**OPERADORES ARITMETICOS**

Las cadenas pueden declararse con comillas simples para ocupar menos caracteres

Esta es una lista de los operadores aritméticos en Python:

print(f'Suma {num + num2}')

print(f'Resta {num - num2}')

print(f'Multiplicacion {num \* num2}')

print(f'Division {num / num2}')

print(f'Division entera {num // num2}')

print(f'Potencia {num \*\* num2}')

print(f'Modulo {num % num2}')

A su vez, se presenta un ejemplo de lo que es un **f-literal**, el equivalente a un template string en JS, que se encargar de imprimir tanto cadenas como variables en una misma. Cabe resaltar que el código que se introduce entre llaves es código Python y se ejecuta como tal.

**OPERADORES DE ASIGNACION**

El operador de asignación por excelencia es el =. Los operadores +=, -=, \*=, /= funcionan igual que en JS.

**OPERADORES DE COMPARACION**

Los operadores de comparación en Python son <, >, ==, >=, <=, !=

**OPERADORES LOGICOS**

Los operadores lógicos en Python son **and, or y not**

**F-STRING CON VARIAS LINEAS**

Funciona igual que el template string de js, solo que cambia la sintaxis para este caso.

Las comillas que se aplican luego de la **f’’’** y al final, son triples

print(f'''

nombre: {nombre}

isbn: {isbn}

precio: {precio}

envio gratis: {envio}

''')

**OTRAS CONVERSIONES**

Por defecto los valores que retorna la funcion input() son de class **string**

Si se quisiese cambiar estos valores a otro tipo podría usarse.

bool(valor)

float(valor)

int(valor)

es importante tener que en cuenta que la conversión bool() como tal, retorna **True** en cualquier caso distinto a una cadena vacía.