Full stack web development using python

# Operator overloading

Saurabh Shukla (MySirG)

# Agenda

1. Define Line class
2. Operator overloading
3. Example
4. Magic Functions.

# Define a class Line

```
class Line:
    def __init__(self, l):
        self.length = l
    def show_length(self):
        print("Length=", self.length)

l1 = Line(10)
l2 = Line(20)
l3 = l1 + l2
l3.show_length()
```

What is the output?

# Operator Overloading

Giving extended meaning beyond their predefined meaning.

We can define all existing operators, but can't define a new operator.

# Example

```
class Line:
    def __init__(self, l):
        self.length = l
    def show_length(self):
        print("Length=", self.length)
    def __add__(self, other):
        return Line(self.length + other.length)

l1 = Line(10)
l2 = Line(20)
l3 = l1 + l2
l3.show_length()
```

# Magic Functions

| | |
|---|---|
| + | __add__ |
| - | __sub__ |
| / | __truediv__ |
| * | __mul__ |
| != | __ne__ |
| == | __eq__ |
| unary - | __neg__ |
| % | __mod__ |
| < | __lt__ |
| > | __rt__ |
| <= | __le__ |
| >= | __ge__ |
| // | __floordiv__ |
| >> | __rshift__ |