

# Condicionales

Los condicionales en Python son estructuras de control que permiten que un programa ejecute diferentes bloques de código según la condición que se evalúe. Podemos tener condicionales de una sola condición como veremos a continuación

```
nombre = "Juan"
if nombre == "Juan":
    print("Hola Juan")
```

También tenemos condicionales para 2 condiciones

```
edad = 10
if edad > 18:
    print("Es mayor de edad")
else:
    print("Es menor de edad")
```

También tenemos condicionales con más de 2 condiciones a la vez

```
numero = 11
if numero > 10:
    print("El número es mayor que 10.")
elif numero == 10:
    print("El número es igual a 10.")
else:
    print("El número es menor que 10.")
```

## Operadores lógicos

Los operadores lógicos se pueden usar para combinar condiciones. Los operadores lógicos disponibles en Python son:

- **and**: Se evalúa como verdadero si ambas condiciones son verdaderas.
- **or**: Se evalúa como verdadero si al menos una condición es verdadera.
- **not**: Se evalúa como verdadero si la condición es falsa.

Por ejemplo, el siguiente código imprime un mensaje si el número es mayor que 10 y menor que 20:

```
numero = 15

if numero > 10 and numero < 20:
    print("El número es mayor que 10 y menor que 20.")
```

## If else en linea / ternarios

Los operadores ternarios en Python son una forma concisa de escribir una expresión condicional. Se pueden usar para asignar un valor a una variable o para realizar una acción dependiendo del valor de una condición.

```
valor_si_verdadero if condición else valor_si_falso
edad = 18

mensaje = "Eres mayor de edad" if edad >= 18 else "Eres menor de edad"

print(mensaje)
# Eres mayor de edad
```

Algunas recomendaciones:

- Evita usarlos para expresiones complejas. Si la expresión es demasiado compleja, es mejor usar una instrucción condicional tradicional.
- Usa comentarios para explicar el código. Los comentarios pueden ayudar a que el código sea más legible y fácil de entender.
- Haz pruebas para asegurarte de que el código funciona correctamente. Las pruebas pueden ayudar a detectar errores y garantizar que el código funcione como se espera.