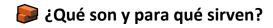
## TECNICATURA UNIVERSITARIA EN PROGRAMACIÓN A DISTANCIA



# Tuplas y Sets en Python



Las **tuplas** y los **sets** son dos estructuras de datos súper útiles de Python que te ayudan a organizar y manejar datos de manera eficiente.

# **1.** Tuplas: el paquete inmutable

# 🔍 ¿Qué es una tupla?

Una **tupla** es una colección **ordenada** de elementos que **no se pueden modificar** después de su creación. Es como un combo cerrado de comida : juna vez que lo pediste, no se cambia nada!

## ¿Cuándo usar una tupla?

- Cuando necesitás proteger tus datos para que no sean modificados.
- Cuando querés usar una estructura como clave de un diccionario (porque las tuplas son hashables).
- Para representar datos que siempre vienen juntos (por ejemplo, coordenadas, fechas, etc.).

## Ejemplo básico:

coordenada = (10, 20)

print(coordenada[0]) # Imprime: 10

# ⚠ Importante:

- No podés hacer coordenada[0] = 15 💢
- Las tuplas pueden contener diferentes tipos de datos.

# ✓ Tip:

Si querés crear una tupla de un solo elemento, ¡no te olvides de la coma!

tupla\_uno = (42,) # Esto sí es una tupla

# Métodos útiles de las tuplas:

## TECNICATURA UNIVERSITARIA EN PROGRAMACIÓN A DISTANCIA



```
mi_tupla = (1, 2, 3, 2)
print(mi_tupla.count(2)) # Cuenta cuántas veces aparece el 2
print(mi_tupla.index(3)) # Devuelve la posición del 3
```

# Desempaquetado de tuplas:

¡Podés asignar sus valores a variables en un solo paso!

x, y = (10, 20)

print(x) # 10

print(y) # 20

# 2. Sets: el club exclusivo

# Qué es un set?

Un **set** (conjunto) es una colección **no ordenada**, **sin elementos repetidos**. Como un club VIP: si ya estás adentro, no te dejan entrar de nuevo

# ¿Cuándo usar un set?

- Para eliminar duplicados.
- Para hacer operaciones matemáticas como unión, intersección, diferencia.
- Cuando no te importa el orden de los elementos.

### Ejemplo básico:

numeros = {1, 2, 3, 2, 1} print(numeros) # Imprime: {1, 2, 3}

# ♠ Importante:

- No tiene índices → no podés hacer numeros[0]
- No mantiene el orden de inserción.

# **Métodos útiles de sets:**

## **TECNICATURA UNIVERSITARIA EN PROGRAMACIÓN A DISTANCIA**



```
a = \{1, 2, 3\}
```

$$b = \{3, 4, 5\}$$

```
print(a.union(b))
                    # {1, 2, 3, 4, 5}
print(a.intersection(b)) # {3}
print(a.difference(b)) # {1, 2}
```

# Agregar y eliminar elementos:

```
colores = {"rojo", "azul"}
colores.add("verde")
colores.remove("rojo")
print(colores) # {'azul', 'verde'}
```

# **Conclusión**

Característica	Tupla	Set
Ordenado	✓ Sí	<b>X</b> No
Elementos únicos	<b>X</b> No	✓ Sí
Mutable	X No (inmutable)	✓ Sí
Acceso por índice	✓ Sí	<b>X</b> No
Uso ideal	Datos fijos, coordenadas	Eliminar duplicados, operaciones de conjuntos

월 "Las tuplas te dan seguridad, los sets te dan limpieza. ¡Usalos con sabiduría y tu código será imparable!"