Responsive Design: Ben Callahan

(Video: 0_Introduction.mp4): **Introduction**

- 00:00:00-00:08:15: Ben describes a moment that changed his life. Receiving his first iPhone and coming to the realization the web is not fixed width.
    - Mobile Device and News Consumption: http://bit.ly/zE1zgp
    - Spark Box: http://seesparkbox.com
    - http://buildresponsively.com
    - Download the course repository: http://bit.ly/LWLTEA

Part 1: Responsive Web Design (RWD) 101 (00:08:16-01:02:14)

(Video: 1_A.mp4): **A Fluid Foundation**

- 00:08:16-00:10:56: The penetration of Laptop/Desktop ownership is the same today as it was in 2007. By the end of 2013, it's predicted more people will be browsing the web on mobile devices. Responsive web design addresses this trend.
    - Read: *Ethan Marcotte's* article on *Responsive Web Design*: http://alistapart.com
- 00:10:57-00:12:54: A fluid foundation honors proportions by creating percentage-based grids instead of fixed-width grids. Percentages are equal to the target divided by a context.
- 00:12:55-00:20:18: Code example demonstrating the difference between a fixed-width layout and a fluid grid system.

(Video: 1_B.mp4): **IE Rounding and the Semantic Grid System**

- 00:20:19-00:23:36: Modern web browsers are getting better at handling percentages like 33.3%. Older versions of Internet Explorer tend to round up and cause columns to wrap unnaturally. Understanding rounding techniques helps create more consistent fluid designs.
- 00:23:37-00:28:24: Some grid systems begin to "lie" to you semantically as the width of the browser gets smaller and smaller. Semantic grid systems always describe the layout accurately.
    - http://semantic.gs
    - http://susy.oddbird.net
    - http://getskeleton.com

(Video: 1_C.mp4): **Flexible Content**

- 00:28:25-00:30:06: Once we have a grid which is based on proportions, the content must also respond. Adding responsive CSS to elements like images can quickly create responsive content.
- 00:30:07-00:35:01: Code example demonstrating responsive image content within the flexible grid system.

- 00:35:02-00:43:44: Responsive video can utilize the same technique as responsive images. However, *iframes* can introduce some issues. Using intrinsic ratios, a responsive container can be built to enable a more responsive *iframe*.

(Video: 1_D.mp4): **Media Queries**

- 00:43:45-00:50:28: When the content and the design are no longer working in harmony, a larger shift in layout may be necessary. Media queries give developers the flexibility to add/override CSS based on media type or media feature.
    - http://www.w3.org/TR/CSS21/media.html#media-types
    - http://www.w3.org/TR/css3-mediaqueries
- 00:50:29-00:53:22: Two approaches for applying media queries are additive and subtractive. With additive, you design for the smallest resolution and add media queries for larger displays. Subtractive begins with the largest displays and add media queries for the smaller viewports.
- 00:53:23-00:56:57: Both additive CSS versus subtractive CSS have pros and cons.
    - Media Query Bookmarklet: http://sparkbox.github.com/mediaQueryBookmarklet
- 00:56:58-01:02:14: Code example using media queries and the media query bookmarklet.

(Video: 1_E.mp4): **Other RWD Considerations**

- 01:02:15-01:10:16: A fluid foundation, flexible content, and media queries are all heavily driven by CSS and the building blocks for any responsive web design. There are other considerations, however. These include touch/target areas, hover states, contrast, and readability.
    - http://lukew.com/ff/entry.asp?1085
-

Part 2: RWD Process (01:10:17-01:37:20)

(Video: 2_A.mp4): **A Myth about Process**

- 01:10:17-01:15:17: It's often thought each client deliverable needs to look more like a final end-product than the previous one. What's better is to deliver organized and prioritized content that functions across multiple resolutions.
- 01:15:18-01:17:08: Deliverables are better categorized as Research, Content, Priority, Style, and Functional. Today, what is often a functional deliverable should be shifted into the priority and style categories.

(Video: 2_B.mp4): **Content Priority Prototype**

- 01:17:09-01:22:56: A content priority prototype replaces the traditional wireframe. It's created in HTML and uses as much real content as possible. Markup is generated by the content/UX people. This layer of semantics that's often lost in design-driven markup

(Video: 2_C.mp4): **Style Prototype**

- 01:22:57-01:31:39: Style prototypes are very fast to build, use accurate web typography and easily show web interactions. Clients preview these prototypes in a browser of their choice. This helps set style expectations for legacy browser support.
    - http://sparkbox.github.com/dr-style-prototype
    - http://bit.ly/Tb7HPr
- 01:31:40-01:36:16: Examples and articles on Style Deliverables. The key is collaboration in the process.
    - http://responsiveprocess.com/yp2012/wireframes/
    - http://bradfrostweb.com/demo/mobile-first/
    - http://rif.superfriend.ly/designs/round2/
    - http://seesparkbox.com/foundry/our_new_responsive_design_deliverable_the_style_prototype/
    - http://alistapart.com/articles/responsive-comping-obtaining-signoff-with-mockups/
- 01:36:17-01:37:20: Priority becomes critical with small displays.

Part 3: Applying RWD Styles (01:37:21-02:38:25)

(Video: 3_A.mp4): **The Basic Structure**

- 01:37:21-01:45:08: Applying responsive style begins with the *viewport* <meta> tag. From there style sheets are loaded ranging from a base set of styles (base.css), to media queries (mq.css), to legacy browser support (nomq.css).

(Video: 3_B.mp4): **CSS files (using Sass**)

- 01:45:09-01:49:31: Example of a base.css file that's using the Sass CSS preprocessor. It contains viewport information, Sass partials for a CSS reset file, smallest display resolution, and a media query for the print CSS.
- 01:49:32-01:51:03: MQ.css begins to load the styles based on the width of the device. Sass partials can be used to dynamically construct mq.css.
- 01:51:04-01:52:14: NOMQ.css handles older versions of Internet Explorer that don't support media queries. They include all styling rules without media query wrappers. Using a CSS preprocessor like Sass makes maintaining multiple CSS files easier.

(Video: 3_C.mp4): **Implementing RWD Styles**

- 01:52:15-01:59:57: Practical example of this structure on Ben Callahan's website.
    - http://bencallahan.com
    - http://mediaqueri.es
- 01:59:58-02:01:36: Looking at the example files included in the course repository, the CSS structure has been recreated without the use of Sass or any other CSS preprocessor.

(Video: 3_D.mp4): **Using EM-Based Media Queries**

- 02:01:37-02:07:33: An em-based approach to media queries allows for a more proportional measurement and layout that adjust based on font-size.
    - http://cloudfour.com

(Video: 3_E.mp4): **RWD Patterns: Navigation**

- 02:07:34-02:19:20: Responsive web design still poses many challenges including complex navigation and tabular data. Looking at how others are handling these challenges begins to outline emerging patterns in responsive design. First, look at navigation.
    - http://siyelo.com
    - http://contentsmagazine.com
    - http://2012.dconstruct.org
    - http://stry.us
    - http://twitter.github.com/bootstrap/
    - http://msj.edu
    - http://dpandl.com
    - http://bostonglobe.com

(Video: 3_F.mp4): **RWD Patterns: Tables and Images**

- 02:19:21-02:23:39: Designing responsive table is often driven by the type of data you are representing. Techniques include scrolling columns, repositioning headers, and toggling column visibility.
    - http://zurb.com
    - http://css-tricks.com
    - http://codepen.io/bencallahan
- 02:23:40-02:30:53: Images add both responsive display and bandwidth challenges. There are varying techniques but not a one-size-fits-all solution. The HTML <picture> element may be the answer. Until supported, using a polyfill like *Picturefill* or other solution is suggested.
    - http://github.com/scottjehl/picturefill/
    - http://docs.sencha.io/current/index.html#1/guide/src
    - http://resrc.it/
    - http://adaptive-images.com

(Video: 3_G.mp4): **RWD Patterns: Off-Canvas Layouts**

- 02:30:54-02:34:54: Off-canvas layouts use the space outside a browser's viewport to hide secondary elements until needed. Animation help users understand where the content originate from as it transitions on and off screen
    - http://lukew.com/ff/entry.asp?1569
    - http://jasonweaver.name/lab/lw/#menu
    - http://jasonweaver.name/lab/offcanvas/
- 02:34:55-02:38:25: A look at the code of an off-canvas implementation.

Part 4: RWD Retrofitting (02:38:26-03:27:21)

(Video: 4_A.mp4): **From Fixed to Fluid**

- 02:38:26-02:43:28: Responsive web design doesn't mean starting from scratch with a web presence. Retrofitting can be a fast, low-risk approach for creating better responsive experiences for existing websites.
  - Techniques: http://github.com/bencallahan/rwd-retrofitting/

(Video: 4_B.mp4): **Retrofitting Twitter.com**

- 02:43:29-02:53:51: Ben gives a demonstration of retrofitting Twitter.com. Using the Chrome developer tool, Ben adds some responsiveness to the design.
- 02:53:52-02:55:43: Adding responsive design to Twitter.com was done using the simple equation: target / context = result.
  - http://alistapart.com/articles/fluidgrids/
  - http://paulirish.com/2012/box-sizing-border-box-ftw/

(Video: 4_C.mp4): **Retrofitting Images**

- 02:55:44-02:59:57: One of the biggest issues with images can be content management systems that write width/height attributes or inline styles. Finding ways to override this functionality is the key to retrofitting images. Be sure to verify browser compatibility with any inline image style techniques.
- 02:59:58-03:01:10: Background images have responsive considerations too. One of the main techniques is to prevent the loading or requesting of backgrounds.
  - http://timkadlec.com/2012/04/media-query-asset-downloading-results/

(Video: 4_D.mp4): **Retrofitting Tables**

- 03:01:11-03:16:09: Retrofitting tables can be extremely difficult, but possible. Ben demonstrates how to style table headers and cells to add a responsive flow to an otherwise ridged design.

(Video: 4_E.mp4): **Retrofitting Media Queries**

- 03:16:10-03:21:06: Media queries in responsive designs typically start typically use a smallest-resolution-first approach. When retrofitting, the approach of small resolution, capped is often applied. This uses the original CSS for the larger viewports and a mobile-first CSS for the small viewports.

(Video: 4_F.mp4): **Client Interaction**

- 03:21:07-03:24:46: Answering some simple questions can help determine if a retrofitting project is right. Remember that retrofitting is a *step* in the right direction, but not a final solution.

- 03:24:47-03:27:21: More fun with retrofitting.

Part 5: JS to the Rescue (03:27:22--03:49:47)

(Video: 5_A.mp4): **Media Queries**

- 03:27:22-03:32:51: Use *polyfills* to help add support for media queries in legacy browsers
    - respond.js: http://github.com/scottjehl/Respond
    - css3-mediaqueries.js: http://code.google.com/p/css3-mediaqueries-js/
- 03:32:52-03:39:21: The *matchMedia* method in the CSS Object Model can be helpful to change the functionality of a website based on the width. Like media queries, polyfills may be required to add support.
    - matchMedia.js: http://github.com/paulirish/matchMedia.js/
    - Harvey: http://github.com/harvesthq/harvey/
    - mediaCheck: http://github.com/sparkbox/mediaCheck/

(Video: 5_B.mp4): **Conditional Loading**

- 03:39:22-03:45:16: Loading content conditionally is a technique to change the amount of content loading for different resolutions. Determining what a user wants to see based on their resolution is not always a good practice though and should be carefully considered.
    - http://github.com/filamentgroup/Ajax-Include-Pattern/
    - http://filamentgroup.github.com/Ajax-Include-Pattern/test/functional/media.html
    - http://m.people.com
- 03:45:17-03:47:39: Earlier, we implemented responsive images using *Picturefill*. This is an example of JavaScript rescuing responsive design. *eCSSential* is another technique that will only load the necessary CSS.
    - http://github.com/scottjehl/picturefill
    - http://github.com/scottjehl/eCSSential
- 03:47:40-03:49:47: *FitText* is a jQuery plugin that inflates your web type to fill the width of a parent element. Responsive video can easily be achieved with *FitVids.js*.
    - http://fittextjs.com
    - http://fitvidsjs.com

Part 6: Lessons Learned (03:49:48-04:14:27)

(Video: 6_A.mp4): **Pricing**

- 03:49:48-04:02:02: Understanding the cost associated with responsive web design is crucial to accurately setting client expectations and budgets. Ben gives some helpful pricing percentages based on his experience.

(Video: 6_B.mp4): **Prioritization**

- 04:02:03-04:08:13: Prioritizing content and functionality leads to better responsive design. Prioritizing doesn't mean restricting. It's putting the most important mobile content where it's needed.

(Video: 6_C.mp4): **Testing, Consistency, and Experimentation**

- 04:08:14-04:13:44: You must test on real devices. You should also be developing in a *Webkit* browser since this makes up a large portion of the market share. Consistent experiences across different resolutions will lead to added user familiarity.
    - http://seesparkbox.com/foundry/cross_width_consistency/
- 04:13:45-04:14:27: Get into the browser quickly because that's where the experimentation occurs.

Part 7: What's next in RWD? (04:14:28-04:30:20)

(Video: 7_A.mp4): **The Responsive Dip**

- 04:14:28-04:24:02: The Responsive Dip is a play on the *Stages of Competence*, but focused toward responsive design. It starts by having a responsive mindset.
    - http://bit.ly/MasqBk
- 04:24:03-04:30:20: Web designers and developers must move beyond the technique and move the industry forward.