One Page Everywhere

Fluid, Responsive Design with Semantic.gs

The Semantic Grid System

- Grid System
- Fluid or Fixed
- Responsive
- Semantic
- Sass or LESS

Grid Systems

Grid System Fixed Size

Grid System Fixed Size

Semantic.gs: Fixed or Fluid

```
// Specify the number of columns and set column and gutter widths

$columns: 12;

$column-width: 60;

$gutter-width: 20;


// Remove the definition below for a pixel-based layout

$total-width: 100%;
```

Grid System Clutter

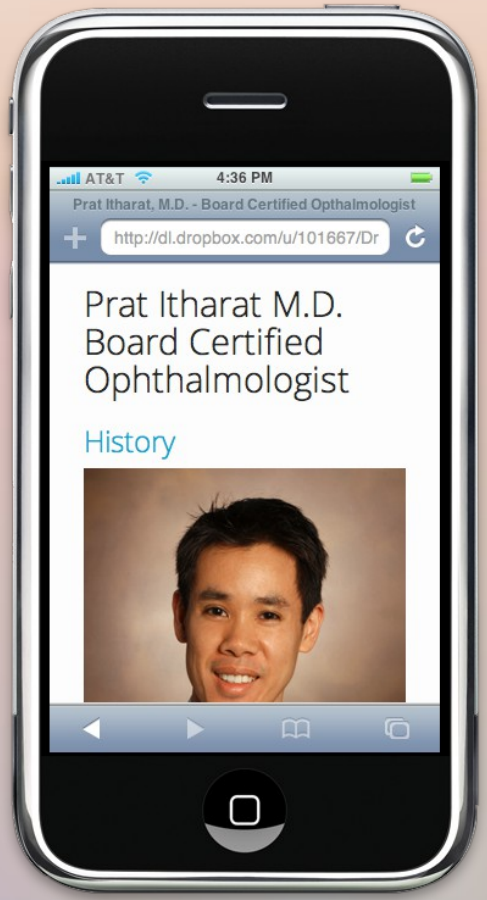
```
<body>
    <div class="container_12">

        <h1 class="grid_4 push_4">
            960 Grid System
        </h1>
        <!-- end .grid_4.push_4 -->

        <p id="description" class="grid_4 pull_4">
            ...
        </p>
        <!-- end #description.grid_4.pull_4 -->
```

Semantic.gs: Layout in Stylesheets

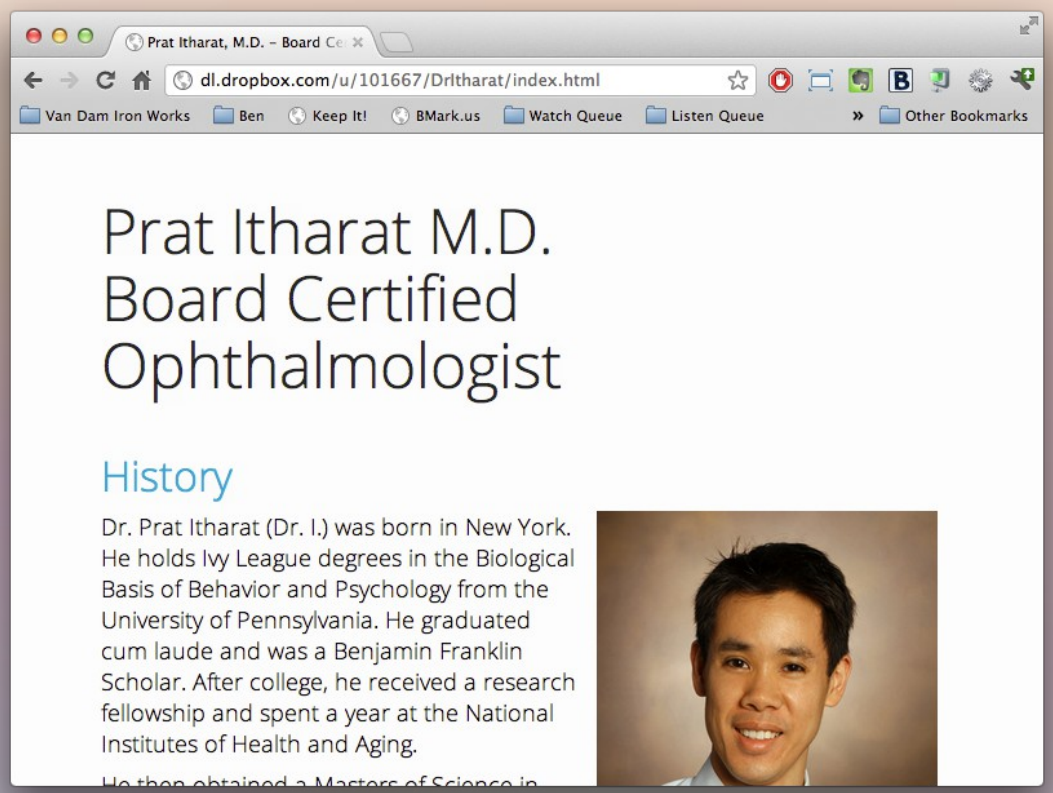
```
<body>
  <div id="main-content">
    <header id="banner">
      <h1> … </h1>
    </header>
    <section id="history">
      <h2>History</h2>
      <p> … </p>
    </section>
```

```
#banner {
    @include column(12);
    padding-top: 3em; }

#history {
    @include column(6); }

#contact {
    @include column(6); }
```

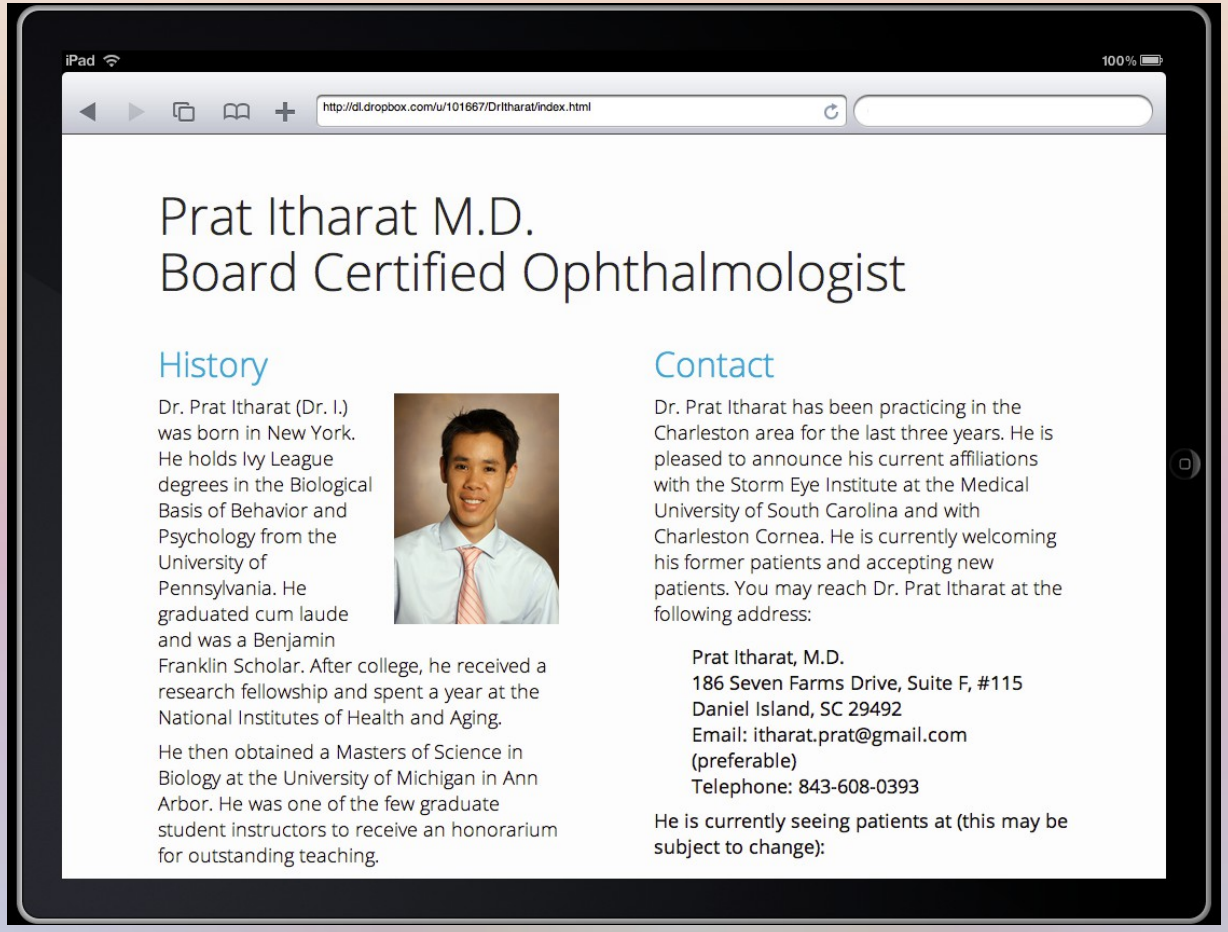
Semantic.gs: Responsive (iPhone)



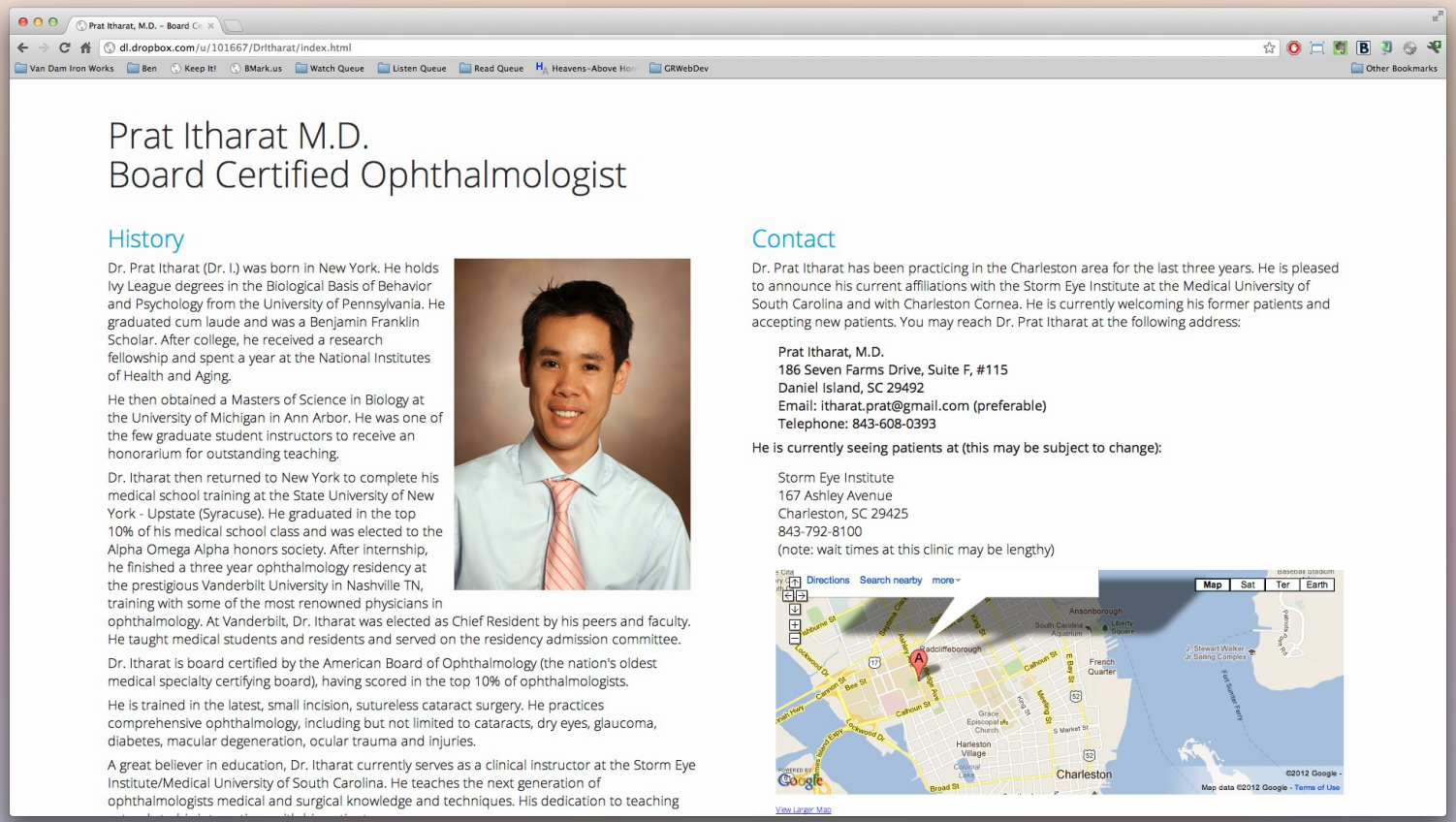
Semantic.gs: Responsive (800x600)



Semantic.gs: Responsive (iPad)



Semantic.gs: Responsive (1920x1080)



Project Walk-through

- File locations
- semantic_gs files

HTML Walk-through

- Single stylesheet
- IE fixes
- Semantic elements
- Few layout divs

SCSS Walkthrough

- Imports
- Variables
- Columns and Gutters
- @include Columns
- Few sizes in pixels, all % and em


Crabby old man kitteh
tellz u boring storeez
ICANHASCHEEZBURGER.COM

Fixes for Old Internet Explorers

Html5shiv

https://github.com/aFarkas/html5shiv

Respond.js

https://github.com/scottjehl/Respond

Things to Watch Out For

- Right and left padding and margins on grid elements can be finicky. Best to just wrap them with a div.
- Image sizing: small screens get large images and have to scale the images down.
- Can't really re-order sections.

Choosing Media Query Breakpoints

1. Start big and go to small
2. Shrink window until something looks funky
3. Create a breakpoint, fix what looks funky
4. Repeat from #2
5. No need to worry about specific devices!

Credits

- The Semantic Grid System http://semantic.gs/
- iPhone screenshots simulated using iPhony
  - http://www.marketcircle.com/iphoney/
- iPad screenshots simulated using iPad Peek
  - http://ipadpeek.com/
- Some slides from http://icanhascheezburger.com/

Some Useful Utilities

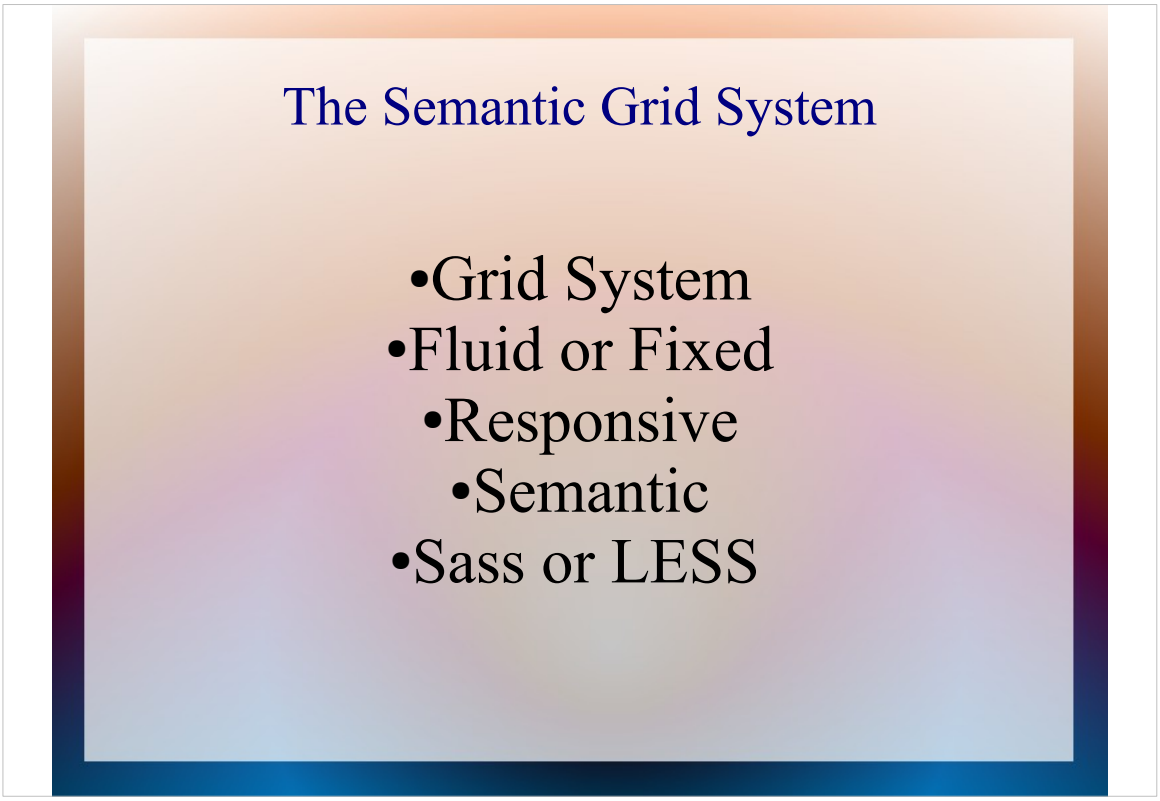
- Chrome Window Resizer Extension
  - https://chrome.google.com/webstore/detail/kkelicaakdanhinjdeammmilcgefonfh
- HTML5 Shiv
  - https://github.com/aFarkas/html5shiv
- Respond.js
  - https://github.com/scottjehl/Respond

Contact Me

- Meatspace: Ben Rousch
- Email: brousch@gmail.com
- Twitter: @brousch
- Google+: https://plus.google.com/102663141609195877664/


One Page Everywhere
Fluid, Responsive Design with Semantic.gs

Greeting



We're going to transform this one page, static website into a fluid, responsive website using The Semantic Grid System.

Semantic.gs is:

12 column, 960 pixel grid system
Which can be fluid or fixed
It is responsive
It is semantic
It uses Sass or LESS

We'll talk about a few of these terms as we go on.

CSS grid systems have been around for a while now. I've used Blueprint in the past, but there are many others available.

How many of you have used a grid system at some point?

Because keeping things lined up in CSS is hard, the idea is that the grid system gives you a framework of columns and gutters, the space in between columns, that let you line things up nicely on your page.

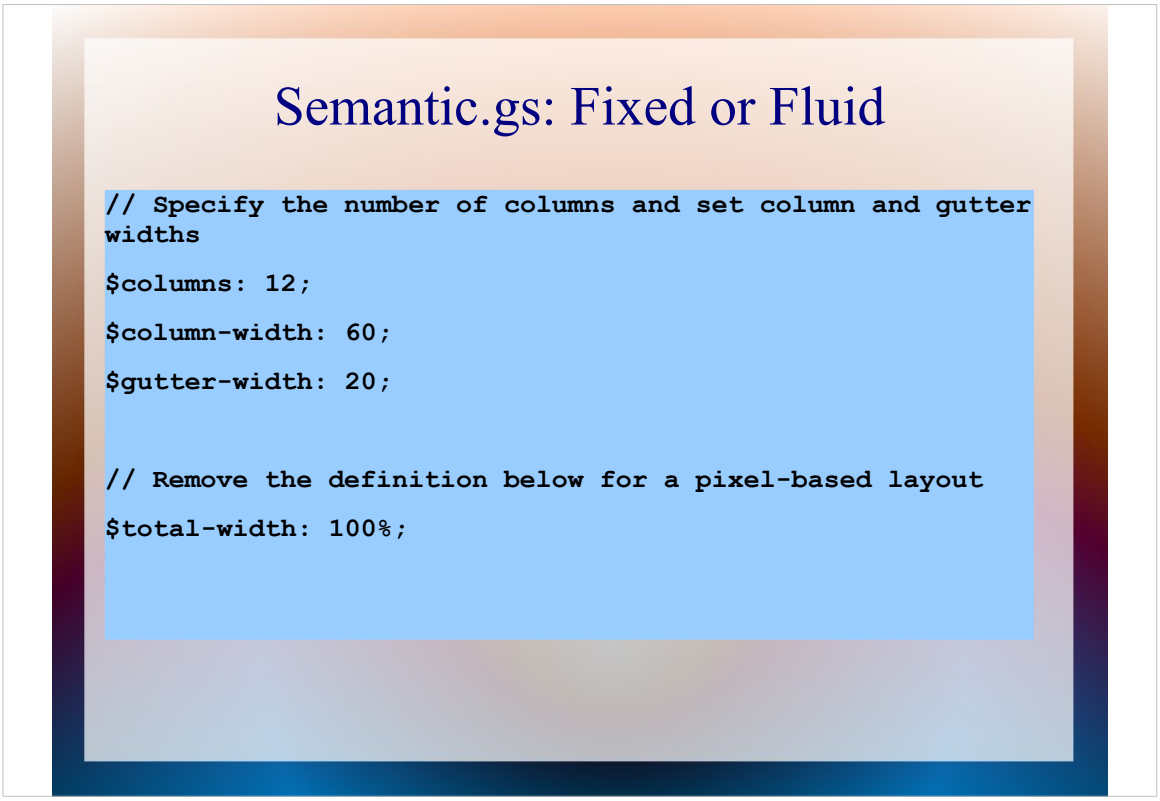
But there are a couple of problems with grid systems:

1. They are usually a fixed width. Often 960 pixels. This does not lend itself well to responsive design, which needs to adapt to many different screen sizes.

Here is the 960 grid on a 1080p monitor. The empty areas on the side are pretty excessive.

Here is the same site on an iPhone.

I don't think I need to say anything.

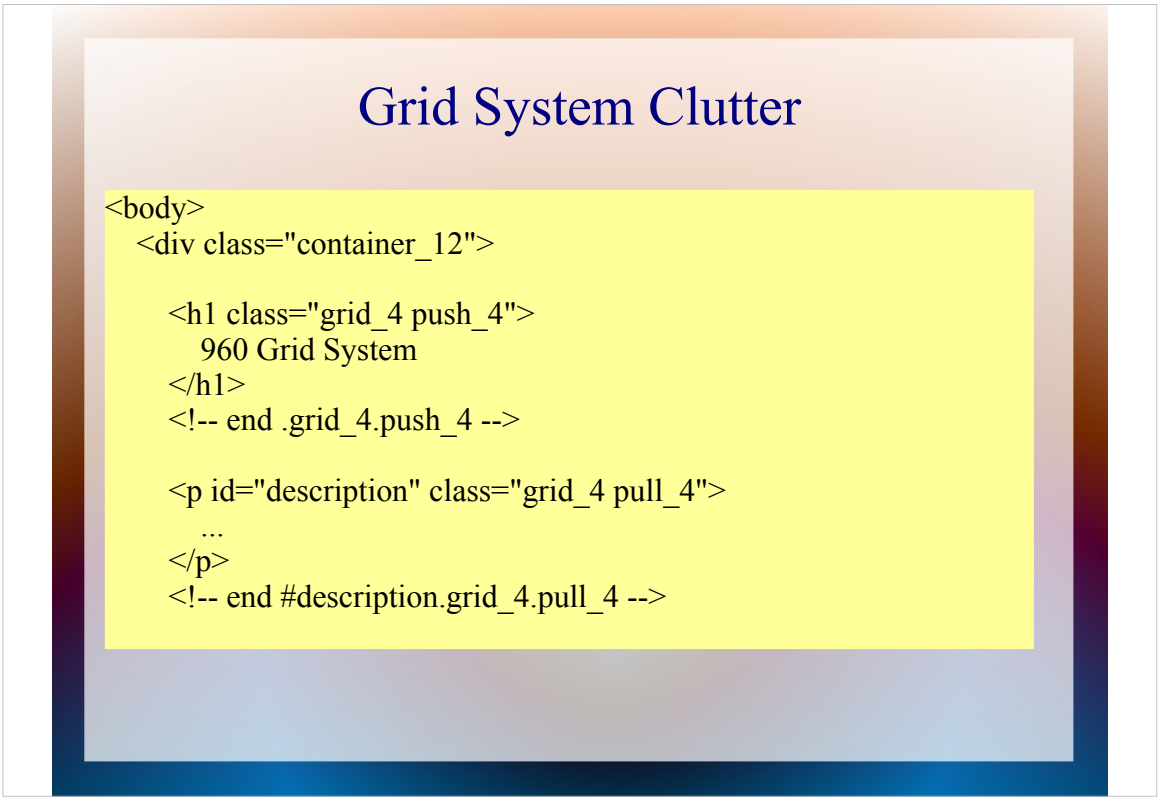


The Semantic Grid System lets you use a fixed size if you want to, but it can also provide a fluid grid for you.

Fluid means that the sizes of elements change for different screen sizes. You use percentages and ems to lay things out instead of pixels.

* Compare fixed DrI to fluid *

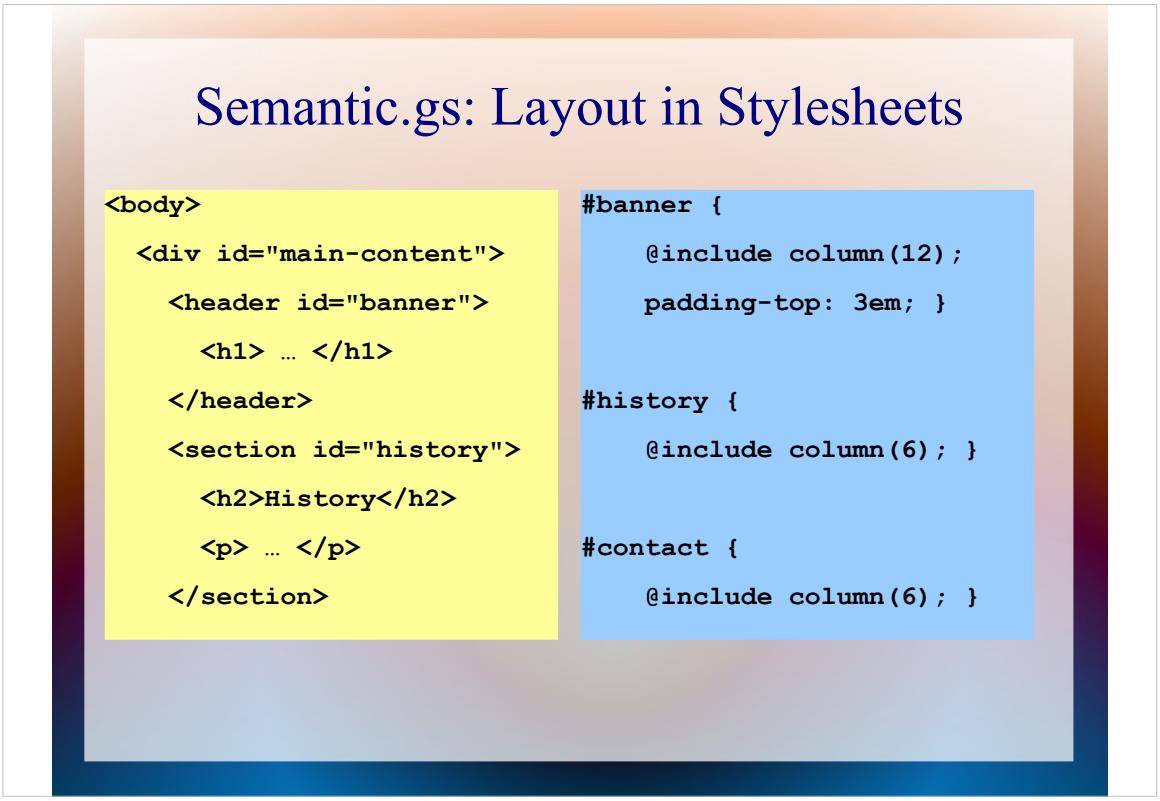
We'll be using the fluid grid today. And I think once you go fluid you won't want to go back to fixed.



Another problem with typical grid systems is they clutter up your HTML with extra divs and span or grid properties.

Here's some of the HTML from that 960Grid page we just saw.

This is layout stuff. It really doesn't belong in your HTML. It belongs in your stylesheets. And that's where The Semantic Grid System puts it.



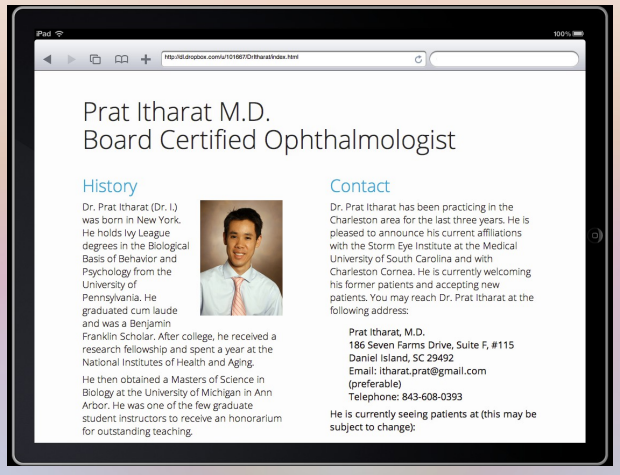
This is a snippet from the final HTML and stylesheet we'll create today. You can see the column information is in the CSS. There's no sign of it in the HTML.

Responsive refers to a website's ability to adapt to different resolutions, screen sizes, and devices.

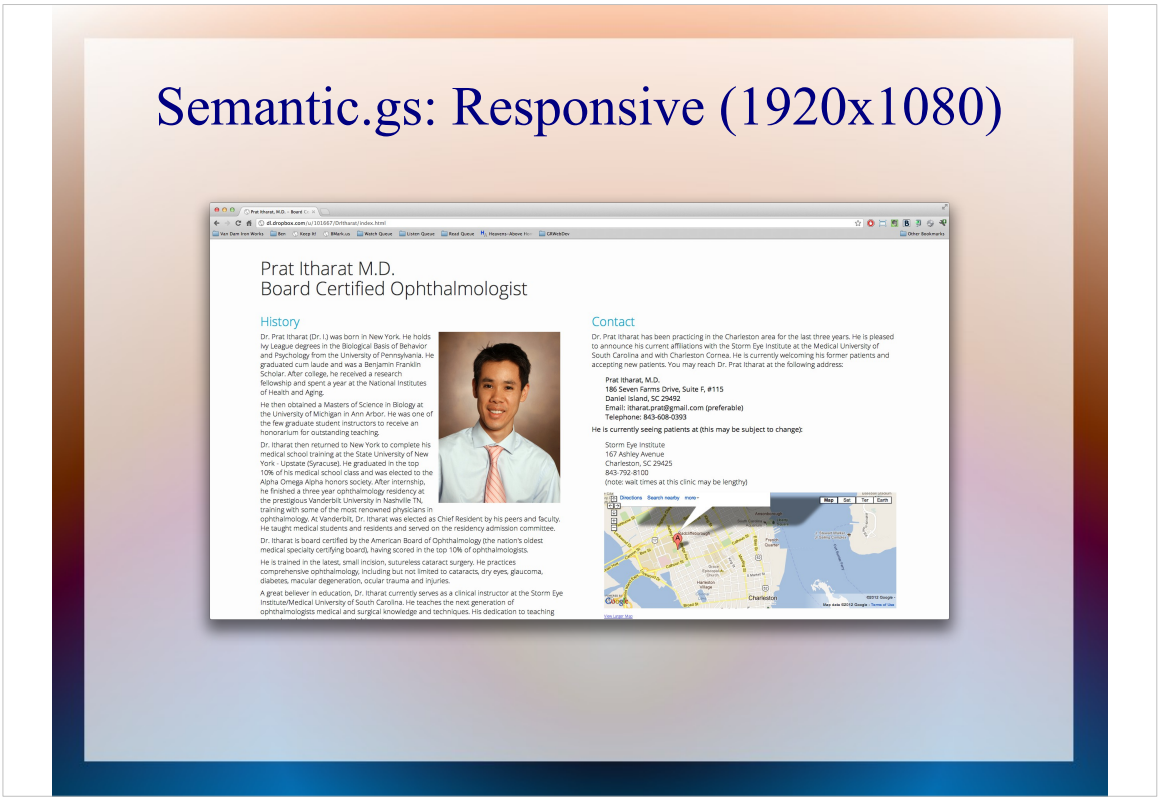
Semantic.gs: Responsive (800x600)

A typical grid system works well for a typical sized desktop or laptop screen, but it's doesn't work well for very large screens or smartphones.

Semantic.gs: Responsive (iPad)



This is partly because the grid is defined in the HTML, which means you can't change it on the fly without resorting to Javascript or different versions of the page for different sized screens..



These are screenshots of our finished website on an iPhone, at 800x600 resolution, on an iPad, and at 1920x1080 resolution.

Let's see them again.

Project Walk-through

- File locations
- semantic_gs files

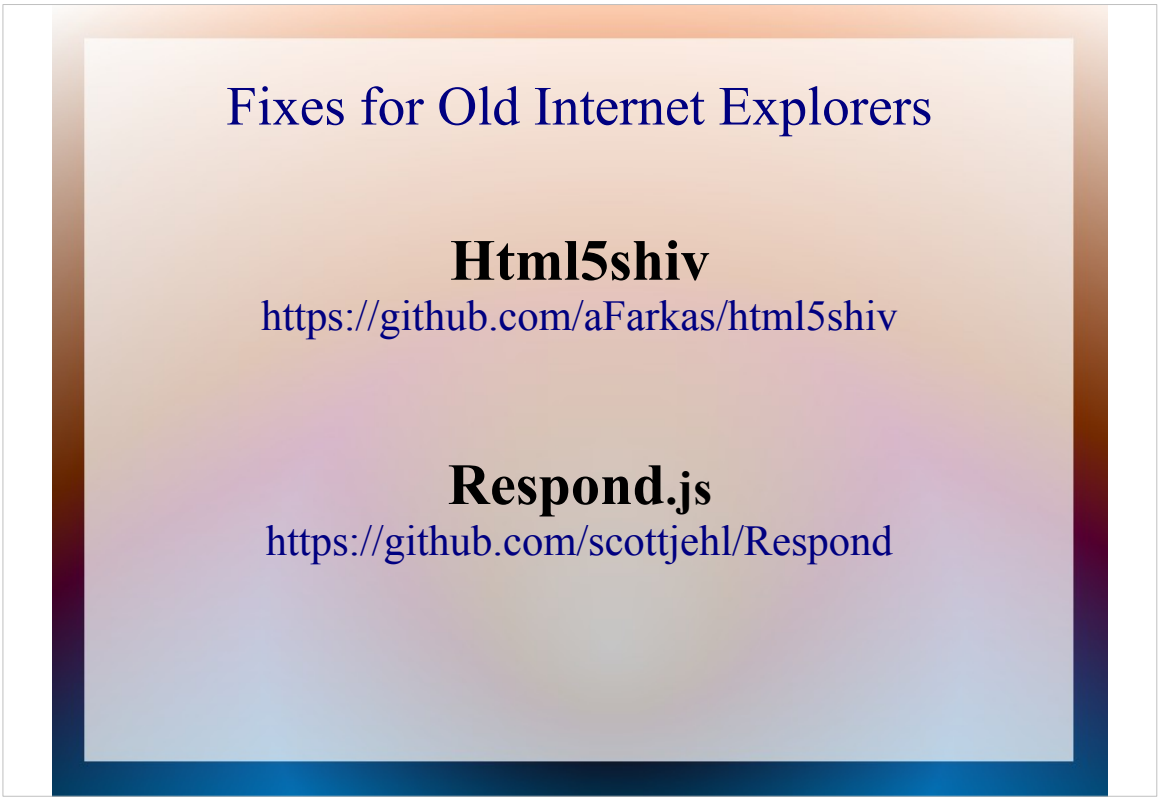
HTML Walk-through

- Single stylesheet
- IE fixes
- Semantic elements
- Few layout divs

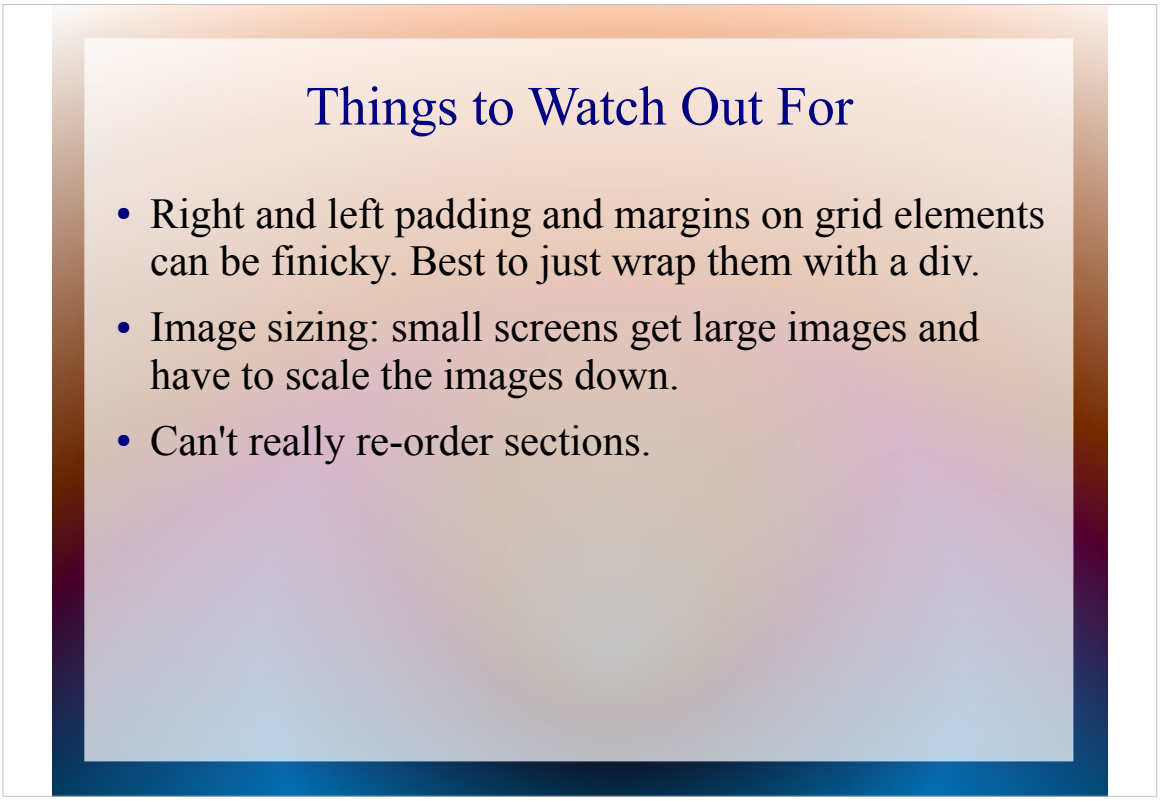
SCSS Walkthrough

- Imports
- Variables
- Columns and Gutters
- @include Columns
- Few sizes in pixels, all % and em

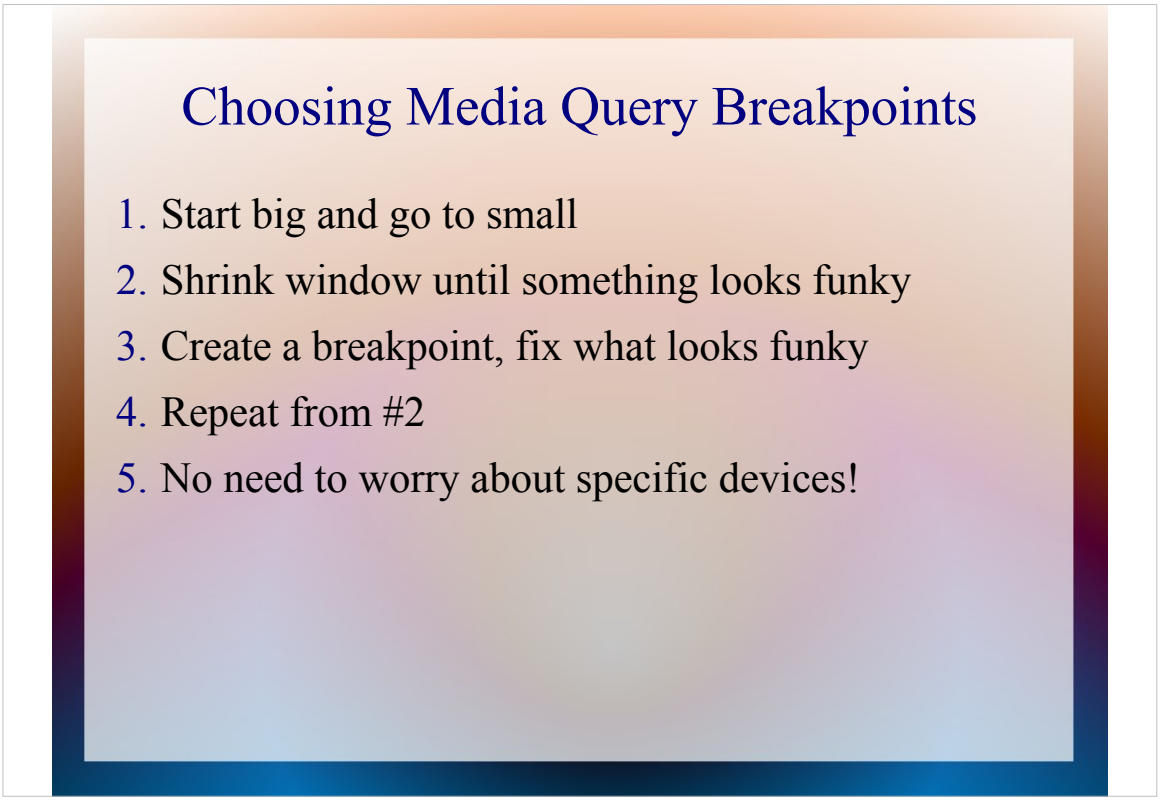
Media queries don't work on these crusty old browsers. So do we just say forget those guys? Of course not. Much of Dr I's audience is likely to be old people still running Windows XP.



Luckily I found an easy fix.



There a few things you need to watch out for as you design your site using The Semantic Grid System.



With a pre-existing site, like to start with the largest version of the website and shrink it. Other people like to go mobile first and make media query breakpoints as the site gets bigger. Try them both, see what you like.

Credits

- The Semantic Grid System http://semantic.gs/
- iPhone screenshots simulated using iPhony
  - http://www.marketcircle.com/iphoney/
- iPad screenshots simulated using iPad Peek
  - http://ipadpeek.com/
- Some slides from http://icanhascheezburger.com/

Some Useful Utilities

- Chrome Window Resizer Extension
  - https://chrome.google.com/webstore/detail/kkelicaakdanhinjdeammmilcgefonfh
- HTML5 Shiv
  - https://github.com/aFarkas/html5shiv
- Respond.js
  - https://github.com/scottjehl/Respond

Contact Me

- Meatspace: Ben Rousch
- Email: brousch@gmail.com
- Twitter: @brousch
- Google+: https://plus.google.com/102663141609195877664/