

Introducción a la Programación con **PYTHON**



José Rodriguez

Unidad 3 Estructuras de Control

UNIDAD 3: ESTRUCTURAS DE CONTROL

INDEX

-
- 1.- La estructura condicional
 - 2.- Bucle While
 - 3.- La estructura FOR
-

1.- La estructura condicional

La estructura básica para establecer una condición es el bloque if. En su versión más simple, dicho bloque se escribe de la siguiente forma:

If condición : acción1;acción2,....

La condición se expresa en términos de un operador lógico aplicado a una o varias variables o valores. En el caso de que el resultado sea “true” se ejecutan las instrucciones que siguen. El bloque de instrucciones viene delimitado por el símbolo “:” y van separadas por “;”

```
Numero=7;
if numero >0: print ("positivo");numero =0;print ("numero")
```

Si numero=7 le asignamos un nuevo valor.

```
numero=-6
if numero >0: print ("positivo");numero =0;print ("numero")
print ("Fin")
```

La instrucción print (“Fin”) está fuera del if con lo cual siempre se ejecutará.

Otra forma de expresarlo es

```
numero=-6
if numero >0:
    print ("positivo")
    numero=0
    print ("numero")
print ("Fin")
```

También se permite la sintaxis clásica de la estructura IF

```
numero=-6
```

```
if numero >0:  
    print ("positivo")  
    numero=0  
    print (numero)  
else:  
    print ("Negativo")  
print ("Fin")
```

También es posible anidar if con la estructura if...elif..else

```
numero=0  
if numero >0:  
    print "positivo"  
  
elif numero<0 :  
    print "Negativo"  
else:  
    print "Cero"  
print "Fin"
```

2.- Bucle While

El bucle While es la estructura mas característica de las iteraciones

```
print ("===== TABLA DE MULTIPLICAR =====")  
numero=0  
while numero<=10:  
    print ("7 x " + str(numero) + " = " + str(numero* 7))  
    numero = numero +1
```

la estructura while no tiene una instrucción de cierre, las instrucciones incluidas en el bucle están identadas por una tabulación. Por ejemplo, si escribimos

```
while numero<=10:  
    print ("7 x " + str(numero) + " = " + str(numero* 7))  
numero = numero +1
```

La instrucción numero = numero +1 no estaría incluida en el bucle y, por tanto, numero siempre sería 0 y el bucle sería infinito.

Podemos incluir bucles que no dependan de un valor final sino que el final del bucle se decida con un if

```
print ("===== TABLA DE MULTIPLICAR =====")  
numero=0  
while True:  
  
    print ("7 x " + str(numero) + " = " + str(numero* 7))
```

```
numero = numero +1  
if numero>10:  
    break
```

La cláusula “continue” se salta las instrucciones que le siguen dentro del bucle y vuelve a la siguiente iteración. En este ejemplo mostramos los números pares menores que 10

```
print ("===== NUMEROS PARES =====")  
numero = 0  
while numero < 10:  
    numero = numero + 1  
    if numero % 2 == 1:  
        continue  
    print(numero)
```

3.- La estructura FOR

La estructura for utilizada en Python es diferentes a la de otros lenguajes de programación. En general es una estructura muy abierta y utiliza el contenido de una lista para iterar sobre sus elementos.

En este ejemplo el bucle recorre con la variable x los valores de la lista a

```
print ("===== TRABAJANDO CON LISTAS =====")  
a= [1,2,3]  
for x in a:  
    print (x)
```

En este otro ejemplo se recorre los valores del array a y se calcula su longitud.

```
a= ["uno","dos","tres"]  
for x in a:  
    print (x, len(x))
```

La segunda estructura de for es la que utiliza la función range() para generar los valores de la iteración que, en este caso, serán numéricos.

```
print ( "===== NUMEROS NATURALES =====")  
for numero in range(1,10):  
    print (numero)
```

Observa en el resultado que no se cuenta el índice superior del rango y por eso sólo se muestra hasta el 9.

La función range tiene la estructura general

range(valor inicial, valor final, incremento)

Si se omite el valor incremento se utiliza la unidad como incremento entre dos elementos consecutivos.

Es obvio que cualquiera de los tres parámetros puede ser negativo ya que range() genera una progresión aritmética en el campo de los números enteros.