Python Operator Overloading

The python operators work for built-in classes.but same operator behaves differently with different type.for example,the + operator will perform arithmetic addition on two numbers,merge two list and concatenate two string.this feature in python,that allows same operator to have different meaning according to the context is called operator overloading the following

Example

```
import math

class Circle:

    def __init__(self, radius):
        self.__radius = radius

    def setRadius(self, radius):
        self.__radius = radius

    def getRadius(self):
        return self.__radius

    def area(self):
        return math.pi * self.__radius ** 2

    def __add__(self, another_circle):
        return Circle( self.__radius + another_circle.__radius )

c1 = Circle(4)
print(c1.getRadius())

c2 = Circle(5)
print(c2.getRadius())

c3 = c1 + c2 # This became possible because we have overloaded + operator by adding a    method named __add__
print(c3.getRadius())
```

Operator Overloading Special Functions in Python

Operator	Expression	Internally
Addition	no1 + no2	no1._add_(no2)
Subtraction	no1 - no2	no1._sub_(no2)
Multiplication	no1 * no2	no1._mul_(no2)
Power	no1 ** no2	no1._pow_(p2)
Division	no1 / no2	no1._truediv_(no2)
Floor Division	no1 // no2	no1._floordiv_(no)
Remainder	no1 % no2	no1._mod_(no2)
Bitwise Left Shift	no1 << no2	no1._lshift_(no2)
Bitwise Right Shift	no1 >> no2	no1._rshift_(no2)
Bitwise AND	no1 & no2	no1._and_(no2)
Bitwise OR	no1 | no2	no1._or_(p2)
Bitwise XOR	no1 ^ no2	no1._xor_(no2)
Bitwise NOT	~no1	no1._invert_()

Comparision Operator Overloading in Python

Operator	Expression	Internally

Operator	Expression	Internally
Less Than	no1 < no2	no1._lt_(no2)
Less Than or Equal to	no1 <= no2	no1._le_(no)
Equal to	no1 == no2	no1._eq_(no2)
Not equal to	no1 != no2	no1._ne_(no2)
Greater Than	no1 > no2	no1._gt_(no2)
Greater Than or Equal to	no1 >= no2	no1_ge_(no2)