

DÍA 4 – LISTAS Y BUCLES EN PYTHON

Objetivo del día

- Entender qué es una **lista** y para qué sirve
 - Manipular listas: acceder, modificar, agregar y eliminar elementos
 - Aprender los bucles **for** y **while** para recorrer listas
 - Aplicar todo junto en ejercicios prácticos + repaso global
-

Parte 1: LISTAS – Tu mochila de Python

Una **lista** es como una **mochila** donde guardas cosas: puede contener números, textos, o incluso otras listas.

```
mochila = ["cuaderno", "bolígrafo", "bocadillo"]
```

Puedes acceder a lo que hay dentro con su **posición (index)**:

```
print(mochila[0]) # cuaderno
print(mochila[-1]) # bocadillo (desde el final)
```

Operaciones básicas con listas

```
# Crear lista vacía
cosas = []

# Añadir elementos
cosas.append("llaves")
cosas.append("teléfono")

# Cambiar un valor
cosas[1] = "móvil"

# Eliminar
cosas.remove("llaves")

# Tamaño de la lista
print(len(cosas))
```

Analogía: la lista es una caja, **append()** mete cosas, **remove()** saca cosas, y **len()** cuenta cuántas cosas hay.

Parte 2: BUCLE FOR

Un **bucle for** repite una acción para cada elemento de una lista. Es como decir:

Por cada cosa en mi mochila, haz esto.

```
mochila = ["libro", "agua", "sandwich"]

for objeto in mochila:
    print("Tienes:", objeto)
```

Visual:

for = bucle = cinta transportadora
cada elemento = una caja sobre la cinta

Parte 3: BUCLE WHILE

while se repite **mientras** se cumpla una condición. Ideal cuando no sabes cuántas veces se repetirá.

```
contador = 0

while contador < 5:
    print("Contando:", contador)
    contador += 1
```

Analogía:

while

es como un reloj de arena que sigue girando mientras haya arena arriba (condición True).

Parte 4: Recorrer listas con for + if

```
numeros = [4, 7, 2, 8, 5]

for n in numeros:
    if n % 2 == 0:
        print(n, "es par")
    else:
        print(n, "es impar")
```

Ejercicios paso a paso (con explicación guiada)

Ejercicio 1: Mostrar mochila

```
mochila = ["mapa", "brújula", "botella"]
for item in mochila:
    print("Llevas:", item)
```

Ejercicio 2: Sumar elementos de una lista

```
numeros = [1, 2, 3, 4]
suma = 0
for n in numeros:
    suma += n
print("Suma total:", suma)
```

Ejercicio 3: Contar números mayores que 5

```
numeros = [3, 7, 2, 9, 4]
contador = 0
for n in numeros:
    if n > 5:
        contador += 1
print("Mayores que 5:", contador)
```

5 EJERCICIOS NUEVOS DE HOY (de menos a más)

1. Crear una lista con 3 frutas y mostrarlas una por una
2. Pedir 5 números al usuario y guardarlos en una lista
3. Calcular el promedio de una lista de números
4. Mostrar los elementos de una lista en orden inverso
5. Dada una lista de edades, contar cuántos son mayores de edad (≥ 18)

REPASO GLOBAL (Funciones + Condicionales + Listas + Bucle)

1. funcion crear_lista() que pida 3 elementos por input y devuelva lista
2. funcion es_par(n): devuelve True si n es par, usando return
3. funcion mayores(lista): recibe lista de edades y devuelve solo los ≥ 18
4. funcion contar_aprobados(lista): cuenta notas ≥ 5
5. funcion resumen(nombre, edad, lista_compras): muestraperfil personalizado cn resumen d compra

FRASE PARA RECORDAR

Las listas guardan datos. Los bucles los hacen bailar. Las funciones los orquestan.