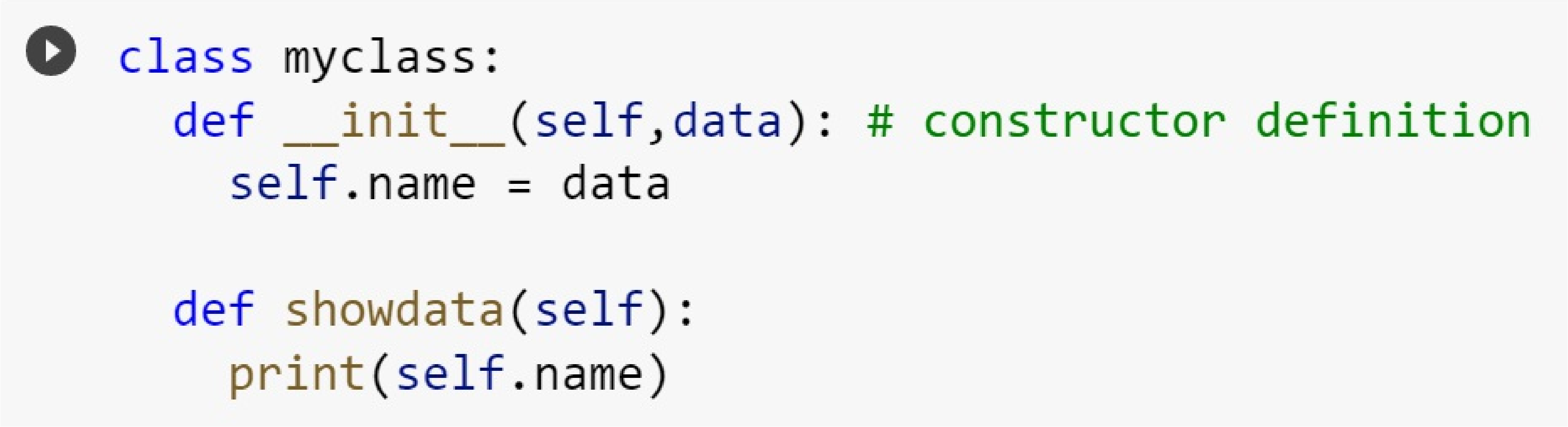
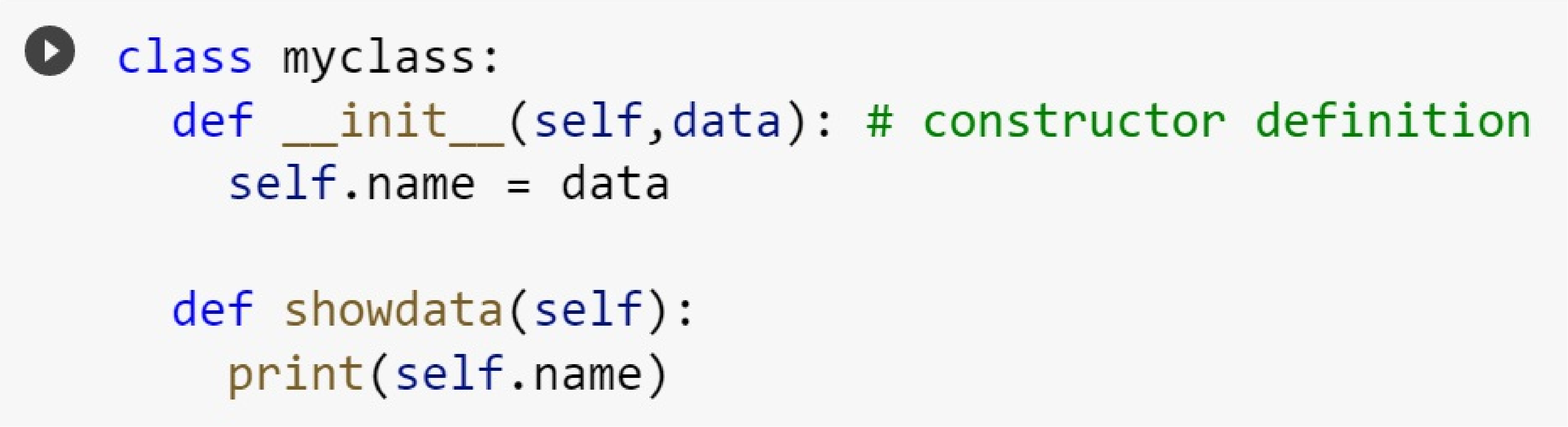
constructor with parameters

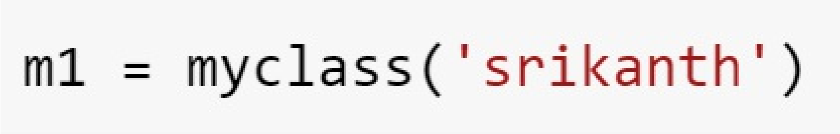


```
class myclass:
  def __init__(self,data): # constructor definition
    self.name = data

  def showdata(self):
    print(self.name)
```

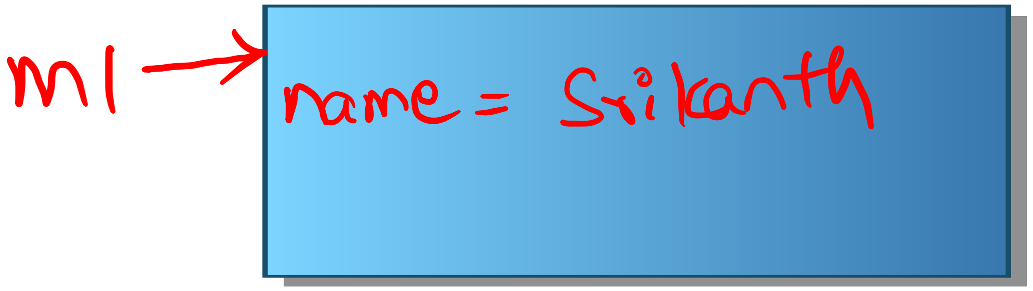
constructor with parameters



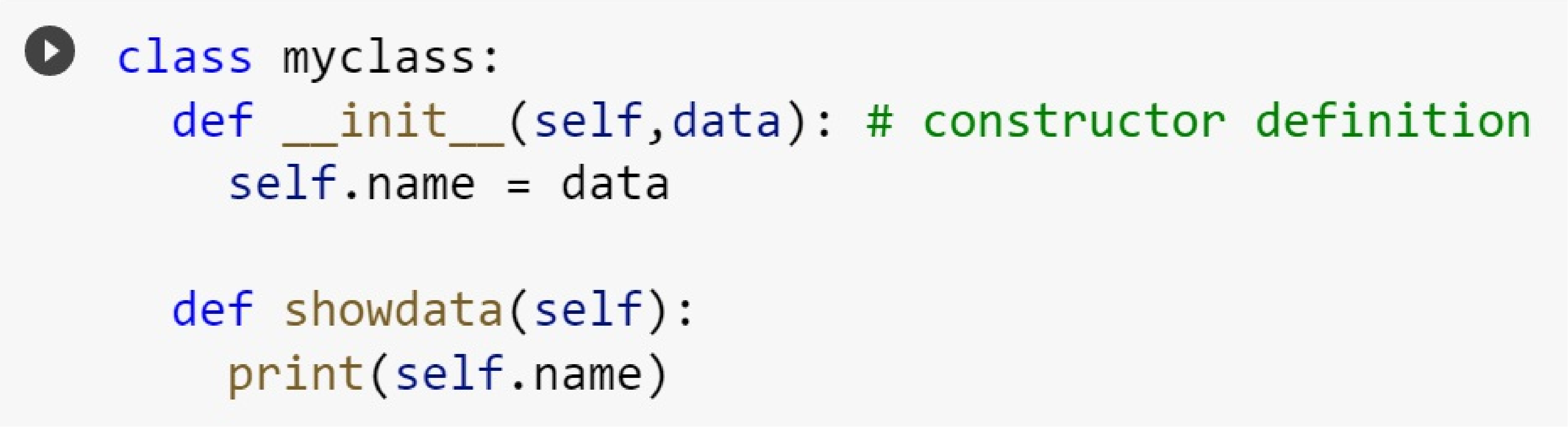
```python
class myclass:
  def __init__(self,data): # constructor definition
    self.name = data

  def showdata(self):
    print(self.name)
```



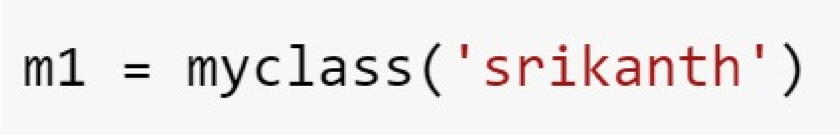
```python
m1 = myclass('srikanth')
```



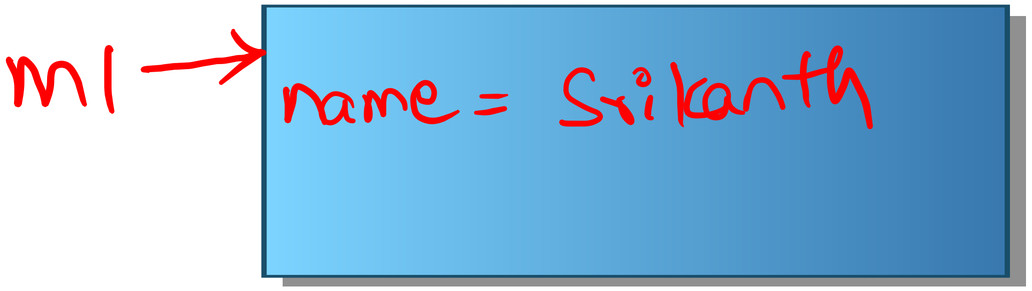
constructor with parameters

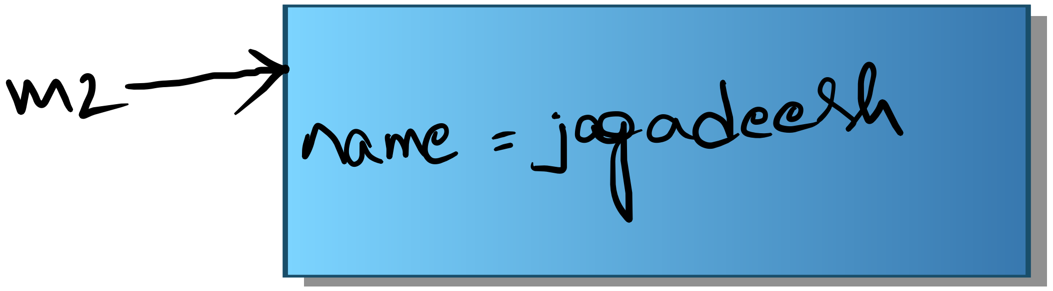


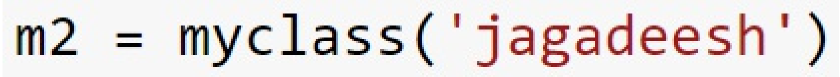
```
class myclass:
  def __init__(self,data): # constructor definition
    self.name = data

  def showdata(self):
    print(self.name)
```



```
m1 = myclass('srikanth')
```

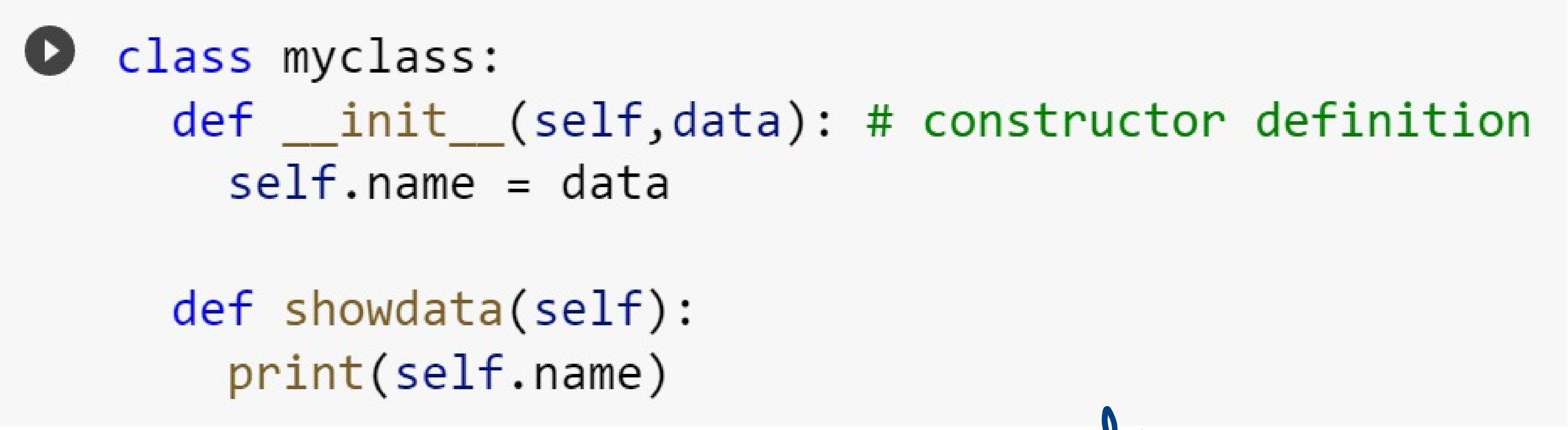




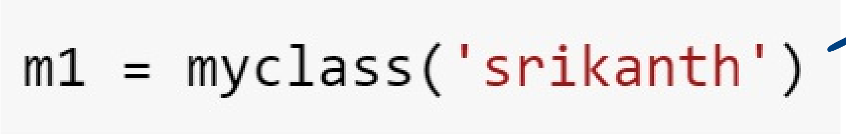


```
m2 = myclass('jagadeesh')
```

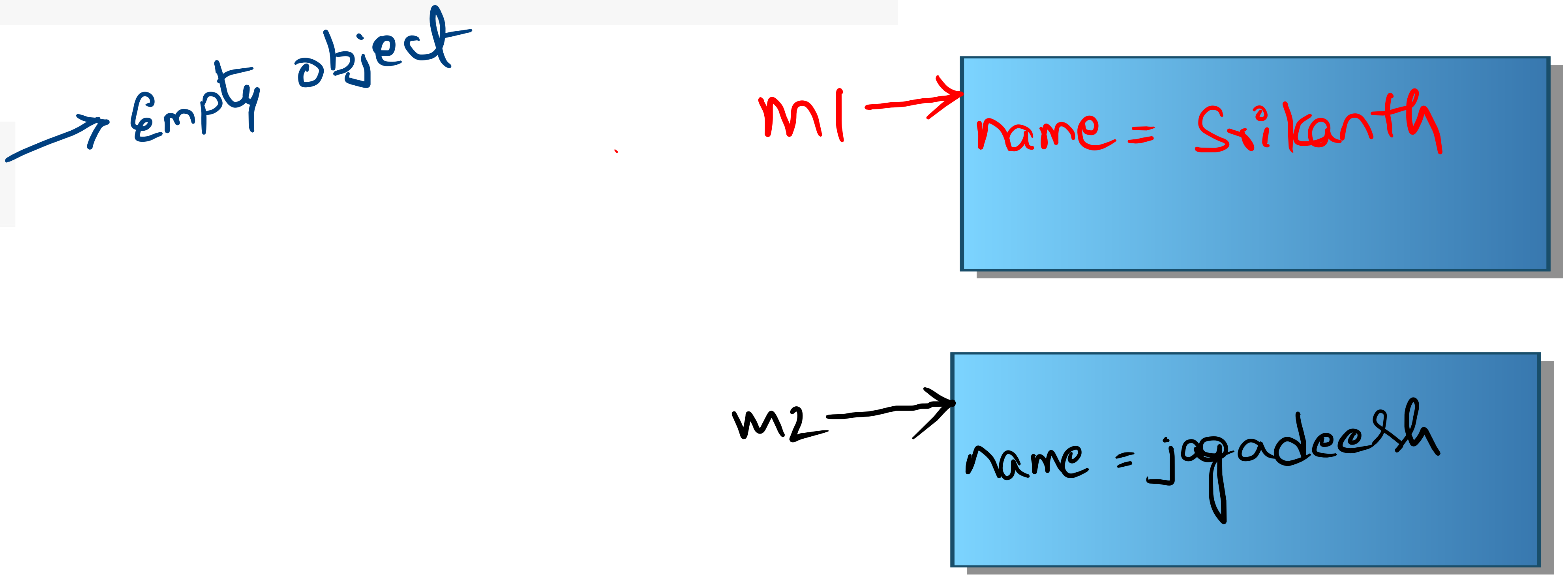
constructor with parameters

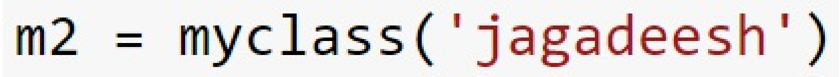


```
class myclass:
  def __init__(self,data): # constructor definition
    self.name = data

  def showdata(self):
    print(self.name)
```



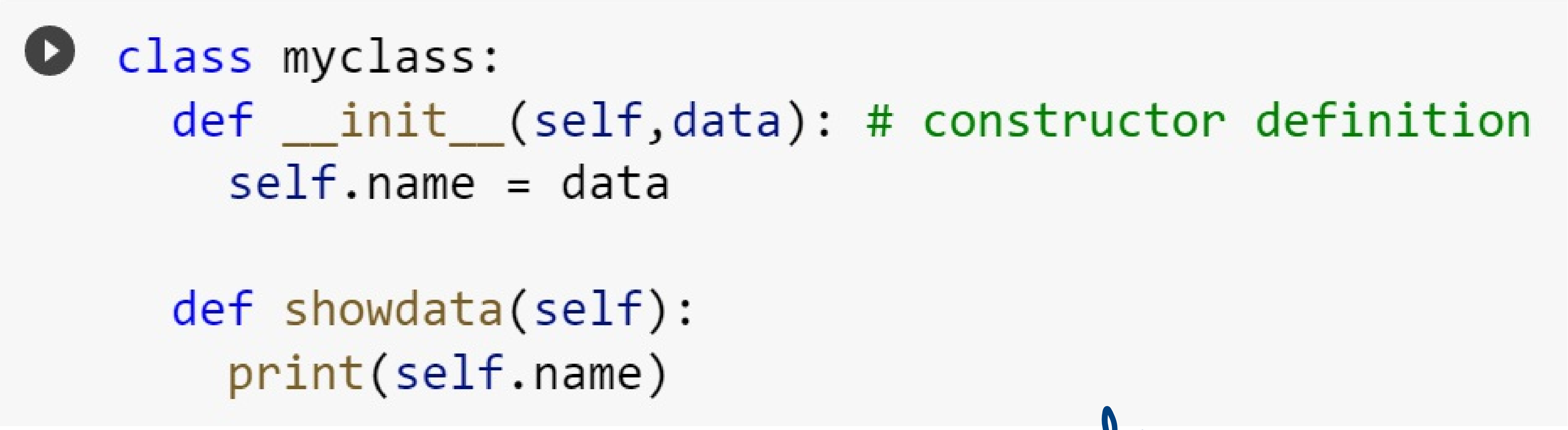
```
m1 = myclass('srikanth')
```



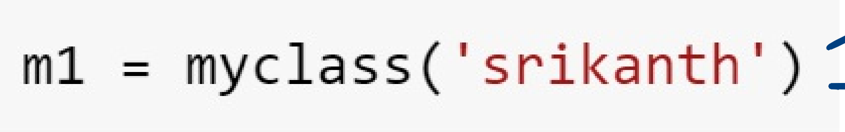


```
m2 = myclass('jagadeesh')
```

constructor with parameters



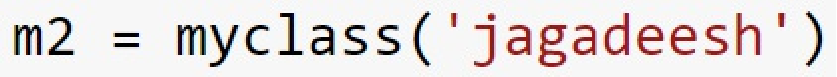
```
class myclass:
  def __init__(self,data): # constructor definition
    self.name = data

  def showdata(self):
    print(self.name)
```



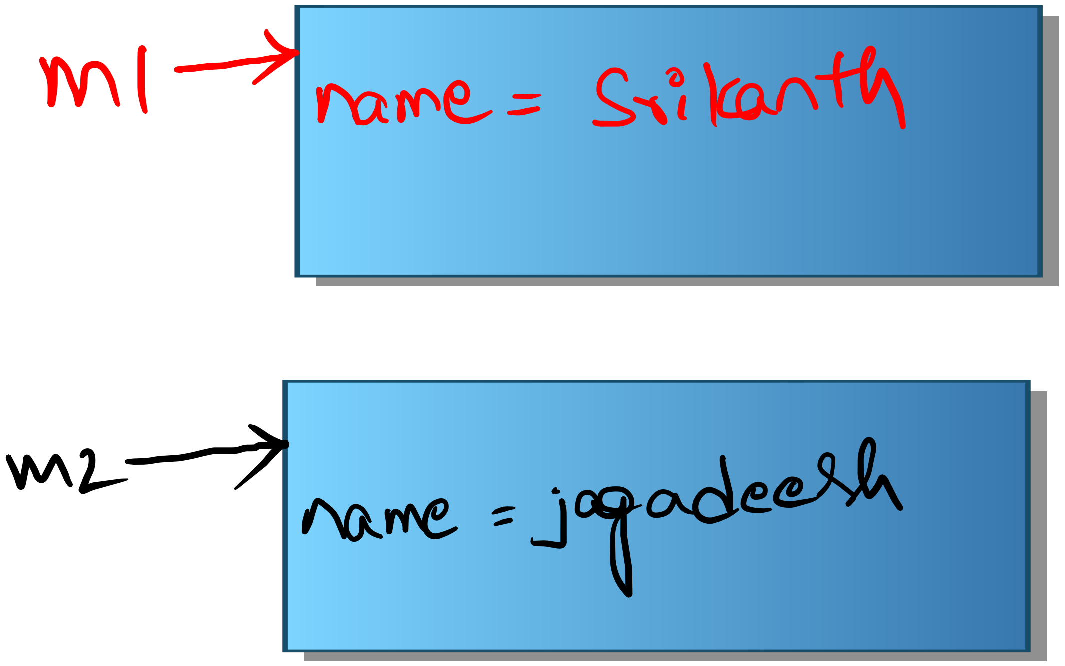
```
m1 = myclass('srikanth')
```

→ Empty object

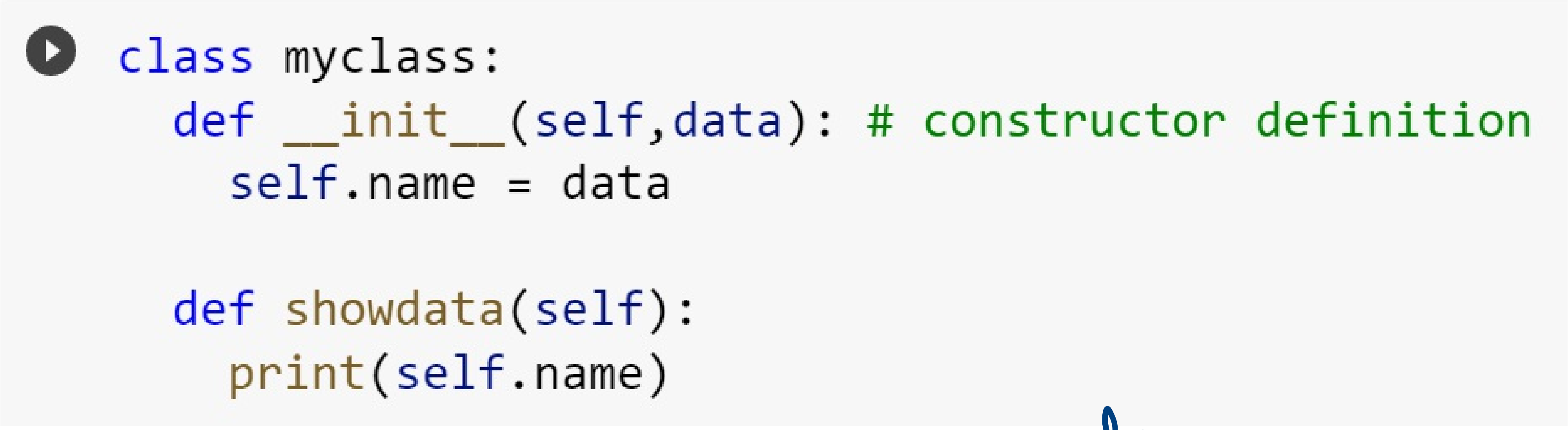
→ Constructor call



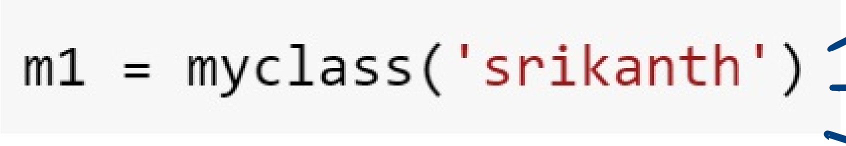
```
m2 = myclass('jagadeesh')
```



constructor with parameters



```
class myclass:
  def __init__(self,data): # constructor definition
    self.name = data

  def showdata(self):
    print(self.name)
```

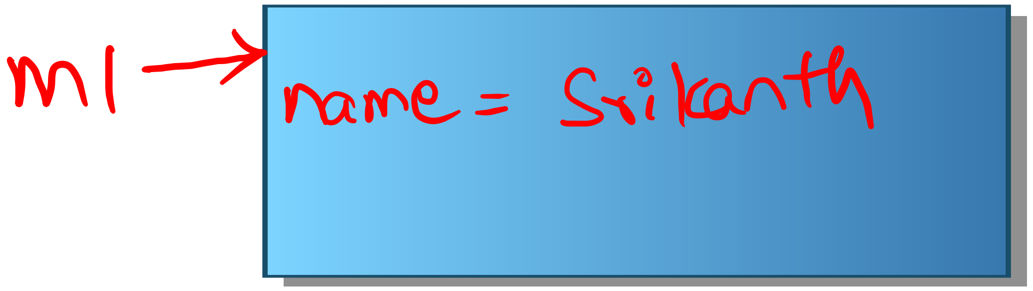


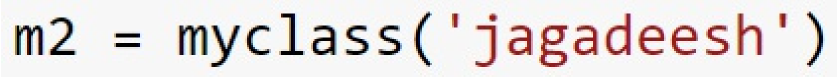
```
m1 = myclass('srikanth')
```

→ Empty object

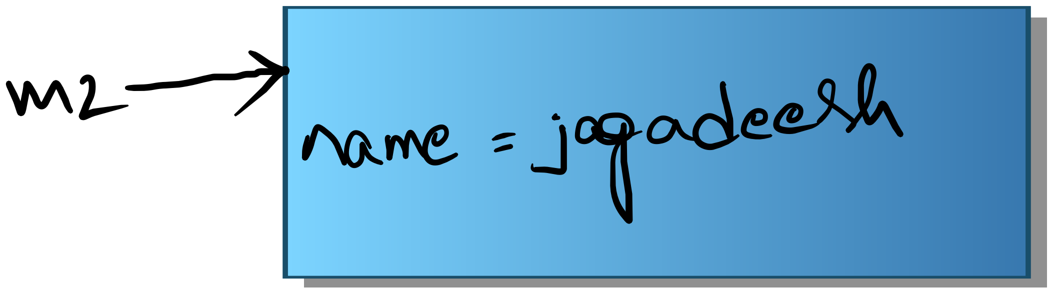
→ Constructor call

→ object initialization





```
m2 = myclass('jagadeesh')
```



m2
name = jagadeesh