

## Chapter 5 - Filling In the Layout

- **5.1 Adding some structure**
  - Downloaded an image with:
    - `$ curl -o app/assets/images/rails.png -OL railstutorial.org/rails.png`
  - Use image with:
    - ``
  - Update Gemfile with:
    - `gem 'rails', '5.0.1'`  
`gem 'bootstrap-sass', '3.3.6'`
  - Add CSS file with 'touch'
    - `$ touch app/assets/stylesheets/custom.scss`
  - Partials
    - Partials are erb files that represent pieces of html (i.e. partial)
    - Add partial file(s), should start with an underscore:
      - `_shim.html.erb`
      - `_header.html.erb`
      - `_footer.html.erb`
    - Modify the application.html.erb to implement these partials with:
      - `<%= render 'layouts/shim %>`
      - `<%= render 'layouts/header' %>`
      - `<%= render 'layouts/footer' %>`
      - NOTE: you do NOT include the underscore
    - Add CSS info for the new footer
- **5.2 Sass and the asset pipeline**
  - Directories for assets:
    - **app/assets**: assets specific to the present application
    - **lib/assets**: assets for libraries written by your dev team
    - **vendor/assets**: assets from third-party vendors
      - All three have sub folders: images, javascript, stylesheets
  - SASS improvements over CSS
    - Nesting
      - ```
.center {
  text-align: center;
  h1 {
    margin-bottom: 10px;
  }
}
```
    - Variables
      - From: `color: #777;` (used multiple places)
      - To: `$light-gray: #777;` (at top)

- color: \$light-gray; (everywhere else)
  - With Bootstrap a shortcut exists where we can use only:
    - color: \$gray-light; because Less already defines it
- 5.3 Layout Links
  - Add the contact page
    - Add the html.erb file
    - Add the def to the controller
    - Add the get to the routes.rb
  - Update routes.rb so we can use shorter references:
    - root 'static\_pages#home'
    - get '/help', to: 'static\_pages#help'
    - get '/about', to: 'static\_pages#about'
    - get '/contact', to: 'static\_pages#contact'
  - Update our partials with the new routes:
    - i.e. about\_path
  - Integration Test for checking links
    - Create with:
      - \$ rails generate integration\_test site\_layout
    - Modify with assertions:
      - test "layout links" do
      - get root\_path
      - assert\_template 'static\_pages/home'
      - assert\_select "a[href=?]", root\_path, count: 2
      - assert\_select "a[href=?]", help\_path
      - assert\_select "a[href=?]", about\_path
      - assert\_select "a[href=?]", contact\_path
      - End
  - Add User scaffolding
    - Add user to routes w/ our new to: structure
    - Update the test to accommodate the new structure
    - Add link to home page
    - Add content to Users web page
  - Wrap everything up as usual
    - \$ git add -A
    - \$ git commit -m "Finish layout and routes"
    - \$ git checkout master
    - \$ git merge filling-in-layout