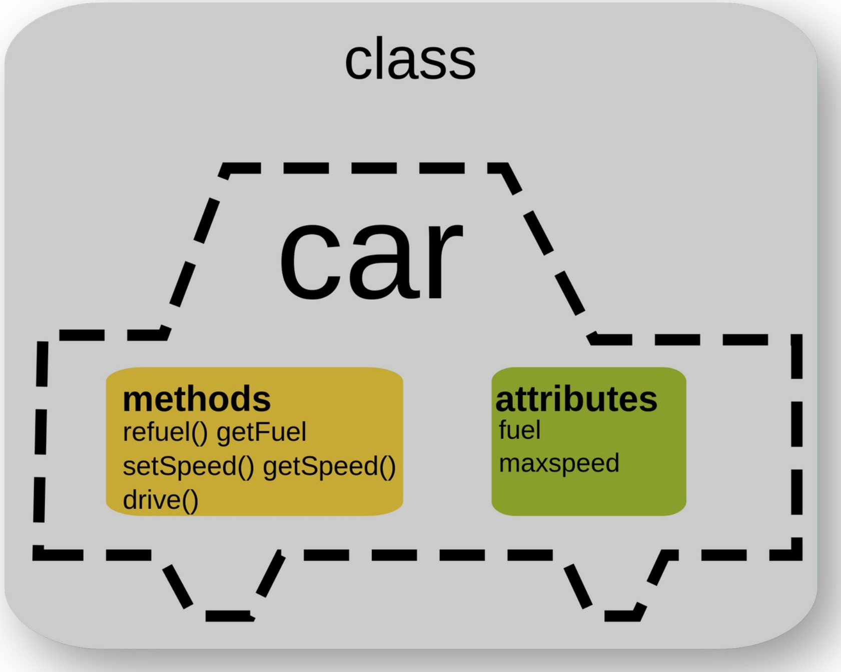
Module 0

Python Programming

Python Basics

Objects and Classes

- Objects are encapsulations of variables and functions into single entity
- Classes provide a way to group information into a single unit
    - Can combine multiple attributes into a single object
- Classes contain methods (functions) and attributes (values)



class

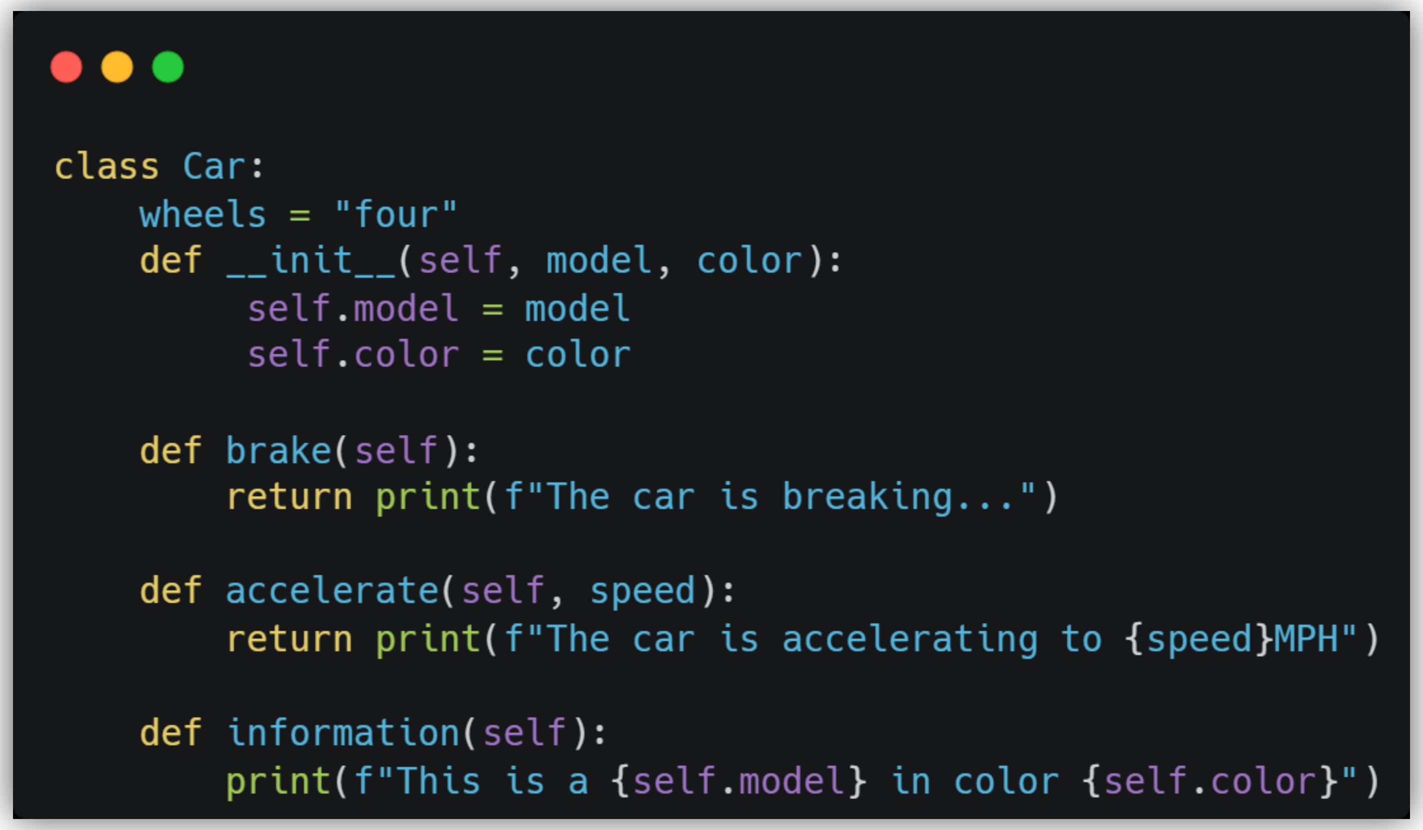
car

methods
refuel() getFuel
setSpeed() getSpeed()
drive()

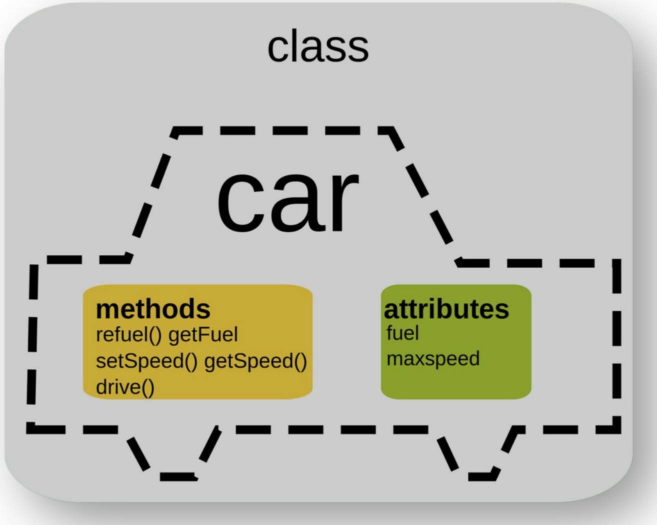
attributes
fuel
maxspeed

Example

- Class Car

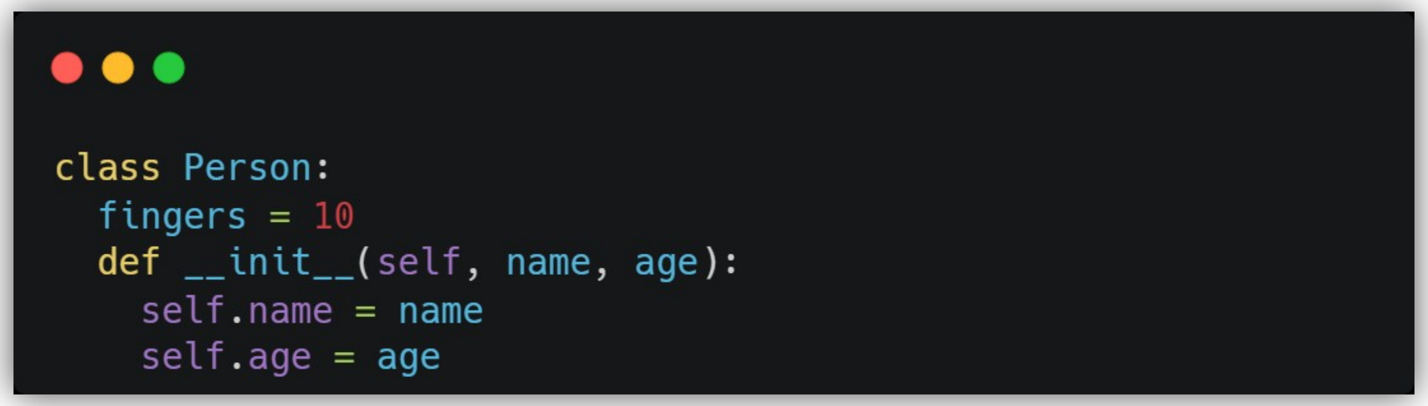


```python
class Car:
    wheels = "four"
    def __init__(self, model, color):
         self.model = model
         self.color = color

    def brake(self):
        return print(f"The car is breaking...")

    def accelerate(self, speed):
        return print(f"The car is accelerating to {speed}MPH")

    def information(self):
        print(f"This is a {self.model} in color {self.color}")
```



Attributes

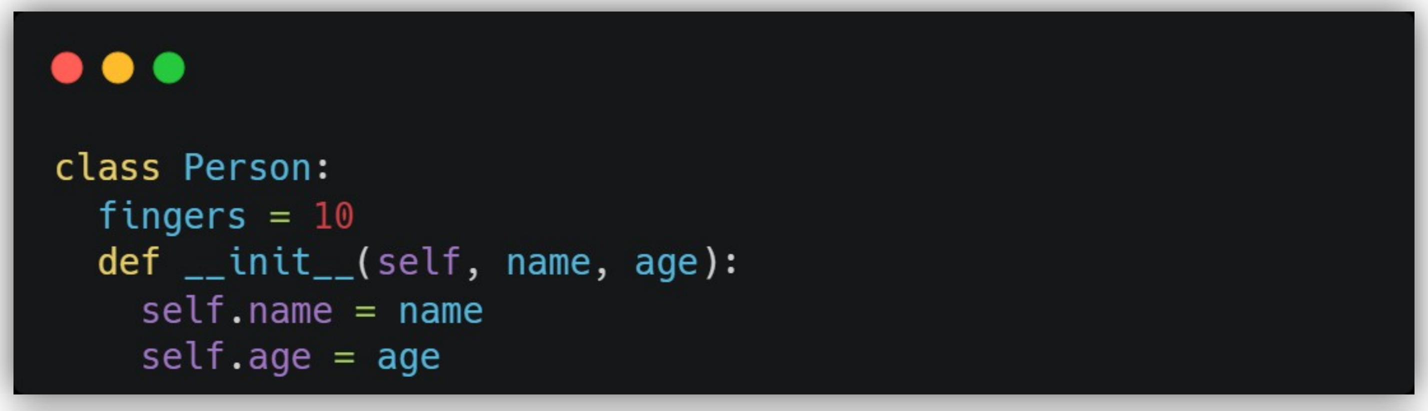
- Characteristics that can be accessed from other objects
- Can be mutable or immutable



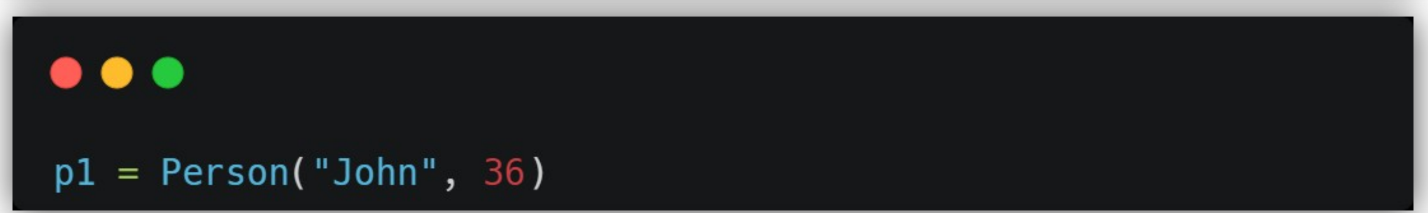
```python
class Person:
  fingers = 10
  def __init__(self, name, age):
    self.name = name
    self.age = age
```

Making an Instance (Instantiate)

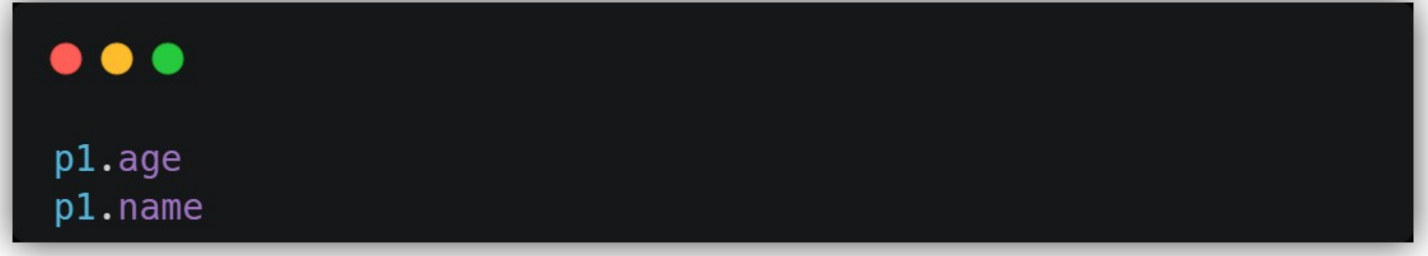
- Creating an instance object from the class



```python
class Person:
  fingers = 10
  def __init__(self, name, age):
    self.name = name
    self.age = age
```



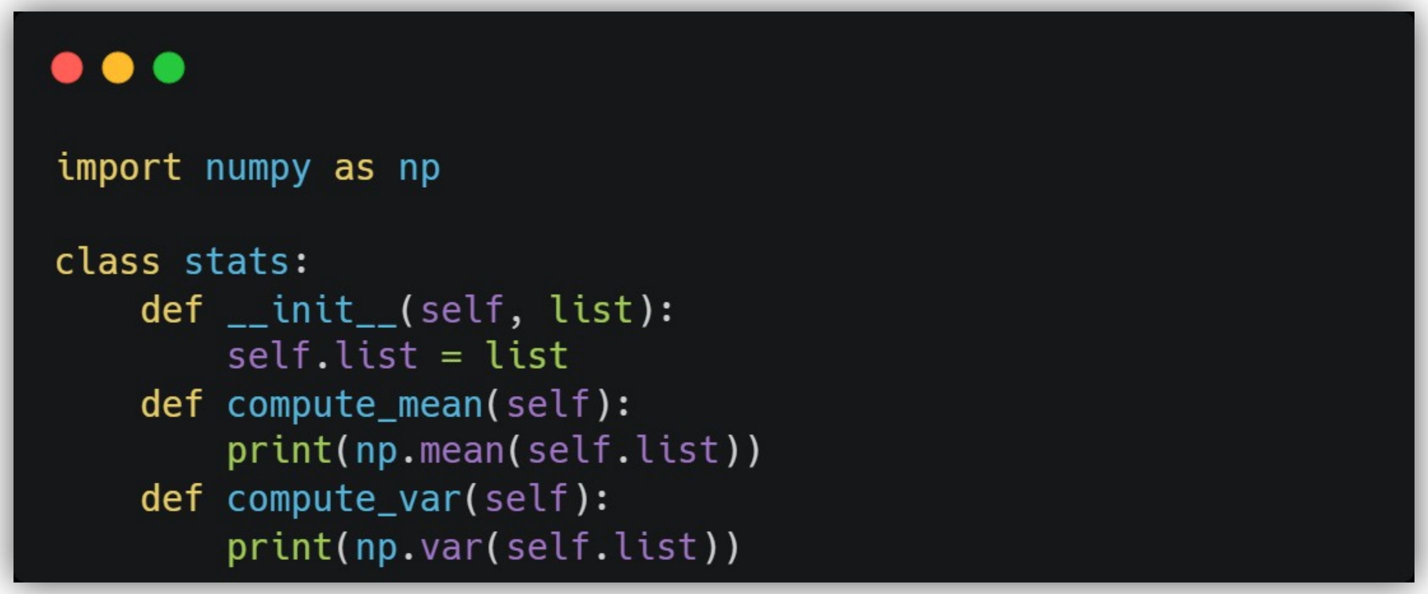
```python
p1 = Person("John", 36)
```



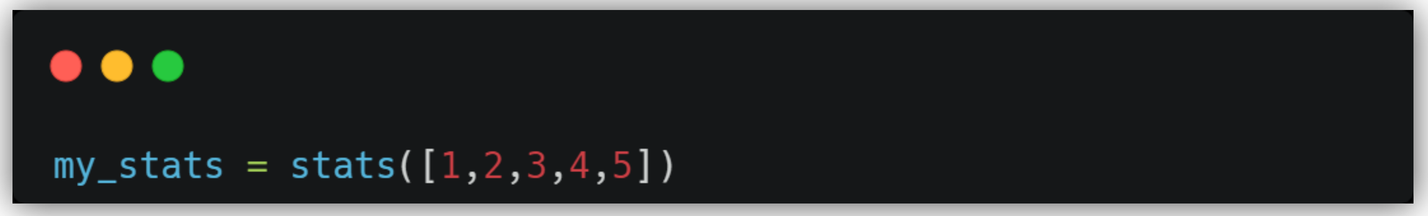
```python
p1.age
p1.name
```

Methods

❑ Enable classes to implement functions



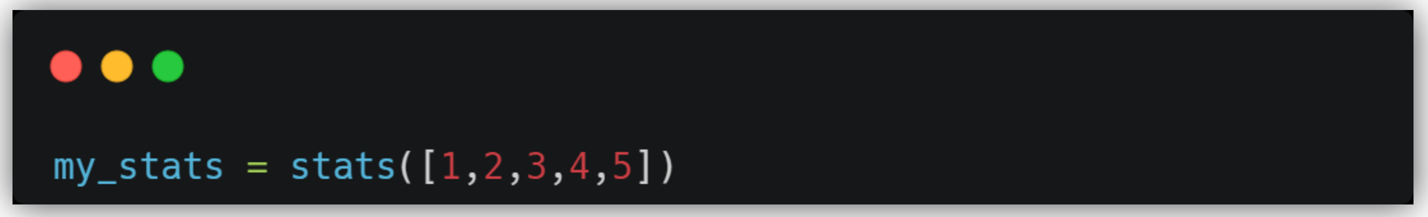
```python
import numpy as np

class stats:
    def __init__(self, list):
        self.list = list
    def compute_mean(self):
        print(np.mean(self.list))
    def compute_var(self):
        print(np.var(self.list))
```



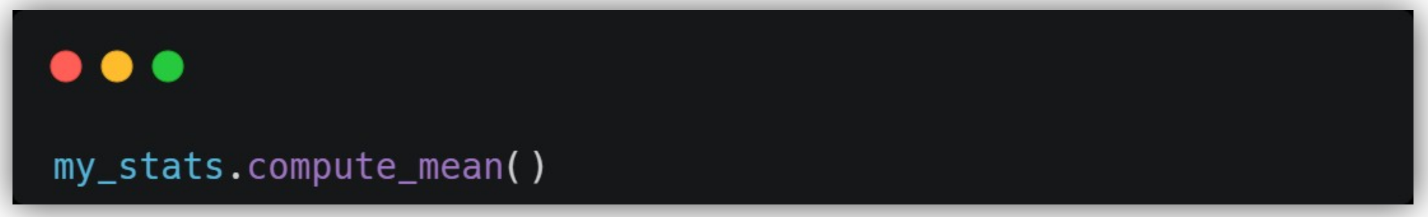
```python
my_stats = stats([1,2,3,4,5])
```

Methods

- Calling a method bound to the class

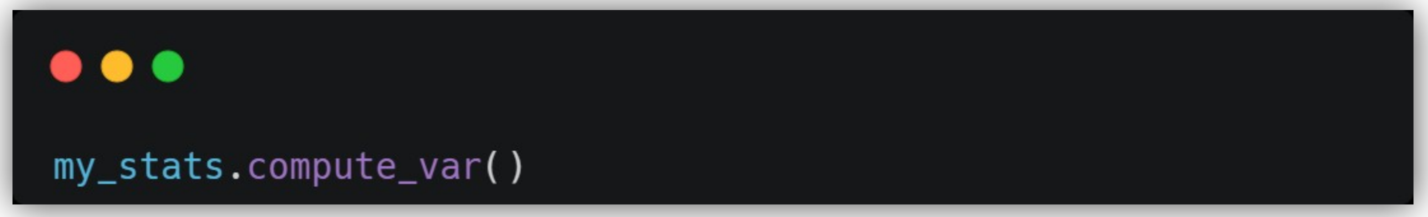


```
my_stats = stats([1,2,3,4,5])
```



```
my_stats.compute_mean()
```

```
3.0
```



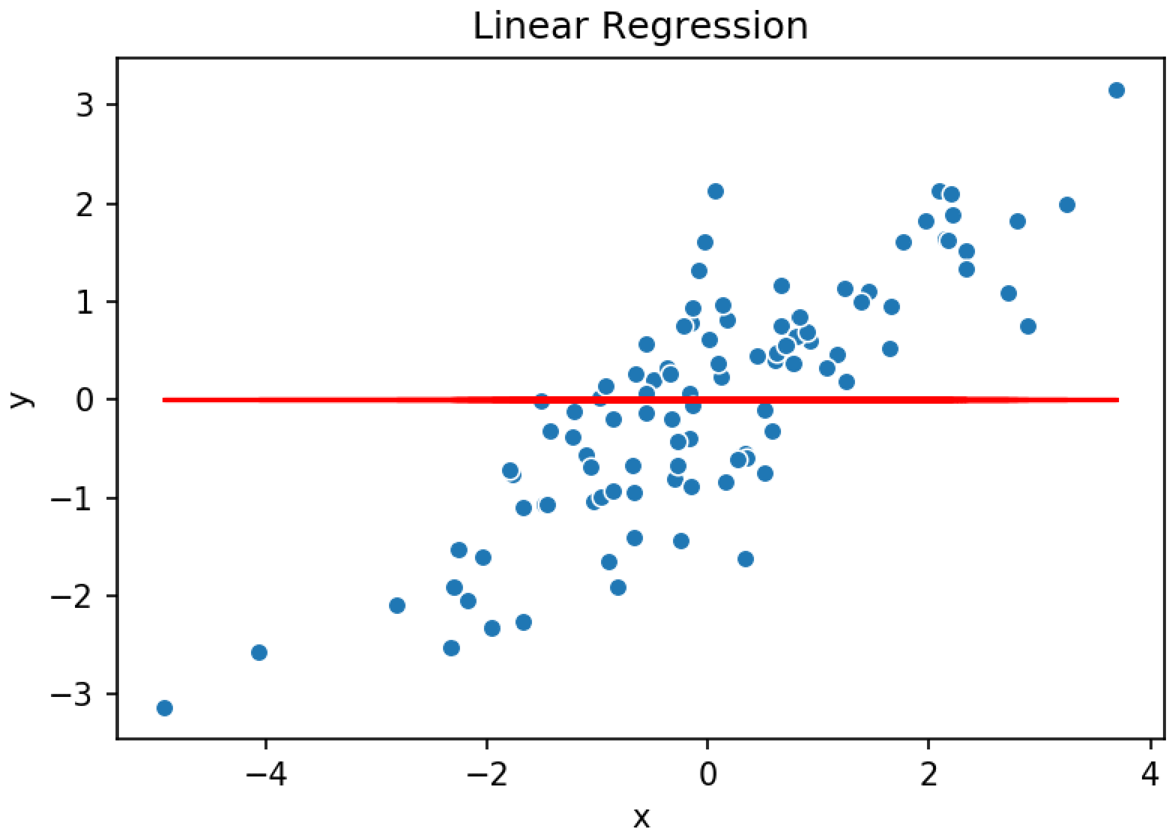
```
my_stats.compute_var()
```

```
2.0
```

Practice

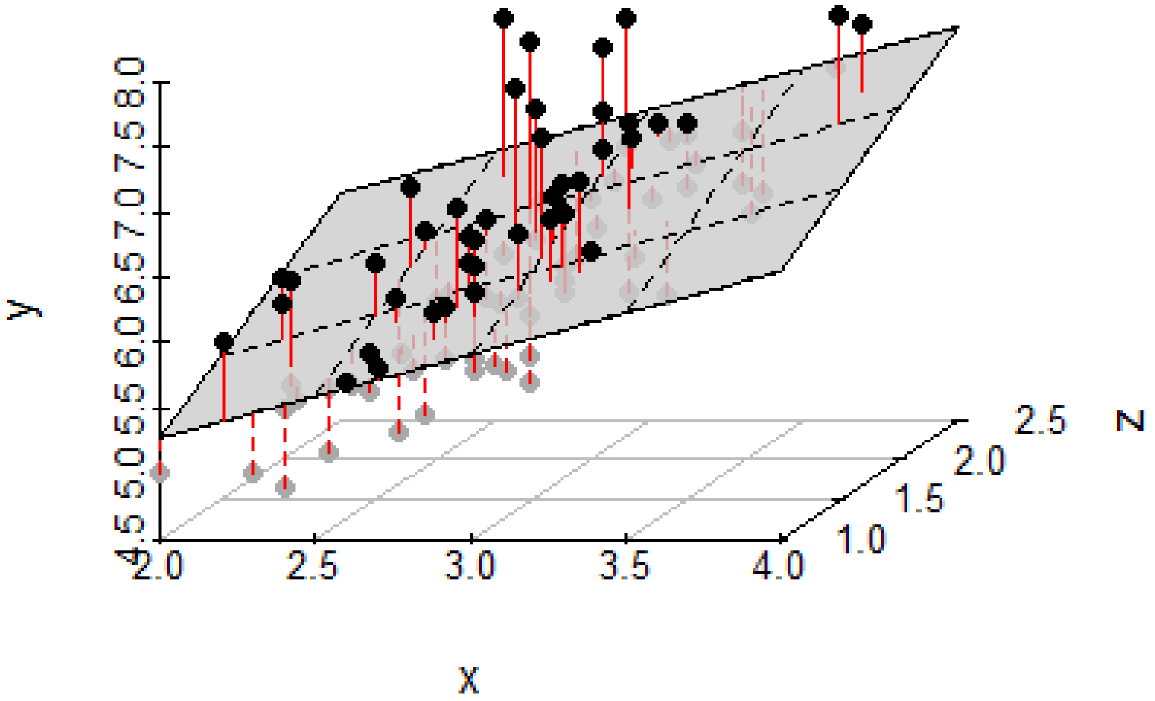
Linear Regression Example

- We minimize the sum of squared errors (SSE).



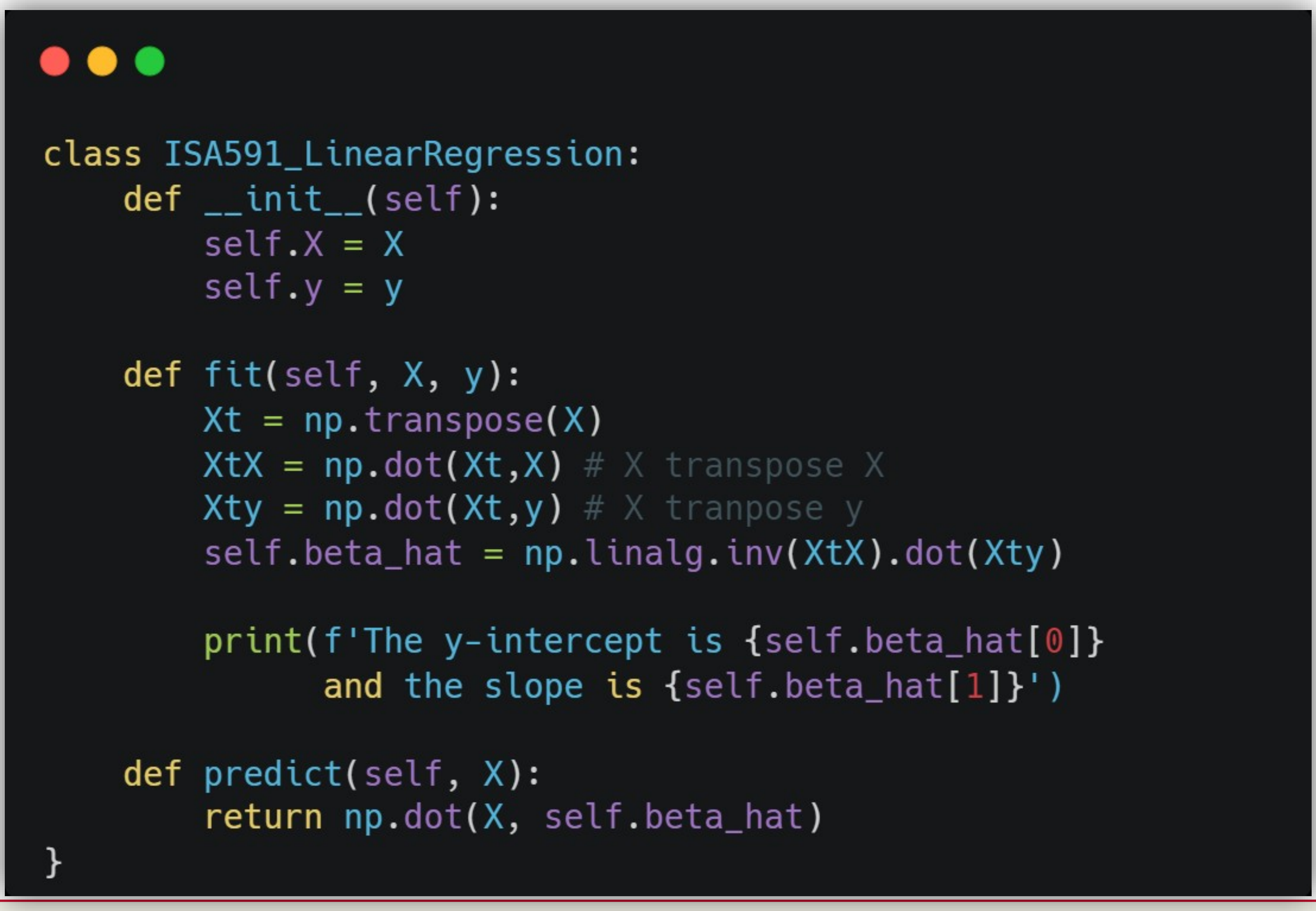
Linear Regression Example

- The vector of estimates can be fit with $\hat{\beta} = (X^\top X)^{-1} (X^\top y)$
- The predictions can be obtained with $\hat{y} = X\hat{\beta}$



Linear Regression Example

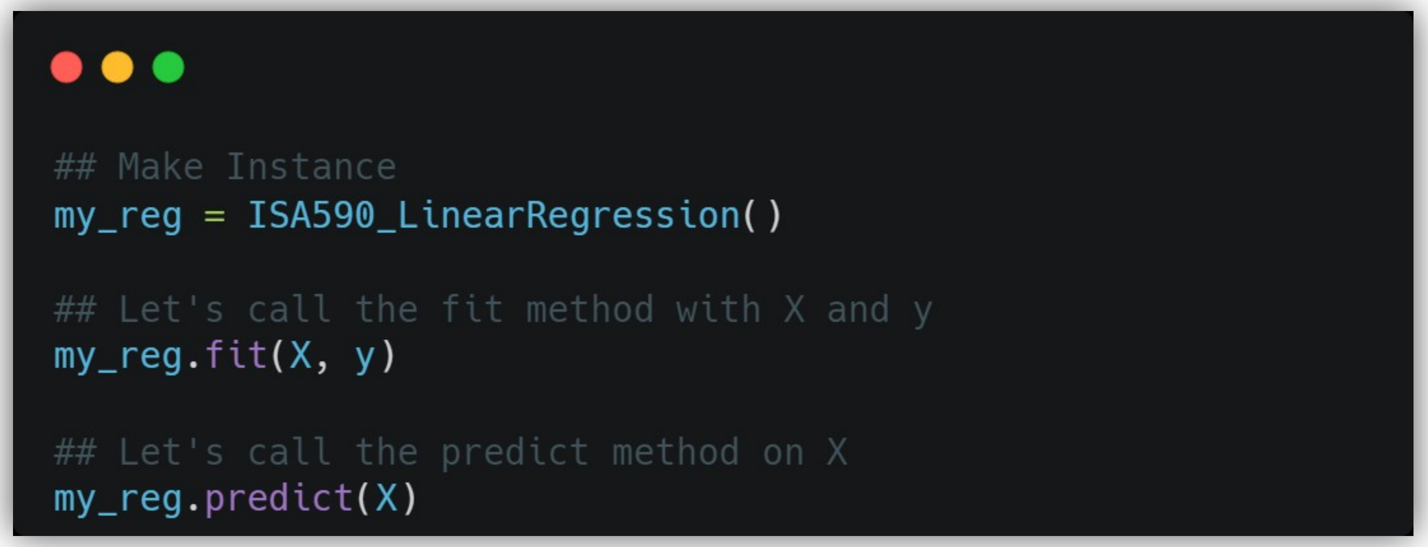
❑ Class implementation



```
class ISA591_LinearRegression:
    def __init__(self):
        self.X = X
        self.y = y

    def fit(self, X, y):
        Xt = np.transpose(X)
        XtX = np.dot(Xt,X) # X transpose X
        Xty = np.dot(Xt,y) # X tranpose y
        self.beta_hat = np.linalg.inv(XtX).dot(Xty)

        print(f'The y-intercept is {self.beta_hat[0]}
              and the slope is {self.beta_hat[1]}')

    def predict(self, X):
        return np.dot(X, self.beta_hat)
}
```

Linear Regression Example

- Class implementation



```
## Make Instance
my_reg = ISA590_LinearRegression()

## Let's call the fit method with X and y
my_reg.fit(X, y)

## Let's call the predict method on X
my_reg.predict(X)
```

Practice