

## 2. Listas Y Tuplas

Listas se definen con corchetes [ ] y se pueden modificar despues de su creaccion

Listas nos permiten almacenar una serie de elementos an una sola variable



Las tuplas se crean utilizando paréntesis ( )





Las tuplas son una estructura de datos similar a las listas, pero con una diferencia clave: **son inmutables**.

Las listas nos permiten realizar operaciones como ordenamiento, filtrado y cálculos estadísticos en conjuntos de datos, mientras que las tuplas nos proporcionan una estructura segura y estable para almacenar información que no debe modificarse.




### Cuadro de métodos – Listas vs Tuplas

 Método	 Listas	 Tuplas	 Descripción / Ejemplo
<code>append(x)</code>	✅ Sí	❌ No	Agrega un elemento al final. <code>frutas.append("uva")</code> → <code>["manzana", "pera", "uva"]</code>
<code>insert(i, x)</code>	✅ Sí	❌ No	Inserta un elemento en la posición indicada. <code>frutas.insert(1, "kiwi")</code> → <code>["manzana", "kiwi", "pera"]</code>
<code>remove(x)</code>	✅ Sí	❌ No	Elimina el primer valor igual a <code>x</code> . <code>frutas.remove("pera")</code>
<code>pop([i])</code>	✅ Sí	❌ No	Elimina y devuelve el elemento en la posición <code>i</code> (o el último si no se indica).
<code>clear()</code>	✅ Sí	❌ No	Elimina todos los elementos de la lista.
<code>sort()</code>	✅ Sí	❌ No	Ordena los elementos (alfabéticamente o numéricamente).
<code>reverse()</code>	✅ Sí	❌ No	Invierte el orden de los elementos.
<code>copy()</code>	✅ Sí	❌ No	Crea una copia independiente de la lista.
<code>count(x)</code>	✅ Sí	✅ Sí	Cuenta cuántas veces aparece <code>x</code> . <code>[1, 2, 2, 3].count(2)</code> → 2
<code>index(x)</code>	✅ Sí	✅ Sí	Devuelve el índice de la primera aparición de <code>x</code> . <code>["a", "b", "c"].index("b")</code> → 1

 Método	 Listas	 Tuplas	 Descripción / Ejemplo
<code>len()</code>	✅ Sí	✅ Sí	Devuelve la cantidad de elementos. <code>len(frutas)</code>
<code>sum()</code>	✅ Sí	✅ Sí	Suma los valores numéricos. <code>sum([1,2,3])</code> → 6
<code>min()</code>	✅ Sí	✅ Sí	Devuelve el valor mínimo.
<code>max()</code>	✅ Sí	✅ Sí	Devuelve el valor máximo.
<code>sorted()</code>	✅ Sí	✅ Sí	Devuelve una <b>nueva lista ordenada</b> (sin modificar la original).
<code>tuple()</code>	✅ (para convertir lista en tupla)	—	Convierte una lista en tupla. <code>tuple([1,2,3])</code> → (1,2,3)
<code>list()</code>	—	✅ (para convertir tupla en lista)	Convierte una tupla en lista. <code>list((1,2,3))</code> → [1,2,3]

## Notas rápidas:

- ✅ Las **listas** son **mutables** → se pueden modificar.
-  Las **tuplas** son **inmutables** → no se pueden cambiar.
- Ambos tipos permiten recorrer elementos con `for`, y acceder mediante índices `[]`.

¿Quieres que te haga este cuadro en **PDF** o en **formato imagen** para que lo uses como ficha visual de repaso?

## Módulo 1 – Lección 02: Tuplas

### Qué es una tupla

- Una **tupla** es una colección **ordenada** e **inmutable** de elementos.
- Se define usando **paréntesis** `()` en lugar de corchetes.
- Se pueden almacenar distintos tipos de datos (números, strings, etc.).

```
mi_tupla = (1, 2, 3)
tupla_mixta = (1, "hola", 3.5, True)
```

## Acceso a elementos

- Igual que las listas, los índices comienzan en **0**.

```
mi_tupla = (10, 20, 30, 40)
print(mi_tupla[0])    # 10
print(mi_tupla[-1])   # 40
```

## Inmutabilidad

Las tuplas **no pueden modificarse**:

```
mi_tupla[0] = 99    # ❌ Error: TypeError
```

## Desempaquetado de tuplas

Permite asignar los valores de una tupla a variables:

```
persona = ("Ana", 25, "España")
nombre, edad, pais = persona

print(nombre)    # Ana
print(edad)      # 25
print(pais)      # España
```

## Métodos útiles de las tuplas

Método	Descripción	Ejemplo	Resultado
count(x)	Cuenta cuántas veces aparece x .	(1,2,2,3).count(2)	2

Método	Descripción	Ejemplo	Resultado
<code>index(x)</code>	Devuelve el índice de la primera aparición de <code>x</code> .	<code>(1,2,3).index(3)</code>	2

---

## Conversión entre tipos

- Se puede convertir una tupla en lista y viceversa:

```
lista = [1, 2, 3]
tupla = tuple(lista)
nueva_lista = list(tupla)
```

---

## Ventajas de las tuplas

- Más rápidas que las listas.
- Pueden usarse como **claves en diccionarios**.
- Útiles cuando los datos **no deben cambiar**.

---

## Módulo 1 – Lección 02: Listas

### Qué es una lista

- Una **lista** es una colección **ordenada y mutable**.
- Se definen con **corchetes** `[]`.

```
frutas = ["manzana", "banana", "pera"]
numeros = [1, 2, 3, 4]
mixta = [1, "hola", True, 3.14]
```

---

## Acceso a elementos

```
print(frutas[0])    # manzana
print(frutas[-1])   # pera
```

## Modificar elementos

```
frutas[1] = "sandía"
print(frutas)    # ['manzana', 'sandía', 'pera']
```

## + Añadir y eliminar elementos

Acción	Método	Ejemplo	Resultado
Agregar al final	<code>.append()</code>	<code>frutas.append("uva")</code>	<code>["manzana", "sandía", "pera", "uva"]</code>
Insertar en posición	<code>.insert(pos, elem)</code>	<code>frutas.insert(1, "kiwi")</code>	<code>["manzana", "kiwi", "sandía", "pera"]</code>
Eliminar por valor	<code>.remove(valor)</code>	<code>frutas.remove("pera")</code>	<code>["manzana", "sandía"]</code>
Eliminar último	<code>.pop()</code>	<code>frutas.pop()</code>	elimina el último
Vaciar lista	<code>.clear()</code>	<code>frutas.clear()</code>	<code>[]</code>

## Otras operaciones

Operación	Ejemplo	Resultado
Concatenar	<code>[1,2] + [3,4]</code>	<code>[1,2,3,4]</code>
Repetir	<code>[0] * 3</code>	<code>[0,0,0]</code>
Sublistas	<code>frutas[1:3]</code>	extrae una porción
Ver longitud	<code>len(frutas)</code>	cantidad de elementos



## Ordenar y revertir

```
numeros = [3, 1, 4, 2]
numeros.sort()      # [1, 2, 3, 4]
numeros.reverse()   # [4, 3, 2, 1]
```

---



## Métodos importantes

Método	Ejemplo	Resultado
<code>.count(x)</code>	<code>[1,2,2,3].count(2)</code>	2
<code>.index(x)</code>	<code>["a","b","c"].index("b")</code>	1
<code>.copy()</code>	<code>nueva = frutas.copy()</code>	copia la lista

---



## Funciones útiles con listas

```
min(lista)
max(lista)
sum(lista)
sorted(lista)
```

---