



LECCIÓN 1

Condicionales



Estructuras de control

Precedencia

Cuando en una expresión aparece más de un operador, el orden de evaluación depende de las reglas de precedencia. Para los operadores matemáticos, Python sigue la convención matemática. El acrónimo PEMDAS es una forma útil de recordar las reglas:

- Paréntesis
- Exponentes
- Multiplicación
- División
- Adición
- Sustracción

Hacer ejercicio con los estudiantes para resolver una expresión matemática

Condicionales

Nota: Para las ejercitaciones de este módulo, se recomienda fuertemente realizar diagramas de flujos previo a la codificación.

Si, es cierto. True

No, esto es falso. False

Para hacer preguntas, Python utiliza un conjunto de operadores. Los operadores binarios requieren dos argumentos para realizar su comparación y verificar si la expresión es verdadera o falsa.

Operador	Nombre
==	Igual
!=	Diferente
>	Mayor qué
<	Menor qué
>=	Mayor o igual
<=	Menor o igual



Operadores lógicos

Existen otros operadores llamados lógicos, están asociados a tablas de verdad en lógica y se presentan a continuación:

Operador	Nombre
and	True si ambos estamentos son verdaderos
or	True si al menos uno de los estamentos es verdadero
not	Niega el resultado, si se obtuvo un True retorna un False

Igualdad

El operador de igualdad `==` se utiliza para preguntar si dos valores son iguales, el operador evalúa los dos argumentos y determina si son o no iguales teniendo en cuenta su tipo de dato y valor

```
[ ] 1 == 1
     1 == '1'
     1 == 1.

True
```

Las anteriores comparaciones son muy sencillas dado que tenemos valores literales, qué sucede si la comparación es `loc == 3`, qué se puede obtener al hacer la comparación?, Para nosotros como programadores puede resultar un tanto complejo, es por esto que Python se encarga de resolver la igualdad y entregar el resultado.

Desigualdad o diferente

Funciona similar a la igualdad pero entrega el resultado contrario, es decir, si son iguales, el resultado de la comparación es `False`. Si no son iguales, es porque son diferentes y el resultado de la comparación es `True`.



```
[ ] 1 != 1  
    1 != '1'  
    1 != 1.
```

True

Mayor que

El operador mayor que > realiza una comparación estricta de los argumentos de la expresión, se obtiene True cuando el argumento de la izquierda es mayor al argumento del lado derecho, False en el caso contrario o cuando son iguales.

```
[ ] 1 > 2  
    2 > 1  
    2 > 2  
    "hola" > "Hola"
```

True

Pueden indentificar el porqué la última comparación entre las cadenas "hola", indican que es verdadero que una es mayor a la otra?

Mayor que

El operador mayor que > realiza una comparación estricta de los argumentos de la expresión, se obtiene True cuando el argumento de la izquierda es mayor al argumento del lado derecho, False en el caso contrario o cuando son iguales.

Mayor o igual que

Es una versión menos estricta al operador mayor qué, ya que este incluye la posibilidad de igualdad, es decir que si el argumento del lado izquierdo es mayor al del lado derecho la salida será True, caso contrario será False, pero en el caso que los argumentos sean iguales la respuesta también será True.

En el siguiente código se encuentran dos temperaturas, una interna y una externa, compruebe la salida de cada una de las comparaciones al modificar la temperatura con un calefactor



```
[ ] temperatura_interna = 18
    temperatura_externa = 22

    print(temperatura_interna >= temperatura_externa)

    calefactor_bajo = 4
    calefactor_alto = 6

    print(temperatura_interna + calefactor_bajo >= temperatura_externa)

    calefactor_alto = 6

    print(temperatura_interna + calefactor_alto >= temperatura_externa)

False
True
True
```

Menor que y Menor o igual

Ahora que ya se conocen los operadores de mayor y mayor o igual, también existen los operadores contrario, es decir menor < y menor o igual <=, igual son operadores binarios por lo que ocupan un argumento a la izquierda y otro a la derecha, si la expresión se evalúa verdadera devuelve True y False caso contrario.

Es posible combinar los operadores para obtener una expresión más compleja que evalúa (de par en par), todos los argumentos dados.

```
[ ] juan = 20
    pedro = 19
    javier = 50
    junior = 1

    print(juan > pedro)
    print(juan > pedro < junior)
    print(juan < pedro > junior)
    print(junior < pedro < juan < javier)
    print(pedro + junior > juan)
    print(pedro + junior >= juan)
    print(juan - junior == pedro)

True
False
False
True
False
True
True
```



Los operadores de condición son muy utilizados dentro de los condicionales o estructuras de control de flujo, sin embargo, también pueden ser almacenados en variables para operar más adelante o controlar la ejecución o no de un bloque de código.

Ejemplo: En un zoológico, es necesario contar con un operador de leones por cada 2 leones, es decir si el zoológico cuenta con 10 leones, debe contar al menos con 5 cuidadores, aunque es un problema simple, es la base de la estructura de las comparaciones que se necesita para conocer su funcionamiento y aplicar una evaluación dentro del programa y así, finalmente, almacenar su resultado.

```
[ ] import math

leones_zoo = 23
cuidadores_leones = 10

cuidadores_requeridos = math.ceil(leones_zoo / 2)

print(cuidadores_requeridos)

cumple_norma = cuidadores_leones >= cuidadores_requeridos

print(cumple_norma)

12
False
```

Ejercitación

Imagine que tiene una bañera cuya capacidad es de 500 litros, la puede llenar un ritmo de 200 litros por hora, pero tenga en cuenta que al llegar a su capacidad maxima el agua se desborda.

Ahora usted quiere ingresar en la bañera sin que se desborde el agua, suponga que usted ocupa un volumen de 120 litros.



Para conocer si luego de 2 horas se puede meter a la bañera, escriba un programa que entregue la respuesta, recuerde crear variables por si alguna condición cambia en el tiempo.

Si la altura de la bañera son 80 centímetros, agregar un sifón de desagüe a una altura de 70 centímetros, permite que al llenarla y meterse a la bañera el agua rebase el límite?

Ejercitación

En España, las rebajas de invierno suelen comenzar entre los días 1 y 7 de enero y finalizan a final de marzo. Por otro lado, las rebajas de verano empiezan durante la primera semana del mes de julio y finalizan durante el mes de septiembre.

Para aprovechar la temporada de rebajas de invierno he salido de compras. He pagado 34€ por unos pantalones que tenían un descuento del 15%. ¿Qué precio tenían antes de aplicar el descuento?

Si el mismo pantalón cuesta 30€ en la temporada de rebajas de verano y viene con un descuento del 5%, es mejor comprarlo en invierno o en verano? compare los precios y retorne True o False

Análisis del problema

- ¿Qué me pide el problema?
- ¿Qué datos necesito?
- ¿Cómo debo responder?