Ruby on rails

1. Setup environment
   1. Install Ubuntu
   2. Install rails
   3. Install DB ( postgresql + pgadmin & mysql + workbench )
   4. Install git : git gui + gitk
   5. Install Guake terminal + ZSH
   6. Exercise : read ruby example & try to do something on ruby

1. Source control
   1. QA for exercise
   2. Create repo on github and git configure (how to use GIT)
   3. Setup and deploy to Heroku
      1. Install heroku:   
         <https://devcenter.heroku.com/articles/getting-started-with-ruby#set-up>
      2. Rename app in heroku:   
         <https://devcenter.heroku.com/articles/renaming-apps>
   4. Exercise:
      1. Read document: how to work with String on ruby : <http://ruby-doc.org/core-2.2.0/String.html>
2. How to build rails app
   1. Exercise: QA
   2. Create rails app + home page
   3. Deploy and run app on heroku
   4. Action flow: route + model + controller + view
   5. Configure css + javascript on rails
3. Active record ORM
   1. Basic: <http://guides.rubyonrails.org/active_record_basics.html>
   2. Migration: <http://guides.rubyonrails.org/active_record_migrations.html>
   3. Validation: <http://guides.rubyonrails.org/active_record_validations.html>
   4. Callbacks: <http://guides.rubyonrails.org/active_record_callbacks.html>
   5. Association: <http://guides.rubyonrails.org/association_basics.html>
   6. Query Interface: <http://guides.rubyonrails.org/active_record_querying.html>
4. Layout and rendering in rails
   1. Creating Responses
      1. Render by default
      2. Using render
         1. Render nothing
         2. Rendering an action view
         3. Rendering an Action's Template from Another Controller
         4. ….
         5. ( Demo )
         6. Finding Layouts
            1. Specifying Layouts for Controllers
            2. Choosing Layouts at Runtime
            3. Conditional Layouts
         7. Avoiding Double Render Errors
      3. Using redirect\_to
         1. Getting a Different Redirect Status Code
         2. The Difference Between render and redirect\_to
      4. Using head To Build Header-Only Responses
   2. Structuring Layouts
      1. Asset Tag Helpers
      2. Linking to JavaScript Files with the javascript\_include\_tag
      3. Linking to CSS Files with the stylesheet\_link\_tag
      4. Linking to Images with the image\_tag
      5. Linking to Videos with the video\_tag
      6. Linking to Audio Files with the audio\_tag
   3. Understanding yield
   4. Using the content\_for Method
   5. Using Partials
      1. Naming Partials
      2. Using Partials to Simplify Views
      3. Partial Layouts
      4. Passing Local Variables
      5. Rendering Collections
      6. Local Variables
      7. Spacer Templates
      8. Collection Partial Layouts
5. Authentication and demo
6. Twitter Bootstrap and scss demo
7. Coffeescript & ajax demo