CSS3: Using media queries to improve the web site experience

November 19, 2011
indieconf
Zoe Mickley Gillenwater | @zomigi
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 on AT&T's design standards team
Visual designer
CSS developer and consultant
My 7 web-enabled devices

- 3 laptops (plus 2 external monitors)
- Android smartphone
- iPod Touch
- Google TV
- Wii
how can our sites accommodate all this DIVERSITY?
Introducing media queries

- Awesome new part of CSS3
- Simple way to feed different CSS based on characteristics of user's device
- Not:
  - for feeding styles based on browser
  - just for feeding styles based on viewport size
Media query syntax: internal

body {
    background: gray;
}

@media screen and (max-width:500px) {
    body {
        background: blue;
    }
}

**English translation:**

*Make the background gray. But up to a maximum width of 500 pixels, make the background blue.*
Media query syntax: internal

body {
    background: blue;
}

@media screen and (min-width:501px) {
    body {
        background: gray;
    }
}

**English translation:**

*Make the background blue. But at and past the minimum width of 501 pixels, make the background gray.*
How it looks

The background will be gray when the viewport is under 501 pixels.

The background will be gray when the viewport is wider than 500 pixels. It will be blue under 501 pixels.
Media query syntax: external

Extend the existing media part of the link element or @import rule:

```html
<link href="narrow.css" rel="stylesheet" media="only screen and (max-width:500px)">

@import url(narrow.css) only screen and (max-width:500px);
```
Internal media queries

Pros:
• No extra HTTP request(s)
• Not out of sight and forgotten

Cons:
• Extra kb in file size for everyone to download
• Have to use JavaScript to make it work with old IE
External media queries

Pros:
• Smaller file size for non-supporting browsers
• Easier to keep organized if CSS extensive
• Can feed to old IE using conditional comments

Cons:
• Extra HTTP requests
• Out of sight so could be forgotten when updating
you now know media query syntax

YAY!

!
but that's not where the CHALLENGES are
Designing for different widths

- Create wireframe comps for placement of major layout areas
- Design details in browser if possible
- Focus on content
- Harder to retrofit existing design
Structuring HTML for reorder

• Build HTML so layout pieces can be rearranged visually as wanted
• This is why retrofitting so difficult
Organizing media queries for maximum efficiency and reach

• Internal or external?
• Overlap or stack?
• Which media features to test against?
• Where to make breaking points?
• Which dimension as starting point?
• Workarounds for non-supporting browsers?
Starting/default width

or

(or something in between)
Starting with desktop styles

Pros:
• No extra work to make majority width appear correctly on IE 6-8
• Easiest way to retrofit existing site

Cons:
• Mobile devices may have to download unneeded desktop assets
• Requires separate style sheets or JavaScript to make mobile design appear in IE Mobile and older mobile browsers
Starting with mobile styles

Pros:

• Prevents mobile devices from downloading unneeded desktop assets
• Older, non-media-query-supporting mobile browsers still get the mobile styles without any extra work

Cons:

• Desktop devices may have to download unneeded mobile assets
• Requires separate style sheets or JavaScript to make majority desktop design appear in IE 6-8
Our starting point
Very narrow: awkward
Wide-screen media query

/*all the other styles up here*/

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

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;
  }

}
Wide-screen design

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 PUFFS, 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 Miedzy Gilewscy as an exercise in the book *Stunning CSS3: A Project-based Guide to the Lotos 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 Sans from *Kernest*
ILLUSTRATIONS:
Yummy by *Jem Eden*
Small-screen media query

/*all the other styles up here*/

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

Remove columns from text

@media screen and (max-width: 760px) {
  h1 + p {
    -moz-column-count: 1;
    -o-column-count: 1;
    -webkit-column-count: 1;
    column-count: 1;
  }
}


Stack feature boxes

@media screen and (max-width: 760px) {
  ...
  .feature {
    float: none;
    width: auto;
    margin: 0 0 1.6em 0;
    padding: 0 0 0 140px;
    background-position: top left;
  }
}
You deserve a sweet little treat every now and then.

Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas.

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 PUFFS, 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
Donec eu libero sit amet quam egestas semper. Aenean ultricies mi vitae est.
pause for CAVEATS & CLARIFICATIONS
Some sites would be better served with a separate site for mobile devices instead of using media queries.
Even if a separate mobile site would be best, using media queries is a **good first step** if a separate site isn't currently feasible.
“The choice is not between using media queries and creating a dedicated mobile site; the choice is between using media queries and doing nothing at all.”

—Jeremy Keith

http://adactio.com/journal/1696/
If you do use them, they're not the only tool you can use—you can **add scripting as well** to further customize the content, functionality, etc.
Media queries are only meant to solve the problem of mobile's small viewports, not all the other things that can make mobile browsing different (such as context, bandwidth, etc.).
“It's making sure your layout doesn't look crap on diff. sized screens.”

–Mark Boulton

http://twitter.com/#!/markboulton/status/50237480368214016
back to CSS
Mobile media query

/*all the other styles up here*/

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

Non-overlapping mobile media queries

@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*/
}
Media features for mobile

min-width
max-width
device-width
min-device-width
max-device-width
orientation
min-device-pixel-ratio
  -webkit-min-device-pixel-ratio
min--moz-device-pixel-ratio
-o-min-device-pixel-ratio
Changing to single column

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

@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, maximum-scale=1.0">
View it live

http://stunningcss3.com/code/bakery/
Dealing with IE 6, 7, and 8

- Conditional comments
- JavaScript
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 mobile IE media query sheet:

```html
<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 mobile IE:

```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-IE browsers using conditional comments:
  <!--[if (lt IE 9)&(!IEMobile 7)]>
  <script src="respond.min.js"></script>
  <![endif]-->
WHAT ELSE

can media queries do?
Swapping images on high-res displays

@media
screen and (moz--min-device-pixel-ratio : 1.5),
screen and (-o-min-device-pixel-ratio : 3/2),
screen and (-webkit-min-device-pixel-ratio : 1.5),
screen and (min-device-pixel-ratio : 1.5) {
}

Swapping images on high-res displays

@media ... screen and (min-device-pixel-ratio : 1.5) {
    .feature {
        -moz-background-size: 64px 64px;
        -webkit-background-size: 64px 64px;
        background-size: 64px 64px;
    }
    #feature-candy {
        background-image: url(icon_candy_128.png); }
    #feature-pastry {
        background-image: url(icon_pastry_128.png); }
    #feature-dessert {
        background-image: url(icon_dessert_128.png); }
}
Examples of sites using media queries

• Gallery: http://mediaqueri.es/
• My own bookmarks:
  – https://gimmebar.com/loves/zomigi/tag/mediaqueries
  – www.delicious.com/pixelsurge/nicedesign+mediaqueries
  – www.delicious.com/pixelsurge/nicedesign+liquid
Learn more

Download slides and get links at
http://zomigi.com/blog/media-queries-presentation
Questions?

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