

Ciclo I

Fundamentos de la programación

Semana 04

Operadores lógicos.

Se utiliza un operador lógico para tomar una decisión basada en múltiples condiciones.

Los operadores lógicos utilizados en Python son:

OPERADOR	DESCRIPCIÓN	USO
and	Devuelve True si ambos operandos son True	a and b
or	Devuelve True si alguno de los operandos es True	a or b
not	Devuelve True si alguno de los operandos es False	not a

```
a, b = True, False
print("Tabla de verdad de and")
print("-"*22)
print(b, b, b and b, sep = '\t')
print(b, a, b and a, sep = '\t')
print(a, b, a and b, sep = '\t')
print(a, a, a and a, sep = '\t')
print("")
print("Tabla de verdad de or ")
print("-"*22)
print(b, b, b or b, sep = '\t')
print(b, a, b or a, sep = '\t')
print(a, b, a or b, sep = '\t')
print(a, a, a or a, sep = '\t')
print("")
print("Tabla de verdad de not ")
print("-"*22)
print(b, not b, sep = '\t')
print(a, not a, sep = '\t')
```

Estructura de control: Sentencias condicionales IF-ELSE y su sintaxis.

IF

Definición

La estructura de control **if** permite que un programa ejecute unas instrucciones cuando se cumplan una condición.

Sintaxis pseudocódigo

```
Si [Condición] Entonces
    Instrucción 1
    Instrucción 2
    .
    .
    .
FinSi
```

Sintaxis lenguaje de programación

```
if condición:
    #aquí van las órdenes que se ejecutan si la condición es cierta
    #y que pueden ocupar varias líneas
```

La ejecución de esta construcción es la siguiente:

- Si el resultado es **True** se ejecuta el bloque de sentencias
- Si el resultado es **False** no se ejecuta el bloque de sentencias

Ejercicios

1. Se desea validar una variable booleana, en caso de que sea **True**, que imprima en pantalla: Esta variable contiene como valor TRUE.
2. Se desea validar una variable booleana, en caso de que **False**, que imprima en pantalla: Esta variable contiene como valor False.
3. Validar si el número que ingresa el usuario es positivo, que imprima en pantalla: El número que ingresó el usuario es positivo.
4. Se tiene como mensaje secreto "**MUNDOHACKER2021**", el usuario debe descubrir dicho mensaje, al descubrirlo que imprima en pantalla: ¡Felicitaciones! Eres un haxor
5. Se tiene dos variables A y B, dichas variables se inicializan de la siguiente manera 32 y 24. Se desea hacer una comparación de que variable es mayor. Imprimir en pantalla: La variable A o B es mayor

IF - ELSE

Definición

La estructura de control **if ... else ...** permite que un programa ejecute unas instrucciones cuando se cumple una condición y otras instrucciones cuando no se cumple esa condición.

Sintaxis pseudocódigo

```
Si [Condición] Entonces
    Instrucción 1
    Instrucción 2
    .
    .
    .
```

```
Instrucción n
SiNo
    Instrucción A
    Instrucción B
    .
    .
    .
Instrucción Z
FinSi
```

Sintaxis lenguaje de programación

```
if condición:
    #aquí van las órdenes que se ejecutan si la condición es cierta
    #y que pueden ocupar varias líneas
else:
    #y aquí van las órdenes que se ejecutan si la condición es
    #falsa y que también pueden ocupar varias líneas
```

La ejecución de esta construcción es la siguiente:

- Si el resultado es **True** se ejecuta el bloque de sentencias 1
- Si el resultado es **False** no se ejecuta el bloque de sentencias 2

Ejercicios

1. En una discoteca, se permite el acceso a ingresos de mayores de edad, ayuda a la persona de SEGURIDAD para validar dicho acceso, en caso de que no sea mayor, lo mandas para su casa.
2. Se desea saber si un número ingresado por el usuario es par o impar.
3. Se tiene dos variables A y B, dichas variables se inicializan de la siguiente manera 32 y 24. Se desea hacer una comparación de que variable es mayor. Imprimir en pantalla: La variable A o B es mayor, La variable A o B es menor.
4. Se desea saber si el número ingresado por el usuario, está en el rango de 10 a 100. En caso de que se encuentre en el rango, imprimir en pantalla "El número si existe", caso contrario, "El número no existe"
5. Escribir un programa que, dado un número entero, muestre su valor absoluto. **Nota:** Para los número positivos su valor absoluto es igual al número (el valor absoluto de 52 es 52) mientras que, para los negativos, su valor absoluto es el número multiplicado por -1 (el valor absoluto de -52 es 52)

IF - ELIF - ELSE

Definición

La estructura de control **if... elif... else ...** permite encadenar varias condiciones. **elif** es una contracción de **else if**.

Sintaxis pseudocódigo

```
Si [Condición 1] Entonces
    Instrucción 1
    Instrucción 2
Si [Condición 2] Entonces
    Instrucción 3
```

```
Instrucción 4
SiNo
    Instrucción 5
    Instrucción 6
FinSi
SiNo
    Instrucción 7
    Instrucción 8
FinSi
```

Sintaxis lenguaje de programación

```
if condición_1:
    bloque 1
elif condición_2:
    bloque 2
else:
    bloque 3
```

La ejecución de esta construcción es la siguiente:

- Si se cumple la condición 1, se ejecuta el bloque 1
- Si no se cumple la condición 1 pero sí que se cumple la condición 2, se ejecuta el bloque 2
- Si no se cumplen ni la condición 1 ni la condición 2, se ejecuta el bloque 3.

Ejercicios

1. Se desea elaborar una calculadora simple, que imprima en pantalla EL OPERADOR ES: [OPERADOR] RESULTADO ES: [RESULTADO]
2. Se desea saber si el número ingresado es múltiplo de 4 y 10, caso contrario que imprima en pantalla, no encuentro múltiplos de 4 y 10
3. Se desea saber si el alumno, Aprobó, Pasó a las justas o Reprobó, donde mayores de 18 es APROBADO, mayores de
4. Requerir al usuario que ingrese un día de la semana e imprimir un mensaje si es **lunes**, otro mensaje diferente si es **viernes**, otro mensaje diferente si es **sábado** o **domingo**. Si el día ingresado no es ninguno de esos, imprimir otro mensaje.
5. Solicitar al usuario que ingrese dos números y mostrar cuál de los dos es menor. Considerar el caso en que ambos son iguales.

Condiciones no booleanas

Definición

Dado que cualquier variable puede interpretarse como una variable booleana, si la condición es una comparación con cero, podemos omitir la comparación.

Ejemplo:

```
numero = int(input("Escriba un número: "))
if numero % 2 != 0:
    print(numero + " es impar")
else:
    print(numero + " es par")
```

También se puede escribir de la siguiente manera, omitiendo la comparación:

```
numero = int(input("Escriba un número: "))
if numero % 2:
    print(numero + " es impar")
else:
    print(numero + " es par")
```

Si el número es impar, **numero % 2** da como resultado **1**. Y como el valor **booleano** de un número diferente de cero es **True** (es decir, **bool(1)** es **True**), la condición se estaría cumpliendo.

Otro ejemplo, si la comparación es una igualdad

```
numero = int(input("Escriba un número: "))
if numero % 2 == 0:
    print(numero + " es par")
else:
    print(numero + " es impar")
```

También se puede escribir de la siguiente manera, omitiendo la comparación:

```
numero = int(input("Escriba un número: "))
if not numero % 2:
    print(numero + " es par")
else:
    print(numero + " es impar")
```

Operador ternario

El operador ternario o `ternary operator` es una herramienta muy potente que muchos lenguajes de programación tienen. En Python es un poco distinto a lo que sería en C, pero el concepto es el mismo. Se trata de una cláusula `if`, `else` que se define en una sola línea y puede ser usado por ejemplo, dentro de un `print()`.

Existen tres partes en un operador ternario, que son exactamente iguales a los que había en un `if else`.

- Tenemos la condición a evaluar,
- el código que se ejecuta si se cumple,
- y el código que se ejecuta si no se cumple.

En este caso, tenemos los tres en la misma línea.

```
# [código si se cumple] if [condición] else [código si no se cumple]
```

Ejemplo:

```
x = 5
print("Es 5" if x == 5 else "No es 5")
#Es 5
```

```
a = 10
b = 5
c = a/b if b!=0 else -1
print(c)
#2
```

```
# Decrementa x en 1 unidad si es mayor que cero
x = 5
x-=1 if x>0 else x
print(x)
```