



# 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.

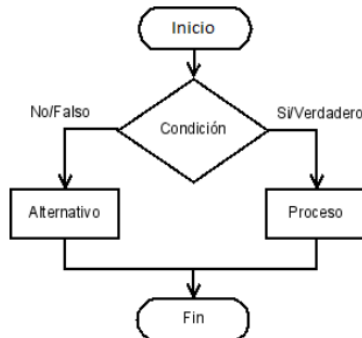
Sintaxis:

**If** condición:

Instrucciones

**Else:**

Instrucciones



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. IF...ELSE

IF anidado. IF dentro de otro IF o de otro ELSE.

```
valor = 5
if valor < 5: #condición 1
    print("Es menor que 5.")
else:
    if valor == 5:
        print("Es igual a 5.")
    else:
        print("Es mayor que 5.")
print("Esta línea la muestro siempre")
```

Es igual a 5.

Esta línea la muestro siempre





# Programación Python

## 7. Estructuras de Control

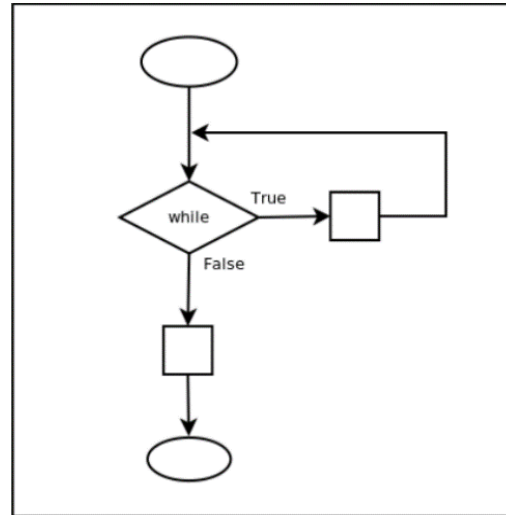
# 7. Estructuras de Control. WHILE

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 / PASS

**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 / PASS

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

```
print("Muestra los numeros del 0 al 10, ambos incluidos.")

for contador in range(11):
    if contador==3:
        continue#no mostrara el 3
    print(contador)

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

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

# 7. Estructuras de Control. BREAK / CONTINUE / PASS

**Pass** sirve cuando necesitamos tener estructuras sin establecer comportamiento

```
for num in range(1, 5):
```

```
    Input In [3]
```

```
    for num in range(1, 5):
```

```
        ^
```

**IndentationError:** expected an indented block

```
for num in range(1, 5):
```

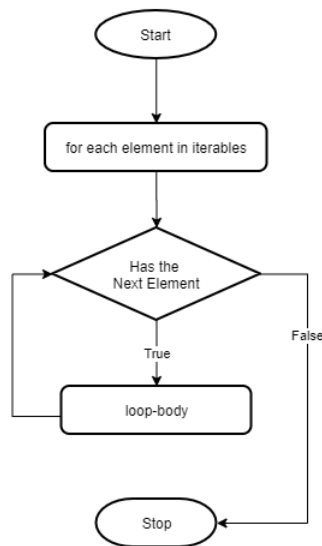
```
    pass
```

# 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. Muestra un mensaje que solicite el nombre del usuario, una vez introducido, se mostrara un mensaje personalizado dando los buenos días.*
- 3. Introducido un número por teclado, nos dirá si es positivo.*
- 4. Introducidos 3 números por teclado nos dirá cuál es el mayor.*
- 5. Pedir un numero por teclado hasta introducir un numero entre 1000 y 1500*
- 6. Introducido un número entre 1 y 10 mostrar su tabla de multiplicar.*

# ...ahora...comencemos!

*A programar!*

**7. Leer 2 números por teclado y mostrar el menú:**

- 1. Sumar**
- 2. Restar**
- 3. Multiplicar**
- 4. Dividir**

***Después, leer por teclado un número entere 1 y 4. Según la opción elegida mostrar el resultado de la operación con los 2 números introducidos.***



# ...ahora...comencemos!

*A programar!*

**8. Dada una nota numérica sin decimales entre 1 y 10 introducida por teclado mostrar un mensaje de texto indicando la nota según este baremo:**

**1-4 Suspenso**

**5-6 Bien**

**7-8 Notable**

**9-10 Sobresaliente**

# ...ahora...comencemos!

*A programar!*

**9. Provocar un bucle infinito que muestre la palabra “ERROR”**

**10. Queremos hacer un juego para 2 jugadores: el primero introduce un número y el segundo debe adivinarlo. El número a adivinar debe estar entre 1 y 1000. Se permite solo 10 intentos para intentar adivinarlo y si no lo acierta el programa le informara sobre si el numero buscado es menor o mayor.**

