Ruby Classes and Modules

Devpoint Labs - Jake Sorce / Dave Jungst

What is a class?

A class is an object that has attributes that exist that can be written or read or both depending on the access.

A class can be instantiated (eg. MyClass.new), but does not need to be in some cases. *(we will go over that later)*

What is a module?

A module acts as a library where you can put code that needs to be shared between different classes or other modules

A module can not and will not be instantiated ever! More examples to come.

What does a class look like?

```
class Person
 attr_accessor :name, :age

 def initialize(name, age)
     @name = name
     @age = age
 end

 def increase_age(number)
     @age += number
 end
end
```

```
jake = Person.new(‘jake’, 25)

jake.name
    => ‘jake’
jake.age
    => 25
jake.increase_age(5)
    => 30
jake.age
    => 30
```

Why use a class?

Main reason to use a class is to have an instance of that class that resides in memory.

Instance variables persist as long as the instance of the class is in memory

What does a module look like?

```
module Math
  def add(num_1, num_2)
    num_1 + num_2
  end

  def subtract(num_1, num_2)
    num_1 - num_2
  end
end
```

```
class MathAssignment
  include Math

  def first_solution
    add(10, 50) + subtract(3, 10)
  end
end
assignment = MathAssignment.new
assignment.first_solution
        => 53
```

Why use a module?

You would use a module when you have common code that would be shared between multiple classes.

You can also use modules to namespace things which we will get into later when we go into web frameworks.

Differences between instance methods and class methods

```
class Dog
 attr_accessor :name, :breed
 def initialize(name, breed)
  @name = name
  @breed = breed
 end

 def info
  puts “#{@name} is of breed: #{@breed}”
 end

 def self.bark
  puts “WOOF”
 end
end
```

```
jake_the_dog = Dog.new
(‘jake’, ‘bulldog’)
```

```
jake_the_dog.info
=> “Jake is of breed: bulldog”
```

```
jake_the_dog.bark
=> no method error
```

```
Dog.bark
=> “WOOF”
```

RubyGems and Pry

RubyGems.org is the Ruby community's gem hosting service. Instantly publish your gems and install them. Use the API to interact and find out more information about available gems. Become a contributor and enhance the site with your own changes.

How to install gems to your system

```
gem install <gem_name>
```

Example:

```
    gem install pry
```

How to use gems in your scripts

At the top of your script file:

```
  require ‘<gem_name>’
```

Example:

```
  require ‘pry’
```