

## PYTHON

### ESTRUCTURAS REPETITIVAS o CÍCLICAS

En Python estas estructuras son bastante parecidas a las que vimos en el lenguaje algorítmico. Sin embargo, sólo tenemos dos, las que antes llamábamos **Mientras** y **Para**.

#### while (Mientras)

Tal como habíamos visto antes, este bucle puede repetir código siempre y cuando se cumpla una condición, en caso contrario, el algoritmo deja de repetir y sigue ejecutando las instrucciones que estén a continuación de la estructura .

En Python será necesario escribir **while**, seguida de una condición (que nos devuelva un valor True o False) y la línea se finaliza con dos puntos (:). Bajo este encabezado, y con su debida indentación, se debe escribir el código que queremos que se repita.

```
n=0  
  
while n<=5:  
    print (n)  
    n=n+1
```

Este código como respuesta dará el siguiente resultado:

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

#### for (Para)

Si bien la idea sigue siendo la misma, en este tipo de bucle encontraremos algunas diferencias con respecto a lo que habíamos visto en el pseudocódigo.

Para usar esta estructura, debemos comenzar escribiendo **for** seguida de una *variable*, luego la palabra **in** y por último un conjunto de ítems que puede ser un rango o secuencia de números, una lista, una cadena de caracteres, etc; la línea como siempre debe finalizar con dos puntos (:). Debajo, se escribirá el código que se debe repetir (con su debida indentación).

La opción más parecida a lo que vimos en el lenguaje algorítmico es usando un rango de números. En este caso debemos utilizar además la función **range()**, que nos permitirá indicar el valor de *inicio*, de *finalización* y el *incremento* o *decremento* de la variable del **for**. Veamos un ejemplo:

```
for i in range(1,10):  
    print(i)
```

Este código dará como resultado→

```
1  
2  
3  
4  
5  
6  
7  
8  
9  
>>> |
```

Es importante tener en cuenta que la función **range(inicio, fin)** siempre considera valores **>=inicio** y **<fin**, por eso en el ejemplo anterior, el ciclo se repite **9** veces (el número 10 no está incluido).

Si en cambio, se escribe **range(10)**, es decir un solo valor, lo tomará como **fin**, asumiendo que el inicio es **0**, por lo tanto el resultado cambiará levemente. Observemos:

```
for i in range(10):  
    print(i)
```

Al ejecutarse mostrará →

```
0  
1  
2  
3  
4  
5  
6  
7  
8  
9  
>>> |
```

Es decir, en este caso el ciclo itera **10** veces, desde **0** a **9**

Finalmente podemos agregar un tercer argumento a la función **range()** para indicar así el *incremento* o *decremento* que precisemos para la variable. Por ejemplo:

```
for i in range(0,14,2):  
    print(i)
```

```
0  
2  
4  
6  
8  
10  
12  
>>> |
```

→

```
for i in range(25,0,-5):  
    print(i)
```

```
25  
20  
15  
10  
5  
>>> |
```