



Módulo 2 – Fundamentos de Programación Python

Ciclos Iterativos

Objetivos



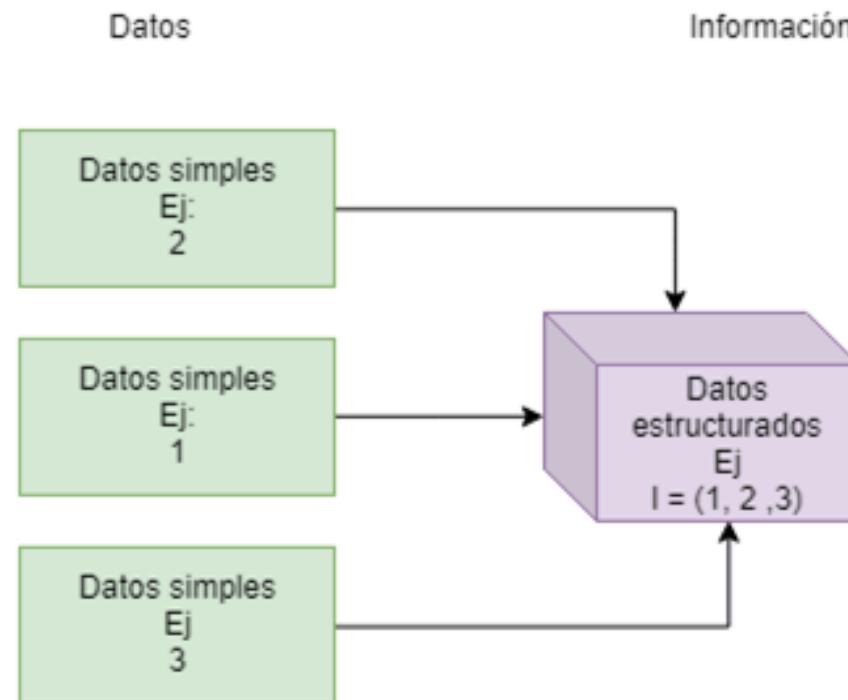
- Utilizar sentencias repetitivas para resolver un problema.

Contenido

1. Sentencias cíclicas



Estructuras de Datos en Python

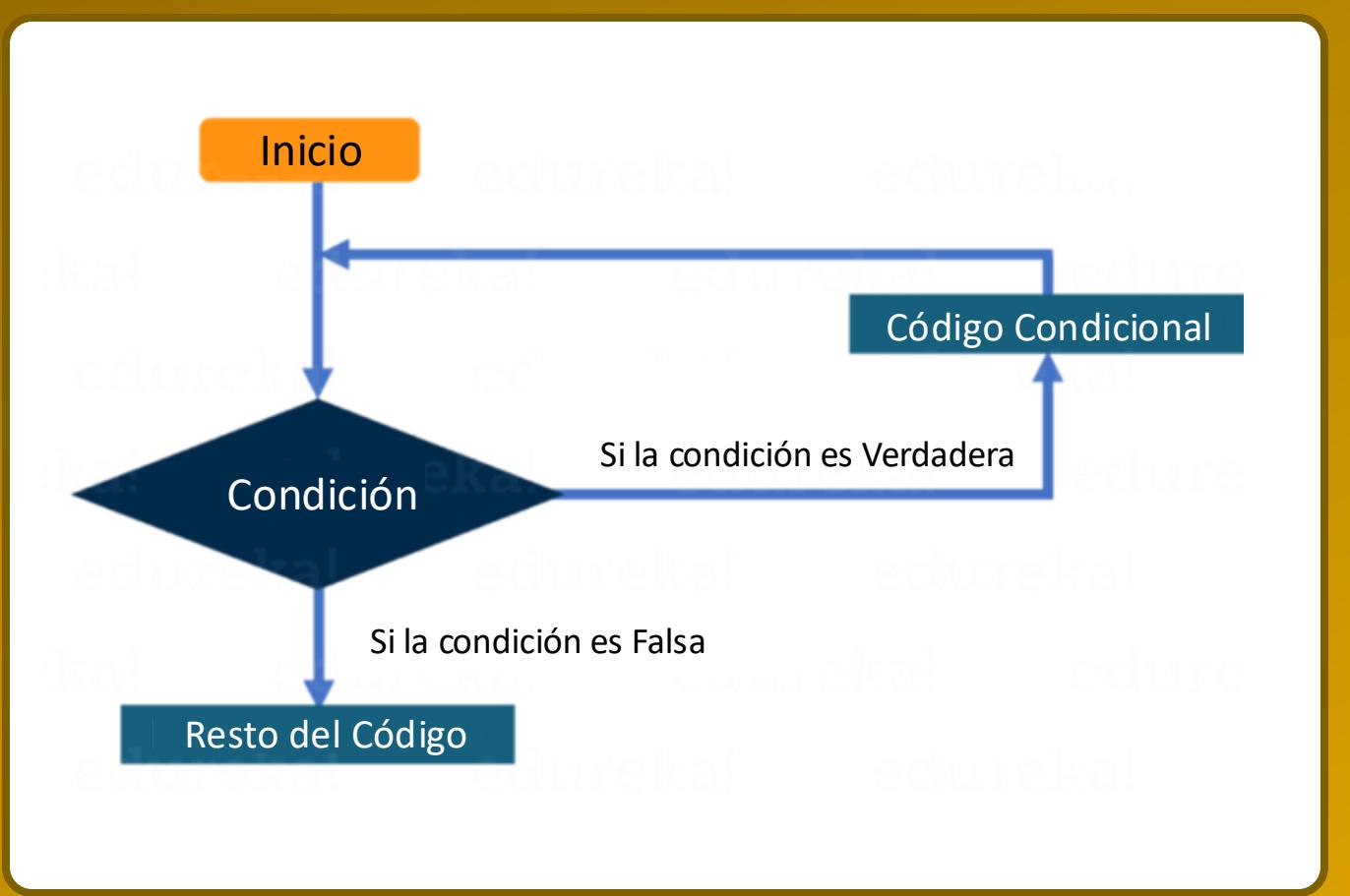


Ciclos/Bucles

Un bucle o ciclo en programación es la ejecución continua de un determinado bloque de código mientras una condición asignada se cumple.

- **Finitos:** Se ejecutan hasta que se cumple cierta condición.
- **Infinitos:** trabajan infinitamente ya que no alcanzan nunca a cumplir la condición.

Python: `for` y `while`



Ciclos

- ➊ **For:** Ejecuta ciertas acciones hasta que la condición ya no se cumpla. **Ciclo Pre-test.**

```
numeros = [1, 2, 3, 4, 5, 6, 7, 8, 9  
, 10, 11, 12, 13, 14] #Creamos la lista con números  
for num in numeros: #En la variable "num" almacenamos los ítem de la lista  
    if num % 2 == 0: #Condición: Si el resto de la división por dos es cero entonces:  
        print (num) # Imprimimos la variable num
```

Resultado: 2, 4, 6, 8, 10, 12, 14

Ciclo

- ➊ **While** : Es un ciclo que se ejecuta similar a **For** pero con la excepción de que no sabes cuántas iteraciones durará la condición.

Post-test ciclo

Sintaxis

```
while condicion:  
    # instrucciones
```

Ejemplo

```
segundos = 10  
while segundos >= 0:  
    print(segundos, end=' -> ')  
    segundos -= 1  
print('Boom!')  
  
10 -> 9 -> 8 -> 7 -> 6 -> 5 -> 4 -> 3 -> 2  
-> 1 -> 0 -> Boom!
```

Break/Continue

- ④ **break:** hace que la ejecución del ciclo finalice y que se continúe con la ejecución de la siguiente instrucción después del ciclo.
- ④ **continue:** hace que se descarte la iteración en curso y el flujo de programa “salte” a la siguiente iteración.

```
names = ["Jimmy", "Rose", "Max", "Nina", "Phillip"]

for name in names: ←

    if len(name) != 4:
        continue →

    print(f"Hello, {name}")

    if name == "Nina":
        break →

    print("Done!")
```

Dudas y consultas



KIBERNUM



Fin presentación