Object-Oriented Programming

Two important concepts

- Objects
- Classes

Objects

- Sometimes pieces of data are clearly related:

```
cat1_name = “Fluffy”
cat1_wt = 13.5
cat1_color = “Brown”

cat2_name = “Mr. Scruffy”
cat2_wt = 9
cat2_color = “Grey”
```

Objects

- Sometimes functions are designed to work on pieces of related data:

```
def print_cat_info(name, wt, color):
  print name, “is a “, color,
  print “cat, and weighs”, wt, “lbs.”
```

Objects

- Objects allow us to *encapsulate* – to store data and functions that are related in a single package

```
cat1 = Cat(“fluffy”)
cat1.color = “brown”
cat1.wt = 14
cat1.print_cat_info()
```

Classes

- *Classes* are blueprints that are used to create objects. They define what data and what functions should be part of the object.
  - Variables, like cat1.name, are called ***members***
  - Functions, like cat1.print_cat_info, are called ***methods***

Classes

```
class Cat:
   def print_cat_info(self):
      print self.name, “is a cat” ...
```

Classes - constructors

```
class Cat:
  def __init__(self, name, weight=10):
    self.name = name
    self.weight = weight

  def print_cat_info(self):
    print self.name, “is a cat” ...
```

Classes - documentation

```
class Cat:
  '''A class to model cats.'''

  def __init__(self, name, weight=10):
    '''Create a cat. By default, the cat will
weigh ten pounds.'''
    self.name = name
    self.weight = weight

  def print_cat_info(self):
    '''Print some basic info about the cat.'''
    print self.name, “is a cat” ...
```