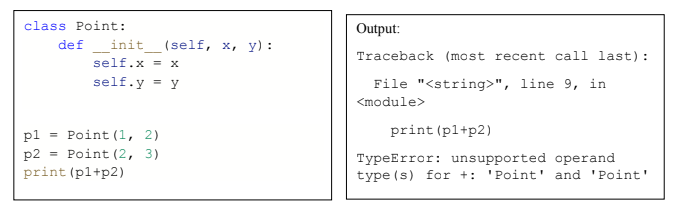
**Operator overloading**

Python operators work for built-in classes. But the same operator behaves differently with different types.

For example, the “ +” operator will perform arithmetic addition on two numbers, merge two lists,

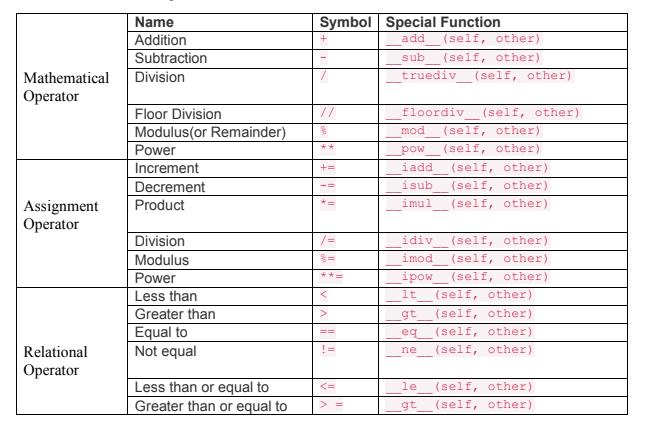
or concatenate two strings. This feature in Python that allows the same operator to have different meanings according to the context is called operator overloading.



**Here, we can see that a “Type Error” was raised, since Python doesn’t know how to add two sum objects together.** However, we can achieve this task in Python through operator overloading. But before that let’s get a notion about special functions.

**Python Special Function:**

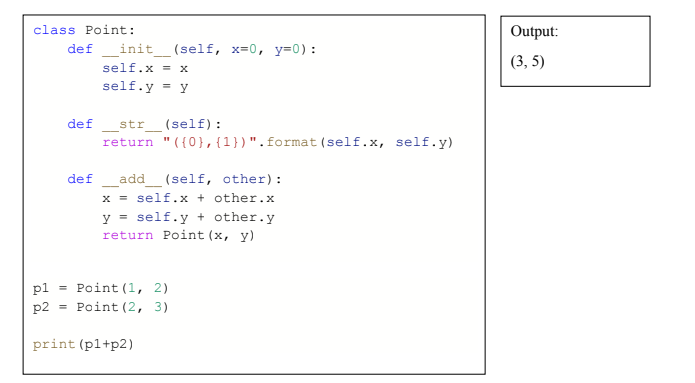
Special functions in python are the functions that are used to perform special tasks. These special functions have \_\_(double underscore) as prefix and suffix to their name as we see in **\_\_init\_\_()** the method which is also a special function.



**Example:**

To overload the “+” operator, we will need to implement \_\_add\_\_() function in the class. In the

below code example we will overload the + operator.

****

In the program above, \_\_add\_\_() is used to overload the + operator i.e. when + operator is used

with two Point class objects then the function \_\_add\_\_() is called.   
\_\_str\_\_() is another special function which is used to provide a format of the object that is suitable for printing. when we use p1 + p2, Python calls p1.\_\_add\_\_(p2) which in turn is

Point.\_\_add\_\_(p1,p2). After this, the addition operation is carried out the way we specified.

