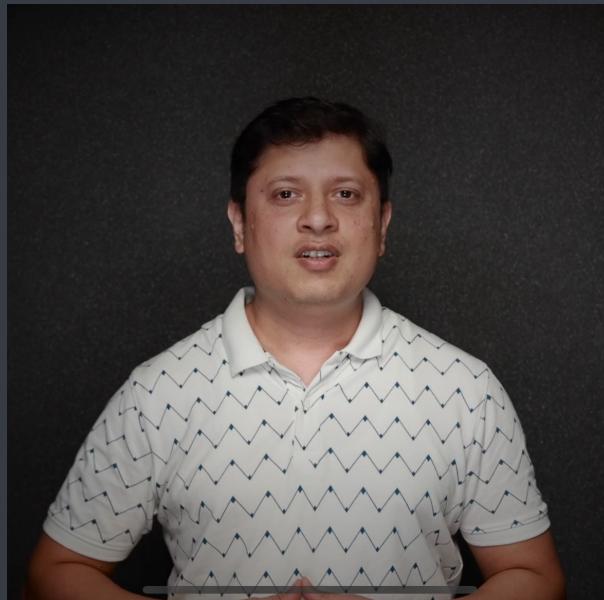


# Full stack web development using python

## name conflict issues



Saurabh Shukla (MySirG)

## Agenda

- ① Instance object variables name conflict
- ② class object variables name conflict
- ③ Instance method name conflict
- ④ static method name conflict
- ⑤ class method name conflict
- ⑥ Instance object variable & class object variable name conflict in single class

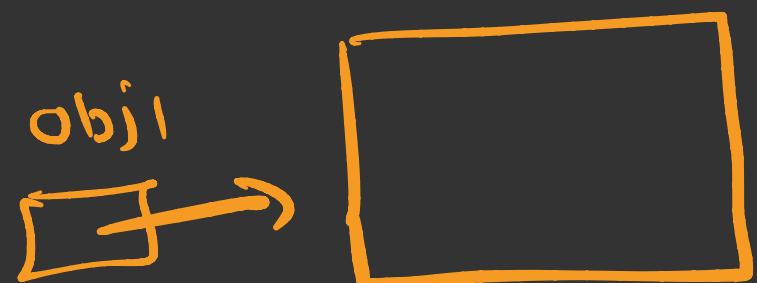
# Instance object variables name conflict

class A :

```
def __init__(self):  
    self.a = 5
```

class B(A) :

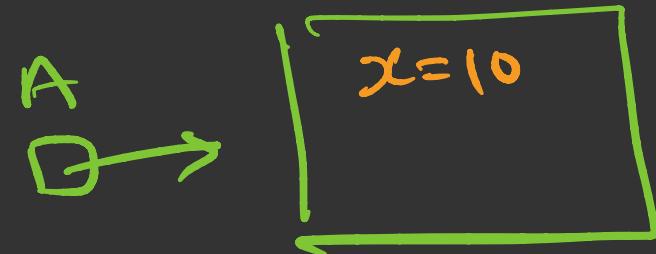
```
def __init__(self):  
    self.a = 4
```



# Class object variables name conflict

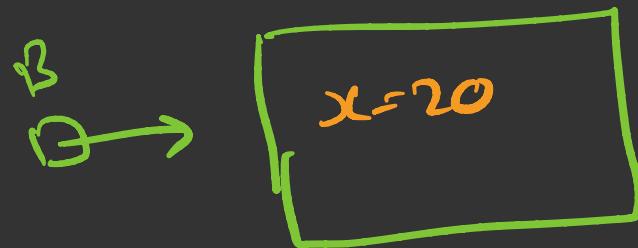
class A :

$x = 10$



class B(A) :

$x = 20$



# Instance method name conflict

## Case 1

class A:

```
def f1(self):  
    print("A")
```

class B(A):

```
def f1(self):  
    print("B")
```

method overriding

## case 2

class A:

```
def f1(self, a):  
    print("A")
```

class B(A):

```
def f1(self):  
    print("B")
```

method hiding

# Static method name conflict

Class A:

```
@staticmethod  
def f1(a,b):  
    print("A")
```

Class B(A):

```
@staticmethod  
def f1(a,b):  
    print("B")
```

Class A:

```
@staticmethod  
def f1(a,b):  
    print("A")
```

Class B(A):

```
@staticmethod  
def f1(a,b,c):  
    print("B")
```



# Class method name conflict

class A:

```
@classmethod  
def f1(cls,a,b):  
    print("A")
```

class B(A):

```
@classmethod  
def f1(cls,a,b):  
    print("B")
```

class A:

```
@classmethod  
def f1(cls,a,b):  
    print("A")
```

class B(A):

```
@classmethod  
def f1(cls,a,b,c):  
    print("B")
```

Instance object variable and  
class object variable name conflict  
in single class

class A :

x = 10

def \_\_init\_\_(self):  
 self.x = 15

class A:

x=10

obj = A()

obj.x=15

print(obj.x)

