



Programación Python

José Javier Galán Hernández :



Programación Python

7. Estructuras de Control

7. Estructuras de Control. IF...ELSE

Evalúa una expresión, si dicha expresión devuelve True realiza las sentencias que contiene ese bloque. En caso contrario las sentencias de dicho bloque no se ejecutarán.

Sintaxis:

If condición:

Sentencias_a_ejecutar

Ejemplo:

```
valor = 3
if valor < 5: #Pregunto si es menor que 5
    print("Es menor que 5.")
if valor > 5: #Pregunto si es mayor que 5
    print("Es mayor que 5.")
print("Esta linea la muestro siempre")
```

Es menor que 5.

Esta linea la muestro siempre



7. Estructuras de Control. IF...ELSE

Opcionalmente, si la condición IF no se cumple, podemos indicar que realice las sentencias de otro bloque de código.

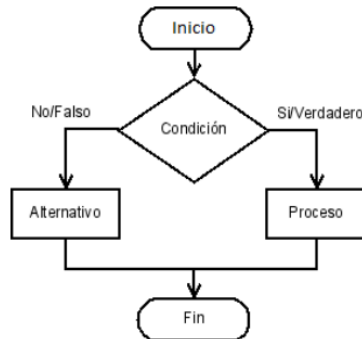
Sintaxis:

If condición:

Sentencias_a_ejecutar

Else:

Sentencias_a_ejecutar



Ejemplo:

```
valor = 3
if valor < 5: #Pregunto si es menor que 5
    print("Es menor que 5.")
else: #Pregunto si es mayor que 5
    print("Es mayor o igual que 5.")
print("Esta linea la muestro siempre")
```

Es menor que 5.

Esta linea la muestro siempre



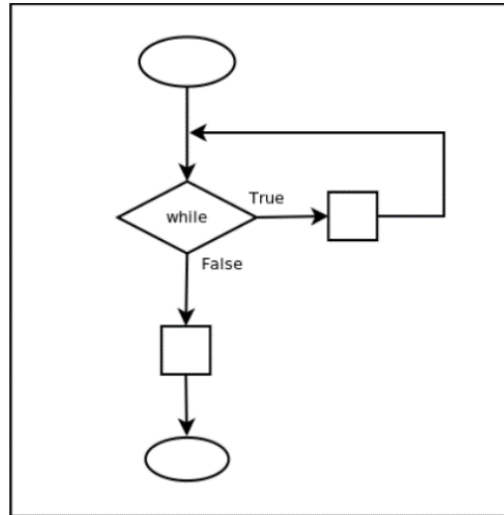
7. Estructuras de Control. WHILE

Es un bucle que repite un bloque de instrucciones mientras se repita una condición. Si la primera vez que evalúa la condición, esta es falsa, nunca realizara el contenido. Si la condición nunca deja de ser verdadera, será un bucle infinito.

Sintaxis:

while condición:

Instrucciones



7. Estructuras de Control. WHILE

Ejemplo:

```
print("Muestra los numeros del 1 al 10, ambos incluidos.")
contador = 1
while contador < 11:
    print(contador)
    contador=contador+1
```

Muestra los numeros del 1 al 10, ambos incluidos.

1
2
3
4
5
6
7
8
9
10



7. Estructuras de Control. WHILE

WHILE Puede contener la sentencia else, que se ejecutara cuando la condición del bucle no sea cierta.

Sintaxis:

while condición:
 Instrucciones

else:
 Instrucciones

```
print("Muestra los numeros del 1 al 10, ambos incluidos.")
contador = 1
while contador < 11:
    print(contador)
    if contador==3:
        contador=20
        contador=contador+1
    else:
        print("El valor ", contador, " no cumple la condición.")
print("continuo")
```

```
Muestra los numeros del 1 al 10, ambos incluidos.
1
2
3
El valor  21  no cumple la condición.
continuo
```



7. Estructuras de Control. BREAK / CONTINUE

Break detiene el bucle

```
print("Muestra los numeros del 1 al 10, ambos incluidos.")
contador = 1
while contador < 11:
    print(contador)
    contador=contador+1
    if contador==3:
        break
else:
    print("El valor ", contador, " no cumple la condición.")
print("continuo")
```

```
Muestra los numeros del 1 al 10, ambos incluidos.
1
2
continuo
```


7. Estructuras de Control. BREAK / CONTINUE

Continue interrumpe el bucle, pero solo en el ciclo en el que se ejecuta.

```
print("Muestra los numeros del 1 al 10, ambos incluidos.")

for contador in range(11):
    if contador==3:
        continue
    print(contador)

print("Resto de programa.")
```

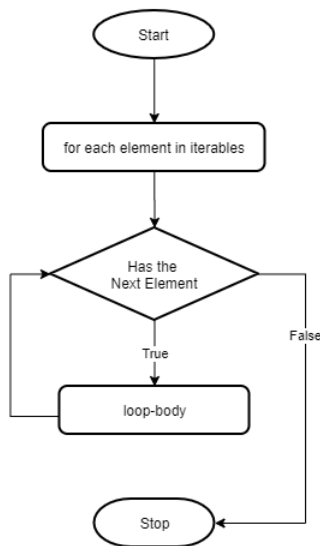
```
Muestra los numeros del 1 al 10, ambos incluidos.
0
1
2
4
5
6
7
8
9
10
Resto de programa.
```

7. Estructuras de Control. FOR

En Python la estructura FOR se utiliza para recorrer uno a uno los elementos de una colección.

Sintaxis:

for elemento **in** listaElementos:
 sentencias



7. Estructuras de Control. FOR

Ejemplos:

```
numeros = [1, 3, 5, 84]  
for n in numeros:  
    print(n)
```

1
3
5
84

```
for i in range(11):  
    print(i)
```

0
1
2
3
4
5
6
7
8
9
10

En Python la estructura FOR se utiliza principalmente con listas y diccionarios que veremos más adelante.

...ahora...comencemos!

¡A programar!

- 1. Crea una variable llamada valor1 que contenga el número 5 y una variable llamada valor2 que contenga el texto "Comiendo a programar".*
- 2. Solicita el nombre del usuario por teclado y, una vez introducido, muestra un mensaje personalizado dándole los buenos días.*
- 3. Pide un número por teclado y muestra un mensaje indicando si es positivo.*
- 4. Solicita tres números por teclado y muestra cuál de ellos es el mayor.*
- 5. Pide un número por teclado hasta que se introduzca un valor comprendido entre 1000 y 1500.*
- 6. Introduce un número entre 1 y 10 y muestra su tabla de multiplicar.*

...ahora...comencemos!

¡A programar!

7. Lee dos números por teclado y muestra el siguiente menú de opciones:

Sumar

Restar

Multiplicar

Dividir

A continuación, pide por teclado un número entero entre 1 y 4. Según la opción elegida, muestra el resultado de la operación correspondiente con los dos números introducidos.

...ahora...comencemos!

¡A programar!

8. Pide por teclado una **nota numérica entera entre 1 y 10**.

Según el valor introducido, muestra un mensaje indicando la calificación correspondiente según este baremo:

1 a 4 → **Suspenso**

5 a 6 → **Bien**

7 a 8 → **Notable**

9 a 10 → **Sobresaliente**

...ahora...comencemos!

¡A programar!

10. Crea un juego de adivinanza para dos jugadores:

El **primer jugador** introduce un número comprendido entre 1 y 1000.

El **segundo jugador** debe adivinarlo en un máximo de **10 intentos**.

Tras cada intento, el programa indicará si el número buscado es **menor** o **mayor** que el introducido.

Si acierta, mostrará un mensaje de éxito; si no lo logra en los 10 intentos, mostrará un mensaje de que ha perdido.

Ejercicios de refuerzo. Opcional.

1. Pide un número entero por teclado y muestra si es par o impar.
2. Pide un número del 1 al 12 e indica a qué mes del año corresponde.
3. Solicita un número entero y muestra la suma de todos los números desde 1 hasta ese número.
4. Muestra todos los números pares entre 1 y 100.
5. Crea un programa que muestre un contador regresivo desde 10 hasta 1 y al final muestre el mensaje "¡Despegue!".

