

Estructuras de control e iteración en Python

Tenemos dos estructuras de selección en Python

if

Condición es de tipo booleano, y cuando ésta se cumple se ejecuta el bloque.

```
if condicion:  
    bloque
```

if else

Condición es de tipo booleano, cuando ésta se cumple se ejecuta el bloque 1, cuando no se cumple se ejecuta el bloque 2.

```
if condicion:  
    bloque 1  
else:  
    bloque 2
```

Además, tenemos la siguiente estructura, que abrevia el if else if:

if elif [else]

```
if condicion 1:  
    bloque 1  
elif condicion 2:  
    bloque 2  
elif condicion 3:  
    bloque 3  
else:  
    bloque 4
```

Iteración

Para realizar bucles, Python nos proporciona dos estructuras.

for

```
for i in range(0,cant):  
    bloque
```

Se ejecuta el bloque cant veces, comenzando la variable i adoptando el valor 0; en las siguientes iteraciones el valor se va incrementando hasta tomar el valor cant-1. Sin embargo, el for en Python permite iterar no solo sobre progresiones aritméticas, sino que recorre los elementos de una secuencia, ya sea lista o cadena, en el orden en que estos aparecen en la misma.

La función range se emplea para generar una lista de enteros

```
range(10)
```

se corresponde con la lista : [0, 1, 2, 3, 4, 5, 6, 7, 8, 9], ya que el valor final indicado no es parte de la lista generada. Se puede hacer que la lista comience en un número dado:

```
range(4,10)
```

[4, 5, 6, 7, 8, 9]: incluso se puede modificar el incremento (step)

```
range(-2,-10,-2)
```

```
[-2, -4, -6, -8]
```

while

El bucle while se ejecuta mientras la condición sea cierta, si la condición es falsa al inicio, el bloque no se ejecuta y se pasan a ejecutar las sentencias que le siguen. En Python, cualquier valor entero distinto de cero es verdadero y 0 es falso. La condición también puede ser una lista o cualquier secuencia, siendo la secuencia vacía falsa. El cuerpo del bucle debe estar indentado, ya que de este modo Python agrupa las sentencias.

```
while condicion:  
    bloque
```