Building Responsive Layouts

by Zoe Mickley Gillenwater
@zomigi

August 28, 2012
Responsive Web Design Summit
What I do

Books

Stunning CSS3: A Project-based Guide to the Latest in CSS
www.stunningcss3.com

Flexible Web Design: Creating Liquid and Elastic Layouts with CSS
www.flexiblewebbook.com

Web

Accessibility specialist at AT&T
Visual designer
CSS developer and consultant
why

responsive web design works
why

what

responsive web design means
why
what
how

to do responsive web design
fluid/liquid layout

uses percentage widths to adapt to size of viewport
**hybrid layout**

one+ column flexible, one+ not
responsive web design

fluid layout and flexible images adapted with media queries

www.alistapart.com/articles/responsive-web-design/
adaptive layouts

multiple fixed-width layouts swapped with media queries

www.netmagazine.com/tutorials/adaptive-layouts-media-queries
Look at this!
This is so tough!
I'm in such peril way up here!
Oh, wait...
How do we make this fluid?
Start with fluid wrapper
Add opposing floats inside

1. `#content-main { float: left; width: 70%; }`

2. `#content-secondary { float: right; width: 25%; }`
3-column layout: nest 2-column layout inside 2-column layout
Percentages are relative

are not the same
Determining nested widths

\[
\text{width of column you want to match} \div \text{width of parent column} = \text{width of nested column}
\]
Determining nested widths

\[
\text{width of column} \div \text{width of parent column} = \text{width of nested column}
\]

\[
\text{target} \div \text{context} = \text{result}
\]
Determining nested widths

\[
\frac{\text{width of column you want to match}}{\text{width of parent column}} = \text{width of nested column}
\]

\[
\frac{\text{target}}{\text{context}} = \text{result}
\]

\[
\frac{20}{80} = .25 \quad \text{which means} \quad 25\%
\]
That's more like it

25% of this

are the same

20% of this
What about fluid grids?

<table>
<thead>
<tr>
<th>12.5%</th>
<th>25%</th>
<th>37.5%</th>
<th>50%</th>
<th>62.5%</th>
<th>75%</th>
<th>87.5%</th>
<th>100%</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td><strong>62.5%</strong></td>
<td></td>
<td><strong>37.5%</strong></td>
<td></td>
</tr>
</tbody>
</table>
Width of this nested block?
Well that's not right…
To the laboratory!

\[
\text{width of column you want to match} \div \text{width of parent column} = \text{width of nested column}
\]

\[
\text{target} \div \text{context} = \text{result}
\]

\[
25 \div 62.5 = .4
\]

which means 40%
There we go

<table>
<thead>
<tr>
<th>12.5%</th>
<th>25%</th>
<th>37.5%</th>
<th>50%</th>
<th>62.5%</th>
<th>75%</th>
<th>87.5%</th>
<th>100%</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td><strong>62.5%</strong></td>
<td></td>
<td><strong>37.5%</strong></td>
<td></td>
</tr>
<tr>
<td><strong>40%</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(of 62.5%)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
widths
between and in fluid columns

spacing
Leave a gap via widths

100% - 70% - 25% = 5% (plus or minus a pixel)
Declaring fluid margin/padding

- Adjust widths so everything adds up to just under 100%
  - Avoids problems due to rounding % to px
- Nesting affects margin/padding values too
  - Use target/context formula to match outer spacing with inner spacing
Using box-sizing

• Makes fixed-width margin/padding on fluid layout **easy**
• Standard box model
  • `box-sizing: content-box`
  • Padding & border added on to width/height
• New box model
  • `box-sizing: border-box`
  • Padding & border subtracted from width/height
Fluid grid, fixed-width spacing

.column {
    float: left;
    -webkit-box-sizing: border-box;
    -moz-box-sizing: border-box;
    box-sizing: border-box;
    padding: 0 20px;
}
Use border as faux margin

.column {
    float: left;
    -webkit-box-sizing: border-box;
    -moz-box-sizing: border-box;
    box-sizing: border-box;
    padding: 0 20px;
    border-left: 10px solid rgba(0,0,0,0);
    -moz-background-clip: padding-box;
    -webkit-background-clip: padding-box;
    background-clip: padding-box;
}

.row {
    margin-left: -10px;
}

Fix box-sizing in IE 7 and 6

• Feed different dimensions based on content-box-model, or
• Use Christian Schaefer's box-sizing polyfill
  https://github.com/Schepp/box-sizing-polyfill
  .column {
    box-sizing: border-box;
    *behavior: url(boxsizing.htc);
  }

I recommend gridpak.com
fluid  hybrid

main column fluid, sidebar fixed
Hybrid layout options

• Easy: sidebars first in HTML
  • Float sidebars, main content moves up between floats
  • But usually not option in responsive design

• Tricky: main content first in HTML
  • Need to float it, but with what width?
  • One solution: negative margins
Fixed-width sidebar starting point
Add wrapper with padding

```css
#content-wrapper {
  padding-right: 290px;
}
```
Lay out main content div

#content-main {
    float: left;
    width: 100%;
}
Float sidebar

#content-secondary {
  float: right;
  width: 250px;
}

---

This page is a fictional site. It was created by Joe Middy. Content is in the book *Styling CSS2: A Project-Based Guide to the Future of CSS*. The use of CSS2 makes it hard to check the design in different browsers.
A positive right margin
A negative right margin
Add negative right margin matching wrapper's right padding

```css
#content-secondary {
    float: right;
    width: 250px;
    margin-right: -290px;
}
```
Success!

You deserve a sweet little treat every now and then.


Candies
TRUFFLES, BARS, CARAMELS, AND MORE

Pastries
CROISSANTS, DANISHES, CREAM PUFFS, AND MORE

Desserts
CAKES, PIES, TARTS, COOKIES, AND MORE
To make sidebar show in IE 6

```css
#content-wrapper {
    zoom: 1;
}
#content-main,
#content-secondary {
    display: inline;
}
```
3-column hybrid layout

• Nest one 2-column layout inside another
• Components of each layout:
  1. Wrapper with padding on one side
  2. Fixed-width sidebar
     • Same width as padding (or smaller)
     • Floated same direction as padding
     • Negative margin matching padding on same side
  3. Fluid column
     • Floated opposite direction
     • Width 100%
fluid layout  media queries

feed different styles based on viewport size
Choosing default styles

• Start "mobile," layer on wider styles?
• Start "desktop," layer on narrower styles?
• Start somewhere in between, layer on both?
• Learn full pros/cons:
  www.zomigi.com/blog/essential-considerations-crafting-quality-media-queries
Starting in the middle
Wide-screen media query

/*all the other styles up here*/

@media screen and (min-width: 1200px) {
  /*styles for larger screens in here*/
}

49
Add third column

@media screen and (min-width: 1200px) {
  #nav-main {
    position: fixed;
    top: 136px;
    width: 13%;
    margin: 0;
  }
  #content-main {
    width: 58%;
    margin-left: 18%;
  }
  #content-secondary {
    width: 20%;
  }
}
Style nav as vertical menu

@media screen and (min-width: 1200px) {
  ...
  #nav-main li {
    float: none;
    margin: 0;
  }
  #nav-main a {
    -moz-border-radius: 0;
    -webkit-border-radius: 0;
    border-radius: 0;
  }
}
You deserve a sweet little treat every now and then.


**Candies**
**TRUFFLES, BARS, CARAMELS, AND MORE**
Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Vestibulum tortor quam, feugiat vitae, ultricies eget, tempor sit amet, ante.

**Pastries**
**CROISSANTS, DANISHES, CREAM PUDDINGS, AND MORE**
Vestibulum tortor quam, feugiat vitae, ultricies eget, tempor sit amet, ante. Donec eu libero sit amet quam egestas semper. Aenean ultricies mi vitae est.

**Desserts**
**CAKES, PIES, TARTS, COOKIES, AND MORE**

**About this Page**
This page is for a fictional site. It was created by Zoe Mapple Gillenwater as an exercise in the book *Stunning CSS3: A Project-Based Guide to the Latest in CSS*. It uses CSS3 media queries to change the design at different screen sizes; resize your browser window or load it up in a mobile device to check it out.

**Credits**
**WEB FONTS:** Nadia Serif from *Kerney*
**ILLUSTRATIONS:** Yummy by *Jon Eden*
Small-screen media query

/*all the other styles up here*/

@media screen and (max-width: 760px) {
  /*styles for smaller screens in here*/
}

/*all the other styles up here*/
Things to fix

too few words per line, so make all one column

each too narrow, so stack instead and put pic on left
Narrow-screen design
Mobile media query

/*all the other styles up here*/

@media screen and (max-width: 550px) {
  /*styles for tiny screens in here*/
}


Non-overlapping version

@media screen and (min-width: 551px) and (max-width: 760px) {
    /*styles for small screens in here*/
}
@media screen and (max-width: 550px) {
    /*styles for tiny screens in here*/
}
Changing to single column

@media screen and (max-width: 550px) {
    #content-main, #content-secondary, #about, #credits {
        float: none;
        width: 100%;
    }
}
Changing feature images

```css
@media screen and (max-width: 550px) {
  .feature { padding-left: 70px; }
  #feature-candy {
    background-image: url(icon_candy_64.png);
  }
  #feature-pastry {
    background-image: url(icon_pastry_64.png);
  }
  #feature-dessert {
    background-image: url(icon_dessert_64.png);
  }
}
```
Mobile design
Viewport meta tag

Forces mobile devices to scale viewport to actual device width

<meta name="viewport" content="width=device-width"/>
Zoom problem in iOS

- The **device-width** on iOS devices always matches **portrait width**
- This means design **doesn't reflow** when you switch to landscape, but instead just **zooms**
Fixing (and adding) zoom issues

• Option 1: add `maximum-scale=1`
  • But disables user scaling
    ```html
    <meta name="viewport" content="width=device-width, maximum-scale=1">
    ```

• Option 2: add `initial-scale=1`
  • Allows user scaling
  • But triggers over-zoom/crop bug when changing from portrait to landscape
    ```html
    <meta name="viewport" content="width=device-width, initial-scale=1">
    ```
The best way to fix zoom issues

- Option 3: add `initial-scale=1` plus script to fix over-zoom bug
  - See http://filamentgroup.com/lab/a_fix_for_the_ios_orientationchange_zoom_bug/

```html
<head>
  ...
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <script src="ios-orientationchange-fix.js">
  ...
  </script>
</head>
```
conditional comments or JavaScript to deal with IE 8 and earlier
Conditional comments

• Split styles into separate sheets and feed applicable sheet to IE based on whether it's IE on desktop or mobile
• Approach varies based on which set of styles are your default
Conditional comment when desktop styles are default

Feed IE Mobile 7 media query sheet:

<link rel="stylesheet" href="global.css" media="all">

<link rel="stylesheet" href="mobile.css" media="all and (max-width: 700px)">

<!--[if IEMobile 7]>
<link rel="stylesheet" href="mobile.css" media="all">
<![endif]-->

Conditional comment when mobile styles are default

Feed older IE media query sheet, hide from IE Mobile 7:

```html
<link rel="stylesheet" href="global.css" media="all">

<link rel="stylesheet" href="desktop.css" media="all" and (min-width: 700px)">

<!--[if (lt IE 9)&(!IEMobile 7)]>
<link rel="stylesheet" href="desktop.css" media="all">
<![endif]-->
```

Source: http://adactio.com/journal/4494/
Pre-fab JavaScript for non-supporting browsers

• Simply add one of these scripts:
  • Respond: https://github.com/scottjehl/Respond
  • css3-mediaqueries.js: http://code.google.com/p/css3-mediaqueries-js/

• Avoid extra HTTP request for non-old-IE browsers using conditional comments:

  <!--[if (lt IE 9)&(!IEMobile 7)]>
  <script src="respond.min.js"></script>
  <![endif]-->
View it live

http://stunningcss3.com/code/bakery/
Learn more

Download slides and get links at http://zomigi.com/blog/responsive-layouts-presentation

Zoe Mickley Gillenwater
@zomigi
design@zomigi.com
zomigi.com | stunningcss3.com | flexiblewebbook.com

Title photo by Steve James (http://www.flickr.com/photos/steeljam/3350481764/)
Rock climbing photo by Justin Johnson (http://www.flickr.com/photos/justinjohnsen/4512815628/)