Responsive Web Design

IT 4213 Mobile Web Development
IT 4403 Advanced Web and Mobile Applications

Jack G. Zheng
Spring 2020

https://www.edocr.com/v/yxdmelxy/jgzheng/responsive-web-design

KENNESAW STATE UNIVERSITY

Overview

This lecture notes summarize basic approaches to develop and deliver mobile websites, with a particular focus on responsive design.

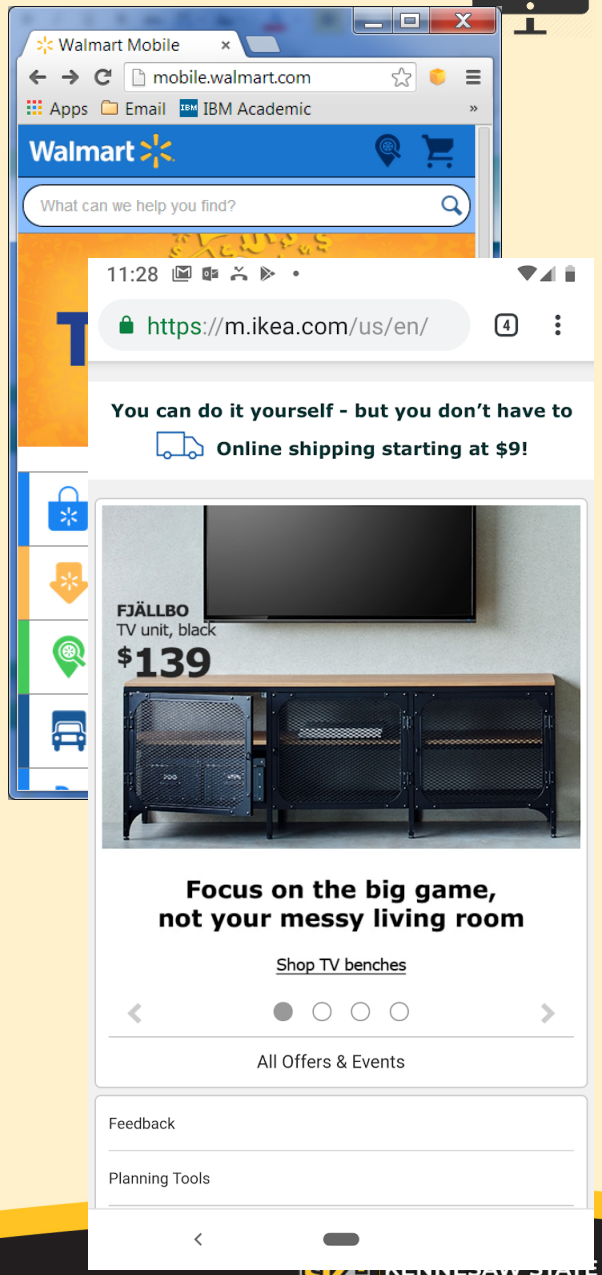
- Two (or four including subcategories) basic approaches
  - Separate mobile site
  - One web
    - Responsive web design
    - Adaptive (dynamic serving)
    - Hybrid: RESS (Responsive + Server Side Components)
- Responsive web design principles and key practices
  - Fluid-grid
  - Fluid image/media
  - Media query

Mobile Website Delivery Strategies

- Separate mobile version site
  - Separate mobile site has its own domain and address, and is independent from the main site.
- One Web
  - One Web means making, as far as is reasonable, the same information and services available to users irrespective of the device they are using, but with different presentations (UI)

Separate Mobile Site

- Separate mobile site has its own domain and address, and is independent from the main site.
- A separate mobile site can best meet mobile user experience. Build a separate mobile-optimized design (or mobile site) if you can afford it.
  - https://www.nngroup.com/articles/mobile-site-vs-full-site/
- Some examples:
  - Wikipedia: https://en.m.wikipedia.org
  - Goodwill: https://mgwdonate.ging.org (used a third party service, more like an app)
- Some companies has moved away from separate mobile site, such as
  - Costco (m.costco.com)
  - Walmart (mobile.walmart.com)
  - Ebay (m.ebay.com)
  - Newegg: (m.newegg.com)
  - Ikea: m.ikea.com/us (jQuery Mobile)
  - Nike: m.nike.com
  - Staples: m.staples.com



Extended reading: Why Separate Mobile & Desktop Web Pages?
http://www.lukew.com/ff/entry.asp?1390

One Web

Quotes from https://www.w3.org/TR/mobile-bp/#OneWeb

- One Web means making, as far as is reasonable, the same information and services available to users irrespective of the device they are using, but with different presentations (UI)
- It does not mean that the content is available in exactly the same representation across all devices.
- The context of mobile use, device capability variations, bandwidth issues and mobile network capabilities all affect the representation.

- One Web approaches
  - Responsive web design (response to screen size)
  - Adaptive (dynamic serving based on device detection and adaptation)
  - Hybrid: RESS (*Responsive Design* + *Server Side Components*) http://www.lukew.com/ff/entry.asp?1392

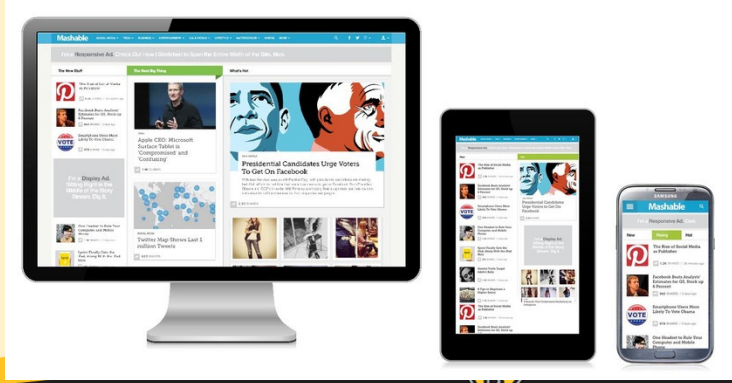
Key readings:
- Design for multi-screen experiences https://www.thinkwithgoogle.com/articles/building-websites-multi-screen-consumer.html
- https://www.wired.com/2013/05/the-two-flavors-of-a-one-web-approach-responsive-vs-adaptive/

Responsive Web Design

Responsive Web Design (RWD) is a Web design approach aimed at crafting sites to provide an optimal viewing experience, easy reading and navigation with a minimum of resizing, panning, and scrolling, across a wide range of screen sizes and devices.

- Ethan Marcotte coined the term responsive web design and defined it to mean fluid grid/ flexible images/ media queries in a May 2010 article in A List Apart
  - http://alistapart.com/article/responsive-web-design
- Basic principles and practices
  1. Fluid grid - no horizontal scrolling
  2. Adaptive/flexible image
  3. Media query

Some interesting infographics about RWD
https://www.webfx.com/blog/web-design/infographics-learn-responsive-web-design/

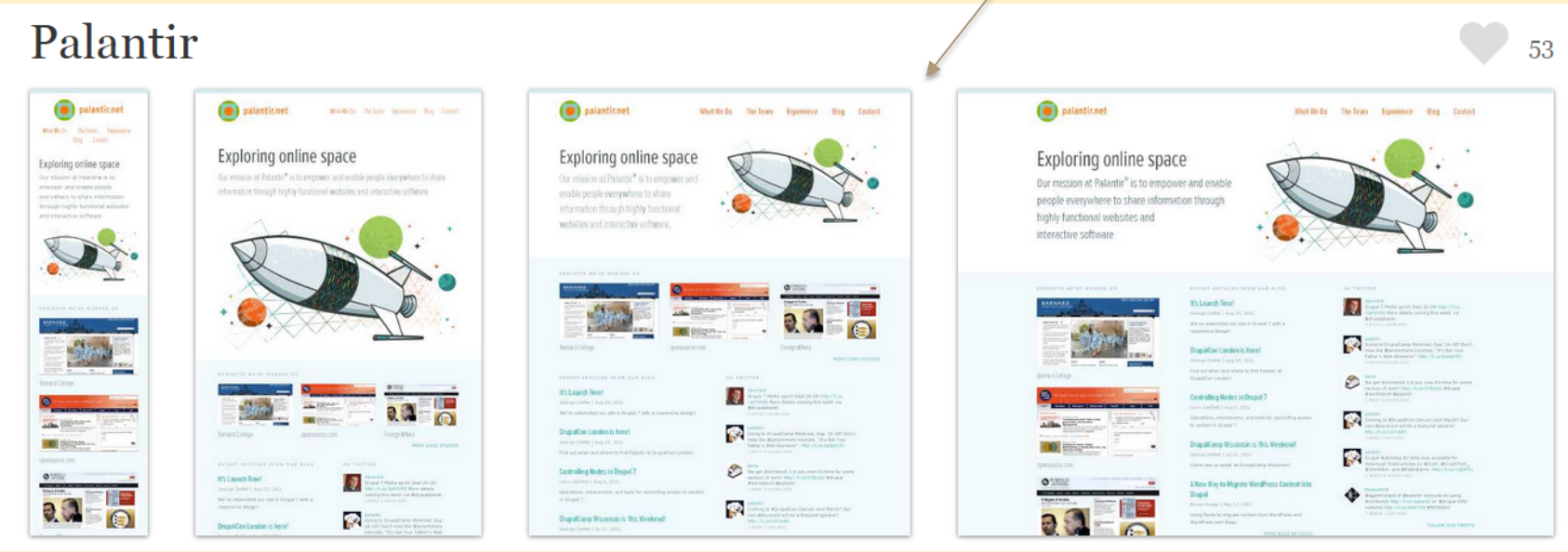


Examples

Adjust the browser width and see the changes or use a tool like responsinator.com or browser developer tools.

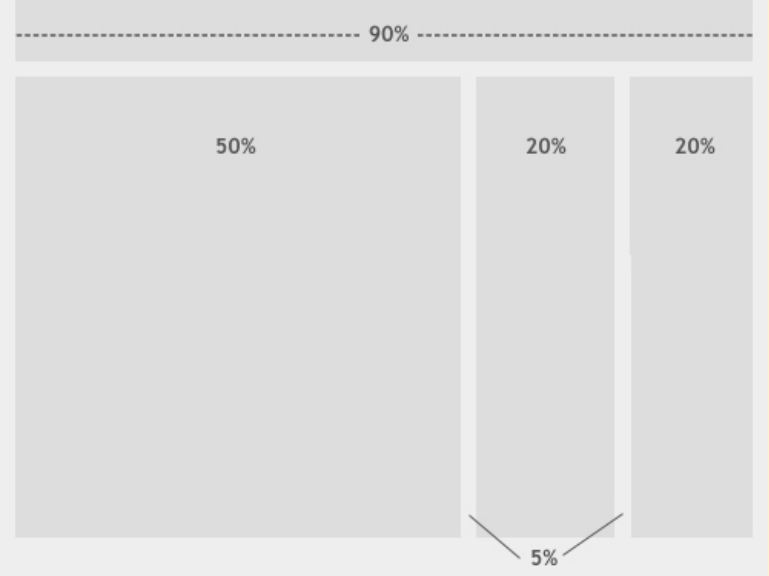
- Responsive examples
  - http://kennesaw.edu
  - https://cobbcounty.org
  - http://mashable.com
  - http://mediaqueri.es (more examples here)

See the webpage layout and content changes as the page width changes. The example is from https://mediaqueri.es



- Non-responsive examples
  - https://www.weather.gov (not mobile friendly)
  - http://yahoo.com (adaptive but not responsive)
  - See more: http://cubicle-h.blogspot.com/2018/08/website-mobile-friendliness-testing.html

Fluid Layout



- Fluid layout, is the practice of building the layout of a website capable of dynamically resizing to any width.
  - using relative length units, most commonly percentages or em units.
  - Set the width using %
  - Set margin and padding using %
- Formula
  - result (%) = target / context (parent size)
- This may not look optimal in some cases especially for margins and paddings. Use it together with media queries.
- Other practices
  - Set max-width and min-width

Extended reading: https://alistapart.com/article/fluidgrids and the example https://alistapart.github.io/code-samples/fluidgrids/examples/grid/final.html

Adaptive Image

- Apply similar fluid practices to images using percentages
  - Code example: https://www.w3schools.com/css/tryit.asp?filename=tryresponsive_image2
- The image can be scaled up to be larger than its original size, which makes the image blurry. A better solution, in many cases, will be to use the max-width property.
  - See this example: https://www.w3schools.com/css/tryit.asp?filename=tryresponsive_image (and compare to the prior example)

```
img
{
  display: block;
  max-width: 100%;
}
```

Don't forget the max-width property, which limit the image size to its original size.

- Reference: https://www.w3schools.com/css/css_rwd_images.asp
- Take it to another level with media queries http://responsiveimages.org

Media Query

- Media query is a CSS3 module allowing content rendering to adapt to conditions such as screen resolution.
- A W3C recommended standard (June 2012)
  - https://www.w3.org/TR/css3-mediaqueries/
- A cornerstone technology of responsive web design
- All major browsers support it
  - http://caniuse.com/#feat=css-mediaqueries

IE	Edge *	Firefox	Chrome	Safari	Opera	iOS Safari *	Opera Mini *	Android Browser *	Opera Mobile *	Chrome for Android
				[1] [2] 3.1 - 3.2						
6 - 8		2 - 3	[1] 4 - 25	[1] 4 - 6		[1] 3.2 - 6.1		[1] 2.1 - 4.3		
[1] 9 - 10	12 - 18	3.5 - 71	26 - 78	6.1 - 12.1	10 - 63	7 - 13.1		4.4 - 4.4.4	12 - 12.1	
[1] 11	79	72	79	13	64	13.2	all	76	46	79
		73 - 74	80 - 82	TP		13.3				

Media Query Basics

- Media queries are based on rules (contexts) when a set of CSS should be applied
- Code example
  - Setting a context in CSS (<style>) block

```
@media screen and (max-width: 300px)
{
    div {background-color: lightblue;}
}
```

Media type

Media features

  - Setting a context in a linked CSS file

```
<link rel="stylesheet" type="text/css" href="..."
media="screen and (max-width: 300px)" />
```

- Three parts in any context
  1. Media type: type of presentation media, like “screen”
  2. Media features: specify the characteristics of media type, like “width”
  3. Logical operator: combine multiple rules

Media Types and Media Features

- Media Queries Level 4 defines four media types

Media Type	Description
all	Used for all media type devices
print	Used for printers
screen	Used for computer screens
speech	Used for devices that “read out” a page

- Major media features
  - width (max-width, min-width)
  - aspect-ratio
  - orientation
  - resolution

- For a full list of types and features, refer to:
  - https://developer.mozilla.org/en-US/docs/Web/CSS/Media_Queries/Using_media_queries
  - http://www.w3schools.com/cssref/css3_pr_mediaquery.asp

Logic in Media Queries

- And

```
@media (min-width: 600px) and (max-width: 800px)
```

- Or

```
@media (max-width: 600px), (min-width: 800px)
```

- Not

```
@media not screen and (color)
```

- See more here
  - http://css-tricks.com/logic-in-media-queries/
  - https://developer.mozilla.org/en-US/docs/Web/Guide/CSS/Media_queries

Media Query Practical Examples

- When the width is between 400px and 600px on screen

```
@media screen and (max-width: 600px) and (min-width: 400px)
```

- When the screen is in portrait orientation and width is less than 640px

```
@media screen and (max-width: 640px) and (orientation:
portrait)
```

- When the screen is narrower than 16:9 aspect ratio and the width is between 480px and 960px

```
@media screen and (max-width: 960px) and (min-width:480px)
and (max-aspect-ratio:16/9)
```

Use the following web app for quick testing:
https://www.w3schools.com/cssref/tryit.asp?filename=trycss3_media2

Identifying Media Query Breakpoints

- Properly set the breakpoints of media queries to allow best viewing and interaction experience based on viewport width.
- Some recommend a common set of breakpoints, such as 576, 768, 992, 1200 (Bootstrap).
- Some (http://bradfrost.com/blog/post/7-habits-of-highly-effective-media-queries/) do not recommend to write media query breakpoints around common viewport sizes as determined by different device resolutions. New devices and resolutions are being released all of the time. Trying to keep up with these changes could be an endless process. Recommendations:
  - Create breakpoints based on content, never on specific devices, products, or brands.
  - Breakpoints should only be introduced when a website starts to break, look weird, or the experience is being hampered.
  - Pick minor breakpoints when necessary in additional to major layout changes.

Arrange a Set of Media Queries

- Media queries are commonly used to set contexts for different sizes of screens, based on the width property
- How to arrange media query contexts? There are mainly two approaches:
  - Graceful degradation
  - Mobile first or progressive enhancement

Extended reading: Generic First
https://www.smashingmagazine.com/2018/12/generic-css-mobile-first/

Graceful Degradation

- Define the default styles for large screen first.
- Then use a set of rules with max-width (in descending order) for smaller screens by order.

Normal CSS styles applied

```
/* Default styles first for the largest
screen; then media queries for smaller
screens */
...

@media screen and (max-width: 1400px) {...}
@media screen and (max-width: 1000px) {...}
@media screen and (max-width: 600px) {...}
@media screen and (max-width: 400px) {...}
```

Later rules override prior rules so you don't need to redefine every CSS style. Only override those needed to be changed.

Mobile First (Progressive Enhancement)

- Starts from default styles for the smallest screen.
  - “The absence for @media queries is in fact the first @media query."
- Then use a set of rules with min-width (in ascending order) for larger screens.
- An example:
  - http://zellwk.com/blog/how-to-write-mobile-first-css/

Use ascending min-width property to “grow” the width

Default CSS styles first for the smallest screen; then media queries

```
/* Default styles first for the smallest
screen; then media queries for larger
screens*/
...
@media screen and (min-width: 400px) {...}
@media screen and (min-width: 600px) {...}
@media screen and (min-width: 1000px) {...}
@media screen and (min-width: 1400px) {...}
```

Later rules override prior rules

Mobile First

- Mobile First is a philosophy created by Luke Wroblewski that highlights the need to prioritize the mobile context when creating user experiences.
  - Design for the smallest mobile device first; then progressively enhance the experience as more screen real estate becomes available.
  - Allows websites to reach more people
  - Forces designers to focus on core content and functionality (What do you do when you lose 80% of your screen real estate?)
  - Let designers innovate and take advantage of new technologies (geolocation, touch events and more)
  - http://www.lukew.com/presos/preso.asp?26

- Extended readings
  - mobile-first responsive web design http://bradfrostweb.com/blog/web/mobile-first-responsive-web-design/
  - A Hands-On Guide to Mobile-First Responsive Design https://www.uxpin.com/studio/blog/a-hands-on-guide-to-mobile-first-design/

Media Query on Mobile Devices

- To make media query correctly recognize the width condition on mobile browsers, viewport needs to be correctly set using the **viewport meta tag**. Otherwise the viewport will be set at 980px by default and media query rules for smaller screens may not be triggered.
- https://www.w3schools.com/css/css_rwd_viewport.asp

Media Query – A Complete Example

- A complete example of using media query is described by Ethan in his original article
  - http://alistapart.com/article/responsive-web-design
- Check out the design results (from the example above) in some basic steps. Each is becoming more responsive.
  1. Fluid design at the beginning: https://alistapart.github.io/code-samples/responsive-web-design/ex/ex-site-flexible.html
  2. Add a breakpoint: https://alistapart.github.io/code-samples/responsive-web-design/ex/ex-site-linearize.html
  3. Set for smaller screens: https://alistapart.github.io/code-samples/responsive-web-design/ex/ex-site-mini.html
  4. Set for larger screens: https://alistapart.github.io/code-samples/responsive-web-design/ex/ex-site-larger.html
  5. Final design: https://alistapart.github.io/code-samples/responsive-web-design/ex/ex-site-FINAL.html

Media Query Uses and Applications

- Set different display styles
  - Change page layout
  - Change appearance/style of elements like text, tables, and forms.
- Set different content
  - Show/hide different part of the page
  - Use (or hide) different image background
  - Use different image with different quality

RWD/Mobile General Principles

- Two key differences of mobile web use is the small screen and touch oriented operations.
- Design for Small Screens: the grand general principle of small screens is to keep the page short
  - Most mobile websites are viewed in portrait mode with narrow width. Stretching a normal multi column webpage will make the page extra long. https://www.viget.com/articles/do-responsive-sites-have-to-be-so-tall-on-mobile/
  - Long stretching pages are more likely to get user lost and difficult to navigate
- Design for touch friendly UI
  - Touch UI poses some challenges but also offers opportunities to design better interaction methods and enable additional functionalities than traditional UI.

These two principles and some best practices will be cover in module 4.

RWD Patterns and Best Practices

- RWD can be applied to many UI elements and content types, like:
  - Page layout
  - Navigation (menu)
  - Content: text, table, list
  - Form and controls
  - Specific content type like data, catalog, article, dashboard, etc.
  - Media like image, video, gallery, etc.
  - Decorative elements like icon, background, etc.
- General patterns and best practices collection
  - http://bradfrost.github.io/this-is-responsive/patterns.html
  - https://responsivedesign.is/patterns/
  - http://ui-patterns.com/patterns
- Dedicated sites for examples
  - http://mediaqueri.es
  - http://www.awwwards.com/websites/responsive-design/
  - http://responsivedesign.is/examples
  - https://designmodo.com/responsive-design-examples/
  - https://www.awwwards.com/50-examples-of-responsive-web-design.html

More specific patterns and best practices for selected content types will be covered in later modules 5 to 9.

RWD Tools

- Responsive design test
  - https://material.io/resizer/
  - http://www.responsinator.com
  - https://www.browsersync.io
  - http://mobiletest.me
- Emulator
  - https://developers.google.com/web/tools/chrome-devtools/device-mode/
  - https://developer.mozilla.org/en-US/docs/Tools/Responsive_Design_Mode
  - http://www.mobilexweb.com/emulators
  - http://lab.maltewassermann.com/viewport-resizer/
- Screen size detection
  - http://mydevice.io
  - http://detectmobilebrowsers.com
  - Screen resolution: http://www.screenresolution.org or http://www.whatismyscreenresolution.com
- Viewport tools (device or visual viewport)
  - https://viewportsizer.com and https://viewportsizer.com/devices/
  - http://whatismyviewport.com
  - https://www.mydevice.io
  - http://lab.maltewassermann.com/viewport-resizer/
- Feature check
  - http://mobilehtml5.org
  - http://caniuse.com

RWD UI Frameworks

- Despite the debate of using frameworks, RWD UI frameworks has been growing in popularity
  - http://www.smashingmagazine.com/2014/02/19/responsive-design-frameworks-just-because-you-can-should-you/
- Major frameworks
  - Bootstrap: http://getbootstrap.com
  - Zurb Foundation: http://foundation.zurb.com
  - Skeleton: http://getskeleton.com
  - Sencha Touch: http://www.sencha.com/products/touch/
- More comparison
  - http://responsive.vermilion.com/compare.php
  - http://www.sitepoint.com/5-most-popular-frontend-frameworks-compared
  - http://mobile-frameworks-comparison-chart.com

This subject will be covered in details in module 10

Adaptive (Dynamic Serving)

- Adaptive design aims to deliver website UI (content, style, function, etc.) and user experience that adapts to the capabilities of the device/browser.
  - Also called dynamic serving as in https://www.thinkwithgoogle.com/marketing-resources/experience-design/building-websites-multi-screen-consumer/
  - Adaptation can be done either at the server side or at the client side - https://www.wired.com/2013/05/the-two-flavors-of-a-one-web-approach-responsive-vs-adaptive/

- Of the Alexa 100 sites, 80% use adaptive web design in their delivery.
  - https://mobiforge.com/news-comment/adaptive-web-design-dominates-in-the-webs-top-brands

- Some examples (notice the webpage does not respond to real-time size change well):
  - http://google.com
  - https://www.yahoo.com
  - http://dealnews.com
  - https://gillette.com
  - https://github.com/vmware

- Responsive design can somewhat be considered as a kind of adaptive design. It adapts UI based on screen size (width mainly). But adaptive design considers more features than just screen size.
  - http://bradfrost.com/blog/post/the-many-faces-of-adaptive-design/

Feature/Device Detection

- Adaptive design heavily uses feature detection and other sound progressive enhancement techniques.
- These features may include
  - Screen size and capabilities (orientation, touch, etc.)
  - Device/OS type
  - Browser type/version
  - Hardware features
  - https://developer.ibm.com/articles/responsive-design-future/
- In the separate mobile site approach, device detection is also often used to redirect users to mobile or desktop version site.
- Feature detection techniques
  - Parsing User-Agent strings: https://deviceatlas.com/blog/user-agent-parsing-how-it-works-and-how-it-can-be-used
  - Using JavaScript testing: Modernizr https://modernizr.com
- More resources: https://deviceatlas.com/knowledge-base

RESS

- RESS: Responsive Design + Server Side Components
- The two approaches are not mutually exclusive. Many sites combine techniques to achieve the best results.
- Some techniques
  - A single set of page templates define an entire Web site for all devices (responsive) but key components within that site have device-class specific implementations that are rendered server side (server side components). http://www.lukew.com/ff/entry.asp?1392
  - Conditional loading (adaptive): http://christianheilmann.com/2012/12/19/conditional-loading-of-resources-with-mediaqueries/
- Comparison of the three:
  - http://www.lukew.com/ff/entry.asp?1509

Key Readings and Learning

- Approaches summary: https://www.thinkwithgoogle.com/marketing-resources/experience-design/building-websites-multi-screen-consumer/
- Responsive web design
  - http://alistapart.com/article/responsive-web-design: the original article focused on concepts.
  - Good introduction with more technical details: https://msdn.microsoft.com/en-us/magazine/hh653584.aspx
  - Media queries: https://developer.mozilla.org/en-US/docs/Web/CSS/Media_Queries/Using_media_queries
- Adaptive web design
  - https://www.wired.com/2013/05/the-two-flavors-of-a-one-web-approach-responsive-vs-adaptive/
  - https://deviceatlas.com/blog/7-myths-about-adaptive-web-design-debunked
- RESS: the original article raising the idea https://www.lukew.com/ff/entry.asp?1392
- Comparison: https://www.lukew.com/ff/entry.asp?1509

Good Resources

- Influencers
  - https://ethanmarcotte.com
  - http://www.lukew.com
  - http://bradfrost.com
- Media resources
  - https://alistapart.com/topic/responsive-design
  - https://www.smashingmagazine.com/category/responsive-web-design
  - https://www.smashingmagazine.com/responsive-web-design-guidelines-tutorials/
  - http://www.hongkiat.com/blog/tag/rwd/
- Tutorials collection
  - https://developer.mozilla.org/en-US/docs/Web/CSS/Media_Queries
  - http://www.quirksmode.org/mobile/
  - https://designwoop.com/responsive-web-design-tutorial
  - http://www.hongkiat.com/blog/responsive-for-mobile-screens/
  - http://www.hongkiat.com/blog/responsive-web-tutorials/
  - https://developer.mozilla.org/en-US/docs/Web_Development/Responsive_Web_design
  - http://www.youtube.com/playlist?list=PLtNErhYMkHnGh691QvRtMp9jMVzpMenL5