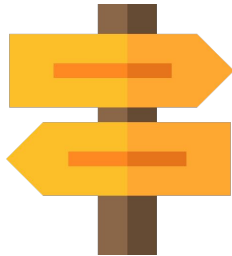


Control de flujo



¿Control de flujo?

A menudo es conveniente tomar una decisión acerca de **cuál es el siguiente paso a tomar** dependiendo de la situación que se presenta.

Este tipo de decisiones dependen de algo llamado “**condiciones**”

Condition



if

La sentencia **if** define si una acción se va a realizar o no.

Está compuesta de la palabra reservada **if** seguida de una condición y dos puntos (:).

```
edad = 13
if edad < 18:
    print("Eres menor de edad")
```

Es importante notar el espacio que hay antes de la impresión (print).

Bloques de código

Los bloques de código se van a diferenciar con la **indentación** que tengan. La indentación es el **espacio** que se encuentra antes de las líneas de código.

line of code
line of code
line of code

line of code
line of code
line of code

line of code
line of code
line of code

line of code
line of code

block 3

block 2

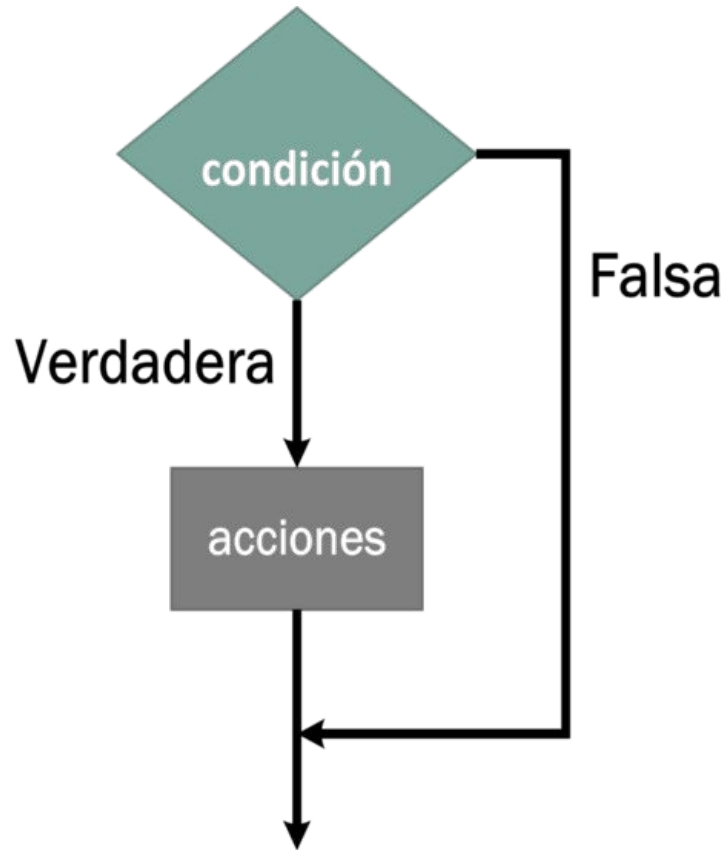
block 1

Bloques de código

La indentación se hace colocando una o más tabulaciones antes de la línea de código o colocando espacios.

No combinar tabulaciones con espacios.

if

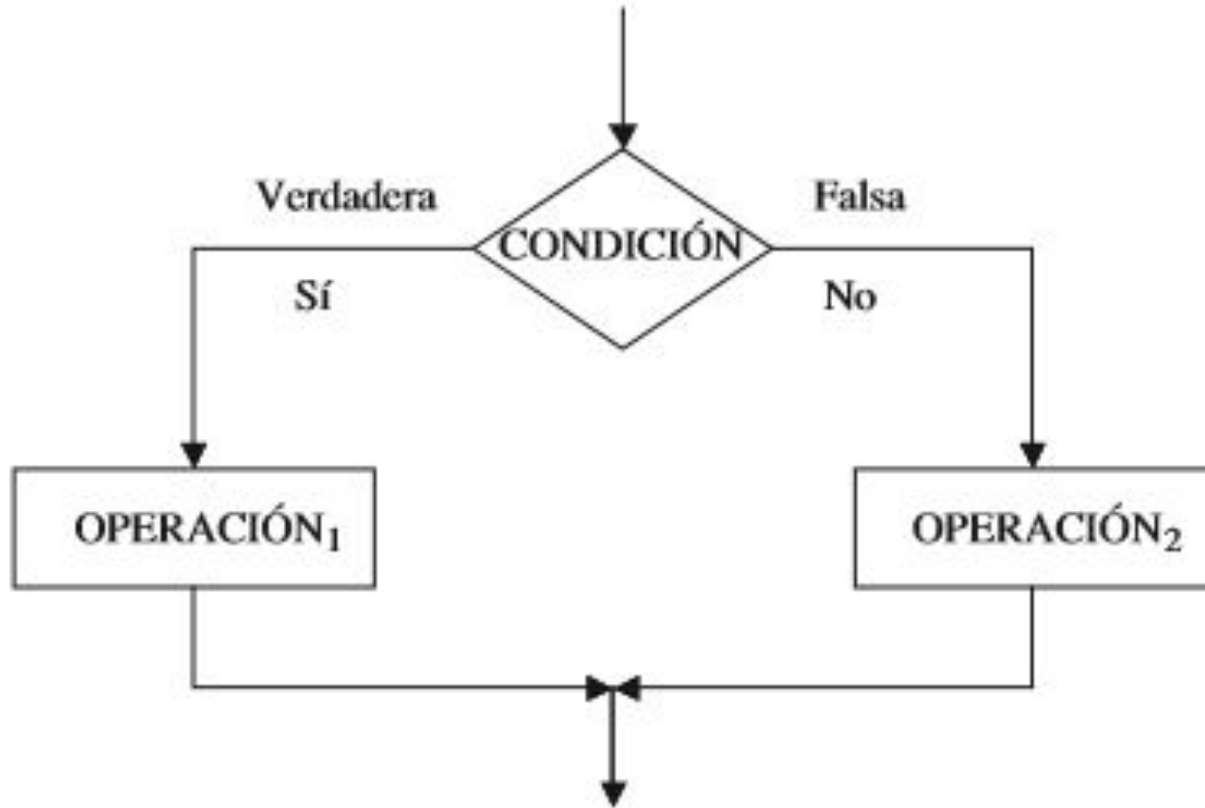


if - else

La sentencia if-else define qué acción se va a realizar.

```
fecha = "Martes"
if fecha = "Lunes":
    print("Hoy es lunes")
else:
    print("Hoy no es lunes")
```

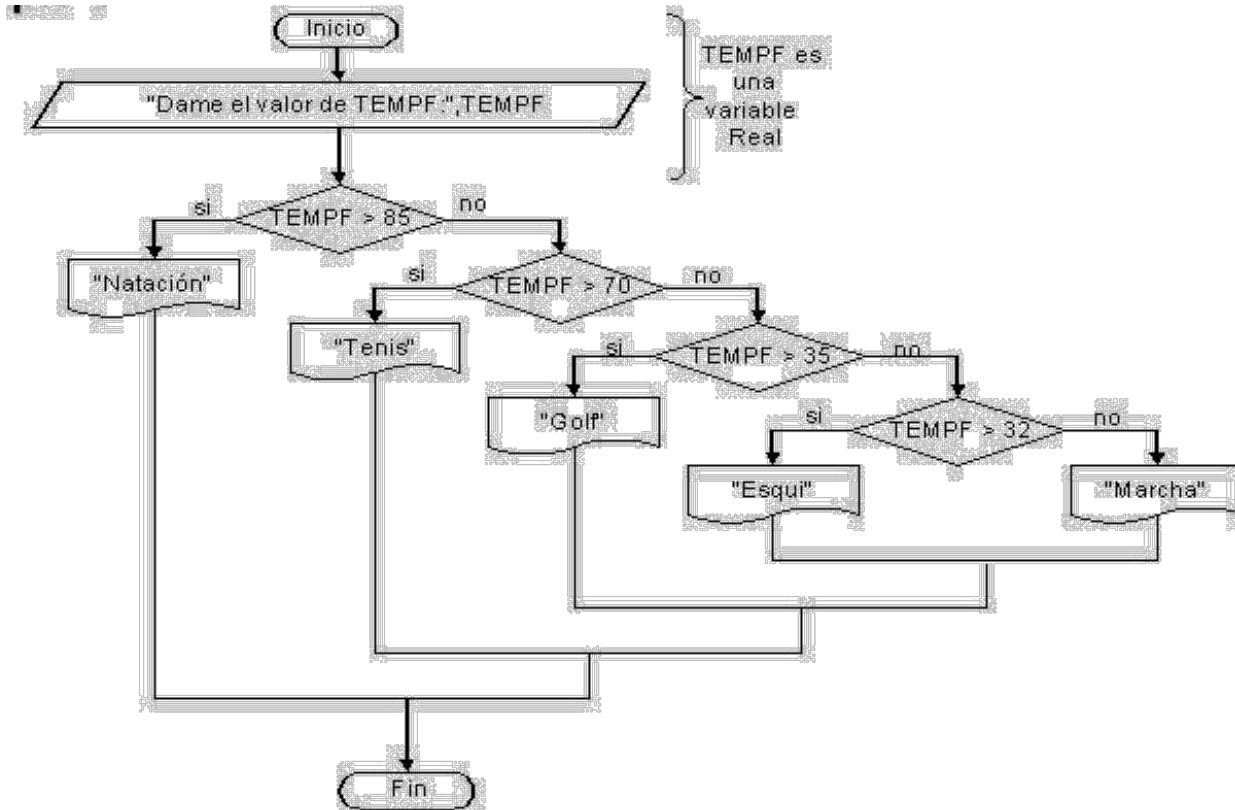

if - else



if - else anidado

```
edad = 10
if edad < 0:
    print("Eso no es posible")
else:
    if edad < 18:
        print("Eres menor de edad")
    else:
        print("Eres mayor de edad")
```

if - else anidado



elif

La sentencia elif nos permite tener un código más limpio al usar if-else anidados.

```
edad = 10
if edad < 0:
    print("Eso no es posible")
elif edad < 18:
    print("Eres menor de edad")
else:
    print("Eres mayor de edad")
```

pass

La palabra reservada `pass` nos permite “no hacer nada”.

```
if equipo == "America":  
  
else:  
    print("Yo tampoco carnal")
```



```
if equipo == "America":  
    pass  
else:  
    print("Yo tampoco carnal")
```



while

La sentencia while hace que una acción se repita mientras se cumpla una condición.

```
respuesta = ""  
  
while respuesta != "salir":  
    respuesta = input("Presiona salir:")
```

continue

La palabra reservada `continue` nos permite saltar lo que resta de un ciclo e iniciar otro.

```
numero = 0

while numero < 10:
    numero += 1
    if numero == 5:
        continue
    print(numero)
```

break

La palabra reservada break nos permite romper el ciclo y terminar el loop.

```
while True:
    respuesta = input("Presiona salir: ")
    if respuesta == "salir":
        break
```


for

La sentencia `for` permite iterar a través de los elementos de cualquier secuencia ordenada o objeto iterable (iterar -> repetir) y ejecutar un bloque de códigos por cada una. El elemento de cada ciclo se almacena en una variable que se encuentra después de la palabra reservada `for`.

for

```
lista = [3, 5, "hola", 4.4]

for elemento in lista:
    print(elemento)
```

for

Entre los objetos sobre los que se puede iterar se encuentra:

- listas
- tuplas
- diccionarios
- cadenas de texto
- 'range'
- archivos

range

Es una herramienta que nos permite **generar una serie de enteros** sobre los que podemos **iterar** en un ciclo for.

```
for i in range(5):  
    print(i)
```

Loop else

Se ejecuta este bloque de código en caso de que el loop **no termine con un break**.

```
respuesta = ""

while respuesta != "salir":
    respuesta = input("Presiona salir: ")
    if respuesta == "romper":
        break
else:
    print("Saliste de manera adecuada")
```