

# Classes

```
class Cat
  def initialize(name, color, age)
    @name = name
    @color = color
    @age = age
  end

  def name
    @name
  end

  # getter
  def age
    @age
  end

  # setter
  def age=(number)
    @age = number
  end

  def purr
    if @age > 5
      puts @name.upcase + " goes purrrrrr..."
    else
      puts "..."
    end
  end

  def meow_at(person)
    puts "#{@name} meows at #{person}"
  end
end

cat_1 = Cat.new("Sennacy", "brown", 3)
cat_2 = Cat.new("Whiskers", "white", 5)
p cat_1 #<Cat:0x007fb6d804cfe0 @age=3, @color="brown", @name="Sennacy">
p cat_2 #<Cat:0x007fb6d6bb60b8 @age=5, @color="white", @name="Whiskers">
p cat_1.name # "Sennacy"
p cat_1.age # 3
cat_1.age = 42
cat_1.purr # "SENNACY goes purrrrrr..."
cat_1.meows_at('Diego')
```

# Class Variables

Let's say we wanted all car instances to have the same number of wheels. We can add a class variable `@@num_wheels`:

```
class Car
  @@num_wheels = 4

  #class method
  def self.upgrade_to_flying_cars
    @@num_wheels = 0
  end

  def initialize(color)
    @color = color
  end

  #instance methods

  # getter for @color instance variable
  def color
    @color
  end

  # getter for @@num_wheels class variable
  def num_wheels
    @@num_wheels
  end
end

car_1 = Car.new("red")
p car_1.num_wheels    # 4

car_2 = Car.new("black")
p car_2.num_wheels    # 4

Car.upgrade_to_flying_cars

p car_1.num_wheels    # 0
p car_2.num_wheels    # 0

car_3 = Car.new("silver")
p car_3.num_wheels    # 0
```

```

class Dog
  def initialize(name, bark)
    @name = name
    @bark = bark
  end

  def self.whos_louder(dog_1, dog_2)
    if dog_1.bark.length > dog_2.bark.length
      return dog_1.name
    elsif dog_1.bark.length < dog_2.bark.length
      return dog_2.name
    else
      return nil
    end
  end

  def name
    @name
  end

  def bark
    @bark
  end
end

d1 = Dog.new("Fido", "woof")
d2 = Dog.new("Doge", "much bork")
p Dog.whos_louder(d1, d2) # "Doge"

```

## Class Constants

```

class Car
  NUM_WHEELS = 4
  .
  .
  .

```

## Wrapping Up

- an `@instance_variable` will be a distinct variable in each instance of a class; changing the variable will only effect that one instance

- a `@@class_variable` will be shared among all instances of a class; changing the variable will effect all instances because all instances of the class
- a `CLASS_CONSTANT` will be shared among all instances of a class, but cannot be changed
- `Class#method_name` means `method_name` is an instance method
- `Class::method_name` means `method_name` is a class method