



## Índice

<b>Índice</b> .....	<b>1</b>
Ejercicio 1: Creación de una lista.....	2
Ejercicio 2: Acceso a elementos de la lista.....	2
Ejercicio 3: Modificación de una lista.....	3
Ejercicio 4: Agregar elementos a una lista.....	3
Ejercicio 5: Eliminar elementos de una lista.....	4
Ejercicio 6: Ordenar una lista.....	4
Ejercicio 7: Slicing (rebanado) de listas.....	5
Ejercicio 8: Listas dentro de listas (listas anidadas).....	5
Ejercicio 9: Comprobar si un elemento está en una lista.....	6
Ejercicio 10: Recorrer una lista con un bucle.....	6



## Ejercicio 1: Creación de una lista

**Objetivo:** Familiarizarse con la creación y visualización de listas.

1. Crea una lista llamada `mis_numeros` que contenga los números: 2, 4, 6, 8, 10.
2. Imprime la lista.

```
# Tu código aquí
mis_numeros = [2, 4, 6, 8, 10]
print(mis_numeros)
```

**Pregunta guiada:**

- ¿Qué sucede si imprimes una lista vacía? Inténtalo con `mi_lista_vacia = []`.
- 

## Ejercicio 2: Acceso a elementos de la lista

**Objetivo:** Practicar cómo acceder a elementos individuales y usando índices.

1. Imprime el primer y el último número de la lista `mis_numeros`.
2. Intenta acceder a un índice que esté fuera del rango (por ejemplo, `mis_numeros[10]`) y observa el error.

```
# Acceder a elementos de la lista
print(mis_numeros[0]) # Primer elemento
print(mis_numeros[-1]) # Último elemento

# ¿Qué ocurre aquí?
# print(mis_numeros[10]) # Índice fuera de rango
```

**Pregunta guiada:**

- ¿Qué sucede si intentas acceder al índice -2? ¿Qué valor obtienes?
-



### Ejercicio 3: Modificación de una lista

**Objetivo:** Aprender cómo modificar valores de una lista.

1. Cambia el tercer valor de `mis_numeros` (actualmente 6) a 7.
2. Imprime la lista actualizada.

```
# Modificar un valor en la lista
mis_numeros[2] = 7
print(mis_numeros)
```

**Pregunta guiada:**

- ¿Qué ocurre si intentas modificar un índice fuera de rango?
- 

### Ejercicio 4: Agregar elementos a una lista

**Objetivo:** Conocer las formas de agregar elementos a una lista (`append` y `extend`).

1. Agrega el número 12 al final de la lista usando el método `append()`.
2. Agrega los números 14 y 16 de una sola vez usando el método `extend()`.
3. Imprime la lista para ver los cambios.

```
# Agregar elementos a la lista
mis_numeros.append(12)
print(mis_numeros) # [2, 4, 7, 8, 10, 12]

mis_numeros.extend([14, 16])
print(mis_numeros) # [2, 4, 7, 8, 10, 12, 14, 16]
```

**Pregunta guiada:**

- ¿Qué diferencia hay entre `append()` y `extend()`?
-



## Ejercicio 5: Eliminar elementos de una lista

**Objetivo:** Practicar la eliminación de elementos de una lista.

1. Elimina el número 8 de la lista usando `remove()`.
2. Elimina el último número de la lista usando `pop()`.
3. Imprime la lista actualizada.

```
# Eliminar elementos de la lista
mis_numeros.remove(8)
print(mis_numeros) # [2, 4, 7, 10, 12, 14, 16]

mis_numeros.pop() # Elimina el último valor
print(mis_numeros) # [2, 4, 7, 10, 12, 14]
```

**Pregunta guiada:**

- ¿Qué pasa si intentas eliminar un número que no está en la lista?
- 

## Ejercicio 6: Ordenar una lista

**Objetivo:** Aprender a ordenar los elementos de una lista.

1. Crea una nueva lista llamada `numeros_desordenados` con los valores [9, 3, 5, 1, 4, 8].
2. Ordena la lista en orden ascendente usando `sort()` y luego imprímela.
3. Ordena la lista en orden descendente.

```
# Crear y ordenar una lista
numeros_desordenados = [9, 3, 5, 1, 4, 8]

numeros_desordenados.sort() # Orden ascendente
print(numeros_desordenados) # [1, 3, 4, 5, 8, 9]

numeros_desordenados.sort(reverse=True) # Orden descendente
print(numeros_desordenados) # [9, 8, 5, 4, 3, 1]
```

**Pregunta guiada:**

- ¿Cómo ordenarías una lista de cadenas de texto?

## Ejercicio 7: Slicing (rebanado) de listas

**Objetivo:** Practicar el slicing para obtener partes de una lista.

1. Crea una lista llamada `letras` con las letras `['a', 'b', 'c', 'd', 'e', 'f', 'g']`.
2. Imprime los primeros 3 elementos.
3. Imprime los últimos 2 elementos.
4. Imprime la lista al revés usando slicing.

```
# Slicing de listas
letras = ['a', 'b', 'c', 'd', 'e', 'f', 'g']

print(letras[:3]) # Primeros 3 elementos
print(letras[-2:]) # Últimos 2 elementos
print(letras[::-1]) # Lista al revés
```

**Pregunta guiada:**

- ¿Qué sucede si haces `letras[:2]`? ¿Qué elementos se obtienen?
- 

## Ejercicio 8: Listas dentro de listas (listas anidadas)

**Objetivo:** Introducir listas dentro de otras listas (listas anidadas).

1. Crea una lista llamada `matriz` que contenga las siguientes tres listas: `[1, 2, 3]`, `[4, 5, 6]` y `[7, 8, 9]`.
2. Imprime el segundo elemento de la segunda lista dentro de `matriz`.
3. Imprime toda la lista `matriz` para visualizar su estructura.

```
# Listas anidadas
matriz = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]

print(matriz[1][1]) # Segundo elemento de la segunda lista
print(matriz)
```

**Pregunta guiada:**

- ¿Cómo accederías al primer elemento de la tercera lista?

## Ejercicio 9: Comprobar si un elemento está en una lista

**Objetivo:** Aprender a usar el operador `in` para comprobar si un elemento está en una lista.

1. Crea una lista de frutas llamada `frutas` con los valores: `['manzana', 'banana', 'naranja', 'uva']`.
2. Comprueba si la palabra `"pera"` está en la lista de frutas.
3. Comprueba si la palabra `"banana"` está en la lista.

```
# Verificar si un elemento está en una lista
frutas = ['manzana', 'banana', 'naranja', 'uva']

print('pera' in frutas) # False
print('banana' in frutas) # True
```

**Pregunta guiada:**

- ¿Qué pasa si buscas `"Banana"` con mayúscula? ¿Es diferente a `"banana"`?
- 

## Ejercicio 10: Recorrer una lista con un bucle

**Objetivo:** Practicar cómo recorrer una lista con un bucle `for`.

1. Crea una lista de nombres: `['Ana', 'Luis', 'Pedro', 'Sofía']`.
2. Recorre la lista con un bucle `for` e imprime cada nombre con un saludo, por ejemplo, `"Hola, Ana"`.

```
# Recorrer una lista con un bucle
nombres = ['Ana', 'Luis', 'Pedro', 'Sofía']

for nombre in nombres:
    print(f"Hola, {nombre}")
```

**Pregunta guiada:**

- ¿Cómo harías para recorrer una lista al revés usando un bucle?
-