

✓ Estructuras de repetición

Se emplean cuando se desea repetir una parte del código, dada una condición.

✓ Listas

- Son un conjunto de tipo de datos,
- *indexado**, separado por comas, y encerrado entre [].
- Todos los elementos dentro de una lista, se empiezan a contar desde 0.

Creación de listas

- Forma general: lista=[]
- Forma manual: animales= ["gato" "perro", zorro, "oso"]
- Por conversión: número=list(range(10))

```
1 lista=[]
```



```
1 type(lista)
```

⇒ list

```
1 zorro= (1,2,3,4)
2 animales=["gato", "perro", zorro, "oso"]
3 animales
```

⇒ ['gato', 'perro', (1, 2, 3, 4), 'oso']

```
1 type(zorro)
```

⇒ tuple

```
1 número=list(range(10))  
2 número
```

⇒ [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]

```
1 range(10)
```

⇒ range(0, 10)

✓ Estructura para-for

Se usa cuando

- Se sabe **cuántas veces** se repite una acción.
- Está compuesta por un contador y una condición.
- **Contador**: Identifica cuántas veces se va a repetir el bucle y con qué frecuencia. v
- **Condición**: Evalúa cuándo se detiene.

Por definición:

```
for variable in range(vi, vf+1, paso):  
aquí se repite parte del código.
```

Donde

1. vi: Condición inicial. Dónde arranco a contar.
2. vf+1: Condición final. Dónde dejo de contar+1
3. paso: Frecuencia de conteo

for para recorrido de listas

```
flor=["amapola", "rosa", "tulipán", "orquidea"]
```

- El para va a iterar tantas veces como elementos tenga la lista.
- La variable dentro del para, toma como valor, cada elemento de la lista

```
for variable in lista:  
variable
```

```

1 #Creando una lista de nombre flor
2 flor=["amapola", "rosa", "tulipán", "orquídea"]
3 flores=5
4 for flores in flor:
5     print(flores)
6     if flores=="tulipán":
7         break

```

⇒ amapola
 rosa
 tulipán
 orquídea

##Estructura while

- * Se emplea cuando ****NO SE SABE**** cuántas veces se debe repetir una acción, pero si se conoce su condición de parada (CP).
- * Condición inicial: Dónde arranca el ciclo
- * Condición final (parada):Que indica dónde se detiene el ciclo
- * Modificador de condición inicial

Estructura general del mientras

```

Condición inicial

```

while (condición==condición final):
modificador de condición inicial
```

```

Modificadores de condición inicial

- * Banderas: son booleanos
- * Son preguntas
- * Son contadores o acumuladores



Estructura while

- Se emplea cuando **NO SE SABE** cuántas veces se debe repetir una acción, pero si se conoce su condición de parada (CP).
- Condición inicial: Dónde arranca el ciclo
- Condición final (parada): Que indica dónde se detiene el ciclo
- Modificador de condición inicial

Estructura general del mientras

```
Condición inicial  
while (condición==condición final):  
modificador de condición inicial
```

Modificadores de condición inicial

- Banderas: son booleanos
- Son preguntas
- Son contadores o acumuladores

1 Comienza a programar o generar con IA.

