

# Estructuras repetitivas For - Python



Ing. Javier Calli Olvea

## ESTRUCTURAS REPETITIVAS



Las estructuras repetitivas se utilizan para realizar un determinado tipo de instrucciones en un número finito de veces. Se caracterizan por tener un punto inicial de partida, una condición la cual se encarga de ejecutar un número determinado de acciones hasta que esta condición no sea válida (no se cumpla).



## Bucle for

En general, un bucle es una estructura de control que repite un bloque de instrucciones. Un bucle for es un bucle que repite el bloque de instrucciones un número predeterminado de veces.

El bloque de instrucciones que se repite se suele llamar cuerpo del bucle y cada repetición se suele llamar iteración.

La sintaxis de un bucle for es la siguiente:

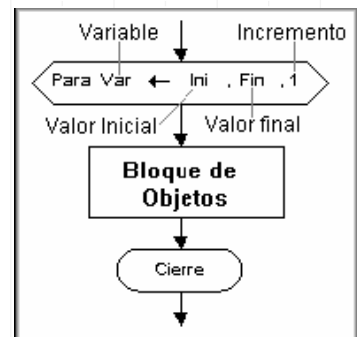
3

## Bucle for



**for** **variable** **in** elemento\_iterable: (lista, cadena, range, etc.)

cuerpo del bucle



4



## Contador

Se entiende por contador una variable que lleva la cuenta del número de veces que se ha cumplido una condición.

```
Acumulador ← Acumulador + cantidad
```

```
Contador ← Contador + 1
```

5



## Ejemplo: (Lista)

```
num = [4, 78, 9, 84]  
for n in num:  
    print(n)
```

```
>> 4  
>> 78  
>> 9  
>> 84
```

6



## Ejemplo: (Cadena)

```
for i in "Python":  
    print(i)
```

```
>> P  
>> y  
>> t  
>> h  
>> o  
>> n
```

7



## Ejemplo: (Cadena)

```
texto = "Python"  
for i in texto[::-1]:  
    print(i)
```

```
>> n  
>> o  
>> h  
>> t  
>> y  
>> P
```

```
texto = "Python"  
for i in texto[:2]:  
    print(i)
```

```
>> P  
>> t  
>> o
```

8



## Ejemplo: range (inicio, fin, salto)

```
for i in (0, 1, 2, 3, 4, 5):  
    print(i)
```

```
>> 0, 1, 2, 3, 4, 5
```

```
for i in range(6):  
    print(i)
```

```
>> 0, 1, 2, 3, 4, 5
```

```
for i in range(8, 12):  
    print(i)
```

```
>> 8,9,10,11
```

```
for i in range(5, 20, 2):  
    print(i)
```

```
>> 5,7,9,11,13,15,17,19
```

9



## Ejemplo: range (inicio, fin, salto)

```
for i in range (5, 0, -1):  
    print(i)
```

```
>> 5,4,3,2,1
```

10



## Ejemplo: For anidados

```
lista = [[56, 34, 1],
         [12, 4, 5],
         [9, 4, 3]]

for i in lista:
    print(i)
```

```
>> [56, 34, 1]
>> [12, 4, 5]
>> [9, 4, 3]
```

Si iteramos usando sólo un for, estaremos realmente accediendo a la segunda lista, pero no a los elementos individuales.

11



## Ejemplo: For anidados

```
lista = [[56, 34, 1],
         [12, 4, 5],
         [9, 4, 3]]

for i in lista:
    for j in i:
        print(j)
```

```
>> 56
>> 34
>> 1
>> 12
>> 4
>> 5
>> 9
>> 4
>> 3
```

Para acceder a cada elemento individualmente, podemos anidar dos for. Uno de ellos se encargará de iterar las columnas y el otro las filas.

12



## Bibliografía.

<https://uniwebsidad.com/libros/python>

<https://www.freecodecamp.org/espanol/news/sentencia-if-else-de-python-explicacion-de-las-sentencias-condiciones/>

<https://www.freecodecamp.org/espanol/news/sentencias-if-elif-y-else-en-python/>



13



Javier Calli Olvea.



javier422@gmail.com

# Gracias!

14