Object-oriented programming

in Python

Fields

```
x = 5
name = “Louis Brandeis”
tax_rate = 0.05
```

Methods

```
def hi():
    print “Hello, class!”
```

Class

The definition of a logical collection of fields and methods (attributes).

Object

An instance of a class. A realization of the class definition.

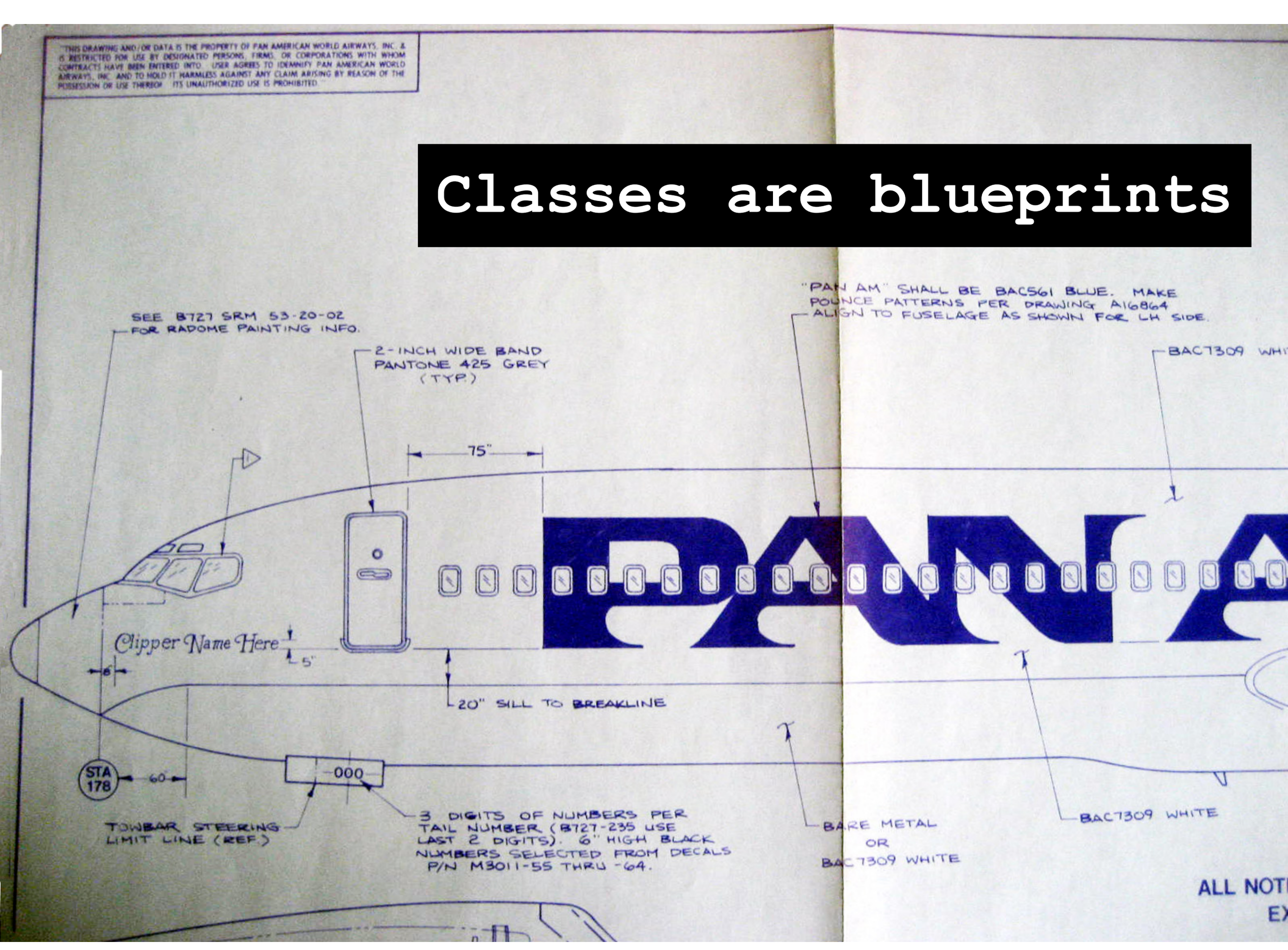
Person

height
weight
eye color
hair color
name

say_hi()

"THIS DRAWING AND/OR DATA IS THE PROPERTY OF PAN AMERICAN WORLD AIRWAYS, INC. & IS RESTRICTED FOR USE BY DESIGNATED PERSONS, FIRMS, OR CORPORATIONS WITH WHOM CONTRACTS HAVE BEEN ENTERED INTO. USER AGREES TO INDEMNIFY PAN AMERICAN WORLD AIRWAYS, INC. AND TO HOLD IT HARMLESS AGAINST ANY CLAIM ARISING BY REASON OF THE POSSESSION OR USE THEREOF. ITS UNAUTHORIZED USE IS PROHIBITED."

Classes are blueprints



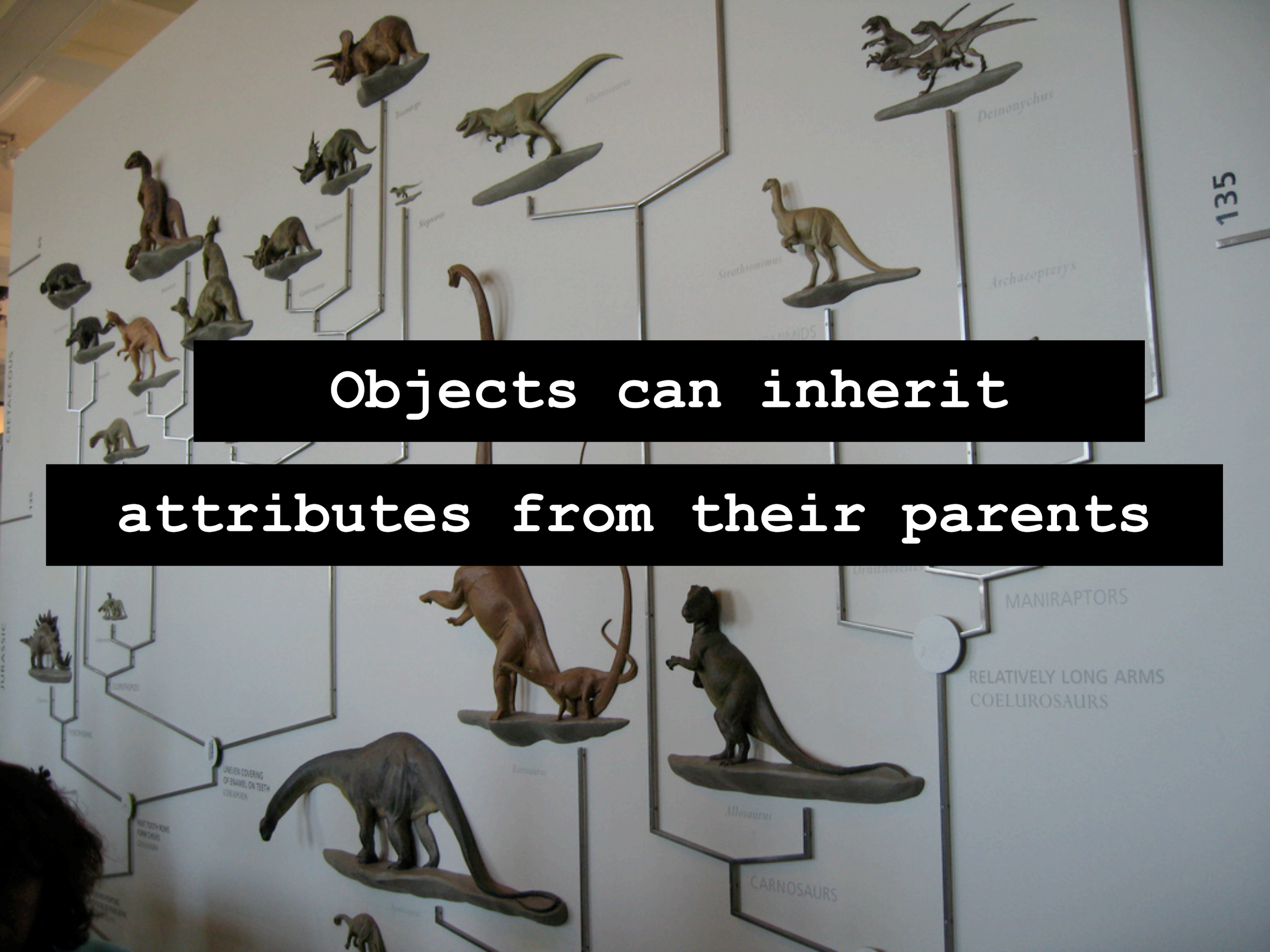
Objects are realizations

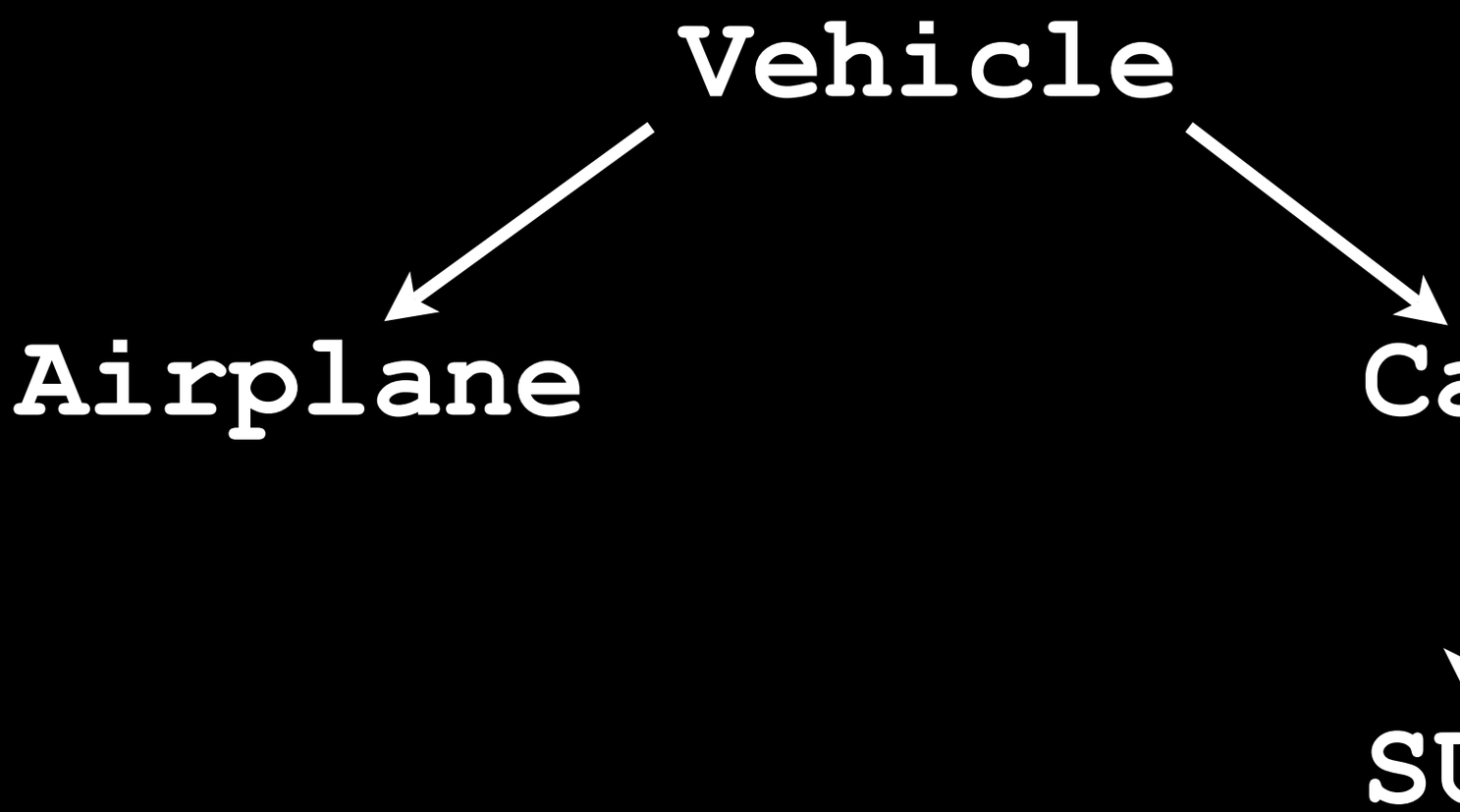
of class blueprints

```
class Person(object):
    name = “”
    def say_hi(self):
        print “Hi, my name is”, self.name

louis = Person()
louis.name = “Louis Brandeis”
louis.say_hi()
# Hi, my name is Louis Brandeis

revorg = Person()
revorg.name = “Revorg the Destroyer”
revorg.say_hi()
# Hi, my name is Revorg the Destroyer
```


Objects can inherit
attributes from their parents
135
Deinonychus
Archaeopteryx
Struthiomimus
MANIRAPTORS
RELATIVELY LONG ARMS
COELUROSAURS
CARNOSAURS
Ulosaurus
UNEVEN COVERING OF ENAMEL ON TEETH
CRETACEOUS
JURASSIC


Vehicle
Airplane
Car
SUV

is a

All Airplanes are Vehicles

~~All Vehicles are Airplanes~~

```
class Vehicle(object):
    pass

class Airplane(Vehicle):
    pass

class Car(Vehicle):
    pass

class SUV(Car):
    pass
```

Everything in Python
is an object

All objects have
special methods

__dunder__

```
mph = 60
print mph.__class__
# <type 'int'>
```

```
tax_rate = 0.45
print tax_rate.__class__

# <type 'float'>
```

```
name = “Louis Brandeis”
print name.__class__

# <type ‘str’>
```

```
def hi():
    print “Hello!”
print hi.__class__

# <type ‘function’>
```

```
class Car(object):
    pass
saturn = Car()
print saturn.__class__

# <type '__main__.Car'>
```

Creating new instances

```
class Restaurant(object):
    name = ""

well_dressed = Restaurant()
well_dressed.name = "Well Dressed Burrito"

cafe1800 = Restaurant()
cafe1800.name = "1800 Cafe"

julias = Restaurant()
julias.name = "Julia's Empanadas"
```

```
class Restaurant(object):
    def __init__(self, name):
        self.name = name
```

```
well_dressed = Restaurant("Well Dressed Burrito")
cafe1800 = Restaurant("1800 Cafe")
julias = Restaurant("Julia's Empanadas")
```

tab
AWESOME!!1!
shift
Q
A
The End