

Estructuras de Control e Iteración

En todo programa que se precie, llega el momento en el que se llega a una bifurcación y que en función de una determinada condición, hay que realizar una serie de cosas u otra.

Esto se hace con el comando if (condición principal), con los opcionales elif (condiciones adicionales, se pueden poner tantas como se quiera) y else (si no se ha cumplido ninguna de las anteriores, sólo se puede poner una vez y al final).

```
posicion= 1

If(posicion == 1):

    Print("Despliega este texto")

Elif(posicion >= 1):

    Print("Despliega este texto")

Else

    Print("Despliega este texto")
```

Condiciones en Python

Las condiciones que se suelen usar con más frecuencia son:

- `a == b` → Indica si a es igual a b
- `a < b`
- `a > b`
- `not` → NO: niega la condición que le sigue.
- `and` → Y: junta dos condiciones que tienen que cumplirse las dos
- `or` → O: junta dos condiciones y tienen que cumplirse alguna de las dos.

While en Python

En ocasiones, tenemos que repetir varias veces una determinada tarea hasta conseguir nuestro objetivo. En Python esto se realiza con el comando while. A modo de ejemplo while en Python se usa así:

```
posicion= 1  
  
while(posicion <= 10):  
    Print("Despliega este texto")  
    posicion+=1
```

For en Python

Sintaxis de la sentencia for

La sintaxis es la siguiente:

```
For iterador in secuencia  
  
    #Codigo a ejecutar
```

El ciclo for en el lenguaje Python tiene cierta ventaja y diferencia con otros lenguajes. Si tienes nociones de programación en lenguajes diferentes a python, tal vez puedas pensar que el ciclo For se define utilizando contadores y rangos en los cuales se ejecutaría el código del for, sin embargo en Python el concepto es un poco diferente.

Cuando usamos la sentencia For en Python, tenemos la capacidad de recorrer una secuencia por medio de “iteraciones”, una secuencia como una lista o una simple cadena de texto, veamos un ejemplo para comprender mejor.

Si quisiéramos declarar una cadena de texto y recorrer cada uno de sus caracteres, podemos usar la sentencia For para ello.

```
For letra in "HOLA"  
  
    Print(letra)
```

Bibliografía

<http://codigoprogramacion.com/cursos/tutoriales-python/ciclo-for-en-python.html#.W5AAesBR2wU>

<https://www.tutorialpython.com/bucles-while-y-for-en-python/>

