Brittany’s Guide to Mobile Web Design

Intro:

Mobile browsing is expected to overtake desktop browsing by 2014 (according to AppSumo.com).

The first thing to understand about Mobile Web Design is that mobile devices do not just provide a different form factor for web usage. They also represent an entirely different “usage scenario,” and thus demand an entirely different browsing experience! Mobile web users will access your site in unpredictable lighting, distracting environments, and often want different information than when browsing on a desktop.

Your user interface needs to be designed with both of these major differences in mind.

Physical Usage Differences

Mobile browsing happens on devices of totally different shapes, sizes, and input methods. Cell phones, PDAs, and tablets range from a viewport width of 170 pixels (an old Nokia N70) to 946 pixels (an iPad viewed in the landscape orientation). Mobile users do not have your standard full-size keyboard/mouse combo. Their input methods are generally less accurate and slower – stylus, mini keyboard, T9, arrowpad, touchscreen, and multi-touchscreen. The Lynda.com recommendation is to “Design for the Finger.” Fingertips vary from 40pixels wide to 80pixels wide, and have no mouseover ability. Inputting information can be a frustrating experience for the mobile user, who often has only one hand free.

Situational Usage Differences

Desktop users generally have a reliable internet connection with few limits on the amount of data they can download per month. Mobile users have to watch their Megabytes closely or face overage charges. Desktop users usually sit at a desk with a chair and good lighting. Mobile users do not have this reliable environment on their side, so we as designers don’t either. Mobile users are out and about, maybe squinting to read their screen under the July sun, or looking up imdb.com facts in a dark movie theater. They may be trying to be discreet so as not to appear rude, or they might even be riding a bike while browsing! A mobile user often wants a quick answer to a question, instead of reading pages and pages of content. What movie do I recognize that actor from? What’s the phone number for that bookstore? Where’s the nearest Thai place? However, sometimes mobile users are waiting for their friends to get to the bar and want to read a lengthy article sitting in a booth. Mobile users are quite unpredictable!

What’s A Designer To Do?

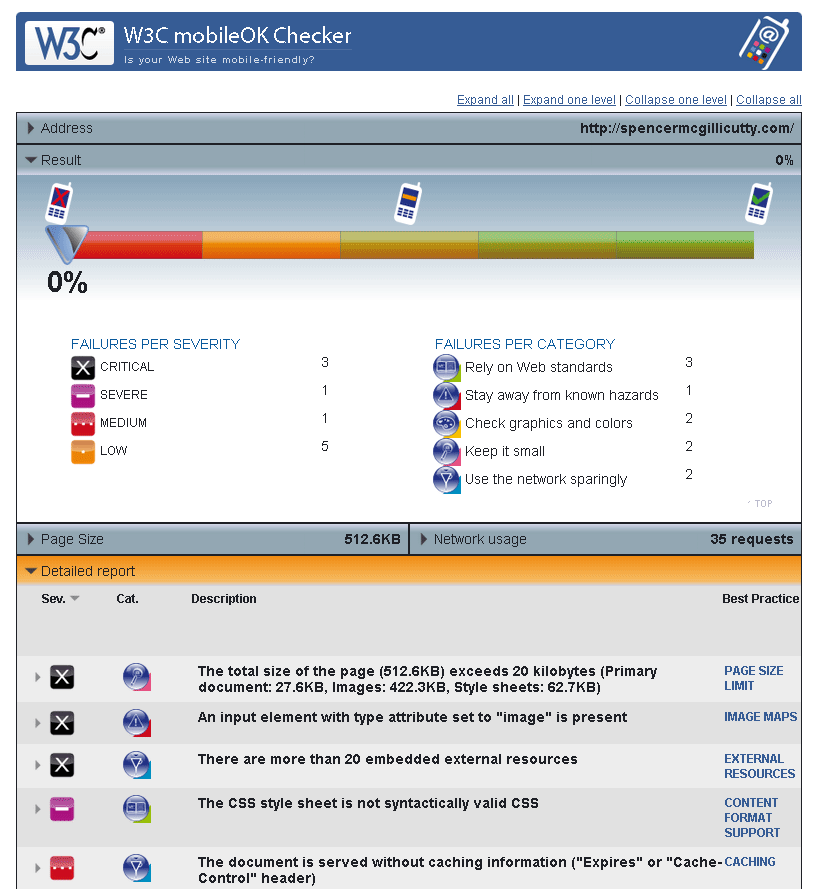
You have three options.

Option #1. Do Nothing. Hope it works okay for mobile.

This is actually a valid option for some sites. You can run your website through the W3C’s “Mobile Friendly” validator here and see how close your site is to working as-is:

<http://validator.w3.org/mobile/>

It will give you a percentile score of how mobile-friendly your website is, and give you notes on how to fix it up. Below is a picture of a total failure (a Wordpress site designed with no thought given to mobile devices).



The “Do Nothing” option could be good for lightweight (small file-size) pages, with flexible, flowing content and traditional vertical columns. It’s not a good idea for sites with complex layouts, or sites that rely on large graphic elements, framesets, or plug-ins like Flash (Lynda.com).

Option #2 - Multi-serve. Separate styling and scripting for each form.

This is the approach I took in my demo. Maintain one HTML file, but link to several different CSS stylesheets, dependent on viewport width or device type. One disadvantage is that it can be difficult to make one HTML file successfully serve two vastly different stylesheets. And, if you use too many varying CSS instructions you risk bloating the filesize to the point of being not worth the trouble.

The “Multi-serve” option is good for pages that have mostly semantic HMTL. It’s not good for complex pages with embedded images or video, or sites that should have vast differences between their mobile and desktop versions. (Lynda.com)

Option #3 - Mobile-specific. Totally split the content into two separate websites.

* You would set up a subdomain, like mobile.mywebsite.com and write in some server-side detection. When the server sees the main webpage (mywebsite.com) is being accessed by a mobile device, it redirects it to the subdomain. This server-side detection can be unreliable, which is a downside to this approach. And you have to maintain two separate batches of content. But, your users get perfectly optimized content no matter how they access your site – and that can be worth a lot. According to AppSumo, 40% of web users report that they’ve sought out a competitor after a frustrating experience with their first-choice’s website. SO, having an optimized web experience is not a bad idea at all. A word of caution: If your site automatically detects whether a visitor is coming from a mobile browser, make sure it’s set up to send that visitor to the link they were trying to visit, otherwise they’re likely to leave and never come back.

The fundamentals of Mobile Web Design

Here are some rules of thumb I picked up.

In general: Users tend to expect one vertical-scrolling column. If you ask them to scroll horizontally, you may confuse them. Make info easy to find and consume, and make searching VERY easy. Use a subtle, readable color scheme (that would be readable in various forms of lighting). Keep graphics to a minimum – and a minimum file size! If you are using Option #2, be sure to upload multiple versions of the same graphic, optimized for desktop and the smaller resolution of mobile. Keep your font usage to just a few fonts, and just a few sizes.

Fingertips are 40-80 pixels diameter

Don't use framesets

Image maps usually rely on hover states which won't work.

Place navigation and search boxes near the top the page

use fewer color and maintain readability

Use consistent font faces and point-size

Avoid complex background images

Resample your images to be smaller filesizes

If you have numerous small images, combine them into one gif and display each one inside a div. like sprites

use whitespace

give user the option to switch between lighter and darker versions for different lighting conditions

display links to suggested pieces of data the user is likely to request based on their location

* lack of JavaScript and cookies (except smartphones)
* no secure connection, flash, pdfs (been changing though)
* Don't use pop-ups or new windows

Strive to keep your design flexible. Flexible width, flexible images, flexible content. Give your user the choice to view your site in the desktop version or the mobile version no matter where they are or what device they use.

To respect the restrictions of mobile input options, you should make your clickable areas large and spaced out. Make navigation easy and clear. Reduce the amount of text a user needs to fill out, and allow the page to “remember” what they input for next time. You should also provide an option to show password characters, because errors are so much more common on mobile devices.

Mobile users want somewhat different content – more streamlined, more quickly accessible, and with fewer pictures. Many mobile users are concerned about data usage – and cell phones have significantly less processing power, storage capacity, and often slower connection speeds than desktop computers. But, Mobile devices do have some cool features that desktop computers don’t! They can actually provide a lot of sweet information to your website. Here are some examples:

* Clock
* ambient light
* compass
* camera
* thermometer
* geolocation
* accelerometer
* microphone
* messaging
* calendar

With all this info available, mobile users expect that a webpage is going to know something about them – like their location, or their local date and time. They want phone numbers on webpages to be setup for instant dialing.

You’re killin’ me, Smalls!

There are quite a few browsing engines out there, and you can’t very easily design for all of them! It is good to know a few of them so you can pick out what you’re aiming for. If your users are primarily younger, technogeek types, make sure you’re designing for the Android and iOS platforms. If your users will be primarily older, more business-minded folks, be sure to design for Windows phones and Blackberries. Lynda.com recommends keeping to the concept of graceful degradation – some older phones may only be able to view plain text!

Current mobile markup language is XHTML-MP, but it is moving toward HTML5.

**BONUS ROUND**

**Responsive Design:**

you can choose media queries, but  you face the hassle of new queries poppping up which jarrs the user experience.   
Data tables should be changed to pie graphs or mini-graphs.

Use context-aware, scalable images???

Fluid layouts (no more fixed widths)

minimal but with strong color

fluid grids

sliding composite images

foreground images that scale with the ayout

hiding and revealing portions of images

The website should accommodate for: Resolution, Image Size, Scripting Abilities

* To maintain fast loading times, use images of a manageable size. Using the width and height attributes to scale them for more text content on smaller devices isn’t a good usability practice.
* An alternative to scaling is **cropping**. The CSS overflow property (e.g. overflow: hidden) gives us the ability to crop images dynamically as the containers around them shift to fit new display environments.
* The option is available to have multiple versions available of the same image and then serve up the appropriate sized version depending on the user.
* Finally, you have the option to **hide images altogether**. Media queries that serve up a stylesheet which sets the display: none property for images takes care of this function.

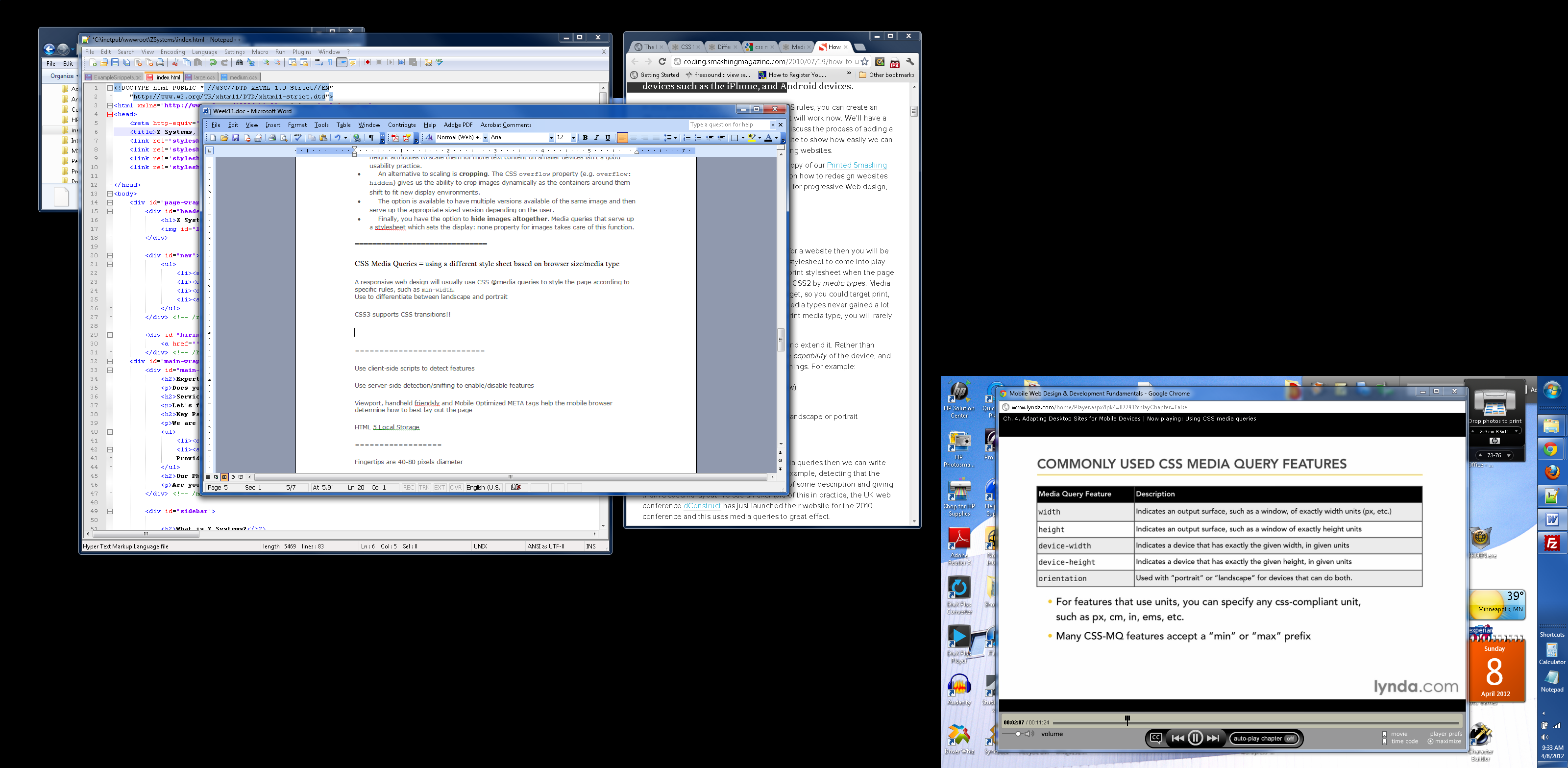
Ideally, you would use the principles of responsive design to make your desktop website automatically crop or resize pics

**BONUS ROUND:**

**CSS MEDIA QUERIES**

A responsive web design will usually use CSS @media queries to style the page according to specific rules, such as min-width.

Use to differentiate between landscape and portrait



On a mobile device, width = device-width, generally.

Define a CSS Media Query directly within the <link> tag.

<link rel=”stylesheet” media=”screen and (max-width:800px)” href=”example.css” />

Or use it in a CSS Style sheet.

@media screen and (min-width: 801px){…};

===========================

Avoid User Agent server-side detection (browser-sniffing) = not future proof

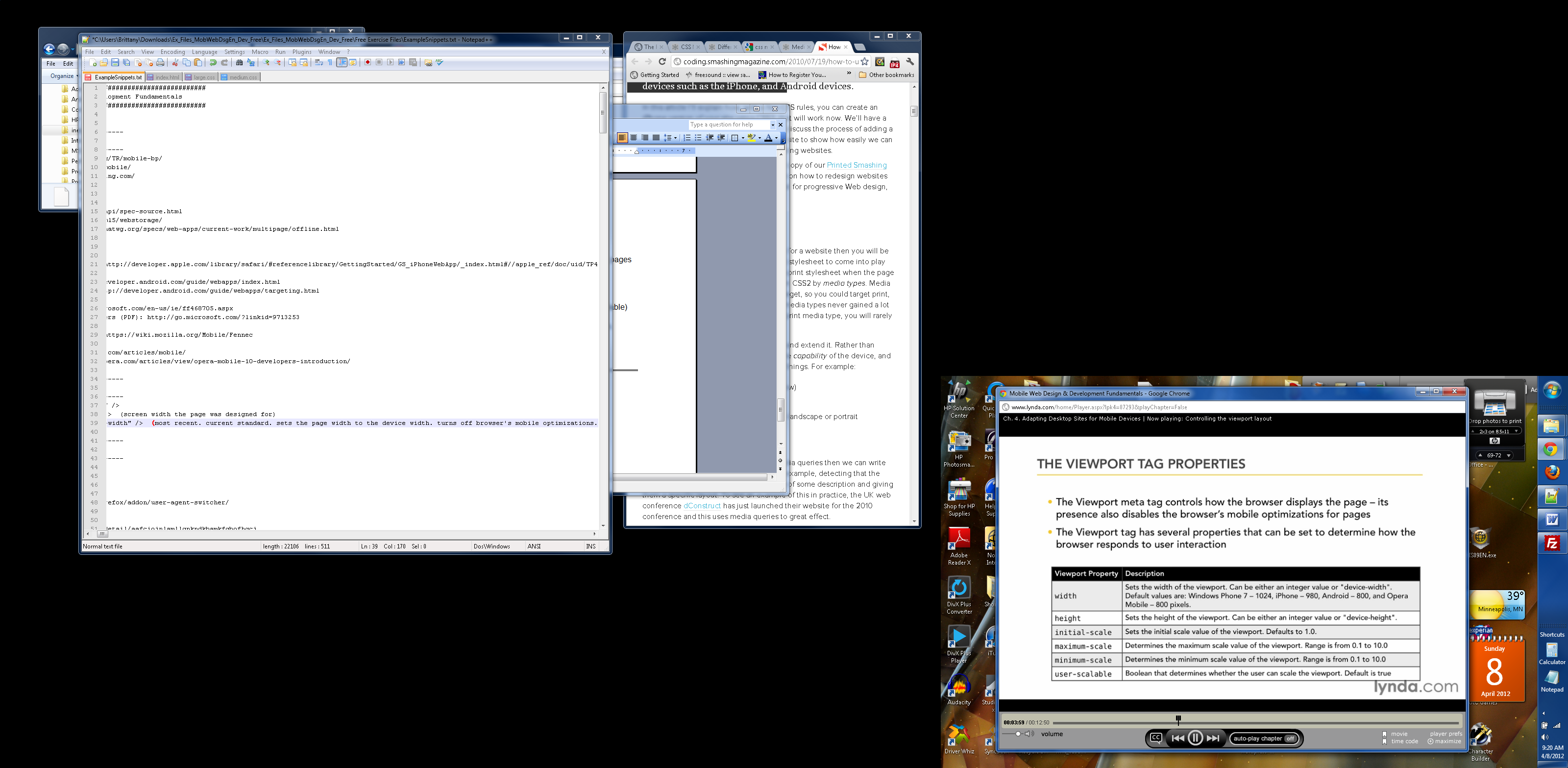
Use user agent switchers to use desktop browsers like a mobile phone.

Use client-side scripts to detect features

Use server-side detection/sniffing to enable/disable features

Viewport, handheld friendsly and Mobile Optimized META tags help the mobile browser determine how to best lay out the page

HTML 5 Local Storage



A NOTE ON FORMS:

FORMS:

Minimize number of screens the user has to load

Try to use native Web form controls instead of custom ones.

Constrain your forms to the screen width and input capabilities

Give the user the option to display password chars

Use top-aligned form labels instead of left or right aligned

Use HTML 5 form input types

free testing tool  tests the mobile-readiness of the site using industry best practices and standards.   MobiReady Report (see [mobiForge](http://en.wikipedia.org/wiki/Mobiforge))

==========================

==========================

Responsive web design...

==============================

CSS Media Queries = using a different style sheet based on browser size/media type

CSS3 supports CSS transitions!!

==================

=========================

**PLAN:**

**Install and setup web-server**

**Install emulators.**

<http://create.msdn.com/en-US> (Click: Download the free tools)

<http://create.msdn.com/en-us/home/getting_started>

Install. Find on your start menu under Windows Phone

http://localhost/

<http://developer.android.com/sdk/index.html>

http://10.0.2.2

<http://www.opera.com/developer/tools/mobile/>

**Resign a website for mobile using best practices… reorganize content for Z Systems page – new homepage? Buttons made of only the word? Resample images**

**New CSS sheet for detecting mobile devices**

**How do I demonstrate it?** <http://www.responsinator.com/?url=b.homes.mcad.edu%2F~bmiller%2FWeek11>

To address the form factor differences, I linked three CSS stylesheets to one HTML file -- one optimized for desktop, one for tablet, one for cellphone. The content is different for the different stylesheets

<http://b.homes.mcad.edu/~bmiller/Week11/>

There is a lot more out there to learn about - such as using CSS3 to crop photos so they fit better on a screen version, or using flexible grids that easily change layouts...or using PHP to check the time and report to your users whether or not the business is currently open.

==================

Webkit --> Safari, Chrome, Blackberry, Palm

Presto --> Opera

Gecko --> Firefox

Trident --> Internet Explorer

Opera = leading mobile engine overall

Webkit = leading mobile engine in North America

iOS = leading mobile engine in Europe

Business users use Windows Mobile and Blackberry

Young/Tech users use iPhone and Android

==========================

Potential Targeting Strategies:

Deliver a lowest common denominator experience.

Choose one or two platforms to optimize for

Don't say ' this site best viewed in Browser X'

Use a tiered approach based on device richness

Mobile devices can detect browser features instead of just browser type

Browsers can be classified into tiers based on their capabilities

Rich/Light/Text-only experience

Approaches:

1. do nothing. Hope it works okay for mobile.
   1. Ok for lightweight pages with flexible, flowing content, traditional vertical columns
   2. Bad for complex layouts, sites relying on plugins, sites with large graphic elements.
2. Multi-serve. Separate styling and script appropriate for the form factor
   1. Desktop.css and mobile.css
   2. Disadvantage: can be difficult to convert existing content.
   3. Careless use of styles can defeat the benefits
   4. Good for pages that are mostly semantic markup and use stylesheets/scripts to define appearance.
   5. Not good for: complex pages with embedded images/video, pages whose content differs greatly between form factors
3. Mobile-specific. Totally split the content.
   1. Involves server-side detection and redirection (can be unreliable)
   2. Good for complex pages. But you have to maintain two sites.

Separate DocType??

Separate URL???

WAP = Wireless Application Protocol, WML = Wireless Markup Language are being phased out

XHTML-MP = preferred markup language with mandated backwards compatibility for WML (as of 2005...)

Use XHTML 1.0, XHTML-MP, HTML 4, CSS 2.1, JavaScript 1.4

Going forward, use HTML5!

**NOTES**

* focused on discrete individual tasks
* Answer these quesitons:
  + Who is using this site? typical profile, age, industry?
  + What are your visitors trying to do at the moment? See if their flight is on schedule and check in. (Not plan a vacation)
  + Where is your site being accessed from? (A bank's site may point out nearby ATMs)
  + When is your site being accessed? (don't show movie showtimes that have already happened. change what phone number is provided depending on the time of day)
  + Why are they coming to your site? Quick answer to a question, or long in-depth article. Provide flexibility for different circumstances.
  + How are they accessing the site? streaming vs text? local storage?
* search and popular topics at the top and bottom
* ability to switch from landscape to portait
* slower than dialup