RWD and CSS

Responsive Web Design

Some material
Yik-Kuang Lu & Hieu Nguyen

History of CSS

Like markup languages and HTML, style sheets were not new due to the Web

- Publishing has a long history…

"Cascading" Style Sheets addresses uniqueness of web:

- No specs or standards, just a newgroups and some folks
- HTML was evolving and included some tags and attrs
  - Remember seeing that " " tag last week?
- In '94 Hakon Wium Lee defined CSS, & by the end'96 CSS1 (w/ B.Bos)
  - But the pollution of HTML3 remained evident for a while
- The *cascading* part came from a recognition that style information was shared between author and user
  - A resolution algorithm was part of CSS1
  - http://www.w3.org/TR/CSS1/

1. Browser default
2. External CSS
3. Internal CSS
4. Inline style

See http://www.w3.org/Style/LieBos2e/history/Overview.html

Fast-forward: CSS3 Now

- Evolution: CSS2 addressed layouts to some extent
  - Folks were using frames and tables like crazy to make grid layouts
  - CSS2 came out in 1998, CSS2.1 evolved from 2004-11
- CSS3 – "Modules"
  - The 2.1 experience demonstrated one spec got too unwieldy, so CSS3 was factored into modules to support its evolution.
  - There are a lot of these, but some of the more notable ones:
    - *Selectors*: lets you to locate HTML elements for styling purposes
    - *Properties*: the attributes that affect presentation of an HTML element
    - *Media Queries*: rules based on attributes of the rendering device/environment
    - *Namespaces*: typical use of these – avoiding conflicts with same element name
    - *Box Model*: Layout management for "box" flow content; get rid of tables!

And a whole lot more – if you review the evolving standards you notice a parallel type of approach with publishing issues, as that is what this really is. But that is OK as it is the intended use of CSS; it is when you cloud its use with HTML5 attributes or Javascript functionality where you can get in trouble

CSS in 1 Slide

Selector rules:

`tag { decl-1; decl-2; …}` where `decl-n` is `property:value`

```
#id SAME      /* select elements with id attr = id */
tag.class SAME /* select [tag] elements of a class */
```

- Selectors may be grouped for economy of representation

Specifying CSS:

```
<link rel="stylesheet" type="text/css" href="s.css">
<style>…</style>  /* specified in the HTML file */
<tag style="p:v;…">…  /* inline style via attribute */
```

- Again, like CSS1, there is a priority to the rules when they clash
  - Brower default
  - External CSS
  - Internal CSS
  - Inline styling (this is highest)

RWD: The Problem

- Rise of mobile devices – new screen sizes
- Website designed for a desktop computer become unusable for a mobile device
- Have separate sites for mobile and desktop Web users

Concept

- Ethan Marcotte coined the term *responsive web design* (RWD) in a May 2010 article in *A List Apart*.
  - "Is the approach that suggests design and development should respond to the user's behavior and environment based on screen size, platform and orientation"
- Provide optimal viewing experience – easy reading and navigation with minimum of resizing, panning, and scrolling across a wide range of devices.
  - Automatically scale and adjust content
  - Responding upon user's needs and devices capabilities
- Has one URL, one HTML code – content is shown according to the defined CSS3 media queries on multiple devices.

RWD

- A flexible, grid-based layout
  - Flexible grid consists of columns expressed in relative widths (e.g. percentages, ems), as proportions of their containing element, rather than in fixed, inflexible pixels.
- Flexible images and media
  - Flexible images move and scale proportionally (shrinking or enlarging) as their flexible container resizes.
- Media queries, a module from the CSS3 specs
  - Media queries are a module from the CSS3 specification that allows building multiple layouts using the same HTML documents.

RWD

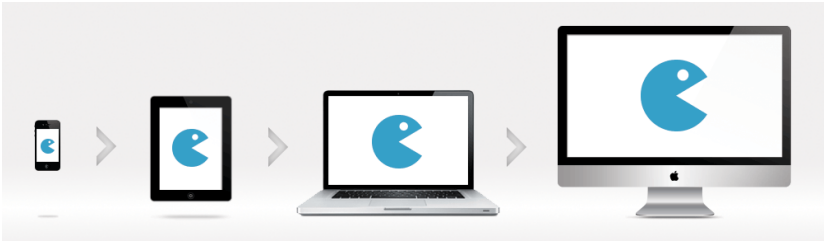
- CSS3 Media Queries
  - Logical true/false rules
  - Examples:

```
<link rel="stylesheet" href="my.css" media="screen">
@media screen and (width: 720px) {
 body { background-color: #00f; }
}
```

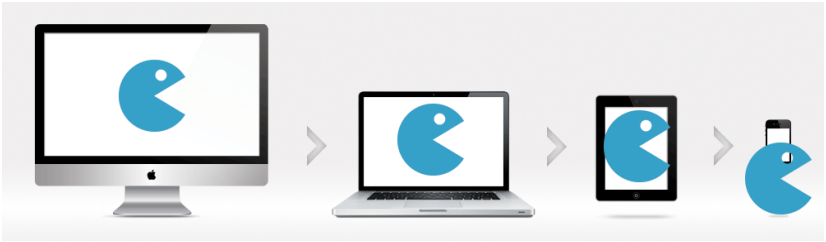
    - The first example grabs a specific stylesheet based on the rendering agent
    - The second example sets the background color if the screen is 720 pixels wide
    - There are a *ton* of conditions you can apply – min/max width/height, aspect ratios, orientation – plus you can and and not them together (or by separate rules)
  - Adaptive versus Responsive Web Design
    - Adaptive creates multiple fixed-width layouts for common screen sizes
    - RWD uses flexible dimensions
      - RWD will also replace objects if they don't fit, whereas AWD tries to scale

Design Approaches

"Designing starts at reference resolutions and with use of media queries adapting to other resolutions"



- Mobile-First
  - Bottom-up
  - Progressive Enhancement
  - Develop with the lowest common denominator then progressively enhanced as the resolution increases



- Desktop-First
  - Top-down
  - Graceful Degradation
  - Use desktop resolution as starting point and gracefully degrades design as the resolution decreases

Pros vs cons of design approach

Desktop-First	
Pros	**Cons**
"Faster development time"	Not "mobile friendly"
Designers & developers are more familiar w/ developing website for desktops	Tablet or mobile version is downgrade vs desktop version
	Prone to accidental removal of vital features and content

Mobile-First	
Pros	**Cons**
Focusing on the most vital parts of website as data speed & screen size is limited	Demand higher learning curve
Focusing on delivering the best browsing experience	Slower to develop
Explore new capabilities for mobile web	

Pros and Cons of RWD

Pros

- SoC – UI Devs and Back-end Devs
- Focus on key factors – content and device-independent website
- Long-term money & time savings; Easy Maintenance
- More consistent user experience and usability
- Better Search Engine Optimization (SEO)
- (Should be) Single URL

Cons

- Overlapping ways to accomplish same thing for UI and Back-end Devs
- Which one should we focus on first? (Desktop or Mobile) ?
- Hugh upfront cost for RWD research and design (BUFD)
- Optimizing RWD site performance is not easy
- Impact on performance of RWD sites
  - Heavy payload results in slow loading of a web page on mobile devices
- Potential incompatibility issues with CSS media queries

Summary

- Mobile devices are forcing change
  - RWD detects device's screen size & rendering features and adjusts
  - RWD may also remap some block structures (HTML5?) to different widgets
- Things to remember:
  - Detecting screen sizes is important during design and implementation
  - Web designers required to have a differently way of thinking
  - Validating and testing RWD designs on as many platforms (simulation / mockup design) as possible
  - Get latest modern development tools
- In practice
  - This is an example of a technology most people "buy"
  - Leave it to specialists to write the CSS, however it is 1) useful to understand the basics so you know good theme structure, and 2) user experience is critically important, you as app developer own it