Rubygems: The Ruby Library System

What is a library? Why use them?

- A library is a collection of code, usually for a specific purpose, that has a well-defined interface
- It allows you to save a lot of work implementing different functionality by using code other people have written
- Many languages, including Ruby, come with much functionality already written known as the **core library**
- If we want to expand Ruby's functionality without writing it from scratch, we'll need to use a library

RubyGems - the Ruby library system

- The system in Ruby for the creation and easy use of libraries is called **RubyGems**
- A Ruby library itself is known as a **gem**

Using RubyGems

- Installation: `$ gem install mygem`
- Removal: `$ gem uninstall mygem`
- List all installed gems: `$ gem list --local`
- When using RVM (Ruby Version Manager) - the sudo command is typically not necessary to install gems
- Only use sudo if you have problems with permissions during gem installation

Ruby projects are now folders

- Our projects are about to get a bit more complicated
- From now on, when you create a new Ruby script, put it in its own folder
- This structure will become useful as our project grows

Bundler

- Bundler is a gem that makes installing a **gem set** for each of your specific projects much easier
- A **gem set** is a set of gems locked at specific versions to be used in your project, a set of **dependencies**
- Bundler is a bit meta: it's a gem that helps install other gems
- To install bundler:

```
$ gem install bundler
```

Gemfile

- A Gemfile is simply a list of all of the RubyGems your project needs to function - your **dependencies**
- Bundler uses the Gemfile to know which gems to install
- The file itself should **always** have the name **Gemfile**, with no extension after it

Gemfile

- At the top of your Gemfile, you need to specify a source for your gems and the version of Ruby you're using on your machine
- Almost every open-source gem is hosted at rubygems.org

```
source 'http://rubygems.org'
ruby '2.1.0'
```

Gemfile

After these initial two lines, you can start listening gems using the following syntax:

```
gem 'gemname'
```

For example:

```
gem 'sinatra'
gem 'haml'
gem 'activerecord'
```

Gemfile

- If you need to fetch a gem from a git repository somewhere, you can use this syntax:

```
gem 'rack-flash3', :git => "git://github.com/treeder/rack-flash.git"
```

© New York Code + Design Academy 2015

Installing the gems in your Gemfile

- Once you've added necessary gems to your Gemfile, `cd` into the directory it resides in and run the `bundle install` command to install the gems with `bundler`:

```
$ cd ~/dev/my_project_directory
$ bundle install
```

Exercise: Working with RubyGems

- Create a new directory for your project
- In Sublime Text, create a Gemfile with the gem named "geolocater" listed in it (note the spelling)
- `cd` to your project directory in the Terminal
- Run `bundle install`

Exercise: Working with RubyGems

- Create your main Ruby file inside the folder, with a name like `app.rb`
- Require the Geolocater gem and some test code:

```ruby
require 'geolocater'
record = Geolocater.geolocate_ip("64.119.207.255")
puts record
```