Handout

Introduction to Ruby 2.0
Lesson 6 - Step 4
Darren Jones

Introduction

In the last step, we created an `Animal` class that allowed us to create lots of animal objects, all with the same methods. Now we're going to create two more classes, a `Rabbit` class and a `Duck` class. Create a file called *inheritance.rb*:

```ruby
require './animal.rb'

class Rabbit < Animal
  def hop
    puts "Hopping is great fun!"
  end
end

class Duck < Animal
  def swim
    puts "I'm quackers about swimming!"
  end

  def say_hello
    "Quack! I'm a duck called #{ @name }"
  end
end
```

Concept of Inheritance

Because rabbits and ducks are both animals they will have many of the same methods found in the `Animal` class. So instead of duplicating all of the code from the `Animal` class what we're going to do is use the concept of **inheritance** to inherit all of the methods.

First of all, we need to require the file containing the `Animal` class.

Next we create the `Rabbit` and the `Duck` classes. This is the same notation that we used before except for this extra bit `< Animal`. It means that the class inherits from the `Animal` class and this means that it gets the same methods and properties.

What we can then do is add any new methods that are specific to these two new classes.

Animal Classes

So I've defined a new method for the rabbit called hop because obviously rabbits like to hop.

Inside the `Duck` class I've defined a new method called `swim` because ducks like to swim, and this will amp up this string here.

We can also override methods that have been inherited from the `Animal` class if we want them to behave in a different way. So for example, you may remember that the `Animal` class had a method called `say_hello`. `Duck` class has this method redefined:

```
def say_hello
  "Quack! I'm a duck called #{ @name }"
end
```