



Making it look nice

Bootstrap, fonts, and assets

Okay, no offense, but our apps are looking laaame

- ☐ **This isn't a front end class, but this whole Craigslist look isn't going to cut it.**
- ☐ **Let's make things look a bit better on the front end today.**
- ☐ **We will touch on HTML, CSS, the asset pipeline (for making logos and such), Google Fonts, and start with the most popular .**
- ☐ **When we finish up, we will use the remaining time to work on our final projects.**

**Let's add a logo and
some style.**

Asset Pipeline

- ☐ Rails CSS and JavaScript assets are compiled and concatenated into single files and cached on the server. This is called the Asset Pipeline.
- ☐ This reduces the number of requests that a client has to make to get all of your images and static files.
- ☐ We won't go too deep into this, just learn how to use it in practice.

CSS

- **Let's take a tour of app/assets**
- **stylesheets is where we will write CSS rules to change the look of the app.**

CSS



- **CSS (Cascading Style Sheets, not Cansei de Ser Sexy) declares rules. The latest, or most specific rule wins. In Rails, you can declare controller specific styles. For simplicity's sake, we will place our css in application.css.**

application.css

- This file is where all of your css concatenates into one file (in production). It gets called on line five of application.html.erb
- This comment means something

```
*= require_tree .  
*= require_self  
*/
```

application.css

- ☐ This essentially means it will concatenate all the stylesheets (CSS, LESS, Sass, SCSS) in that directory, then any CSS in application.css itself.
- ☐ After that comment, any css you write will override the other stuff, such controller specific stylesheets or a front end framework.

- ☐ **Open up the bloggr application that we started, lets futz with some styles.**

Images in rails

- ☐ Images in the asset pipeline are useful for logos, front page images and such.
- ☐ User uploaded images have to be hosted somewhere else, such as Amazon's S3.
- ☐ Don't go completely nuts loading your app with images.

Image Tag

- ☐ Rails has a helper method for inserting images from the asset pipeline.
- ☐ Just pop in `<%= image_tag 'logo.png' %>`
- ☐ It looks for logo.png in app/assets/images.
- ☐ If the logo is in app/assets/images/logos/ then use `<%= image_tag 'logos/logo.png' %>`

Introducing Bootstrap

Front end frameworks

- ☐ There some problems that need to be solved by everyone, and there is no point in everyone wasting their time hand rolling everything.
- ☐ Front end frameworks usually come with a ready-made starting point for your own styling.

Front end frameworks

- ☐ Any good framework will include stuff like:
 - ☐ Grid layout
 - ☐ Cross-browser normalization (read: making it not look like poo on Internet Explorer)
 - ☐ Modals
 - ☐ Navbars
 - ☐ Typography, Icons, Glyphs

Front end frameworks

- ☐ There are plenty of full service front end frameworks:
 - ☐ Twitter Bootstrap
 - ☐ Zurb Foundation
 - ☐ Skeleton
 - ☐ Semantic UI
 - ☐ UIKit

Bootstrap

- ☐ Probably the most popular framework out there.
- ☐ Open source, actively developed, very well documented, tutorials abound.
- ☐ Plenty of 'flavors' downstream such as Bootflat, Flat UI etc that are based on Bootstrap but don't look exactly like Bootstrap.

Bootstrap

- ☐ Created in 2010 by some developers and designers at Twitter.
- ☐ Open source
- ☐ Mobile-first, responsive

Bootstrap on Rails

- ☐ Browsers understand CSS, but CSS is kind of annoying. Preprocessors such as Less and Sass (syntactically awesome style sheets) are nice, and Bootstrap comes in either flavor. Rails compiles them into CSS so browsers can understand them.
- ☐ I am not interested in stoking the flames of the Less/Sass holy war. Take it outside. Count to ten.
- ☐ Let's use Less for now.

Bring Bootstrap into your Rails app

- In the Gemfile of Bloggr:

```
gem "therubyracer"  
gem 'less-rails'  
gem "twitter-bootstrap-rails"
```

- You might need `gem 'less-rails', github: 'metaskills/less-rails'` if that doesn't work. A few days ago there was an issue with the gem in Ruby 2.2 and the fix wasn't released when I wrote this.

Install Bootstrap

```
bundle install  
rails generate bootstrap:install less
```

Now reload your app.

Bootstrap

- ☐ **Whoa! The app is already quite stylish.**
- ☐ **Now we have access to Bootstrap's massive library of components**
- ☐ **Let's code together to add a nice navbar for our links and make those flash messages cooler.**

container

col-xs-3

col-xs-9

col-xs-6

col-xs-6

col-xs-6

col-xs-6

col-xs-6

col-xs-6

Your turn

- ☐ These buttons from the devise views for sign up now look, how shall we say, garbage.
- ☐ Look at the bootstrap docs for how to make buttons look cool.
- ☐ Also, the buttons are like right up against the fields. Fix that with some CSS.
- ☐ Work with a buddy. I will now play the song "CSS Suxx" by CSS.
- ☐ You have until the song "Alala" ends

Google Fonts

Google fonts

- ☐ Let's make the typography awesomer than the default bootstrap stuff.
- ☐ There are all kinds of places to get free web fonts that you never even have to download.
- ☐ Let's check out google web fonts

Quiet assets

Quiet assets

- ☐ Look at the server logs
- ☐ Ugh. Full of crap about assets because of bootstrap.
- ☐ Let's use the gem Quiet Assets to quiet it down so we can see more of the actual errors that we need to look at in the logs.

With any time that we have left:

- ☐ **Work on your projects**
- ☐ **Give it some style with Bootstrap (unless you want to be a unique little snowflake with a different framework)**
- ☐ **Give it a logo, maybe some fonts from Google.**