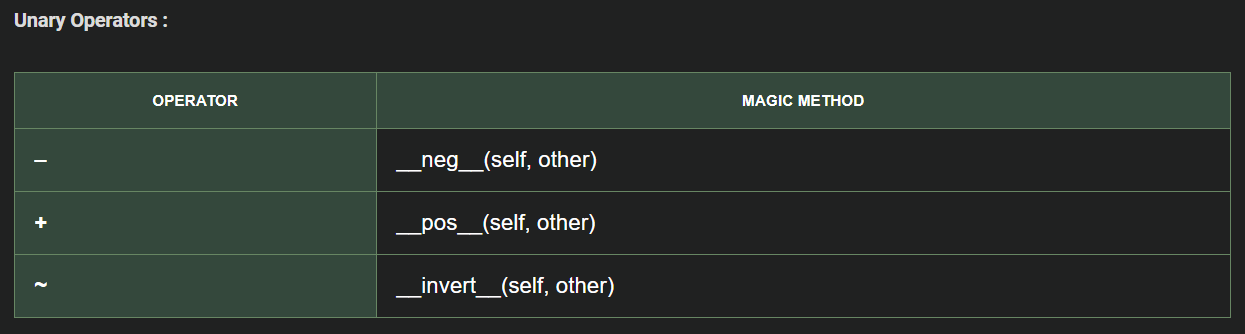
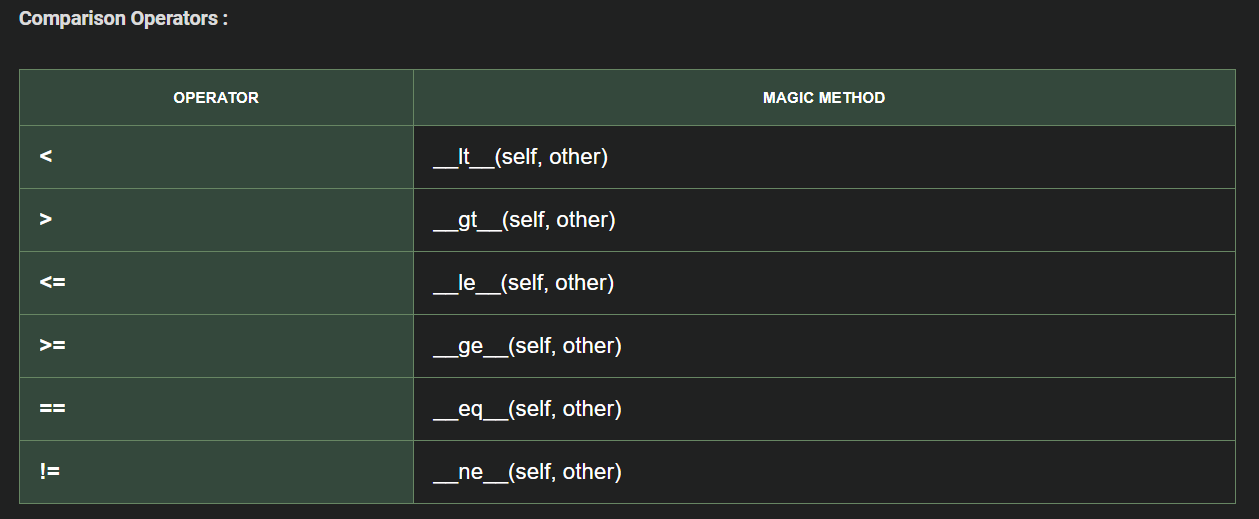
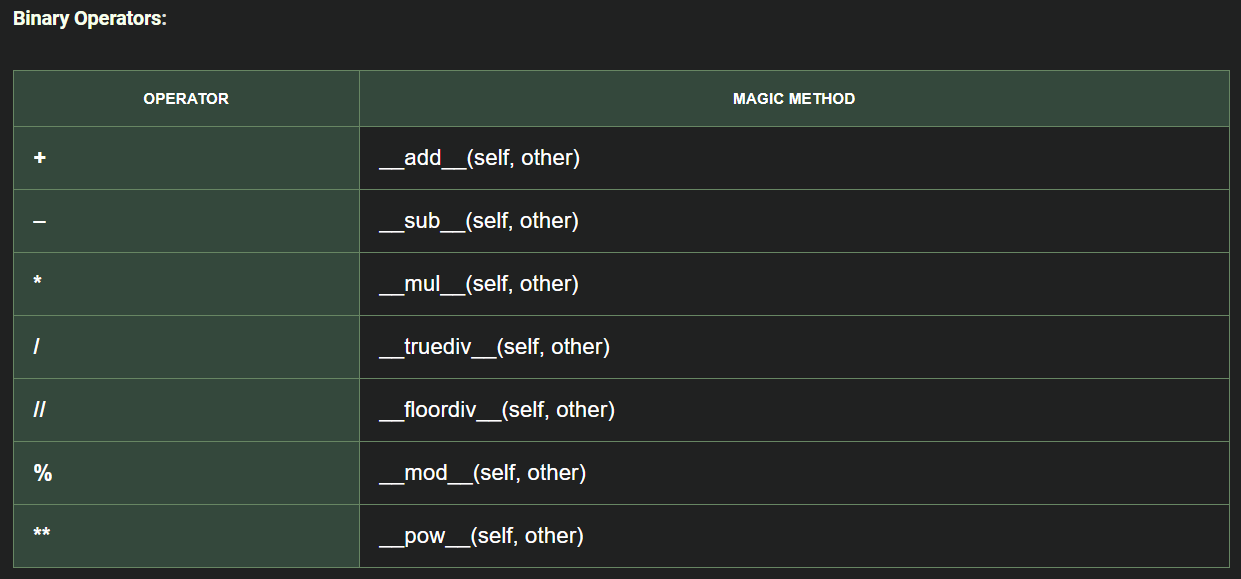
**SOBRECARGA DE OPERADORES EN PYTHON**

El concepto de sobrecarga de operadores implica que un mismo operador pueda comportarse de diferentes formas, dependiendo la situación.

Por ejemplo, **+** puede concatenar, operar o hasta incluso unir 2 listas, dependiendo de la situación.

La magia ocurre al operar con clases.

Estos son los métodos que hay que sobrescribir para operar con las mismas



Estos métodos al ser dunder (\_\_.\_\_) se heredan de la clase **object**

**SOBRESCRIBIENDO ADICION PARA LAS CLASES**

Para este ejemplo, se diseña una clase Human y el método \_\_add\_\_() se encargara de concatenar los valores de sus atributos

class Human:

    def \_\_init\_\_(self, name, age) -> None:

        self.name = name

        self.age = age

    def \_\_add\_\_(self, other):

        if(isinstance(other, object)):

            age = self.age + other.age

            name = f'{self.name}{other.name}'

            return (f'Age: {age} , Name: {name}')

obj1 = Human('Agustin', 18)

obj2 = Human('Nacho', 6)

Ahora la operación sería la siguiente:

suma = obj1 + obj2

print(suma)

Cuando se invoca a la suma de dos objetos, lo que se hace internamente es:

obj1.\_\_add\_\_(obj2)

Solo que el IDE simplifica la sintaxis para que podamos escribir obj1 + obj2