FEWD Week 3 • Class 6: Responsive Web Design

but first...

GitHub

Review Time!

What We'll Cover

- Responsive Web Design with Media Queries
- Support Queries
- Pseudo Elements

Responsive Web Design

Objectives

- Describe Responsive Web Design
- Use media-queries to create an adaptive design
- Explain why the viewport meta tag is important
- Use flexible images
- Add media-queries to your grid to adjust for small format screens

Responsive Web Design

Responsive designs respond to changes in browser width or orientation by adjusting placement of content to fit in the available space.

Hallmarks of Responsive Web Design

- A Fluid Layout
- Media Queries
- Flexible Images

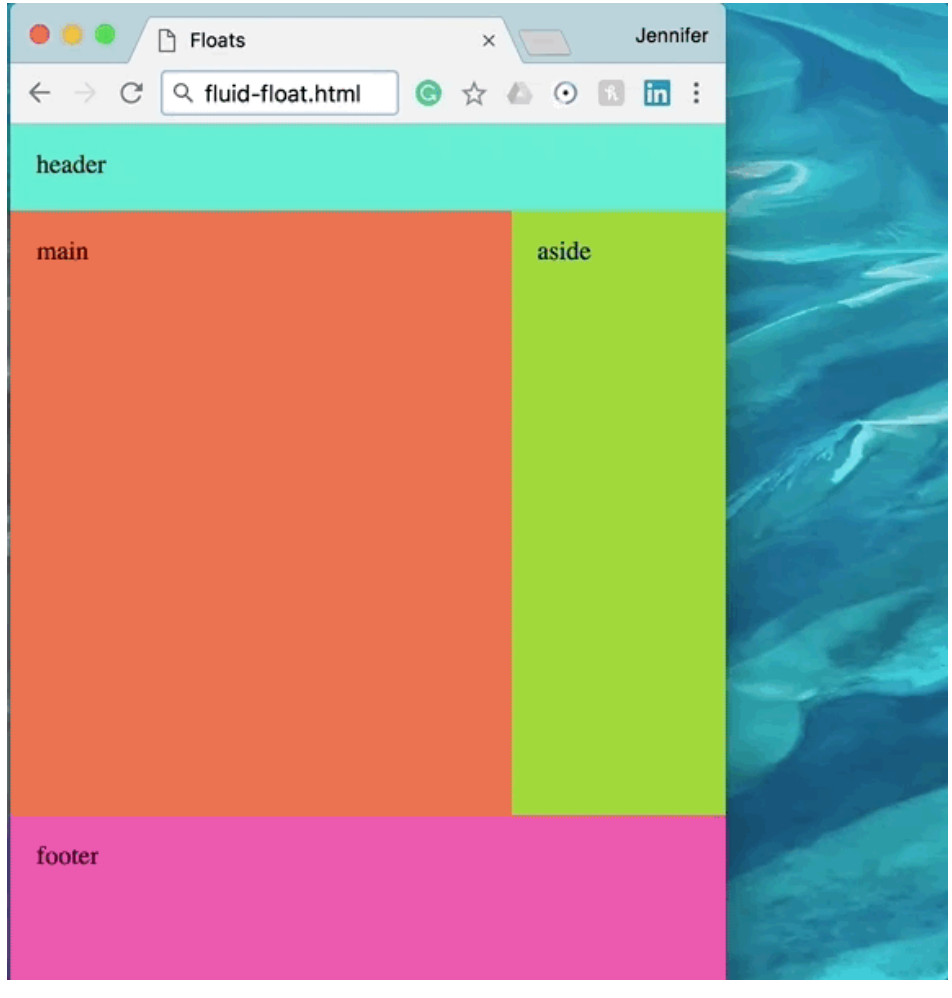
Fluid Layout

Fluid Layout

Fluid layouts can be accomplished with floats, flexbox and grid. They are built with relative units of measure so that the page can adapt easily to different screen sizes.

Fluid Floats Example

```
header {
  width: 100%;
}
main {
  width: 75%;
  float: left;
}
aside {
  width: 25%;
  float: left;
}
footer {
  width: 100%;
}
```



Media Queries

Media Queries

Media Queries allow us to write CSS rules that are applied for specific screen sizes or orientation.

When the conditions of the query are satisified the styles within are applied.

Media Queries can also be used for adapting to other conditions, such as `print` and `speech`.

Media Query Example

```
div {
  width: 100%;
  background-color: blue;
}

/*  Media queries need to follow the base styles
    to be applied correctly   */

@media (min-width: 768px) {

  div {
    width: 75%; /* Cascades and overrides the width: 100% rule */
  }

}
```

Media Features

Media features allow us to selectively apply styles based on a feature of the device or browser in use. They include: width and height, the orientation of the device, the aspect-ratio or resolution of the device.

```
@media (orientation: landscape) {
  /* orientation is a media feature */
}

@media (max-width: 768px) {
  /* max-width is a media feature */
}
```

Media Types

We can also target the delivery mode with *media types*. There are: `screen`, `speech`, `print`, and `only`.

```
/*  Exclusively used when the page is printed */

@media only print {
  ...
}
```

Logical Operators

Media features and types can be combined using operators:
`and`, `or`, `not`.

```
/* All of our extra small, small and medium devices in landscape mode */
@media (orientation: landscape) and (min-width: 767.98px) {
  body {
    flex-direction: row;
  }
}

/* Some of our small, all medium and some large devices */

@media (max-width: 950px) and (min-width: 600px)  {...}
```

Combining Media Types and Media Features

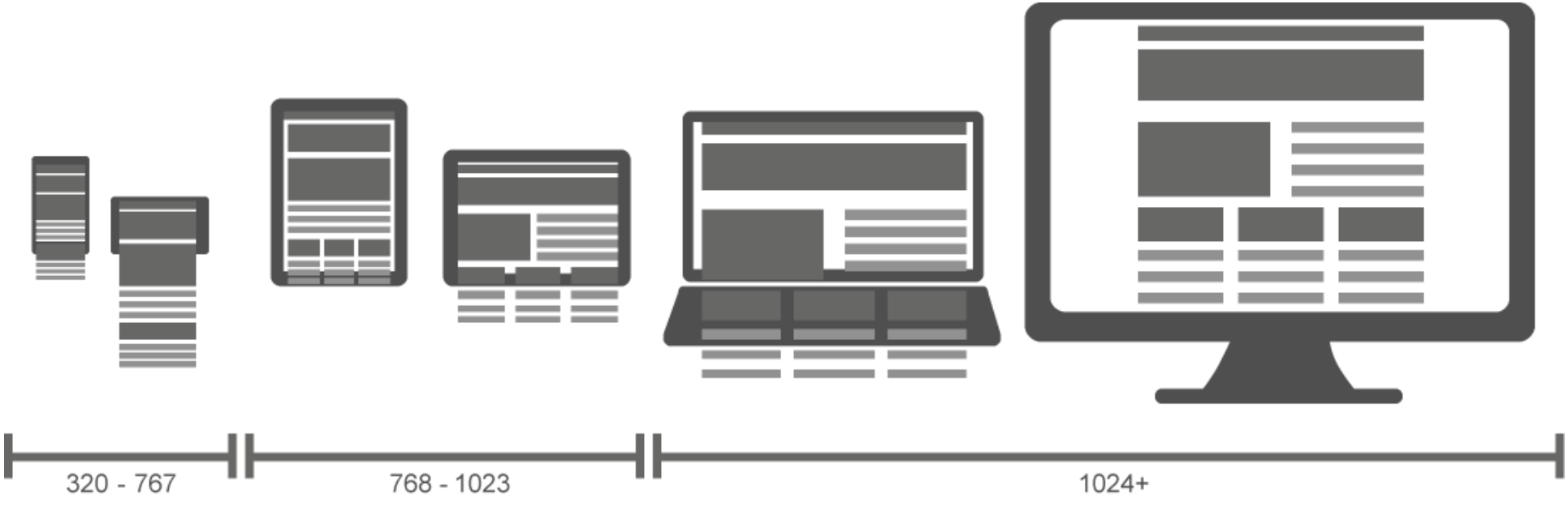
```
/* On devices under 768px */
.col-left, .col-center, .col-right  {
  width: 100%;
}

@media only screen and (min-width: 768px) {
    [class*="col-"] {
        width: calc(100%/3);
        float: left;
    }
}
```

Media Queries

Mobile First

Mobile First Concept

Generally, we build for the smallest devices, then add media queries for common **breakpoints**.



Why Mobile First?

It's simple: Mobile devices are our **primary devices for viewing web content** today.

A Mobile First Approach

```
/* Extra small devices (portrait phones, less than 576px)
   are the default (Mobile First) */

/* Small devices (landscape phones, 576px and up) */
@media (min-width: 576px) { ... }

/* Medium devices (tablets, 768px and up) */
@media (min-width: 768px) { ... }

/* Large devices (desktops, 992px and up) */
@media (min-width: 992px) { ... }

/* Extra large devices (large desktops, 1200px and up) */
@media (min-width: 1200px) { ... }
```

Adding Media Queries to Matchmaker

Viewport Tag

Viewport Meta Tag

- Without instructions, mobile devices render pages at typical desktop screen widths, and then scale the pages to fit the mobile viewport
- If you don't set the viewport meta tag, your media queries don't fire
- Thank you Apple!

Scaled vs. Responsive

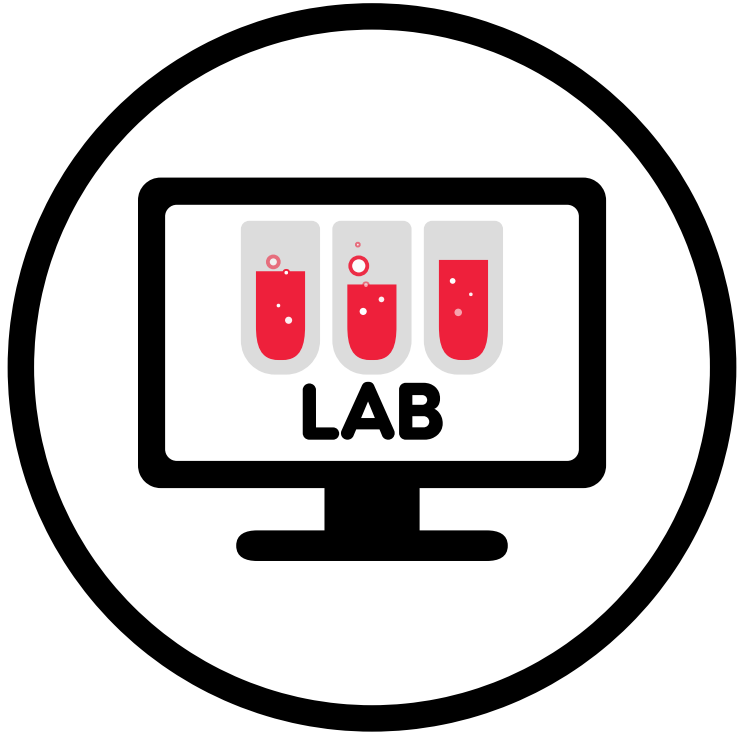

T-Mobile
1:44 PM
en.m.wikipedia.org
WIKIPEDIA
Today's featured article
Monroe Edwards (1808–1847) was an American slave trader who was convicted of forgery after a well-publicized trial in 1842. Originally from Kentucky, Edwards moved to New Orleans, then settled in Texas. He smuggled slaves into Brazil in 1832 and into Texas in 1836. After a swindle, he was forced to flee the Republic of Texas to the United States. He tried to scam money out of various abolitionists in the United States and the United Kingdom, partly with forged letters of introduction. He then forged letters from cotton brokers in New Orleans, which he used to secure bank drafts for large sums. His fabrications caught

Viewport Meta Tag

```
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width,
                               initial-scale=1, shrink-to-fit=no">
...
</head>
```

- `width=device-width`: sets the width of the viewport to the width of the device
- `initial-scale=1`: sets the initial zoom level when visiting the page
- `shrink-to-fit=no`: tells the device browser to reflow content instead of shrinking it

Adding Our Viewport Tag



Experiment with Media Queries

Support Queries

CSS @supports

Support queries let us check if a feature is available and apply styles accordingly! 🎉 👏 😀

Use @supports

```
@supports(prop:value) {
  /* more styles */
}

/* Check for flexbox */
@supports (display: flex) {
  div { display: flex; }
}
```

Logicals Operators Also Work

```
@supports (display: -webkit-flex) or
          (display: -moz-flex) or
          (display: flex) {

    /* use styles here */
}
```

A Real Life Example

```
section {
  float: left;
}

@supports (display: -webkit-flex) or
          (display: -moz-flex) or
          (display: flex) {

    section {
      display: -webkit-flex;
      display: -moz-flex;
      display: flex;
      float: none;
    }
}
```

Change our Floats to Grid

Pseudo Elements

It’s Kind of an Element

- Pseudo-classes can be used to style an element based on its state
- Keeps your HMTL semantic
- Careful about accessibility

You'll Use These

`::before`

Inserted as the first child

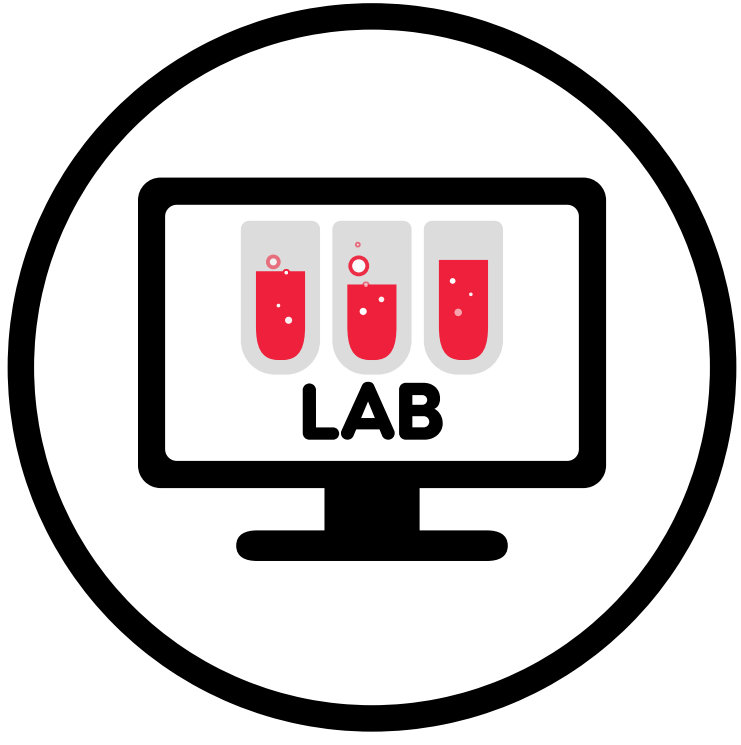
```
/* Add hearts before links */
a::before {
  content: "♥";
  padding-right: 10px;
}
```

`::after`

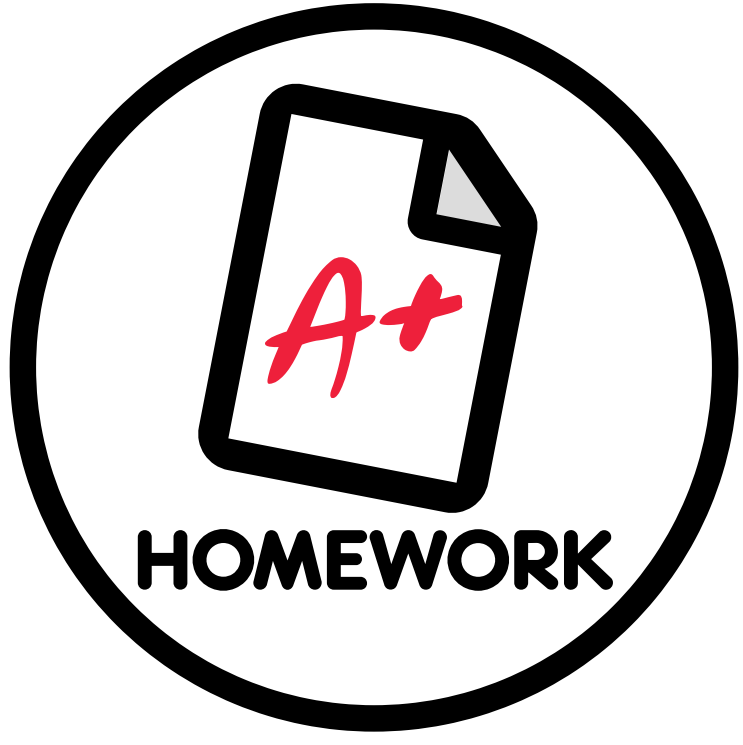
Inserted as the last child

```
/* Add copyright to the footer */
footer::after {
  content: "\0000A9 2018 Acme Corp."
}
```

Pseudo Elements in the Real World



Add a Copyright to Matchmaker



Finish Making Matchmaker Responsive

Go Build Awesome Things!