

# Unidad IV: Tipos de Datos

Lcdo. Diego Medardo Saavedra García. Mgtr.

2023-08-31

# Section 1

## Unidad IV: Tipos de Datos

# String y Números

**Objetivo:** Aprender sobre los tipos de datos básicos en Python.

## String (Cadena de Texto)

- Representa texto y se encierra entre comillas simples o dobles.
- Ejemplo: nombre = "María".

```
# Strings  
mensaje = "Hola, bienvenido al curso de Python."  
nombre = 'María'
```

## Números

- Tipos numéricos: entero (int) y flotante (float).
- Ejemplo: edad = 25; altura = 1.75.

```
# Números
```

```
edad = 25
```

```
saldo = 1500.75
```

## Listas

- Estructura que almacena varios elementos en un orden específico.
- Ejemplo: `colores = ["rojo", "verde", "azul"]`.

```
frutas = ["manzana", "banana", "naranja", "uva"]  
primer_fruta = frutas[0]  
segunda_fruta = frutas[1]
```

## Tuplas

- Similar a las listas, pero no se pueden modificar después de creadas.
- Ejemplo: `coordenadas = (10, 20)`.

```
coordenadas = (3, 5)
x = coordenadas[0]
y = coordenadas[1]
```

## Range

- Secuencia de números generados automáticamente.
- Útil para iteraciones.
- Ejemplo: `numeros = range(1, 6)`.

```
# Generación de secuencias de números
```

```
secuencia1 = range(5)           # 0, 1, 2, 3, 4
```

```
secuencia2 = range(2, 10)       # 2, 3, 4, 5, 6, 7, 8, 9
```

```
secuencia3 = range(1, 11, 2)    # 1, 3, 5, 7, 9
```

```
# Conversión a lista
```

```
lista_secuencia1 = list(secuencia1)
```



## Diccionarios

- Estructura clave-valor para almacenar información.
- Ejemplo: `persona = {"nombre": "Carlos", "edad": 30}`.

```
# Creación de un diccionario
```

```
persona = {  
    "nombre": "Juan",  
    "edad": 30,  
    "ciudad": "México"  
}
```

```
# Acceso a valores utilizando claves
```

```
nombre = persona["nombre"]  
edad = persona["edad"]
```

## Booleanos

- Tipo de dato para representar verdadero (True) o falso (False).
- Útil en expresiones condicionales.
- Ejemplo: `es_mayor_de_edad = True`.

```
# Variables booleanas
```

```
es_mayor_de_edad = True
```

```
tiene_tarjeta = False
```

```
# Expresiones lógicas
```

```
puede_ingresar = es_mayor_de_edad and tiene_tarjeta
```

## Ejemplo Práctico

Crea una lista de tareas pendientes e imprime cada tarea en una línea.

```
# Lista de tareas pendientes
tareas = ["Sacar la basura", "Barrer la entrada", "Pasear al perro"]

# Imprimir cada tarea en una línea
for tarea in tareas:
    print(tarea)
```

# Referencias:

- Python.org. (2023). Built-in Types.  
<https://docs.python.org/3/library/stdtypes.html>
- Downey, A. B. (2015). Think Python. O'Reilly Media.