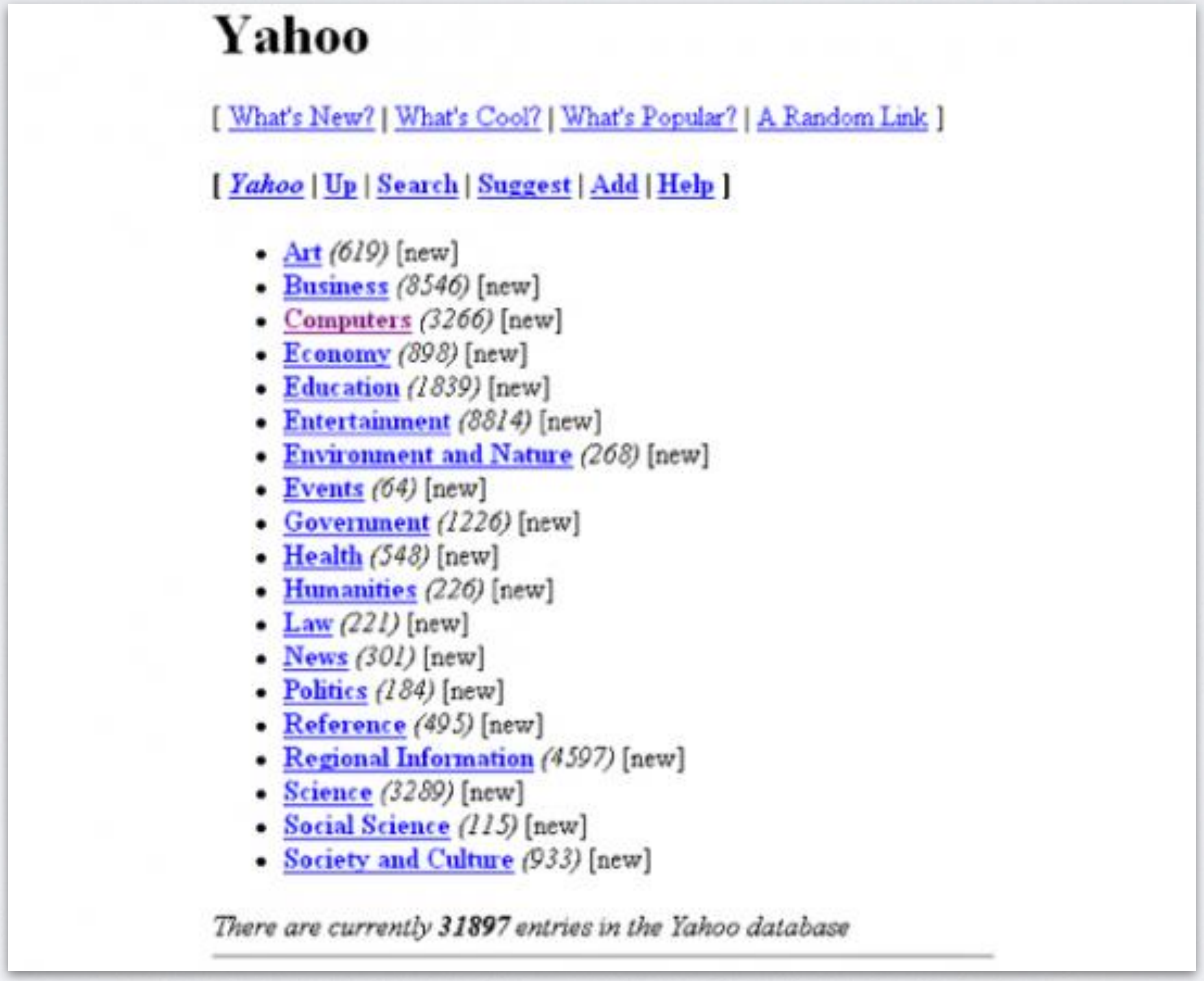
RESPONSIVE WEB DESIGN

THE MOBILE WEB AND MORE

TOPICS

- Web Page Layout: a historical perspective
- Grid Layouts
- The Mobile Web / Mobile First
- Responsive Design

NO STYLE!



Yahoo

[What's New? | What's Cool? | What's Popular? | A Random Link]

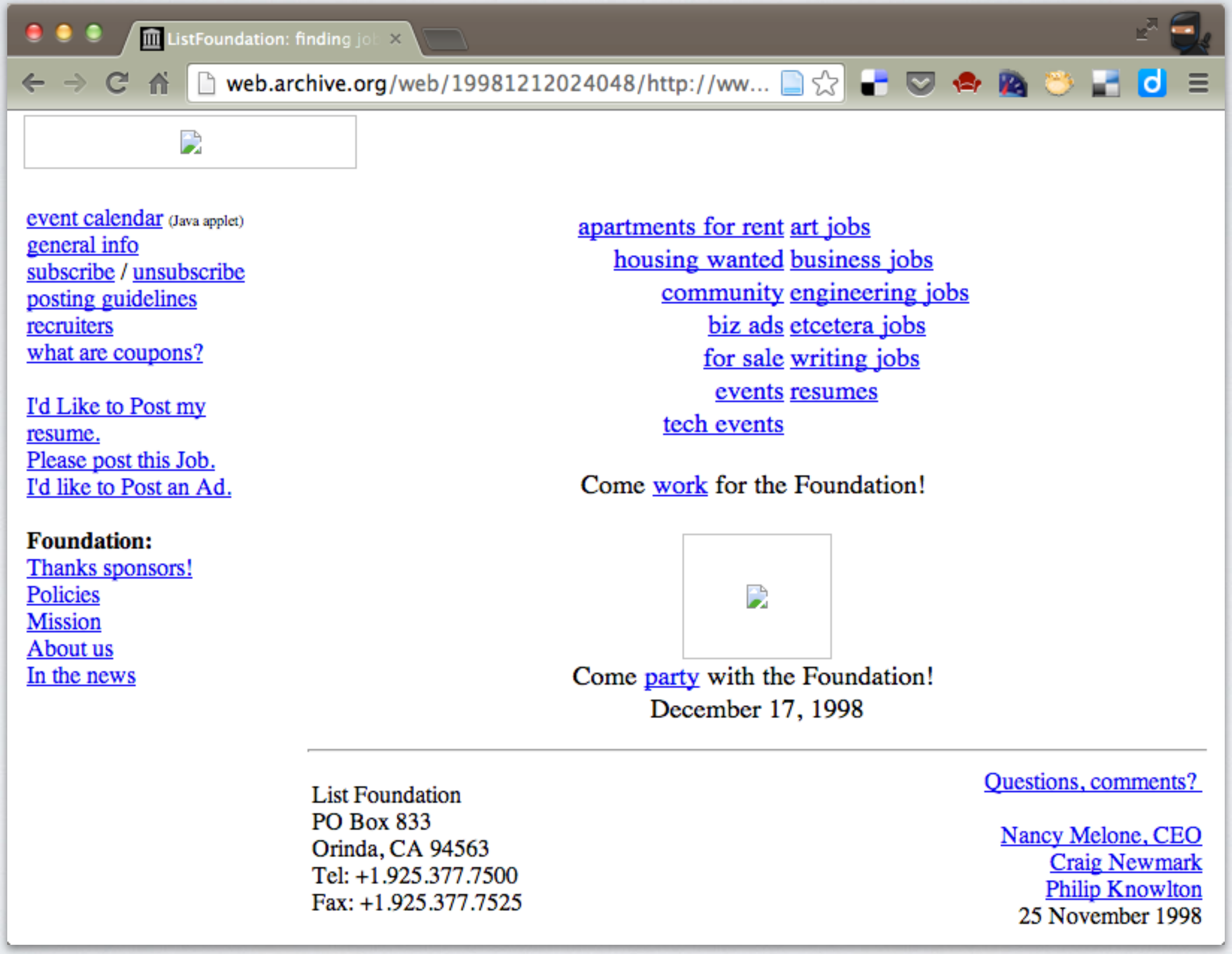
[*Yahoo* | Up | Search | Suggest | Add | Help]

- **Art** *(619)* [new]
- **Business** *(8546)* [new]
- **Computers** *(3266)* [new]
- **Economy** *(898)* [new]
- **Education** *(1839)* [new]
- **Entertainment** *(8814)* [new]
- **Environment and Nature** *(268)* [new]
- **Events** *(64)* [new]
- **Government** *(1226)* [new]
- **Health** *(548)* [new]
- **Humanities** *(226)* [new]
- **Law** *(221)* [new]
- **News** *(301)* [new]
- **Politics** *(184)* [new]
- **Reference** *(495)* [new]
- **Regional Information** *(4597)* [new]
- **Science** *(3289)* [new]
- **Social Science** *(115)* [new]
- **Society and Culture** *(933)* [new]

*There are currently **31897** entries in the Yahoo database*

yahoo.com circa 1995

NO STYLE!


ListFoundation: finding jo
web.archive.org/web/19981212024048/http://ww...

event calendar (Java applet)
general info
subscribe / unsubscribe
posting guidelines
recruiters
what are coupons?

I'd Like to Post my resume.
Please post this Job.
I'd like to Post an Ad.

Foundation:
Thanks sponsors!
Policies
Mission
About us
In the news

apartments for rent art jobs
housing wanted business jobs
community engineering jobs
biz ads etcetera jobs
for sale writing jobs
events resumes
tech events

Come work for the Foundation!

Come party with the Foundation!
December 17, 1998

List Foundation
PO Box 833
Orinda, CA 94563
Tel: +1.925.377.7500
Fax: +1.925.377.7525

Questions, comments?

Nancy Melone, CEO
Craig Newmark
Philip Knowlton
25 November 1998

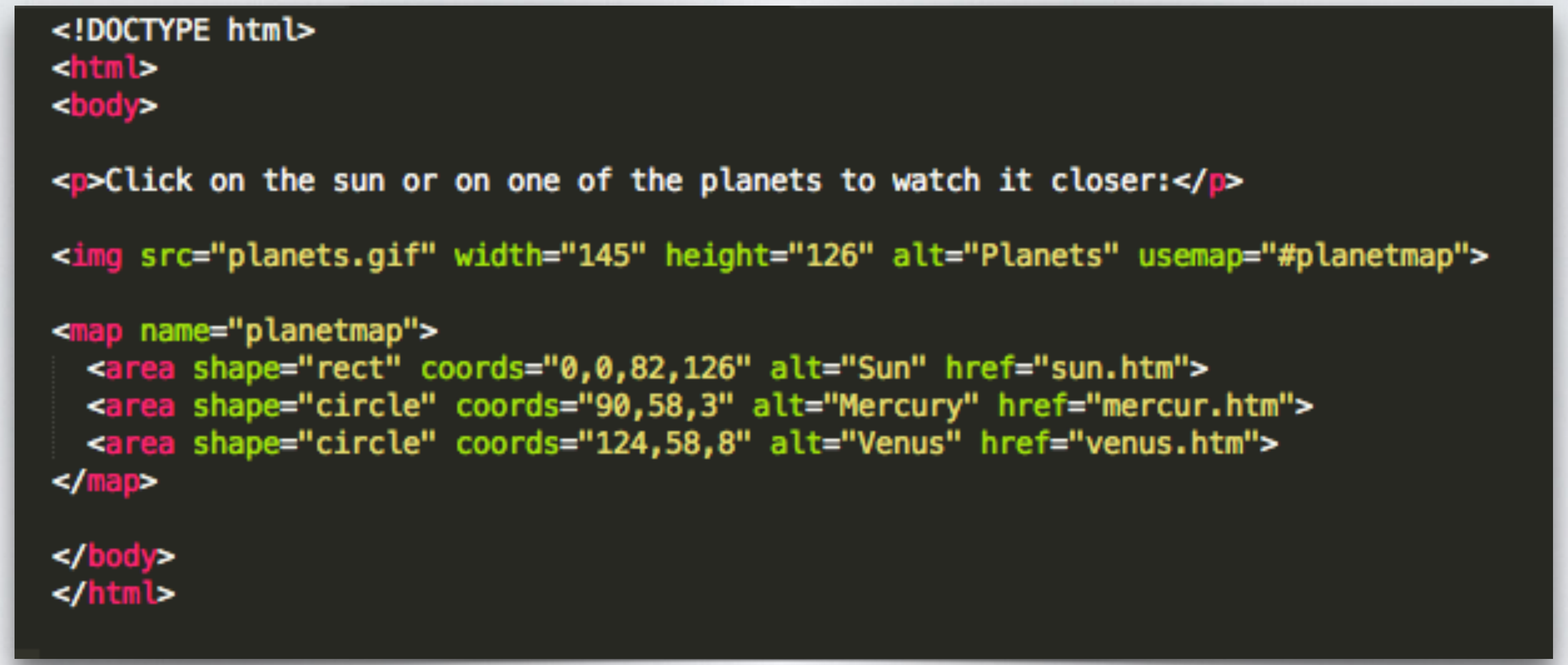
craigslist.org circa 1998

See: Why the Web Hasn't Birthed a Prettier Craigslist

EARLY ATTEMPTS AT STYLING

- Images: introduced in HTML 2.0 (1994)
- Image maps:
  - Mosaic 1.1: server side image maps (1995)
  - HTML 3.2 client side image maps. (1996)

CLIENT SIDE IMAGE MAPS



```html
<!DOCTYPE html>
<html>
<body>

<p>Click on the sun or on one of the planets to watch it closer:</p>

<img src="planets.gif" width="145" height="126" alt="Planets" usemap="#planetmap">

<map name="planetmap">
  <area shape="rect" coords="0,0,82,126" alt="Sun" href="sun.htm">
  <area shape="circle" coords="90,58,3" alt="Mercury" href="mercur.htm">
  <area shape="circle" coords="124,58,8" alt="Venus" href="venus.htm">
</map>

</body>
</html>
```

- Used to be used on almost every web page!

  - Used to implement navigation schemes

IMAGE MAPS - BAD IDEA

- Problems
  - Longer load times (think mobile)
  - Horrible from an accessibility perspective.
  - Painful to create.
  - Really out of style from a design perspective.

MORE ATTEMPTS AT STYLING

- Adobe Flash (introduced in 1996)
    - required a browser plug-in
    - allowed complex highly interactive websites to be built.
    - broke HTTP concept of addressability

Apple struck a very controversial chord when they banished Adobe Flash from their iOS platform. Was this primarily a technical decision or a business decision?

See Thoughts on Flash by Steve Jobs.

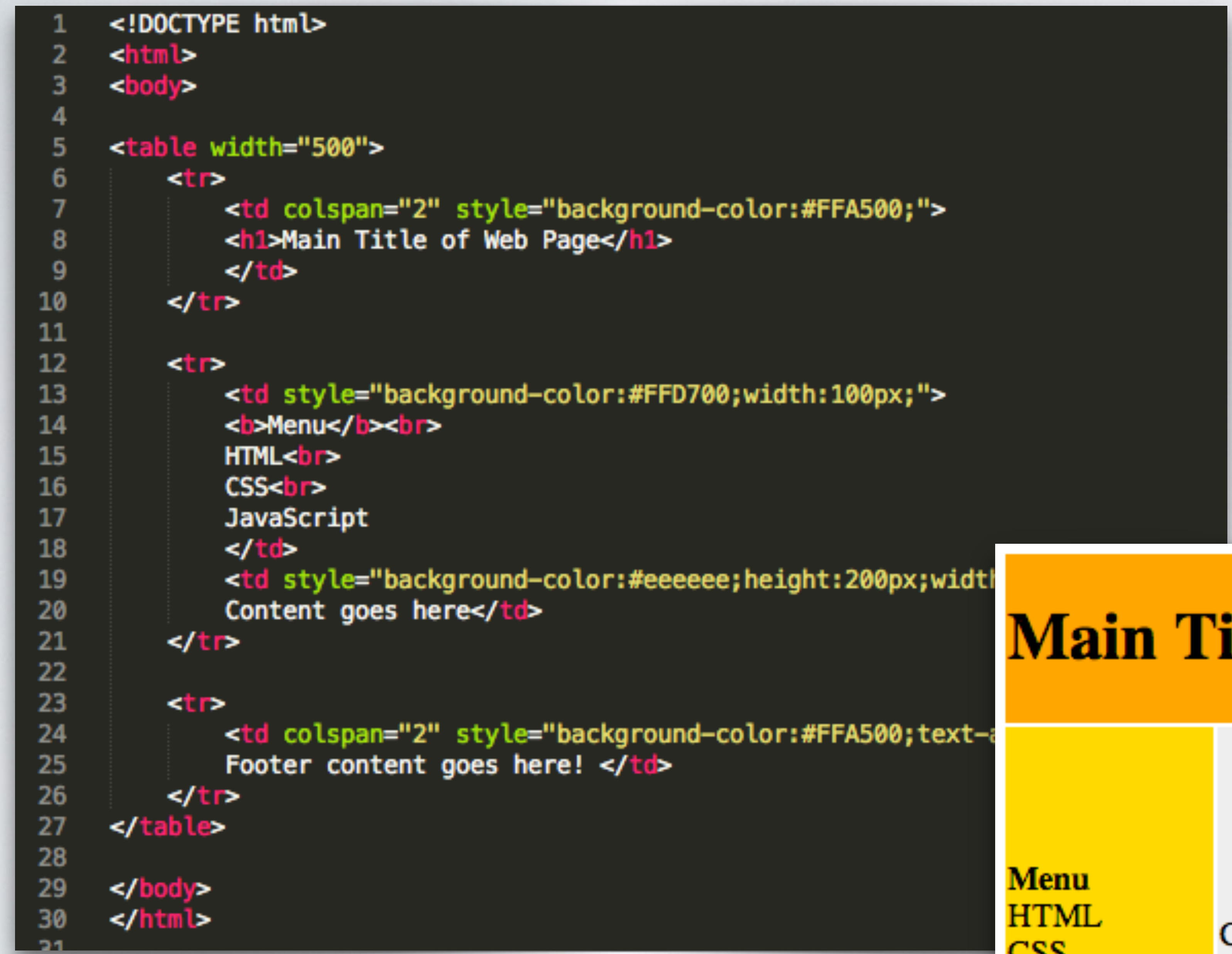
MODERN FLASH SITE EXAMPLES

- Flash is still being used… some interesting sites:
  - http://cloudsovercuba.com/
  - http://wechoosethemoon.org/

YET MORE ATTEMPTS AT STYLING

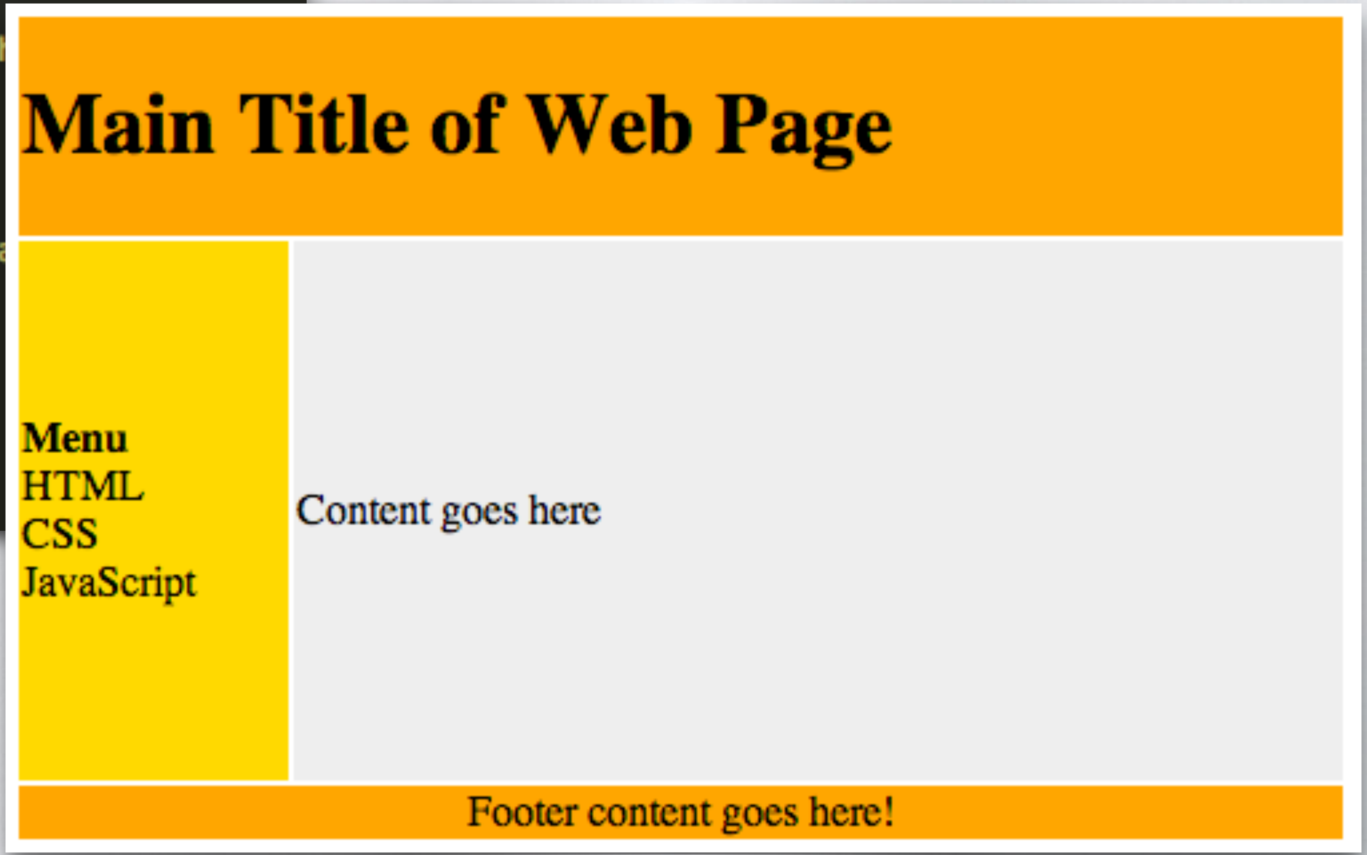
- Tables: introduced in “HTML 3.0” for display of tabular info. Quickly subverted for page layout. (1995)
- fonts, colors, background images, etc. HTML 3.2 (1996)

LAYOUT WITH TABLES

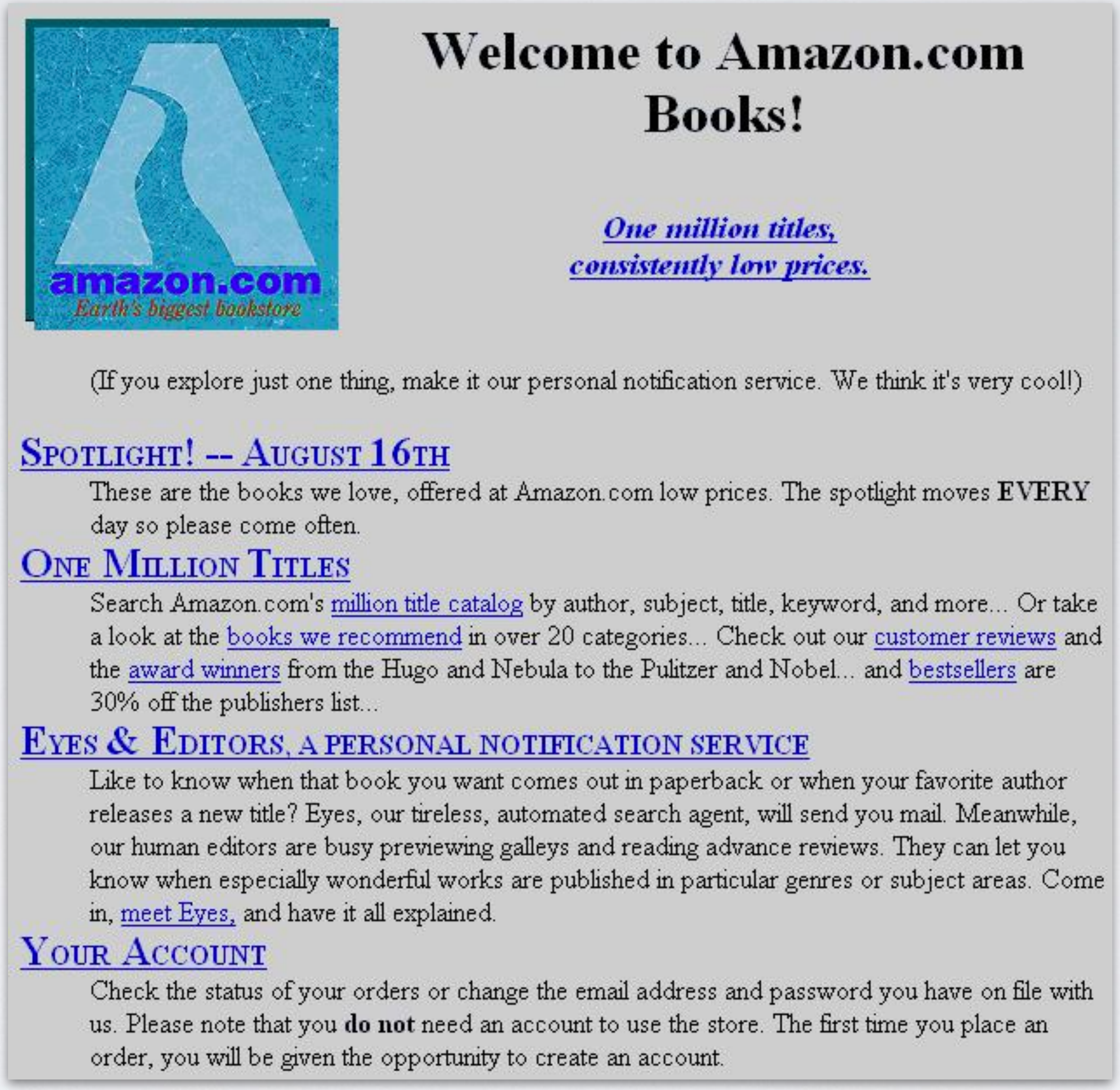


```
<!DOCTYPE html>
<html>
<body>

<table width="500">
    <tr>
        <td colspan="2" style="background-color:#FFA500;">
        <h1>Main Title of Web Page</h1>
        </td>
    </tr>

    <tr>
        <td style="background-color:#FFD700;width:100px;">
        <b>Menu</b><br>
        HTML<br>
        CSS<br>
        JavaScript
        </td>
        <td style="background-color:#eeeeee;height:200px;widt
        Content goes here</td>
    </tr>

    <tr>
        <td colspan="2" style="background-color:#FFA500;text-a
        Footer content goes here! </td>
    </tr>
</table>

</body>
</html>
```

(Note that CSS wasn't widely used until years later.)



PRIMITIVE LAYOUT WITH TABLES



Welcome to Amazon.com Books!

One million titles, consistently low prices.

(If you explore just one thing, make it our personal notification service. We think it's very cool!)

SPOTLIGHT! -- AUGUST 16TH

These are the books we love, offered at Amazon.com low prices. The spotlight moves **EVERY** day so please come often.

ONE MILLION TITLES

Search Amazon.com's million title catalog by author, subject, title, keyword, and more... Or take a look at the books we recommend in over 20 categories... Check out our customer reviews and the award winners from the Hugo and Nebula to the Pulitzer and Nobel... and bestsellers are 30% off the publishers list...

EYES & EDITORS, A PERSONAL NOTIFICATION SERVICE

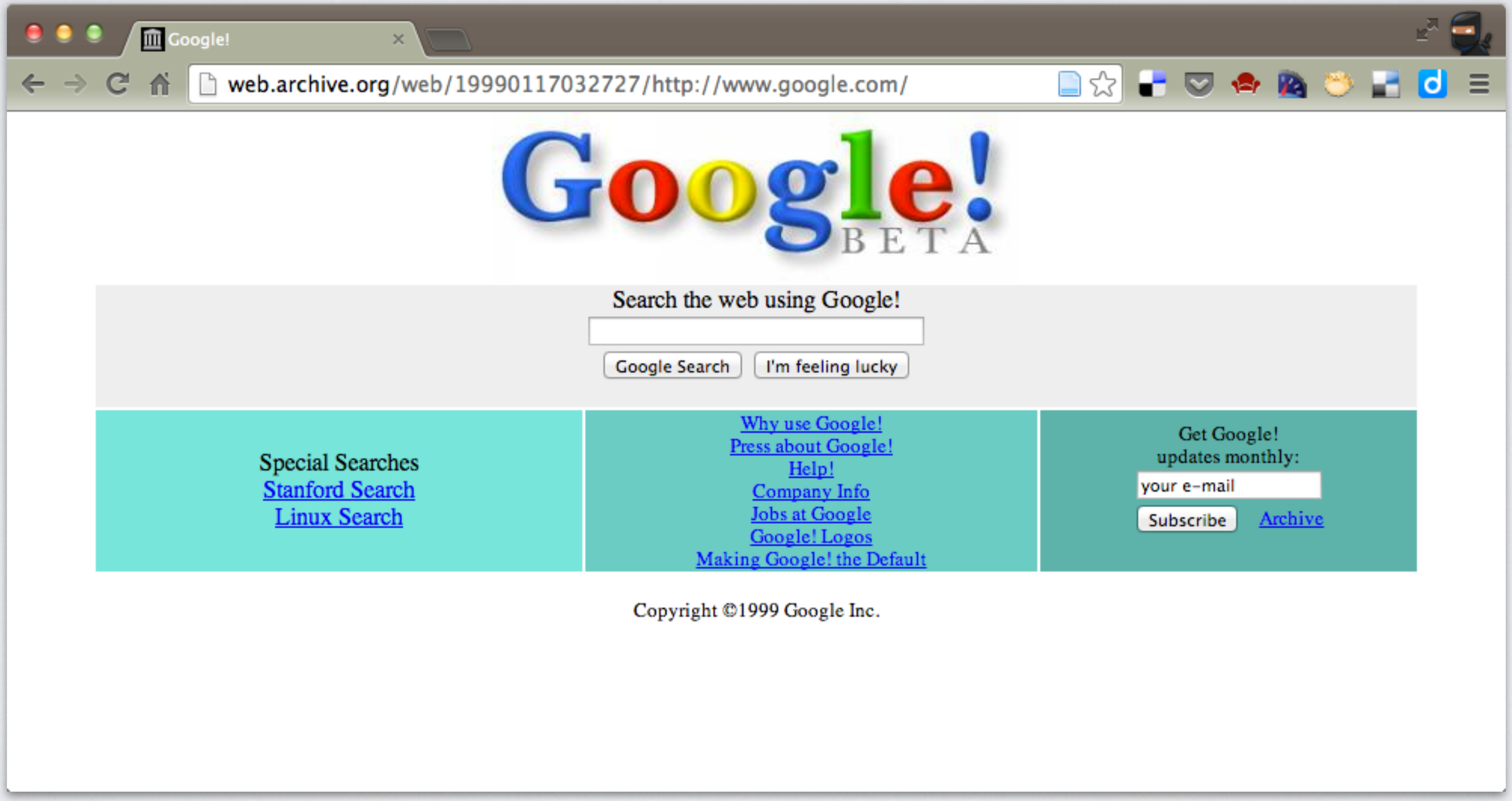
Like to know when that book you want comes out in paperback or when your favorite author releases a new title? Eyes, our tireless, automated search agent, will send you mail. Meanwhile, our human editors are busy previewing galleys and reading advance reviews. They can let you know when especially wonderful works are published in particular genres or subject areas. Come in, meet Eyes, and have it all explained.

YOUR ACCOUNT

Check the status of your orders or change the email address and password you have on file with us. Please note that you **do not** need an account to use the store. The first time you place an order, you will be given the opportunity to create an account.

amazon.com circa 1996?

VINTAGE GOOGLE!



Google!

web.archive.org/web/19990117032727/http://www.google.com/

Google! BETA

Search the web using Google!

Google Search | I'm feeling lucky

Special Searches
Stanford Search
Linux Search

Why use Google!
Press about Google!
Help!
Company Info
Jobs at Google
Google! Logos
Making Google! the Default

Get Google!
updates monthly:
your e-mail
Subscribe | Archive

Copyright ©1999 Google Inc.

google.com circa 1999

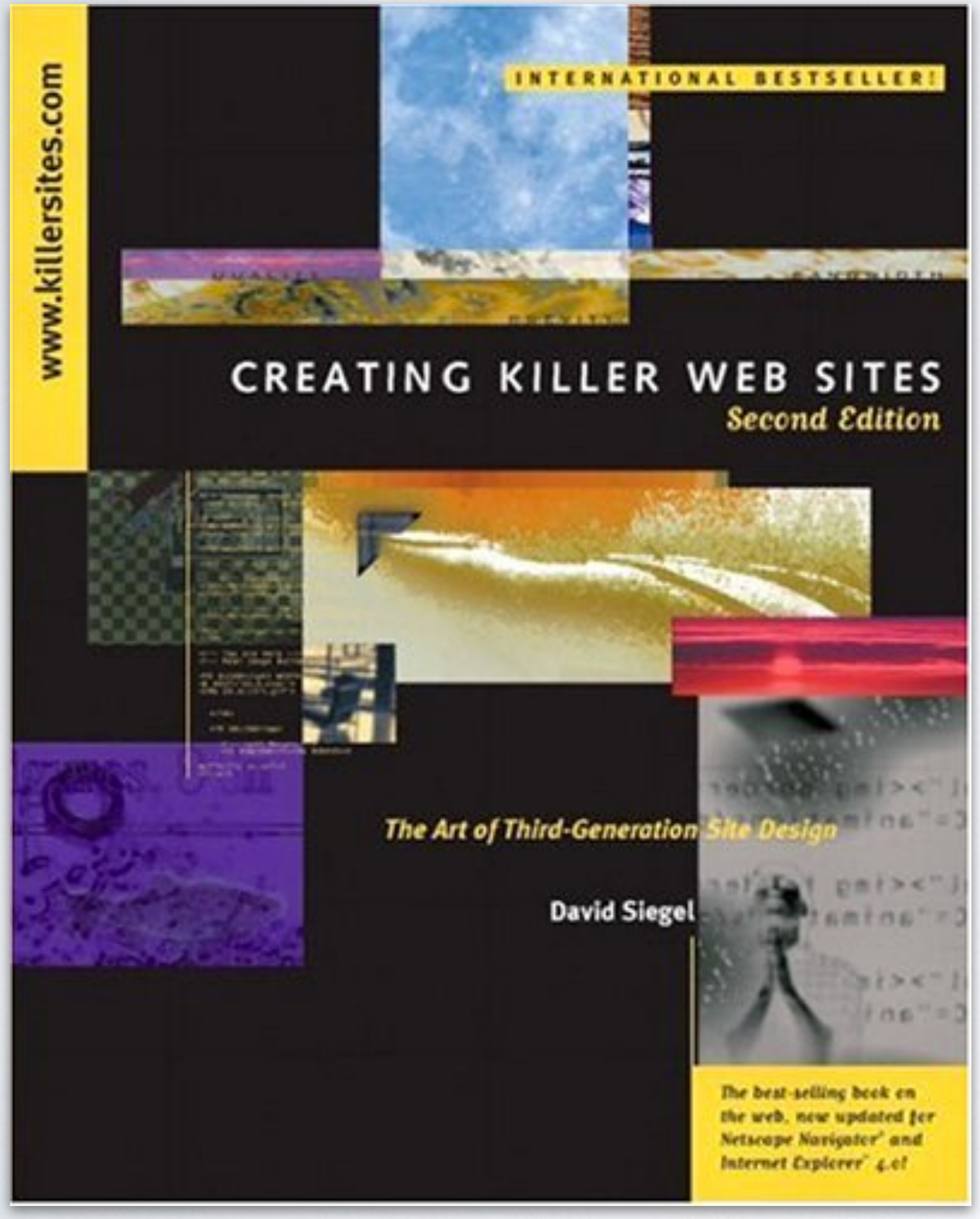
DON'T USE TABLES FOR LAYOUT!

- Using HTML Tables for layout purposes is NOT a best practice:
  - mixes presentational data in with your content. (why is this bad?)
  - makes redesigns of existing sites and content extremely labor intensive
  - makes it extremely hard (and expensive) to maintain visual consistency throughout a site.

DON'T USE TABLES FOR LAYOUT (CONTINUED)!

- Using HTML Tables for layout purposes is NOT a best practice (continued):
  - hard to get same content to display well on smaller screens.
  - does not lend itself well to accessibility.

HISTORICAL FOOTNOTE: A PREVIOUSLY POPULAR BOOK!


www.killersites.com
INTERNATIONAL BESTSELLER!
CREATING KILLER WEB SITES
Second Edition
The Art of Third-Generation Site Design
David Siegel
The best-selling book on the web, now updated for Netscape Navigator® and Internet Explorer® 4.0!

- Originally published in 1996 and one of the most popular books on the net.
- You still find people using Siegel's work arounds!

WEB PAGE LAYOUT TODAY

- HTML + CSS has largely eclipsed using HTML table elements or Flash for laying out web pages.

What are the advantages of using HTML/
CSS instead of HTML tables, etc.?

REASON #1

- By removing presentational markup from your pages, redesigns of existing sites and content is much less labor intensive.

- To change the layout of the site, all you need to do is change the style sheets; you do not need to edit the pages themselves at all.

Check out the CSS Zen Garden.

REASON #2

- Easy to maintain visual consistency throughout a site.
- Since pages use the same CSS document for their layout, they are all formatted the same.
- Improve usability and overall quality of the site.

REASON #3

- Improved accessibility for users with disabilities and users accessing with smartphones / tablets.
- separating content from presentation boosts device independence.

REASON #4

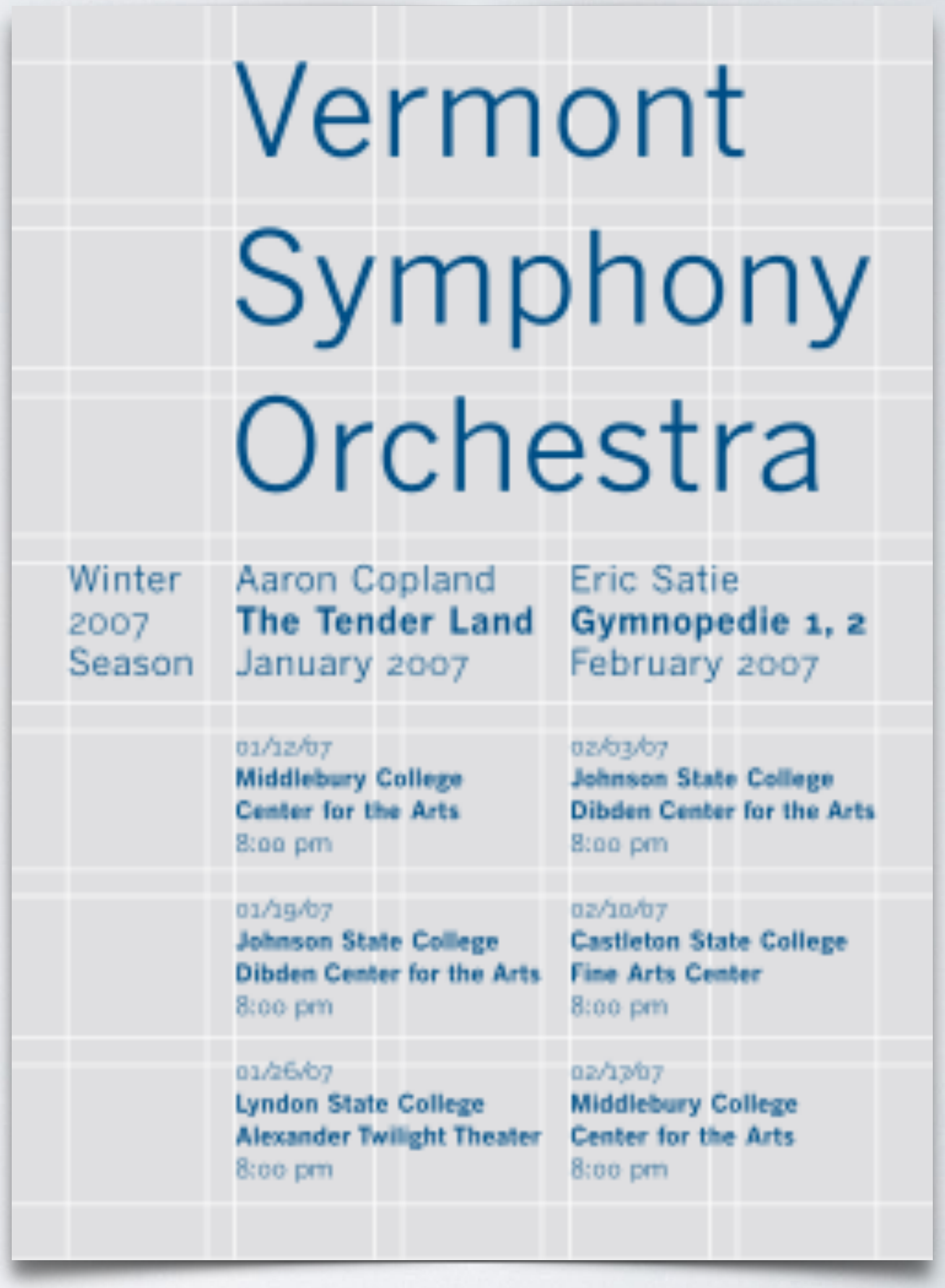
- Using HTML/CSS reduces the file sizes of your pages, as users no longer need to download presentational data with each page they visit.
- Stylesheets can be cached
- Faster page loads and less data used (big + for mobile)!

REASON #5

- minimizing markup and using header tags properly **improves search engine ranking**.
    - reduce the ratio of code to content
    - use keywords in the header tags
    - replace header GIFs with actual text

GRID BASED LAYOUTS

- Technique developed for and widely used in traditional print publishing.
- Popularized in web design by a number of recent web UI frameworks



Vermont Symphony Orchestra

Winter 2007 Season

Aaron Copland
The Tender Land
January 2007

Eric Satie
Gymnopedie 1, 2
February 2007

01/12/07
Middlebury College
Center for the Arts
8:00 pm

02/03/07
Johnson State College
Dibden Center for the Arts
8:00 pm

01/19/07
Johnson State College
Dibden Center for the Arts
8:00 pm

02/10/07
Castleton State College
Fine Arts Center
8:00 pm

01/26/07
Lyndon State College
Alexander Twilight Theater
8:00 pm

02/17/07
Middlebury College
Center for the Arts
8:00 pm

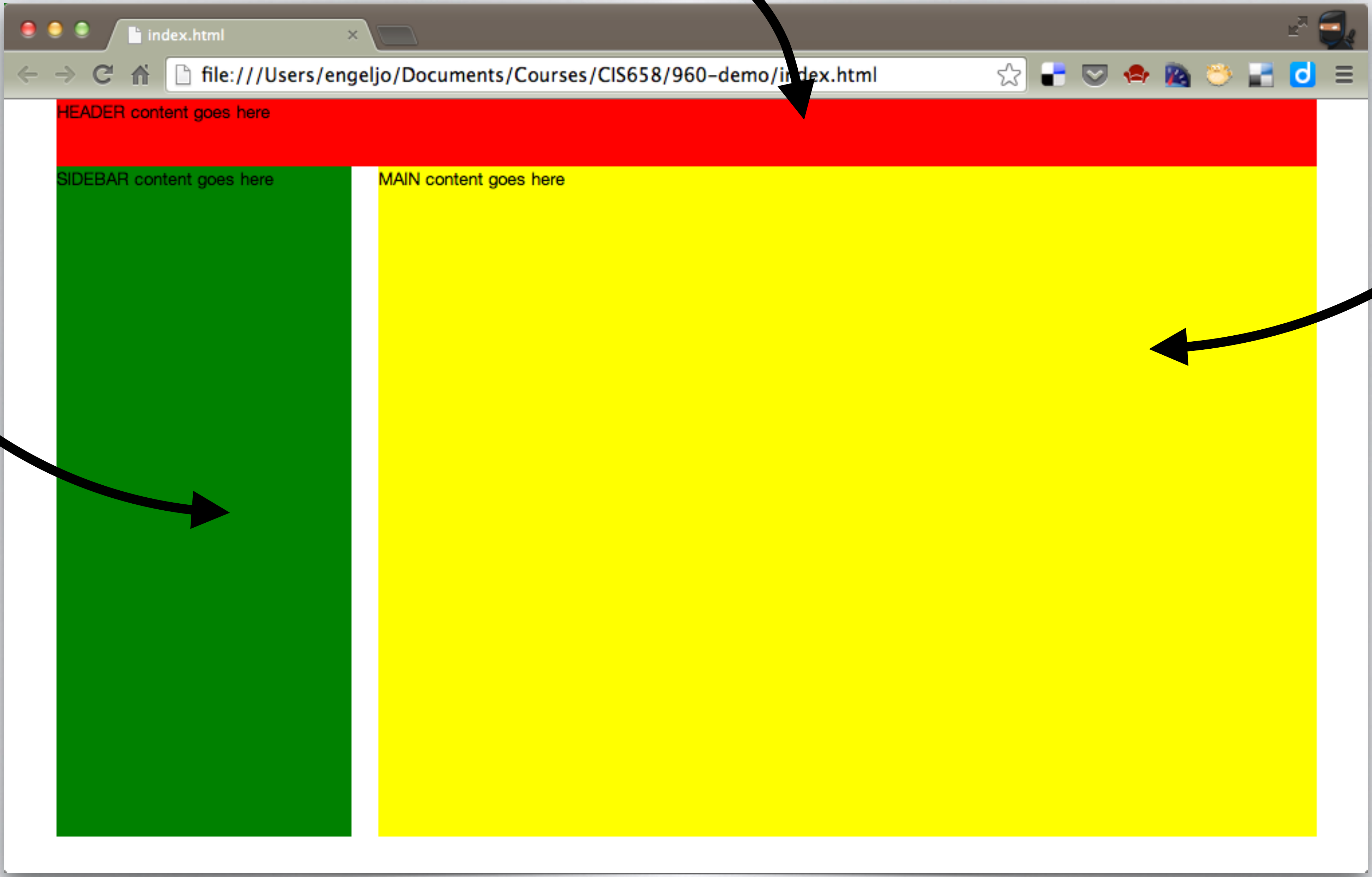
Source: http://en.wikipedia.org/wiki/Grid_(graphic_design)

960 GRID SYSTEM

- Popular CSS layout framework.
- modern monitors support at least 1024 × 768 pixel resolution.
- 960 is divisible by 2, 3, 4, 5, 6, 8, 10, 12, 15, 16, 20, 24, 30, 32, 40, 48, 60, 64, 80, 96, 120, 160, 192, 240, 320 and 480, making it a flexible number to work with.

960 GRID LAYOUT EXAMPLE

940px + 20px margins = 960 wide

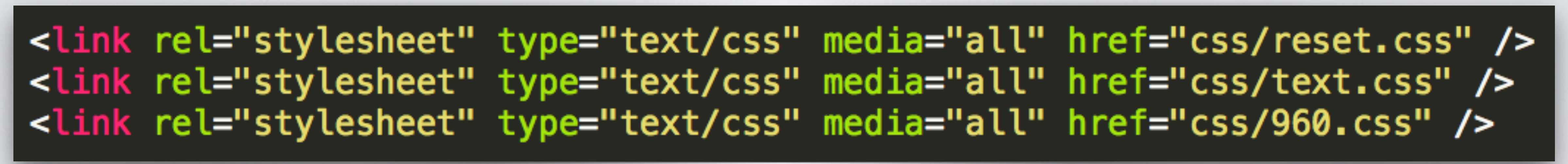

index.html
file:///Users/engeljo/Documents/Courses/CIS658/960-demo/index.html
HEADER content goes here
SIDEBAR content goes here
MAIN content goes here

700px

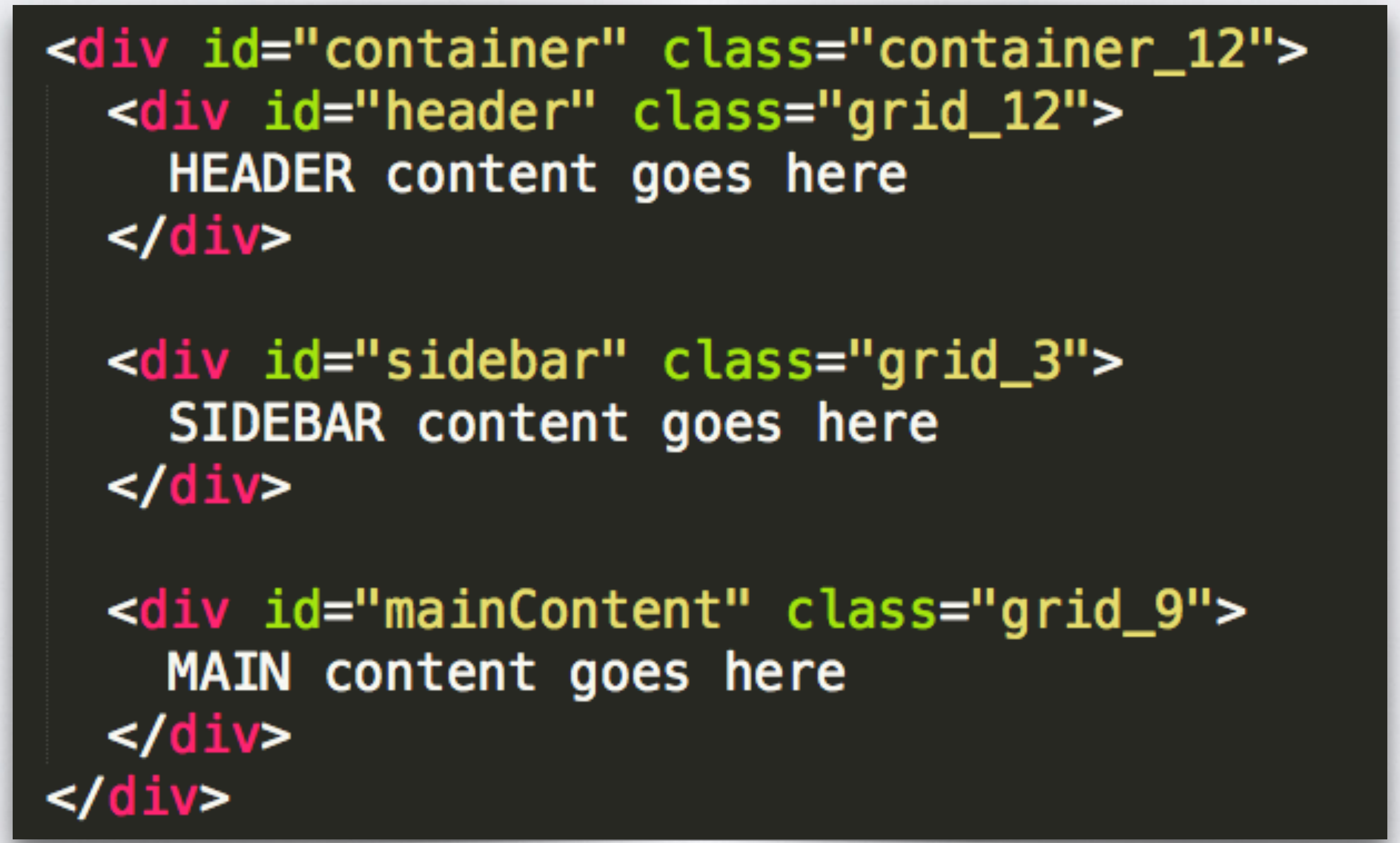
220px

12 column layout: sidebar is 3 column, body 9

960 GRID LAYOUT CODE EXAMPLE



```html
<link rel="stylesheet" type="text/css" media="all" href="css/reset.css" />
<link rel="stylesheet" type="text/css" media="all" href="css/text.css" />
<link rel="stylesheet" type="text/css" media="all" href="css/960.css" />
```



```html
<div id="container" class="container_12">
  <div id="header" class="grid_12">
    HEADER content goes here
  </div>

  <div id="sidebar" class="grid_3">
    SIDEBAR content goes here
  </div>

  <div id="mainContent" class="grid_9">
    MAIN content goes here
  </div>
</div>
```

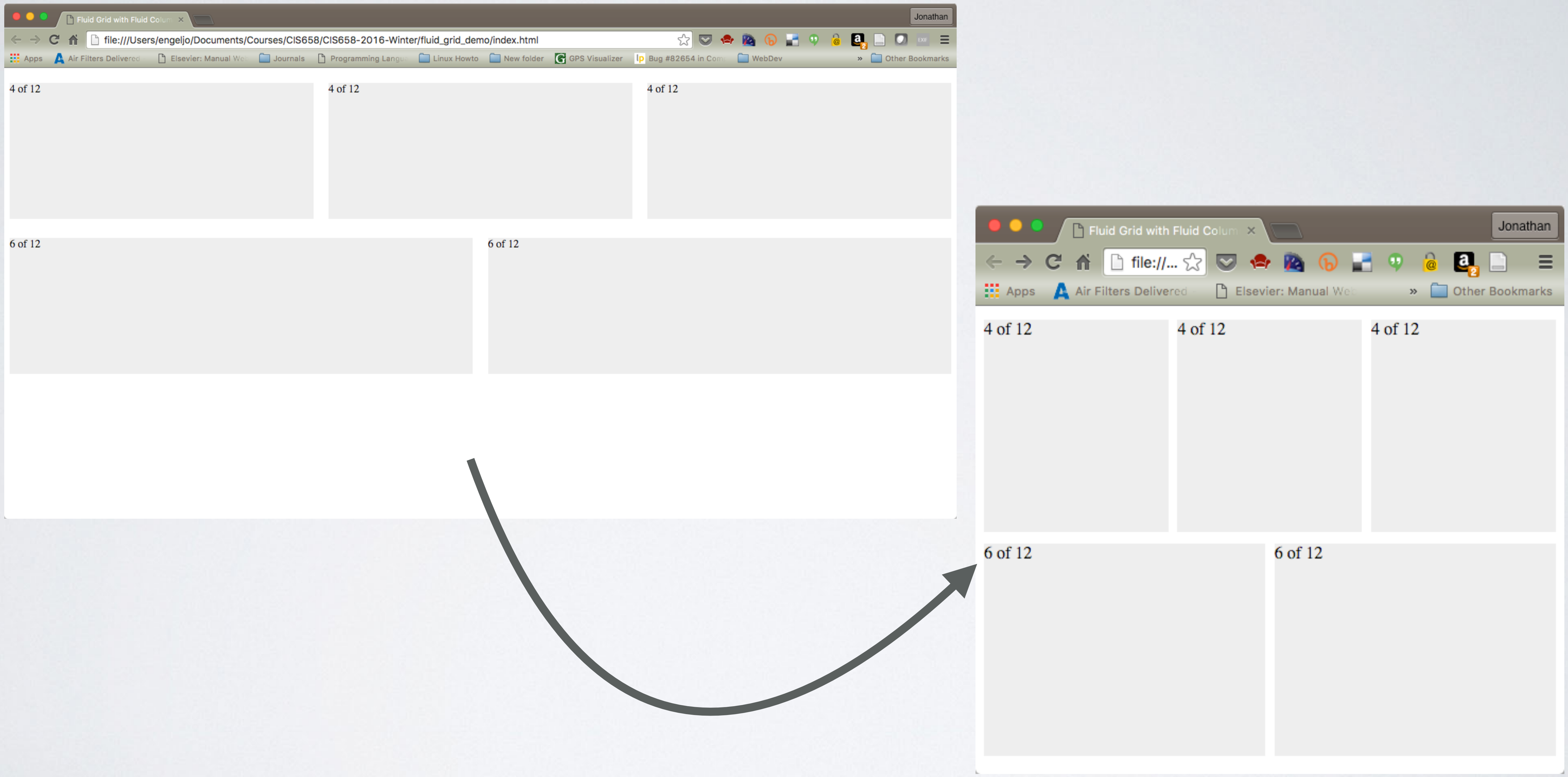
GRID LAYOUT TAKEAWAYS

- Much less verbose than table layouts
- Styling separate from content
- But what if screen size > 960 or screen size < 960?

FLUID GRIDS

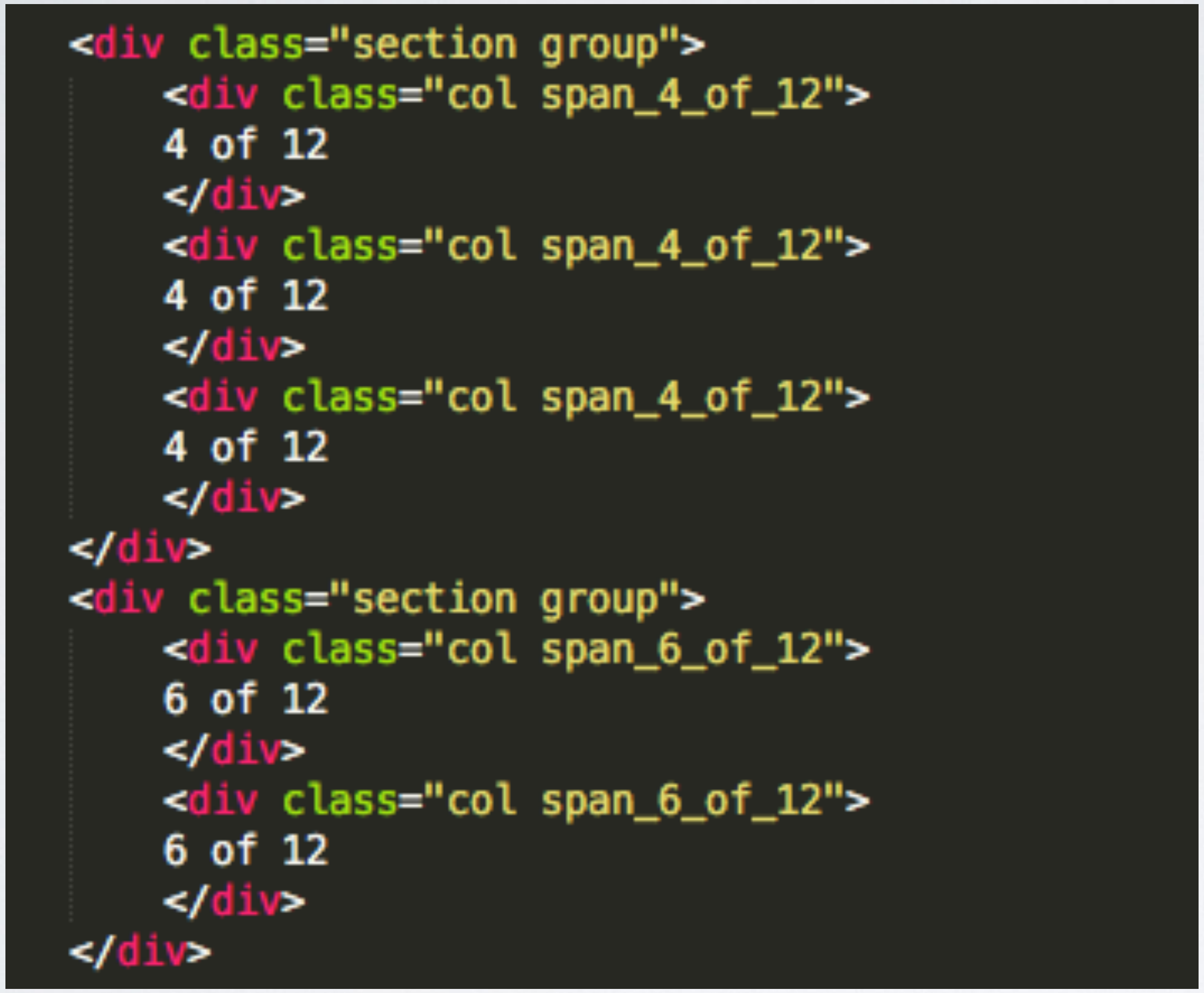
- graceful (fluid like) support for different size screens.
- designer specifies a max container width
- layout is scaled in proportion to the container (widths specified as percentages).

EXAMPLE: FLUID GRID

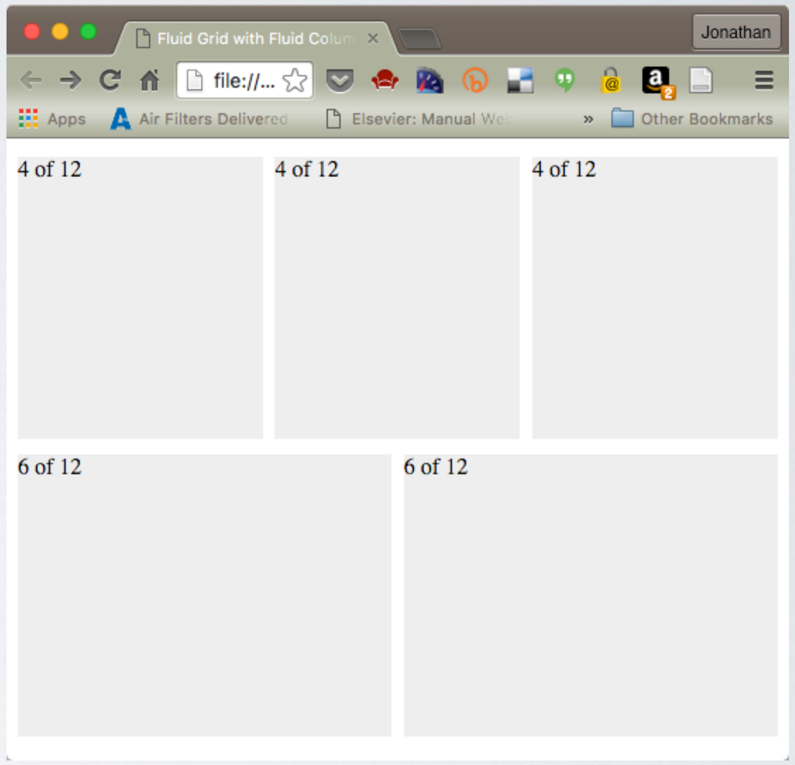


Example coded with: http://www.responsivegridsystem.com/calculator/

EXAMPLE: FLUID GRID



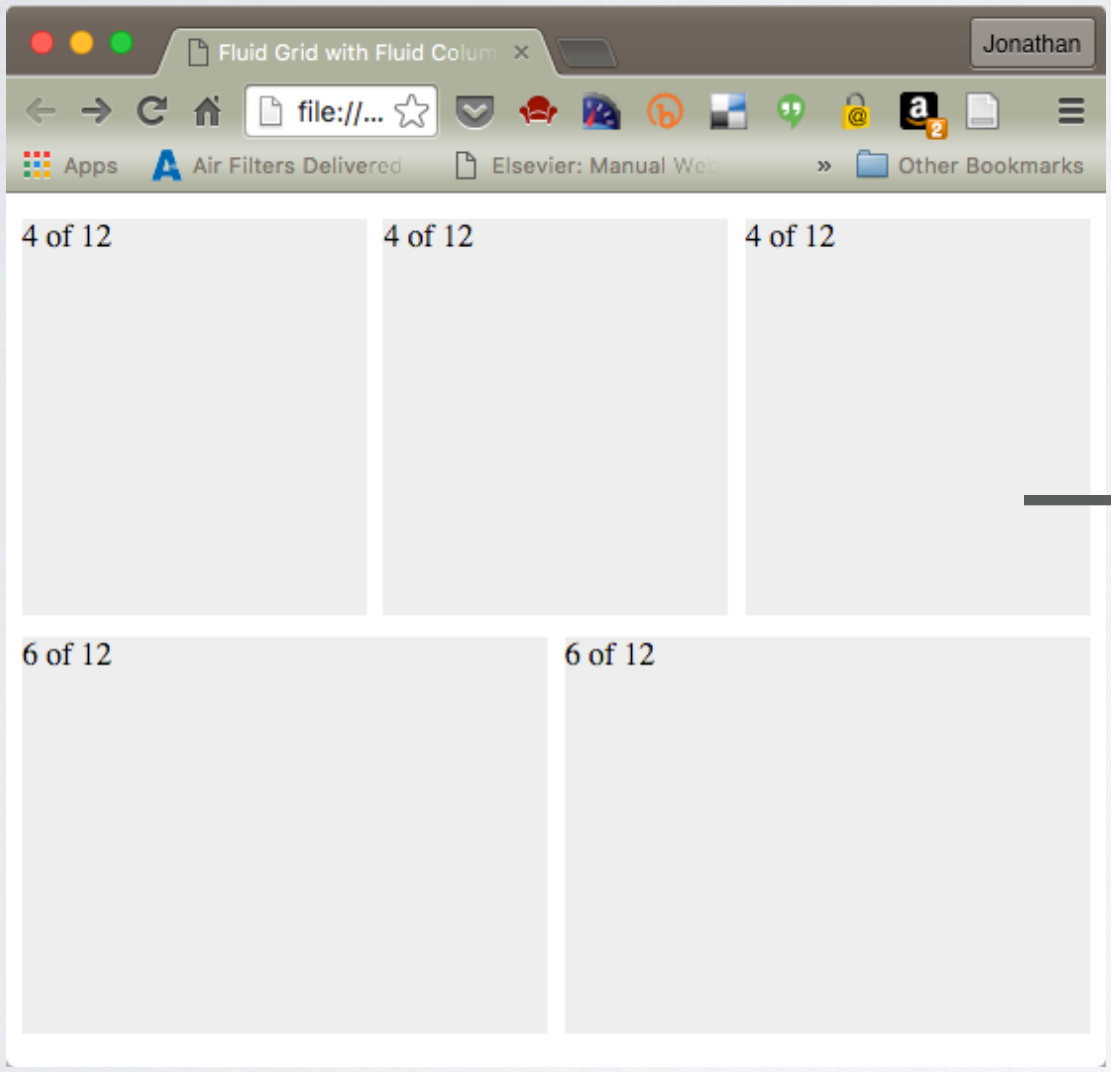
```
<div class="section group">
    <div class="col span_4_of_12">
    4 of 12
    </div>
    <div class="col span_4_of_12">
    4 of 12
    </div>
    <div class="col span_4_of_12">
    4 of 12
    </div>
</div>
<div class="section group">
    <div class="col span_6_of_12">
    6 of 12
    </div>
    <div class="col span_6_of_12">
    6 of 12
    </div>
</div>
```

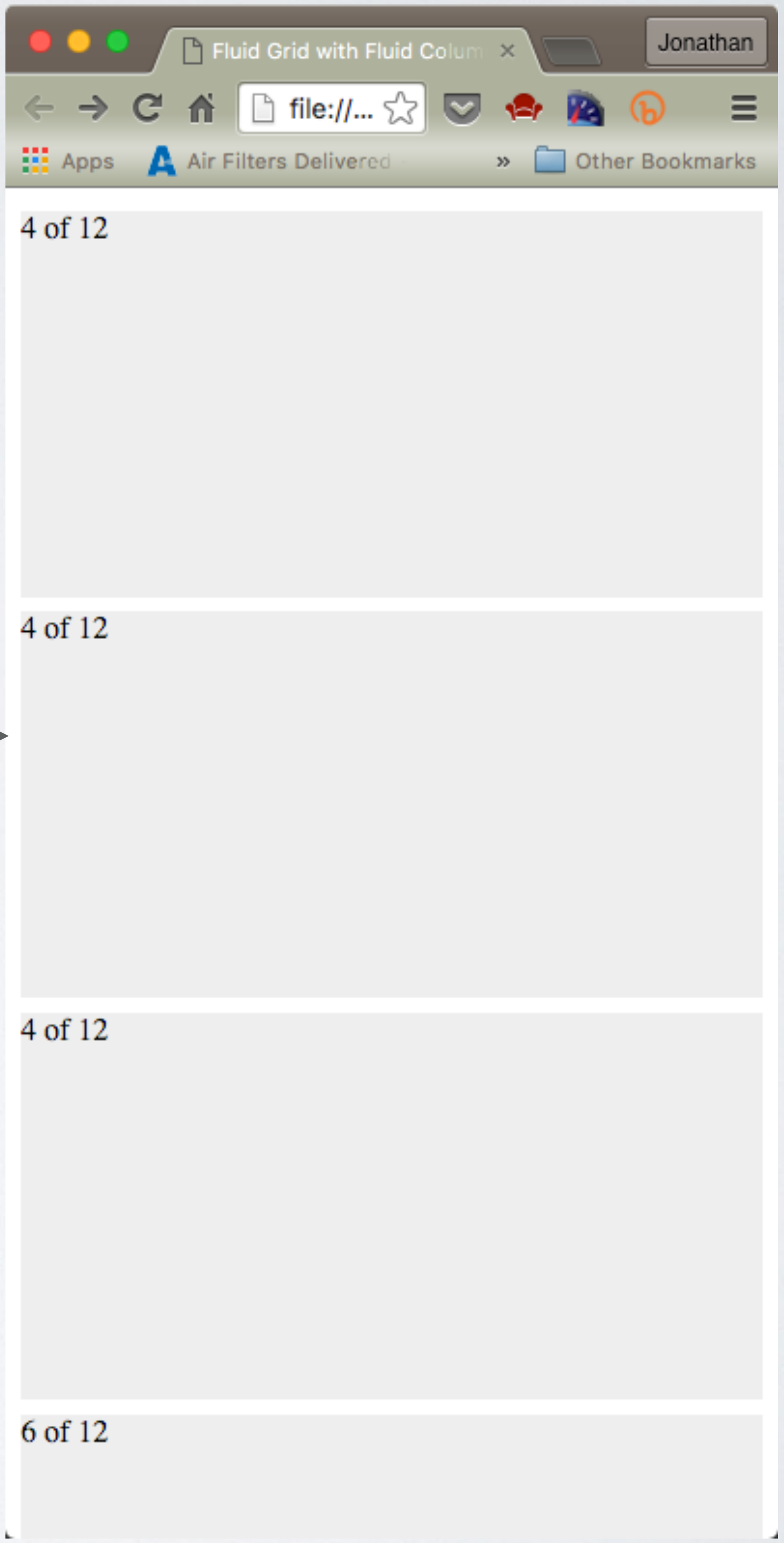


- What happens if content cannot be scaled any further?

Example coded with: http://www.responsivegridsystem.com/calculator/

RESPONSIVE DESIGN





THE BIG LAYOUT CHALLENGE

Smartphones
Tablets

Computers

Big Screens

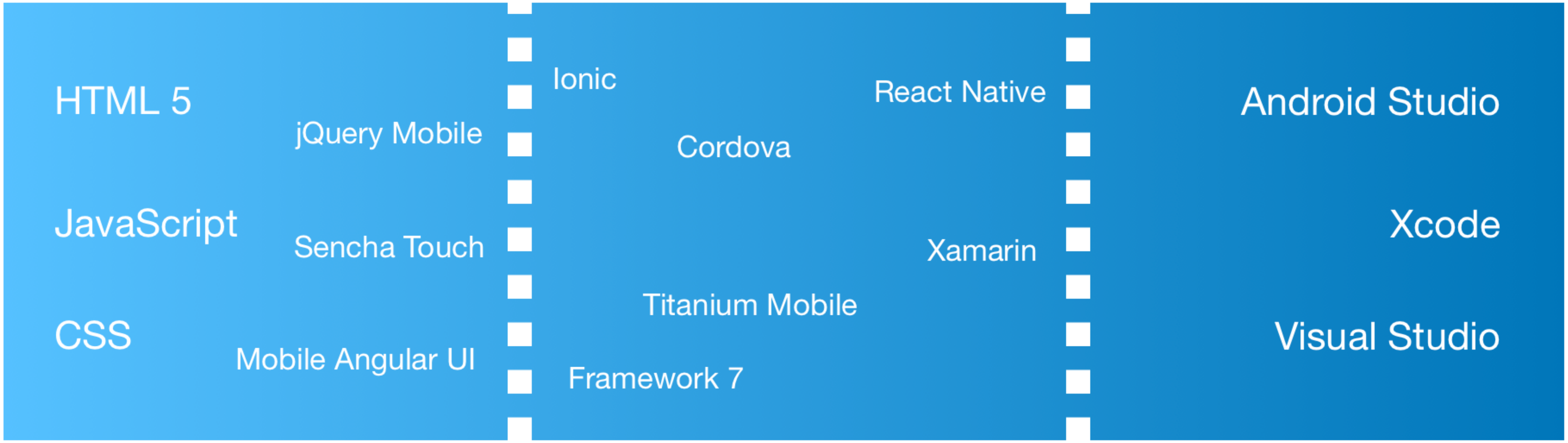
Small
Medium
Large

How can we develop a single web app and have it look good on every possible screen?

What are our options for getting content onto mobile devices and tablets? Native app ecosystems are hugely popular. Do we even need to use the web to deliver content to smartphones and tablets?

MOBILE APP DEVELOPMENT

Mobile Web Apps **Hybrid Apps** **Native Apps**



Web apps

Frameworks that generate native apps, typically based on web technologies. (Xamarin and React Native use native components)

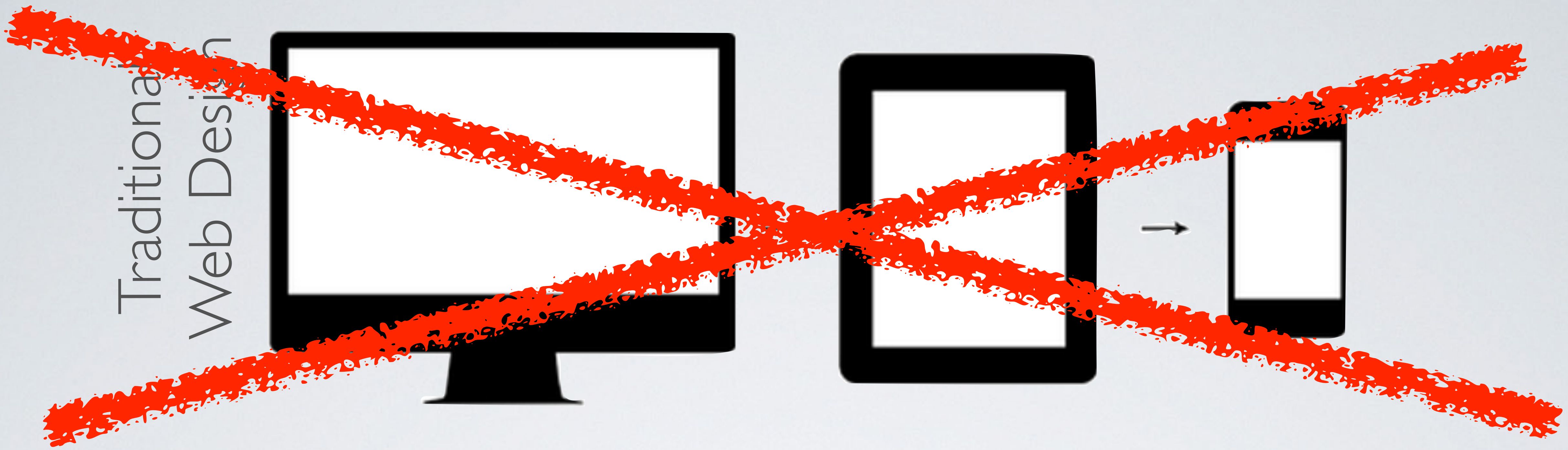
Apps specifically written for a single platform.

GO NATIVE OR NOT??

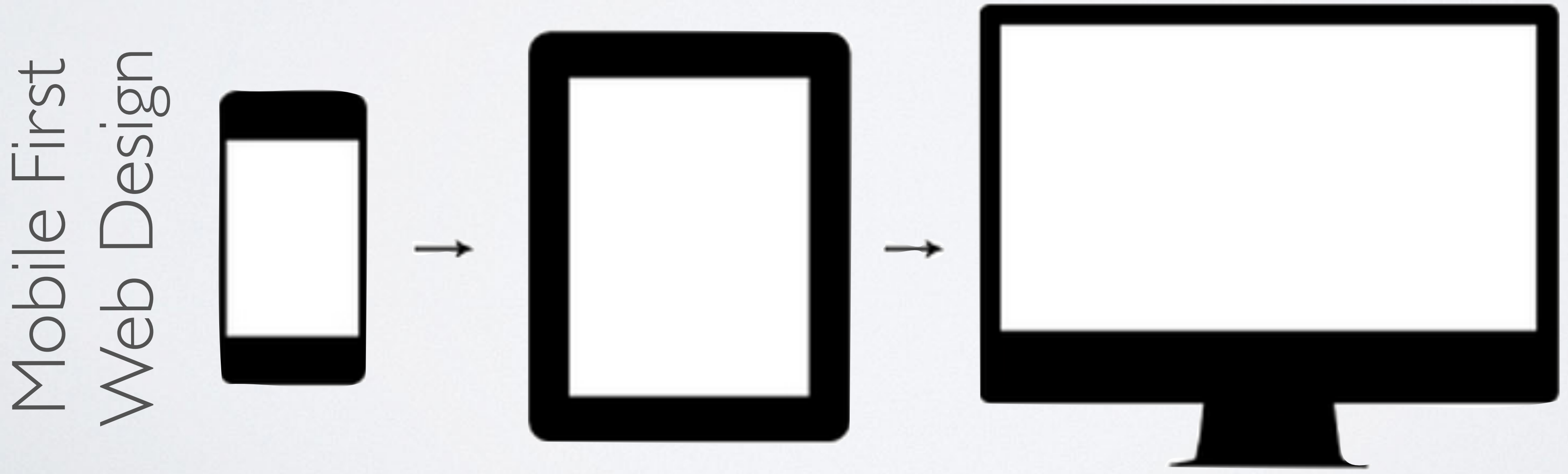
- Gartner's "Rule of Three": use Mobile Enterprise Application Platform (MEAP) when:
  - there are 3 or more mobile applications
  - there are 3 or more targeted operating systems or platforms
  - they involve the integration of 3 or more back-end systems

ENGELSMA'S TAKE

- Economics are important, but be user centric (as budget allows):
  - If possible, go native for consumer or customer facing apps
  - Hybrid / Web approach should be considered when you have a "captive audience", e.g. employee-only app.
  - Web vs. Hybrid: "app store" distribution is most familiar to end users today.



Definition: Mobile First - when designing a new web app, first begin with the mobile design and then work up to a larger desktop version.



MOBILE FIRST

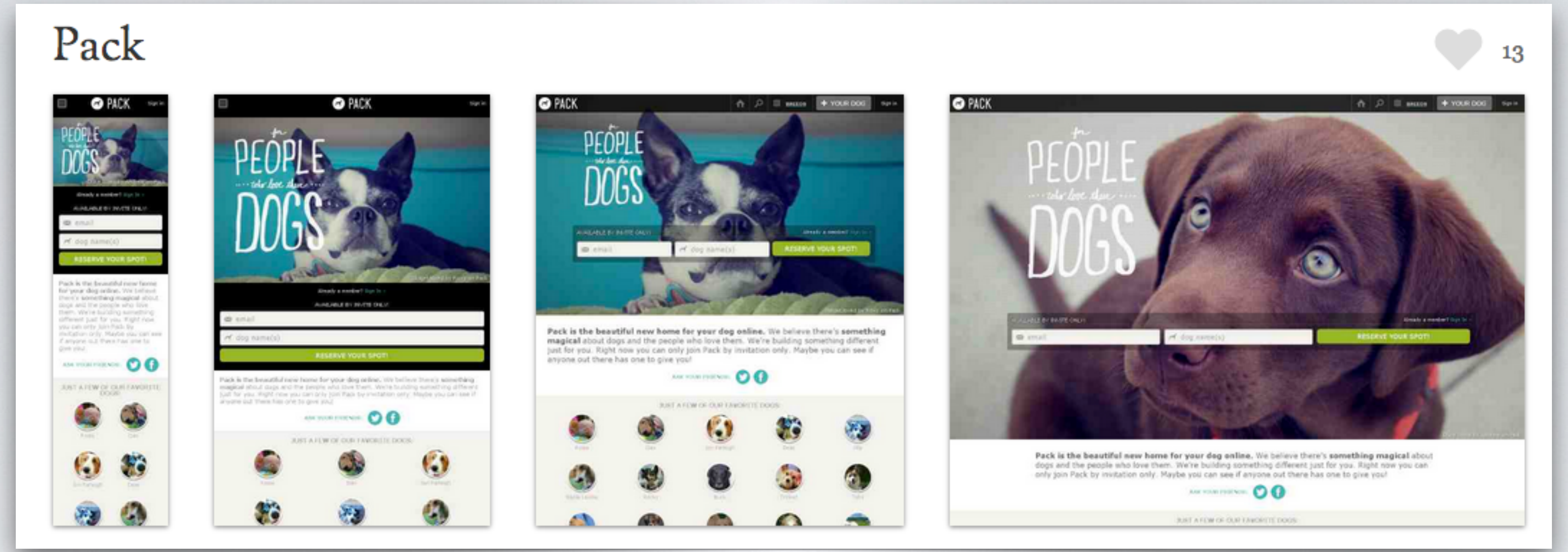
- Why?
  - Prioritizes content
    - only have room for the important stuff!
  - Allows progressive enhancement
    - Easier to fill up space than to compress smaller?
  - Why not?
    - Designing for all screens is difficult!

RESPONSIVE WEB DESIGN

- a design approach aimed at crafting sites to provide an optimal viewing experience - easy reading and navigation with minimum resizing, panning, and scrolling, across a wide range of devices, from small screens to large screens.

Source: http://en.wikipedia.org/wiki/Responsive_Web_Design

RESPONSIVE WEB DESIGN

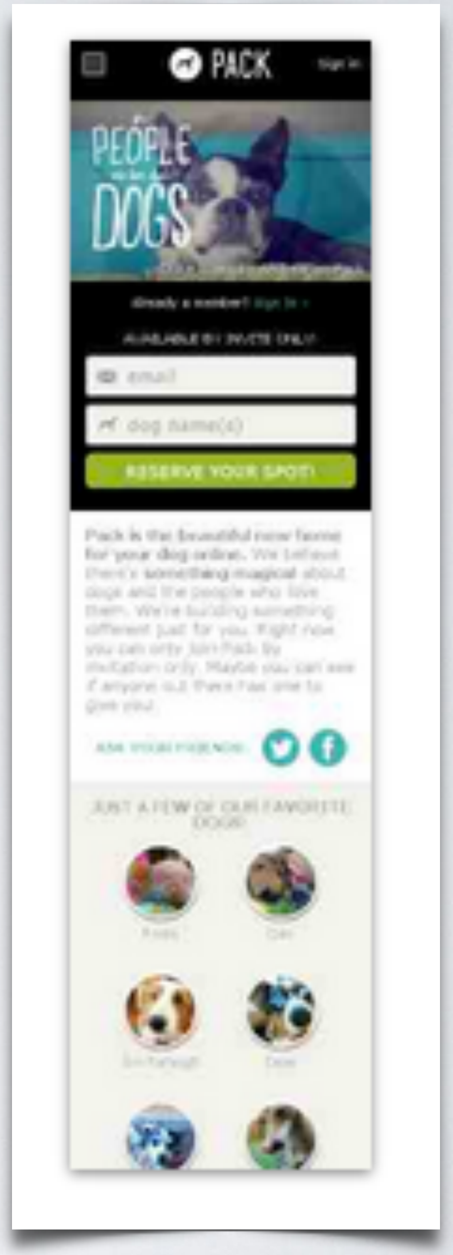


Pack is the beautiful new home for your dog online. We believe there's something magical about dogs and the people who love them. We're building something different just for you. Right now you can only join Pack by invitation only. Maybe you can see if anyone out there has one to give you!

Source: http://mediaqueri.es/

KEY COMPONENTS OF RESPONSIVE WEB DESIGN

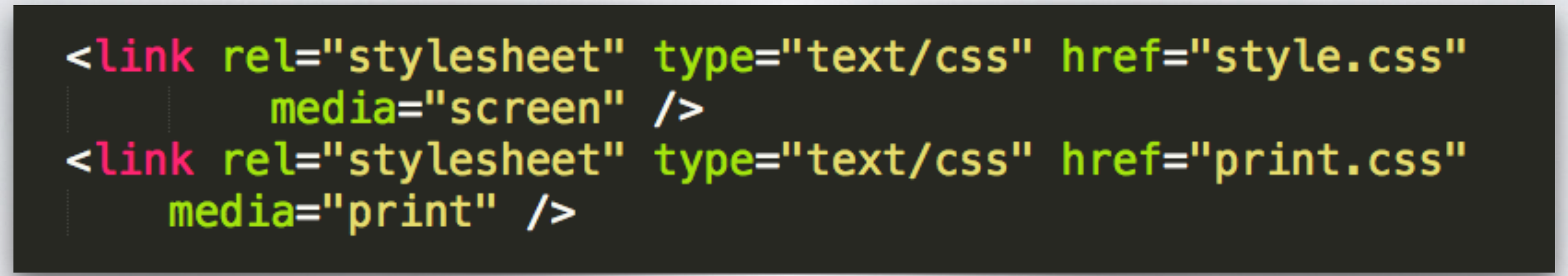
- Fluid grid (relative sizing)
- Flexible images (see http://alistapart.com/article/fluid-images)
- Media queries to switch out styling rules based on viewing device.


PACK
Sign in
PEOPLE
DOGS
Already a member? Sign in
AVAILABLE BY INVITE ONLY!
email
dog name(s)
RESERVE YOUR SPOT!
ASK YOUR FRIENDS!
JUST A FEW OF OUR FAVORITE DOGS!

Pack is the beautiful new home for your dog online. We believe there's something magical about dogs and the people who love them. We're building something different just for you. Right now you can only join Pack by invitation only. Maybe you can see if anyone out there has one to give you!

MEDIA QUERIES (1)

- CSS 2.1 introduced a concept called *media types*:



```
<link rel="stylesheet" type="text/css" href="style.css"
        media="screen" />
<link rel="stylesheet" type="text/css" href="print.css"
    media="print" />
```

- Other media types included "braille", "speech", "handheld" "tv" and more.

- Most were never implemented by browsers!

Reference: http://www.w3.org/TR/CSS21/media.html#media-types

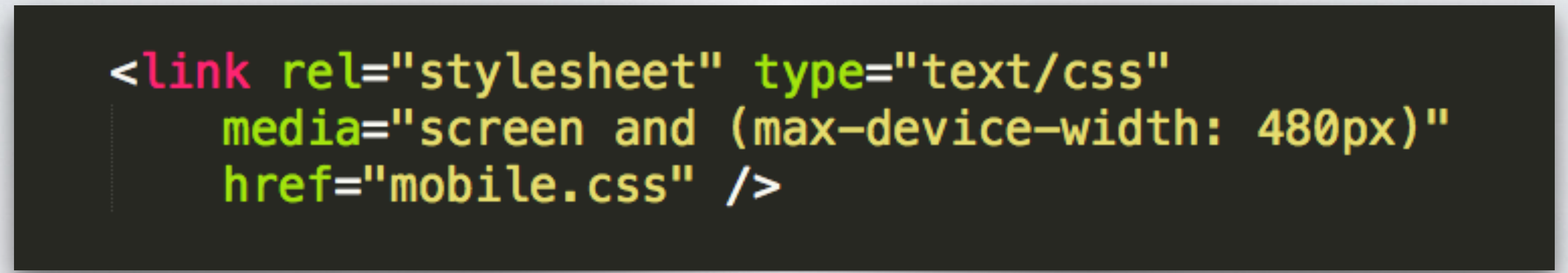
MEDIA QUERIES (2)

- CSS3 introduced a concept called *media queries*:
- Finer control, e.g., target not only the device class, but its physical characteristics.

Reference: http://www.w3.org/TR/css3-mediaqueries/#media1

MEDIA QUERIES (3)

- Media Query example:

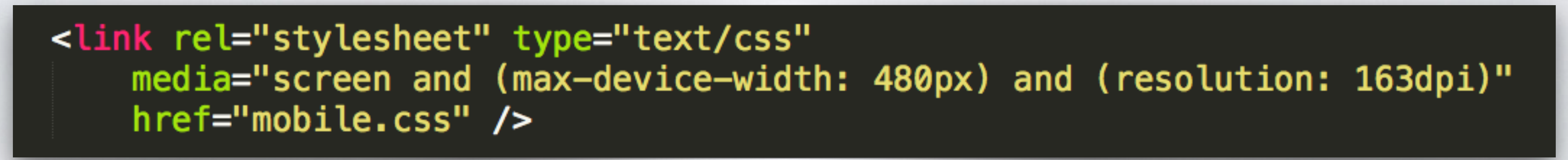


```
<link rel="stylesheet" type="text/css"
    media="screen and (max-device-width: 480px)"
    href="mobile.css" />
```

- Media query contains 2 items:
  - a media type (in this case "screen")
  - a query referring to a media feature, in this case max-device-width and the value 480px.

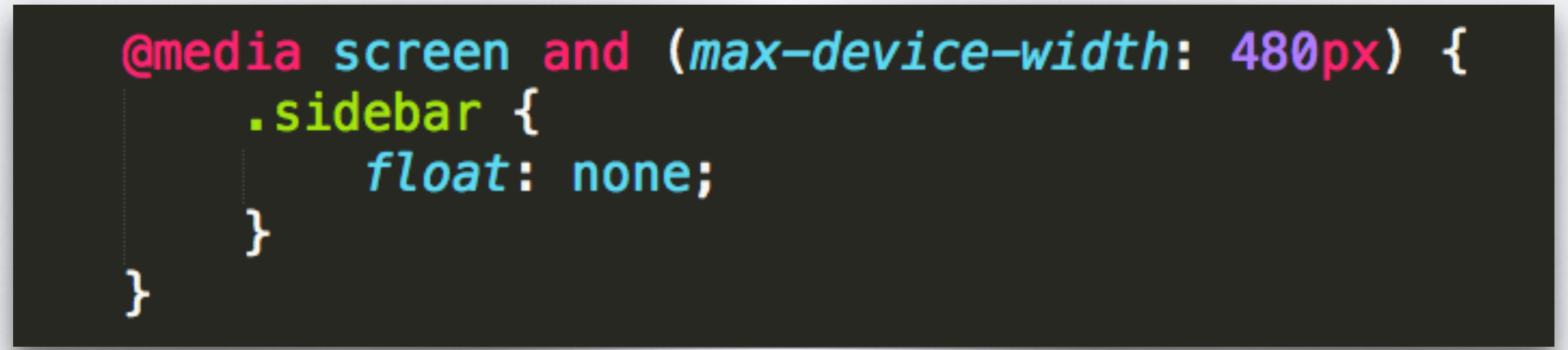
MEDIA QUERIES (4)

- Multiple device features can be chained in a single media query:



```html
<link rel="stylesheet" type="text/css"
    media="screen and (max-device-width: 480px) and (resolution: 163dpi)"
    href="mobile.css" />
```

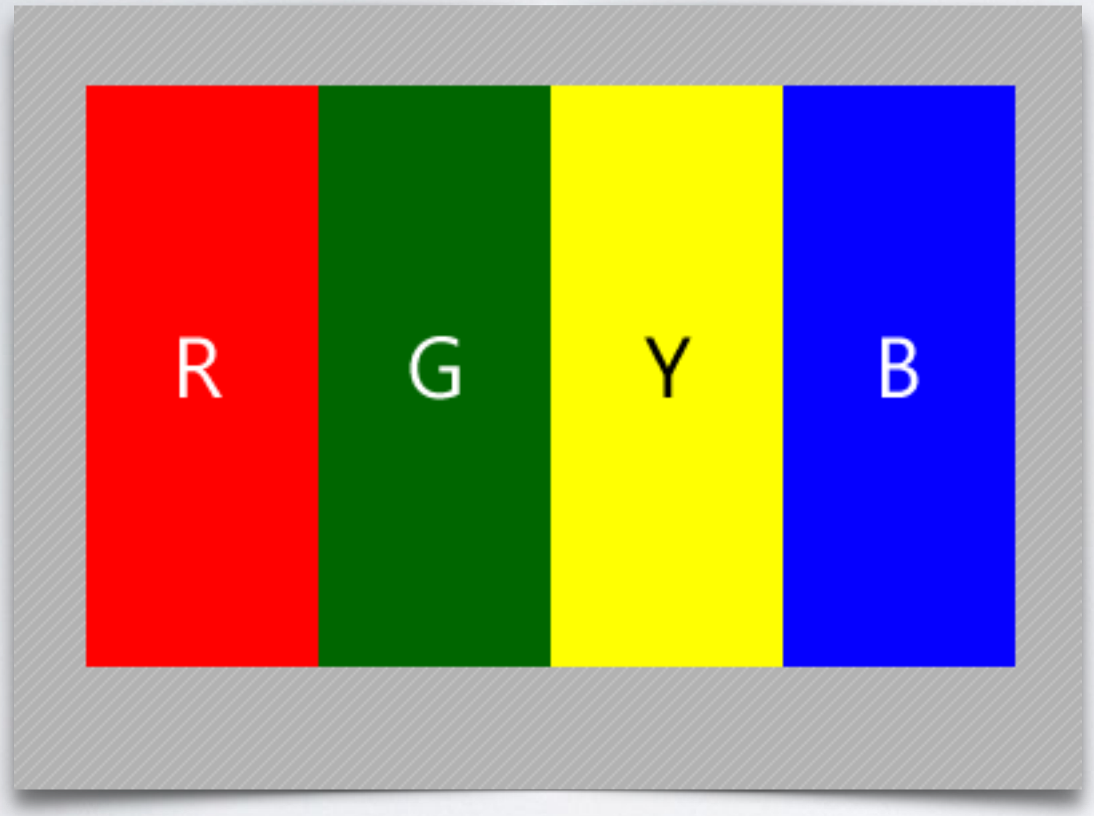
- Media queries can also be embedded right in the stylesheet:



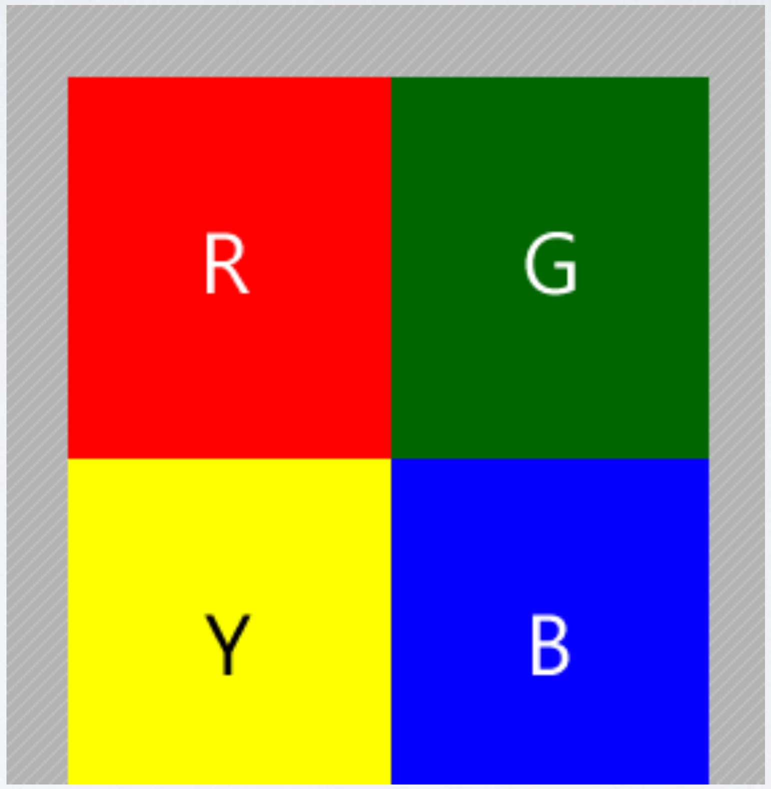
```css
@media screen and (max-device-width: 480px) {
    .sidebar {
        float: none;
    }
}
```

SIMPLE RESPONSIVE EXAMPLE

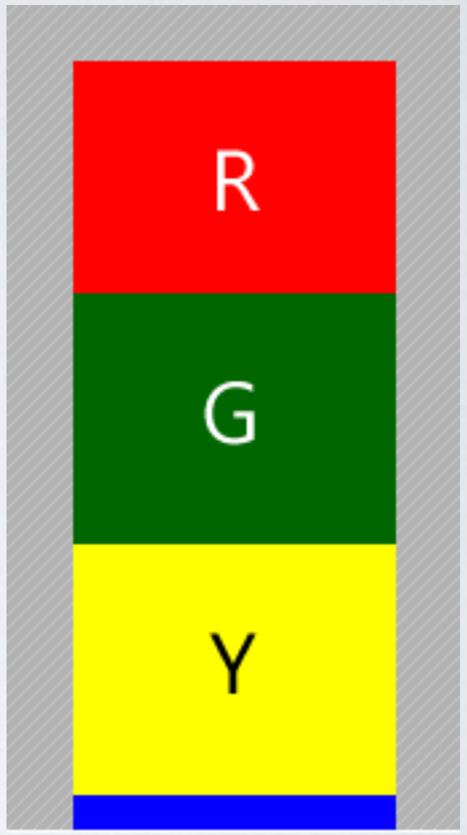
PC Layout



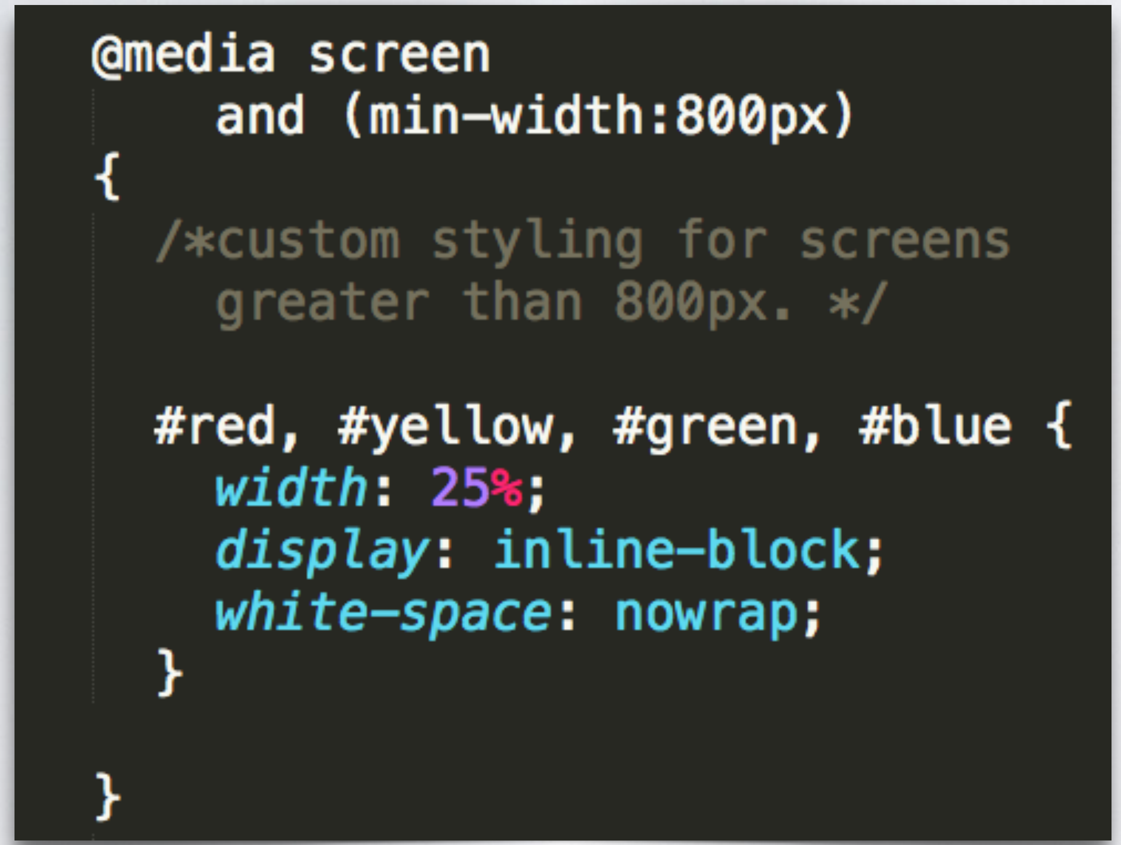
Tablet Layout



Phone Layout

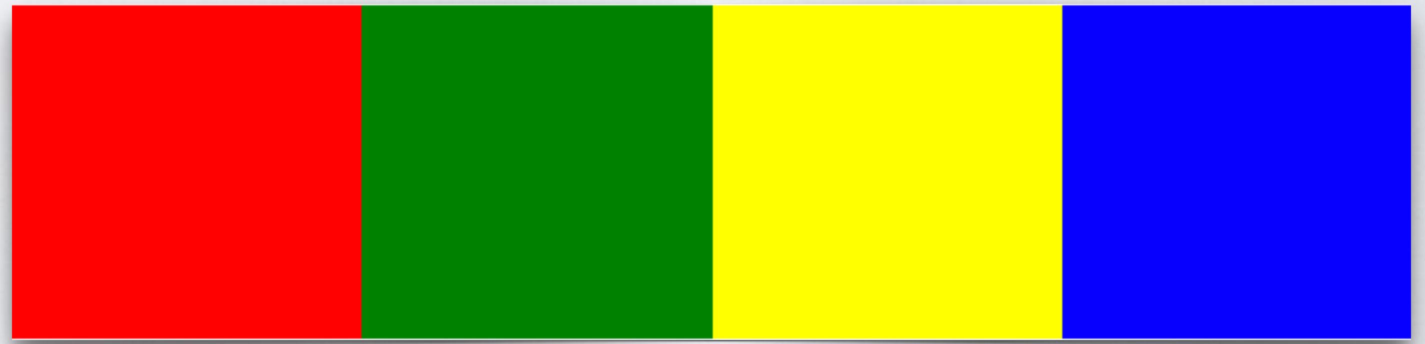


SIMPLE RESPONSIVE EXAMPLE (2)

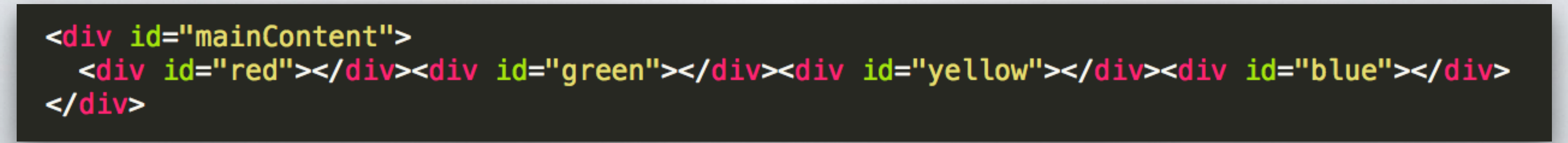


```
@media screen
    and (min-width:800px)
{
  /*custom styling for screens
    greater than 800px. */

  #red, #yellow, #green, #blue {
    width: 25%;
    display: inline-block;
    white-space: nowrap;
  }

}
```

If screen width is > 800px



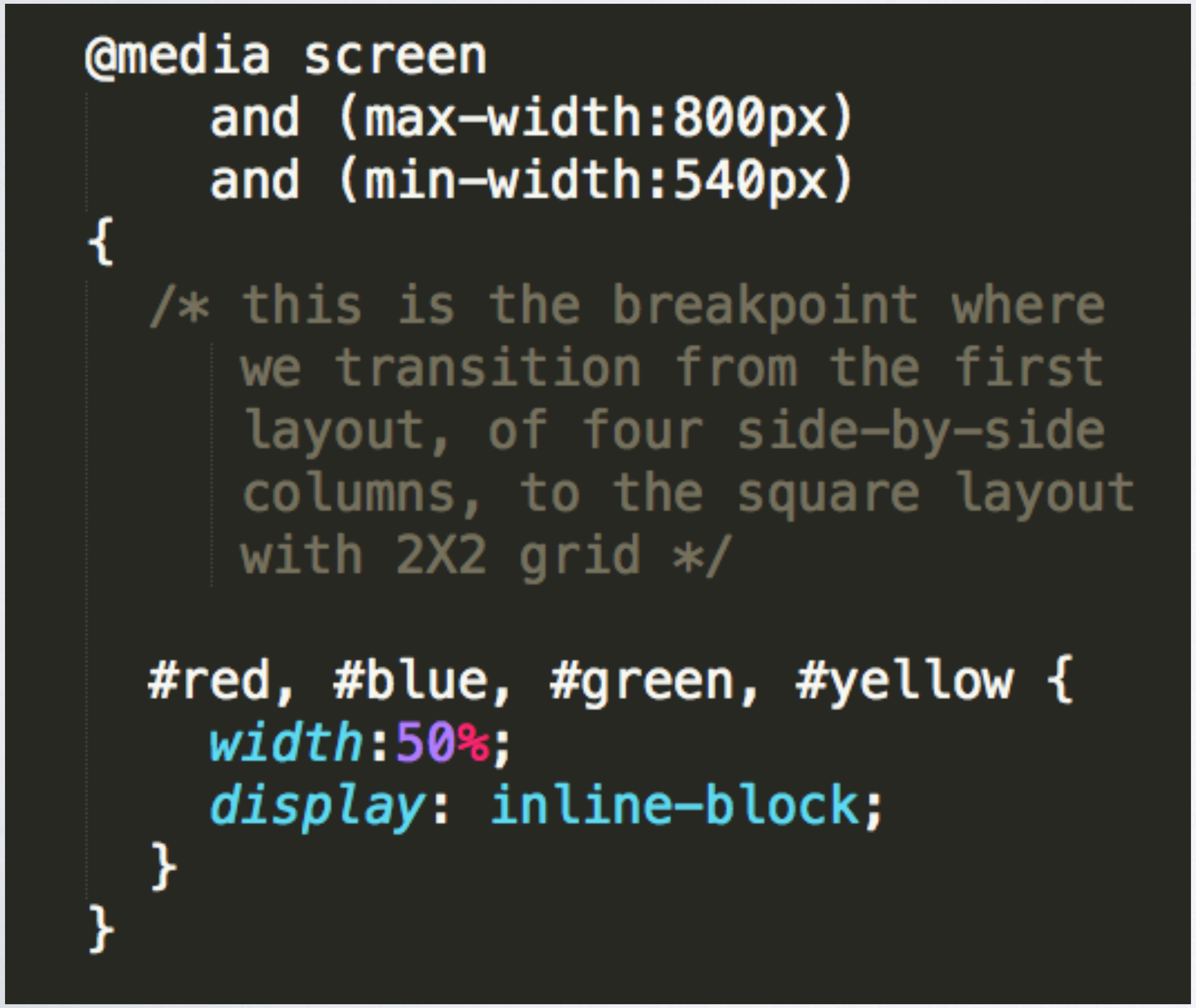
The content:



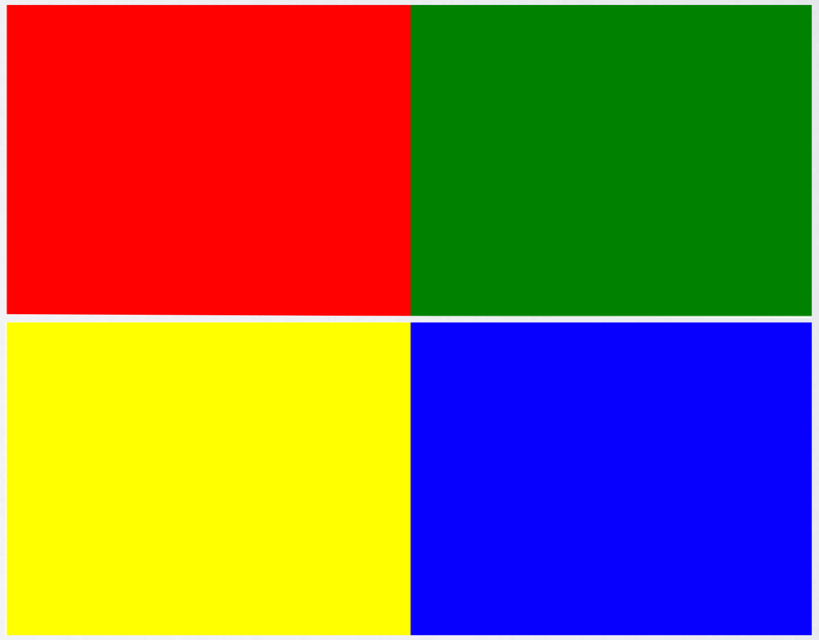
```
<div id="mainContent">
  <div id="red"></div><div id="green"></div><div id="yellow"></div><div id="blue"></div>
</div>
```

SIMPLE RESPONSIVE EXAMPLE (3)

- If 540px <= width <= 800px we get 4 X 4 layout:

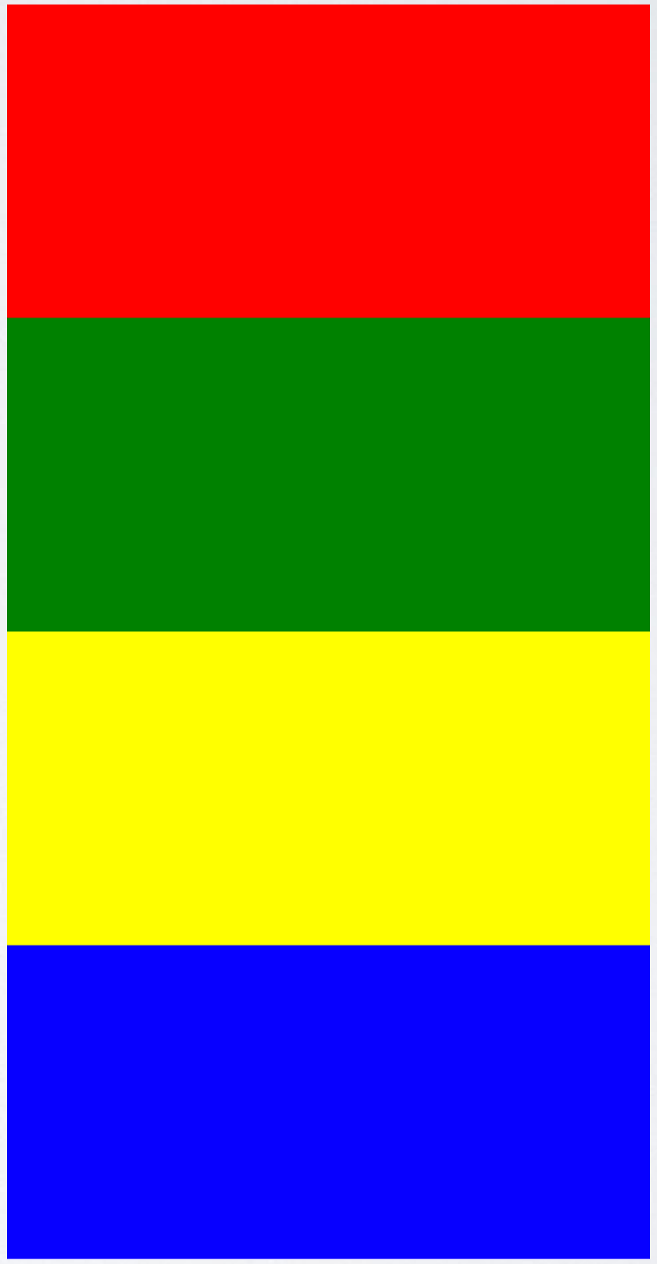


```
@media screen
    and (max-width:800px)
    and (min-width:540px)
{
  /* this is the breakpoint where
     we transition from the first
     layout, of four side-by-side
     columns, to the square layout
     with 2X2 grid */

  #red, #blue, #green, #yellow {
    width:50%;
    display: inline-block;
  }
}
```



MEDIA QUERIES (4)

- if width < 540px, neither of the media queries match, so blocks are laid out in the room available (vertically)



RESPONSIVE FRAMEWORKS

- The techniques discussed are widely implemented in popular CSS frameworks. Some examples:
  - Bootstrap: http://getbootstrap.com/
  - Pure: http://purecss.io/
  - Foundation: http://foundation.zurb.com/
  - HTML5 BoilerPlate: http://html5boilerplate.com/
  - More: https://hackr.io/blog/top-bootstrap-alternatives

CONCLUSIONS

- Huge progress has been made in efficiently creating highly styled and maintainable web apps.
- There’s no excuse for having a website that doesn’t look decent on every screen!

SASS

- Syntactically Awesome Style Sheets
- Supports two syntaxes SASS and SCSS
    - Same engine, different inputs
- Preprocessor for CSS
    - (i.e., 'compiles' down to CSS)
    - Provides helpful features
        - variables
        - nesting
        - importing / mixin / extend
    - The above features can be done with CSS, but it is just more verbose.
- SASS syntax
    - Older.
    - More terse
    - Indent/whitespace based
    - Similar to .haml
- SCSS syntax
    - Newer
    - Similar to CSS
    - Every CSS file is valid SCSS