lista de caracteres → ['H', 'o', 'l', 'a']
```

2️⃣ Ejemplos básicos de comprensión (genéricos)

🔶 Ejemplo 1: Conversión de tipos

```
a = "25"
b = 5
```

```
resultado = int(a) + b
print(resultado)
```

> 📘 Convertimos `a` (cadena) a entero antes de sumarla con `b`.

🔶 Ejemplo 2: Indexación y slicing

```
texto = "Python"
print(texto[0]) # Primer carácter
print(texto[-1]) # Último carácter
print(texto[0:3]) # Desde índice 0 hasta antes del 3 → 'Pyt'
```

> 📘 Python indexa desde `0`. Los negativos cuentan desde el final.

🔶 Ejemplo 3: Crear y acceder a listas

```
numeros = [10,20,30,40]
print(numeros[1]) # Acceso al segundo elemento
print(numeros[-2]) # Penúltimo elemento
```

🔶 Ejemplo 4: Modificar listas

```
colores = ["rojo","azul","verde"]
colores[1] = "amarillo"
print(colores)
```

> 📘 **Salida esperada:** `['rojo', 'amarillo', 'verde']`

🔶 Ejemplo 5: Operaciones básicas con listas

```
lista = [5,1,9,3,7]
print(max(lista)) # Máximo → 9
```

```
print(min(lista)) # Mínimo → 1
lista.sort()
print(lista)      # Orden ascendente
lista.reverse()
print(lista)      # Lista invertida
```

3️⃣ Ejercicios prácticos

🧱 Ejemplo 1: Fusión de listas

```
nombres = ["Ana","Luis","Sofía"]
edades = [20,22,19]
fusion = nombres + edades
print(fusion)
```

> 📘 El operador `+` concatena listas.

🧱 Ejemplo 2: Agregar e insertar elementos

```
frutas = ["manzana","pera"]
frutas.append("naranja")      # Agrega al final
frutas.insert(1,"plátano")    # Inserta en posición 1
print(frutas)
```

🧱 Ejemplo 3: Eliminar elementos

```
animales = ["gato","perro","loro","pez"]
animales.remove("loro")                  # Elimina por valor
del animales[0]                          # Elimina por índice
print(animales)
```

🧱 Ejemplo 4: División (slicing)

```
numeros = [10,20,30,40,50,60]
print(numeros[:3])          # Primeros 3
print(numeros[3:])          # Desde el 4to al final
```

🧱 Ejemplo 5: Comparación de listas

```
a = [1,2,3]
b = [1,2,3]
print(a == b)    # True (elementos iguales)
print(a is b)    # False (objetos distintos)
```
