GENERAL ASSEMBLY

THE BASICS: RESPONSIVE DESIGN

Kim Goulbourne

AGENDA

- Review
- Responsive Design
- Fluid vs Fixed
- Em Typography
- Media Queries
- Handing responsive in JS
- Lab Time

REVIEW

RESPONSIVE DESIGN

RESPONSIVE DESIGN

I heard you want to be a web developer
Here are a few devices to test your site

WHAT IS RESPONSIVE DESIGN?

Responsive web design (RWD) is an approach to web design aimed at crafting sites to provide an optimal viewing and interaction experience across a wide range of devices (from desktop computer monitors to mobile phones).

We are going to learn how to BUILD responsive sites.

"*One site for every screen.*"

RESPONSIVE EXAMPLES

http://thenextweb.com/
http://elespacio.net
http://time.com/
http://foundermantras.com

What makes these responsive?

WHAT'S HAPPENNING?

- Columns are Changing Size
- Images are Scaling
- Dimensions are Changing Columns
- Navigation Items are Being Rearranged
- Elements Are Being Hidden and Shown
- Typography is Changing Size / Ratios

RESPONSIVE DESIGN

FIXED VS FLUID

WHAT IS A “FIXED” LAYOUT?

A “fixed” layout uses pixels to set box model values (height, width, margin, padding).

It doesn’t care what size the browser is, it will always honor the pixel value set.

WHAT IS A “FLUID” LAYOUT?

A “fluid/flexible” layout uses percentages (for height, width, margin, padding) to achieve relative sizes based on the browser size.

Fluid layout is our first step toward Responsive Design.

* Note: You cannot set a % border-width.

LET'S SEE THE DIFFERENCE

Fixed: https://jsfiddle.net/kimgoulb/x0bcwjoh/1/
Fluid: https://jsfiddle.net/kimgoulb/x0bcwjoh/3/

Combo: https://jsfiddle.net/kimgoulb/8Lzmb8hv/1/ (uses pixels for gutter and spacing between elements)

BREAK 10 MINS

BREAK!

RESPONSIVE DESIGN

MEDIA QUERIES

WHAT IS A MEDIA QUERY?

A media query is use to define different style rules for different media types/ devices. It’s like an “if” statement for CSS.

Media queries can be used to check many things, such as:

- width and height of the viewport or device
- orientation (is the tablet/phone in landscape or portrait mode?)
- resolution and much more

http://www.w3schools.com/cssref/css3_pr_mediaquery.asp

USING THE @MEDIA KEYWORD

```
@media only screen and (check size/resolution/orientation) {
    /* style blocks */
}
```

- “only screen” targets just screens

```
@media all and (check size/resolution/orientation) {
    /* style blocks */
}
```

- “all” targets both screen and print

BROWSER WIDTH/HEIGHT MEDIA QUERIES

```
@media only screen and (min-width: 1024px) { }
```

- targets browsers greater than or equal to 1024px wide

```
@media only screen and (max-width: 1024px) { }
```

- targets browsers less than or equal to 1024px wide

```
@media only screen and (min-height: 600px) { }
```

- targets browsers greater than or equal to 600px in height

```
@media only screen and (max-height: 600px) { }
```

- targets browsers less than or equal to 600px in height

RESPONSIVE DESIGN

```
// all sizes
p {
  font-size: 16px;
}

// what will this media query target?
@media only screen and (min-width: 1024px) {
  p {
    font-size: 24px;
  }
}
```

USING MULTIPLE CONDITIONS

Separate multiple condition statements with “and”:

```
@media only screen and (min-width: 320px) and (max-width: 480px) {
  /* These styles will only apply on devices between 320 and 480 pixels wide. */
}
```

** Note: both min-width and max-width are inclusive so it will match the numbers chosen vs being 1 less.*

WHAT ARE BREAKPOINTS?

Breakpoints are target widths used to change the layout. Classic breakpoints to target are:

1440px - desktop large
1280px - desktop
1024 - tablet landscape
768 - tablet portrait
568px - small tablet / large mobile
320px or 375px - mobile (based on iphone5 or 6)

TARGETING TABLET PORTRAIT

```
p {
 font-size: 16px;
}

@media only screen and (min-width: 768px) and (max-width: 1023px) {
  p {
   font-size: 24px;
  }
}
```

TARGETING TABLET PORTRAIT

```
.element {
 width:30%;
}

@media only screen and (min-width: 768px) and (max-width: 1023px) {
  .element {
   width:50%;
  }
}
```

RESPONSIVE LAYOUTS EXERCISE

We need to update the layouts on tablet and mobile using the 320px and 768px breakpoints.

EM TYPOGRAPHY

WHAT IS AN “EM”?

An “em” is a unit of measurement. Just like pixels, ems can determine the size of elements on a web page. Unlike pixels, which are fixed, ems are relative to their parent’s font size.

1em = inherited font size (default on body is 16px)

** If the font size of a <div> is set to 16px, 1em within that <div> is equivalent to 16px.*
** If the font size of that <div> changes to 20px, 1em within that <div> is equivalent to 20px.*

WHAAAAATT???

Check out this demo: https://jsfiddle.net/kimgoulb/gurh7upb/

- The body is set to a font-size of 40px so on this page:
- 1em = 40px //// 2em = 80px //// 3em = 120px

```
.element {
  font-size: 20px;
  line-height: 1.2em; //i.e. 24px
}
```

HOW TO USE EMS

Set a font-size on the <body> using pixels.

Set your typography based on ems.

Then, in your media queries at smaller sizes, set that <body> font size to be smaller. The other values will cascade down.

RESPONSIVE DESIGN

TLDR;

IMPORTANT RESPONSIVE TAKEAWAYS

- Use percentages on box model styles (width, height, margin, padding) to create a more fluid layout
- Use media queries to update your layout a different breakpoints
- Use ems on font styles (i.e font size and line height)

HANDLING RESPONSIVE IN JS

DETECTING SCREEN WIDTH IN JS

We can use the following methods to get the size of the browser window at any time:

```
$(window).width();
$(window).height();
```

* You can also get and set the width of other elements using these methods. Eg. $(‘.element’).width() or $(‘.element’).width(200);

THE .RESIZE() EVENT

The resize event is used to track and make changes in the browser when the user is manually resizing the window (for w.e reason). This is an opportunity to check the screen width or height and update elements if necessary.

```
var winH = $(window).height();
$(window).on(‘resize’, function(){
   winH = $(window).height(); // update
   $(‘.hero’).height(winH);
});
```

THE .RESIZE() EVENT

```
var winW = $(window).width();

$(window).on(‘resize’, function(){
  winW = $(window).width(); // update
  if (winW < 768) {
    $(‘aside’).hide();
  } else {
    $(‘aside’).show();
  }
});
```

FINAL PROJECT

FIRST DRAFT JS DUE THURS, NOV 3

LAB TIME

LET’S MAKE A HAMBURGER MENU

HOMEWORK

FINISH THE HAMBURGER MENU