



## Ciclo 1

# Semana 4

## Estructuras de control: repetición o iteración

---

### Lectura 3 – Sentencia for



## | Sentencia for

Algunas veces es necesario utilizar un ciclo para recorrer uno a uno los elementos de una secuencia, por ejemplo, una lista de palabras, las líneas de un archivo, o una lista de números. En estos casos se puede utilizar la sentencia **for**. A la sentencia **while** se le conoce como bucle indefinido, porque se repite hasta que cierta condición se hace falsa, mientras que el **for** es un ciclo definido porque normalmente se repite a través de un conjunto conocido de elementos, de tal forma que se ejecuta tantas veces como elementos haya en el conjunto.

La sintaxis de este ciclo es similar al **while**, tiene la palabra reservada **for**, la secuencia a recorrer y un cuerpo de instrucciones que se repite y que debe estar indentado a la derecha. Por ejemplo:

```
amigos = ['Joseph', 'Glenn', 'Sally']  
for amigo in amigos:  
    print('Feliz año nuevo:', amigo)  
print('¡Terminado!')
```

```
Feliz año nuevo: Joseph  
Feliz año nuevo: Glenn  
Feliz año nuevo: Sally  
¡Terminado!  
  
Process finished with exit code 0
```

En términos del lenguaje Python la variable amigos es una lista de tres cadenas de texto y el ciclo **for** la recorre y ejecuta su cuerpo de instrucciones, una vez para cada una de ellas. Si se piensa en la variable amigos como un conjunto, se puede entender el **for** como si se le dijera al computador “Ejecute este ciclo una vez para (**for**) cada amigo que esté en (**in**) el conjunto llamado amigos”. En este ejemplo **for** e **in** son palabras reservadas en cambio amigo y amigos son nombres de variables, amigo una cadena de texto y amigos un conjunto de cadenas de texto. Para este ejemplo amigo es la variable de iteración del **for**, cambia en cada iteración del ciclo y

controla cuando se termina. Esta variable se desplaza sucesivamente a través de cada una de las cadenas de texto de la variable amigos.

```
for i in "AMIGO":  
    print("Dame una ", i)  
print("¡AMIGO!")
```

```
Dame una  A  
Dame una  M  
Dame una  I  
Dame una  G  
Dame una  O  
¡AMIGO!
```

En muchos otros lenguajes de programación la estructura **for** tiene la forma:

*for i = from 0 to 10*

cuerpo del ciclo.      En java por ejemplo sería:      }

*for (i = 0; i <= 10; i++){*

En Python el enfoque es ligeramente diferente, los valores de cero a 10 son representados por un rango (**range**). Realmente es una función que generará el rango de valores a ser usado como la secuencia en el ciclo **for**. Esto ocurre porque el lenguaje es muy flexible y puede no solamente iterar sobre valores numéricos, sino también iterar en cualquier conjunto de valores que sea mantenido en estructuras de datos, como listas de enteros o cadenas de texto. El formato del ciclo **for** para el caso de rangos es el siguiente:

*for variable in range (...):*

sentencias (cuerpo del ciclo).

```
#Ejemplo de rango en Python  
print('Mostrar los valores de un rango')  
for i in range(0, 10):  
    print(i, ' ', end='')  
print()  
print('Hecho')
```

```
Mostrar los valores de un rango  
0 1 2 3 4 5 6 7 8 9  
Hecho  
  
Process finished with exit code 0
```



## Semana 4

Estructuras de control:  
repetición o iteración

Como se puede ver en el ejemplo, la variable *i* es la variable de iteración y tomará los valores del rango desde 0 hasta 9, no se incluye el valor final 10, es muy importante tener eso en cuenta en los casos en que se utilice los rangos, además el incremento de la variable en cada iteración es 1, si se requiere otro valor es necesario dar otro argumento o parámetro al rango cómo se ve en el siguiente ejemplo, donde el incremento es de 2. En este caso los argumentos del rango son tres: primero el valor inicial (0), segundo el valor final (10) que no se incluye y por último el incremento (2).

```
#Ejemplo de rango con incremento diferente de 1
print('Mostrar los valores de un rango con incremento de 2')
for i in range(0, 10, 2):
    print(i, ' ', end='')
print()
print('Hecho')
```

```
Mostrar los valores de un rango con incremento de 2
0 2 4 6 8
Hecho

Process finished with exit code 0
```