MOBILE DEVICE EXPLOSION

- Study from 8/14/13 - Nearly 1 in 3 Web site Visits Come from Mobile
- Web visits coming from mobile devices increased 73 percent since the second quarter in 2012
- 77% of world’s population has a mobile device

HOW MOBILE WEB DIFFERS

- Mobile browser support is inconsistent
- Mobile devices are smaller and slower
- Different input tools (stylus, fingers)
- Different browsers
- 320 x 480 pixels is a common resolution for mobile

DEVICE RESOLUTION IN PIXELS PER INCH

- Nexus One/ Droid Incredible - 800 x 480
- HTC Evo 4G/ HTC Desire HD- 800 x 480
- Droid 2 - 854 x 480
- Samsung Galaxy - 800 x 480
- iPhone 3GS - 480 x 320
- iPhone 4 - 960 x 640
- iPhone 5 - 1136 x 640
- iPad - 1024 x 768
- BlackBerry PlayBook -1024 x 600

DEVELOPMENT OPTIONS

- Mobile-responsive Web site
- Progressive Enhancement
- Mobile-specific Version of a Web site

GRACEFUL DEGRADATION
• Create a site using newest features - then make sure older browsers don't choke on the code and display the content if not all the features

• As long as it displayed the content - no matter how painful the experience might be - successfully practiced graceful degradation

• Not a good experience for the older browser user, but also not forward thinking to the mobile devices that use less capable browsers

PROGRESSIVE ENHANCEMENT

• Instead of building for latest browsers first - create a baseline experience first
• use semantic markup and structure
  ■ present the content in a clean, useable form
  ■ layer presentation and interactivity in a way that preserves the baseline experience
  ■ provide richer experience for more capable browsers
• Layers - peanut M&M analogy from Aaron Gustafson
  ■ content - peanut
  ■ presentation is the CSS - chocolate
  ■ JavaScript for interactivity - the shell
  ■ content can stand by itself, more rich experience as you layer functionality on top

ADAPTIVE & RESPONSIVE WEB DESIGN

• Attempt to deal with the mobile device explosion
• One Web approach - same info and service available irrespective of the device

• 2 approaches - each with their own strengths and weaknesses
  ■ Responsive
  ■ Adaptive, client-side and server-side flavors

RESPONSIVE WEB DESIGN

• most commonly used - becoming de facto standard
• defined by Ethan Marcotte
  ■ A List Apart article in 2010
  ■ book - A Book Apart - Responsive Web Design
• a responsive site is no more a mobile site than it is a desktop site or a tablet site
  ■ "adopting a more flexible device-agnostic approach to designing for the web" - Ethan Marcotte
• fluidly change and respond to fit any screen or device size
• deliver same HTML and CSS to all devices
  ■ only custom CSS for elements that need to be treated differently
• akin to Progressive Enhancement on steroids

RESPONSIVE DESIGN CONTAINS 3 ASPECTS
1. **fluid grids** - built using a relative unit of measure like ems or percent instead of pixels

2. **flexible images and media** - display larger on a big-screen, and they scale down as the size reduces
   - change image sources in CSS

3. **media queries** and the `@media` rule
   - included in the CSS3 standard
   - allows you to define different styles for different screen width
   - determine what to apply to current environment

**FLUID LAYOUTS**

- Default body font size of 100%
- Adjust percentages for widths
- Set images and objects to a max width of 100%

```
img, object { max-width: 100%; }
```

- Don't use IMG width and height attributes

**ADVANTAGES OF RESPONSIVE WEB DESIGN**

- single template for all devices
- CSS used to determine how content is rendered on different screen sizes
- Open-source toolkits - simplify process
  - Bootstrap
  - Foundation
  - Boilerplate

**DISADVANTAGES OF RESPONSIVE WEB DESIGN**

- need complete site re-build
- performance hits - so much CSS
- performance matters
  - smartphones: 3.5% drop with one second wait
  - 3 seconds = 57% drop
- How to handle images

**ADAPTIVE WEB DESIGN**
something of a precursor to responsive design
uses a fixed width grid instead
coined by Aaron Gustafson - book of same title
will change to fit a predetermined set of screen and device sizes
pre-defined set of layout sizes based on device size + CSS and JS
implementation in earlier Bootstrap, the layout of the screen resizes once specific pixel widths are reached
the design may not fill the entire screen at all times the way a fully-responsive layout does

ADVANTAGES OF ADAPTIVE WEB DESIGN

• Advantage: ability to reuse one set of HTML and JS across devices
• Specifically target devices
• Only required resources are loaded by client's device

CLIENT-SIDE ADAPTIVE

• adaptations occur on the client side in user's browser
• device and feature detection switched to mobile device itself
• JS to enrich sites with advanced functionality and customization
• delivery retina-quality images only to retina displays
• don't have to re-build site from ground up
• need a solid grasp of JS
• layer on top of your existing code base

SERVER-SIDE ADAPTIVE

• server does detecting and loading correct template
• server-side plugins and custom user agent detection
• distinct templates for each device enabling more customization
• manage multiple templates can be costly
• smaller mobile pages that load faster
• requires significant changes to back-end systems - lengthy and costly implementation
• heavy load on server can cause performance issues

COMPARISONS OF AWD VS RWD
• methods differ in their delivery of the responsive or adaptive structures
• RWD relies on flexible and fluid grids
• AWD relies on predefined screen sizes
• RWD takes more code and implementation strategies
• AWD has streamlined, layered approach utilizing scripting
• terms overlap: *On Adaptive vs. Responsive Web Design*

**MOBILE FIRST**

• Philosophy created by Luke Wroblewski - book: *Mobile First*
• Prioritize design for mobile devices
• Use Progresive Enhancement web strategy to cover all users

**MOBILE FIRST RESPONSIVE WEB DESIGN**

• RWD techniques that start from a mobile template, add progressive enhancement
• **Smallest phones** - begin with basic HTML, simple layout, small images, limited CSS and JS
• **Smart phones** - add newer HTML5 features if supported, simple layout, small images, but bigger than featurephone size, more CSS and JS
• **Tablets** - with more room, add optional content (sidebars), multiple column layouts, larger images
• **Desktops** - add widescreen layouts, larger images

**MEDIA QUERIES IN CSS2**

• Previously starting in CSS2 - used media type to differentiate devices
  media="screen | print | handheld | ...
• Let you specify a type of media to target
• never gained a lot of support by devices other than print
• Now, so many devices, this method needed extending

**DEVELOP FOR DIFFERENT SCREEN SIZES**

• Multi-column makes sense for laptops and desktops
• Perhaps two column layout for tablets
• Perhaps one column layout for smart phones
MEDIA QUERIES IN CSS3

- instead of type of device, look for capability of device
- width and height of browser window
- orientation - in landscape or portrait mode
- resolution

MEDIA QUERIES LINK CODE

```html
<link href="... .css" rel="stylesheet"
media="only screen and (max-width:900px),
only screen and (max-width: 683px)
and (orientation:landscape">
```

ALTERNATE APPROACHES

- External linked CSS
  ```html
  <link ... media="only screen and (max-width:900px),
  only screen and (max-width: 683px)
  and (orientation:landscape">
  ```
- Imported CSS
  ```html
  @import url (... .css) screen and (max-width:900px);
  ```
- Media blocks within <style>
  ```html
  @media screen and (max-width:900px) {
  /* styles here */
  }
  ```

MEDIA FEATURES

- Height, width [[min-width]:900px]
  - Viewport of browser/ user agent
- Device-height, device-width [[max-device-height]:480px;]
  - Actual value for device itself
- Orientation [portrait | landscape]
- Aspect-ratio [4/3 16/9]
- Color [2]
- Resolution [[min-resolution:]300dpi]
- And many more - cssmediaqueries.com, W3C explanations

CSS MEDIA QUERIES & IE

```html
<!--[if (lt IE 9)&(!IEMobile)]>
<link rel="stylesheet" type="text/css" href="layout.css"
```
Not supported in IE8 and earlier

Use conditional comment

**USING VIEWPORT**

```html
<meta name="viewport" content="width=device-width, initialscale=1, maximum-scale=1">
```

• When visiting a website via a mobile browser - assumes viewing from desktop, will be wanting to see all of it, not just the top left corner.

• Sets the viewport width at (in the case of iOS Safari) 980px, shoe-horning everything into its little display.

• **Don’t Forget the Viewport Meta Tag**

**VOCABULARY**

• **Progressive Enhancement**

• **Graceful Degradation**

• **Responsive Web Design**

• **Adaptive Web Design**

• **Mobile First** - prioritize design for mobile devices

• **Media Queries** - allow you to control which styles are applied based on device properties such as width, orientation, resolution

• **Breakpoint** - the point at which a new media query is applied (e.g. 980px)

**MORE VOCABULARY**

• **Viewport** - the browser’s visible screen area

• **Framework** - structure or scaffold that provides some base functionality intended to serve as a support or guide for building applications

• **Less** - dynamic stylesheet language used by Bootstrap; extends CSS with dynamic behavior such as variables, mixins, operations, functions
  - **Using Less with Bootstrap**

• **Normalize** - Bootstrap utilizes to reset CSS to make browsers render all elements more consistently and in line with modern standards

**DREAMWEAVER FLUID GRIDS EXAMPLE**

• **Building fluid grid layouts in Adobe Dreamweaver CS6**