

Responsive Web Design

birds of feather

Approaches to Mobile Development

1. “No Mobile” Approach
2. Native Mobile Applications
3. Mobile Websites
4. Responsive (universal) design

“No Mobile” Approach

- Website that does not offer a tailored mobile experience (either app or website)
- Can still be viewable on most devices, but not particularly usable (Having to zoom in/out and pan around = BAD USER EXPERIENCE)

Not always an issue of ignorance, but budget or other shortcomings

Old style Mobile Websites

- “Browser sniffing”
 - –Method that identifies which browser and operating system you are using
- Requires maintaining a list of browsers and operating systems
- Features/content left out for mobile users
 - Might make sense, but too often due to assumptions instead of user studies
- If redirects are not properly set up, sharing links can be problematic
- Maintaining several code bases

Native Mobile Applications

- **Barrier to entry**
 - – Device and even OS version
- **Separation of content and features**
- **Costly**

What is good about Mobile Websites and Native Apps

- Both offer experience tailored to mobile devices
- Native applications can take advantage of advanced device capabilities

Web browsers are catching up!

RWD

Proposed by Ethan Marcotte on A List Apart in May 2010

<http://www.alistapart.com/articles/responsive-web-design/>

One website for all devices!

Optimized for different contexts using:

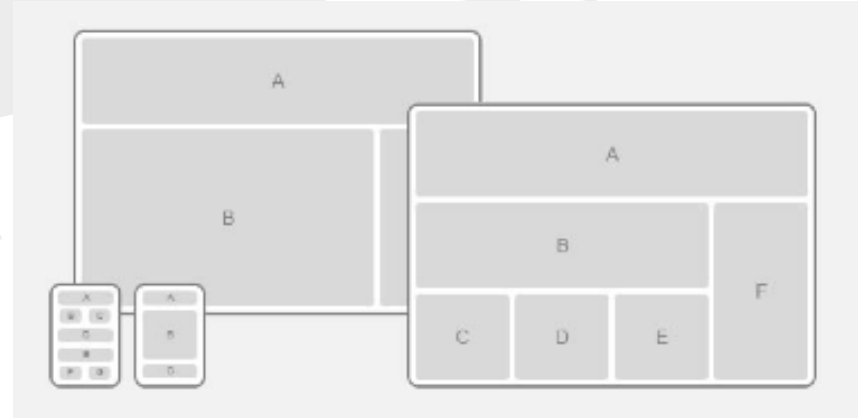
- Fluid grids
- Flexible media
- CSS Media Queries

Grid Systems

A way of organizing different pieces of information along vertical and horizontal axes.

Fluid grid = width of boxes is defined in percentage rather than fixed units (pixels, em)

- Can grow or shrink as the screen width changes
- Allows for utilizing all available space
- Avoids issue of horizontal scrolling



Flexible Media

Similar concept to fluid grids, but applied to images and movies so dimensions of media can change depending on screen size.

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

Can result in problems in older browsers that don't support max-width (Internet Explorer 7)

- Set width to 100%

Image considerations

- If images are going to be viewed at small sizes, no point in serving large resolution images?
- Existing images look blurry on displays with high pixel density
- May serve different images based on media queries?

Media Queries

- Part of CSS3 specification
- Extends existing media type functionality that allowed style sheets for print, screen, etc.
- Gives more granular control as to when different CSS rules are applied
 - Based on media features such as screen width/height, screen orientation, pixel density, etc...

Media Query Examples

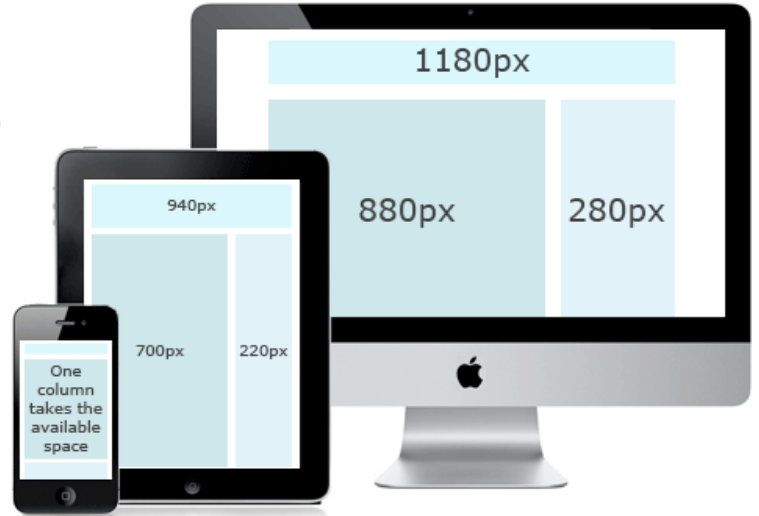
```
@media screen and (min-width: 650px) and (max-width: 960px) {...}
```

```
@media (min-width:800px) and (max-width:1200px) and (orientation:portrait) { ... }
```

```
@media screen and (-webkit-device-pixel-ratio: 1.5),  
screen and (resolution: 144dpi) { ... }
```

Breakpoints

- Breakpoints are defined resolution points (typically width) specified in media queries at which different CSS styles are applied.
- Breakpoints example:
 - Below 650 (small screen)
 - 650-960 (tablet)
 - Above 960 (desktop)
- Should be chosen based on your content rather than known resolutions of popular devices



Media Query Support

- **Mobile browsers**
 - –iOS Safari (3.2+)
 - –Android Browser (2.1+)
- **Desktop browsers**
 - –Internet Explorer (9+)
 - –Firefox (3.5+)
 - –Chrome (4+)
 - –Safari (4+)

Full list at <http://caniuse.com/css-mediaqueries>

How To Handle Lack of Media Query Support

- **Mobile-first approach for other browsers**
 - Default CSS = single column layout
 - Introduce additional complexity inside media queries (unsupported browsers will just ignore this)
- **Respond.js - solid but limiting** (no support for device-width, device-height, orientation, aspect-ratio, color, monochrome or resolution)
- **CSS3-MediaQueries-js** - more supported features but slow to load
- **Conditional IE Style Sheets**
 - Your media queries are simple enough to include in a single style sheet;
 - You do not have to support more legacy desktop browsers.

Advantages of RWD

- one content site to manage
 - content is always current
 - less Maintenance
- development costs are low(er)
- adjusts to different width/sizes
- device OS independent
 - Smartphone, Tablet, iPhone, Android, etc
- plays well with current initiatives on campus (Drupal, need for short development times)
- clear need for an alternative to app development

Drupal and RWD

Options to start with:

- Use existing responsive theme (the most common approach)
 - [Bootstrap \(Stanford\)](#)
 - [Boilerplate](#)
 - [Responsive Skeleton](#)
 - [Omega](#)
 - [Zen](#) (mobile first)
- Develop your own

Compare

Existing Themes

Pros	Cons
Standardization	Size
Support	Complexity
Communal Accountability	Learning Curve
Best Practices	Assumptions

Custom Themes

Pros	Cons
Small Footprint	Lack of standards
Maximum Control	Lack of Support
Project Specificity	Lack of Flexibility
Quick Implementation	Need for Accountability

Drupal Modules (for handling images)

- Adaptive Image

Handles images by serving device appropriate versions from your site's image fields. It's also very easy to configure, all you need to do is add the adaptive effect to an image style and specify some breakpoints.

- Picture

Uses the proposed (but not implemented) HTML5 picture element and delivers alternate image sources depending on the device capabilities. This will prevent devices operating on bandwidth-challenged networks from downloading large images.

- Retina Images

Solves this problem for Drupal by adding an option to the image effects that come with Drupal core to make them output high resolution images that look good on these Retina displays

Drupal Modules

- FitVids

Handles videos for your fluid layout so you don't have to worry about that. You know, when you embed YouTube or Vimeo videos they come with a specific size. Well this takes care of that and makes them resize automatically based on the container they are in.

- Tinynav

Implements the tinynav.js jQuery library to transform your menu into a select dropdown on small devices.