FEWD - Responsive Basics

James Gallichio

Agenda

- Review
- Responsive Layouts
- Fonts - rem/em
- Media Queries

Review

Boxes

Responsive Design

Fixed Layout

- What we've mostly been doing till now
- Relies on a container of fixed width
- Usually around 960px or 980px

Responsive Layout

- Different styles for different screen widths
- Uses an elastic/fluid layout
- Flexbox and percentages

Fixed vs Responsive

Checkout these **Fixed** sites

- Google.com
- Spacex.com
- UPS.com

Checkout these **Responsive** Sites

- Generalassemb.ly
- Dwolla.com
- Shapeways.com
- Relayrides.com

Mobile Boxes

Font Sizing

Px

- Based on display resolution pixels (rather than physical screen pixels)
- Simple, consistent size metric
- May not scale as well on older browsers if user changes font settings

EM

- Relative size metric
- Sized based on the width of the letter 'M'
- A multiplier of the parent element's size
- 1em=100% font-size; 1.4em = 140% font-size;

EM

```css
.someElement {
    font-size: 16px;
}

.someElement .child {
    font-size: 2em;
}

.someElement .child .grandchild {
    font-size: 1.5em;
}

/* .child's font size is 32px */
/* .grandchild's font size is 48px */
```

REM

- 'Root' em
- Same as em but based on the font-size of `<html>` element

REM

```css
html {
    font-size: 14px;
}

.someElement .child {
    font-size: 2rem;
}

.someElement .child .grandchild {
    font-size: 1.5rem;
}

/* .child's font size is 28px */
/* .grandchild's font size is 21px */
```

What should I use?

- `px` is easiest to read
- `rem` is easier to scale and maintain
- Avoid `em` for fonts (fine for padding or margins though)

Media Queries

Media Queries - Specify the conditions

```css
@media screen
@media print
@media (max-width: 1280px)
@media (min-width: 480px)
@media (orientation: portrait)
@media (orientation: landscape)
```

Combine multiple conditions

Separate multiple clauses with 'and'

```css
@media screen and (max-width: 600px){
    /* mobile specific css in here */
}

@media (min-width: 600px) and (max-width: 960px) {
    /* mobile specific css in here */
}
```

Usage

```css
.box {
    display: inline-block;
    width: 30%;
}

@media only screen and (max-width:768px) {
    .box {
        display: block;
        width: 100%;
    }
}
```

Mobile Display

We can also specify how a mobile browser displays the document:

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

<meta name="viewport" content="width=device-width, initial-scale=0.8, maximum-scale=2">
```

Things to keep in mind

- Start with mobile first.
  - It's easier to scale up a simpler design than scale down a complex one.
  - You're forced to decide what your site *really* needs.
- Design for sizes not devices.
- Think about how user experience might differ across devices.

Responsive Boxes

Further Reading

- w3schools - Responsive Web design
- MDN - Using media queries
- CSS-Tricks - Media Queries for Standard Devices