CSS Animations & Responsive Design

CISC-2350-R01 | Fall 2017 | Week 10-2

Ruta Kruliauskaite

Today’s Agenda

- Attendance
- Review: CSS grids
- Web review presentations (Michaela, Jake)
- More transition examples
- Homework Show & Tell (CSS transition tricks)
- CSS Animations
- CSS Responsive Design
- Homework Assignment

Review: grids

Keep in mind...

- Link files properly
- Make sure you downloaded the correct file and named it properly
- Make sure you're using correct class names
- Do not modify grid stylesheet
- Size images properly
- Add same height for images on the same row

700px

700px

Skeleton grid

- Can be found here: http://getskeleton.com/
- Download skeleton.css and link it in your HTML file
- Use classes:
  - container
  - one column
  - two columns, etc.
- Add images to additional with class="row" to keep track of what goes where

Skeleton example 1: text

Skeleton example 2: images

Skeleton example 2: images2

More transition examples

Hover effect example

Making things fly off the page

Change cursor example

Homework show & tell

http://bit.ly/2xzLHvt

CSS Animations

An animation lets an element gradually change from one style to another.

CSS Animations

- You can change as many CSS properties as you want
- Must first specify some keyframes for the animation
- Keyframes hold what styles the element will have at certain times (from beginning to end)
- Name the animation and bind it to HTML element

```
@keyframes pic {
    from {background-image: url(“images/pic1.jpg”);}
    to {background-image: url(“images/pic2.jpg”);}
}
div {
    width: 300px;
    height: 300px;
    background-image: url(“images/pic1.jpg”);
    animation-name: pic;
    animation-duration: 3s;
}
```

Example

CSS Animations

- Instead of *from* and *to* you can also specify %
- animation-delay
- animation-iteration-count (number / infinite / reverse / alternate)
- animation-timing-function (ease / linear / ease-in / ease-out / ease-in-out)
- also change position of HTML element

```
@keyframes pic {
        0% {background-image: url(“images/pic1.jpg”);}
        25% {background-image: url(“images/pic2.jpg”);}
        50% {background-image: url(“images/pic3.jpg”);}
        75% {background-image: url(“images/pic4.jpg”);}
        100% {background-image: url(“images/pic5.jpg”);}
}
div {
        width: 300px;
        height: 300px;
        background-image: url(“images/pic1.jpg”);
        animation-name: pic;
        animation-duration: 4s;
}
```

Example

CSS Animation libraries

- animate.css
- CSShake
- bounce.js (JavaScript)
- anime.js (JavaScript)

Animate.css

- Animate.css
- Cross-browser animations to use on HTML elements
- More documentation here: https://github.com/daneden/animate.css
- Download animate.css file, link it in your HTML page and animate!
- To use the animation, add the class "animated" to the element plus one of the following class names...

Animate.css class names

bounce
flash
pulse
rubberBand
shake
headShake
swing
tada
wobble
jello
bounceIn
bounceInDown
bounceInLeft
bounceInRight
bounceInUp
bounceOut
bounceOutDown
bounceOutLeft
bounceOutRight
bounceOutUp
fadeIn
fadeInDown
fadeInDownBig
fadeInLeft
fadeInLeftBig
fadeInRight
fadeInRightBig
fadeInUp
fadeInUpBig
fadeOut
fadeOutDown
fadeOutDownBig
fadeOutLeft
fadeOutLeftBig
fadeOutRight
fadeOutRightBig
fadeOutUp
fadeOutUpBig
flipInX
flipInY
flipOutX
flipOutY
lightSpeedIn
lightSpeedOut
rotateIn
rotateInDownLeft
rotateInDownRight
rotateInUpLeft
rotateInUpRight
rotateOut
rotateOutDownLeft
rotateOutDownRight
rotateOutUpLeft
rotateOutUpRight
hinge
jackInTheBox
rollIn
rollOut
zoomIn
zoomInDown
zoomInLeft
zoomInRight
zoomInUp
zoomOut
zoomOutDown
zoomOutLeft
zoomOutRight
zoomOutUp
slideInDown
slideInLeft
slideInRight
slideInUp
slideOutDown
slideOutLeft
slideOutRight
slideOutUp

Animate.css example

CSShake

- csshake.css
- CSS Animation library to shake your HTML elements
- More documentation here: https://github.com/elrumordelaluz/csshake
- Download csshake.css file, link it in your HTML page and animate!

csshake.css example

CSS Animations Resources

- https://www.w3schools.com/css/css3_animations.asp
- https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_Animations/Using_CSS_animations
- https://css-tricks.com/almanac/properties/a/animation/
- https://www.sitepoint.com/our-top-9-animation-libraries/

Looking forward...

Congrats!

Though it may feel like all this new terminology is just starting to sink in, you have the basic skills to be a web designer.

...but nowadays, web design doesn't start and stop with HTML and CSS.

- Need to make your site mobile-compatible!
- Need to make it cross-platform!
- Need to get worry about hosting!
- Need some JavaScript code to make it responsive to all situations, and to finish off the forms we began making in HTML!

What else should I know?

If you want to continue understanding web design, you'll need to be able to at least recognize and know how the following works:

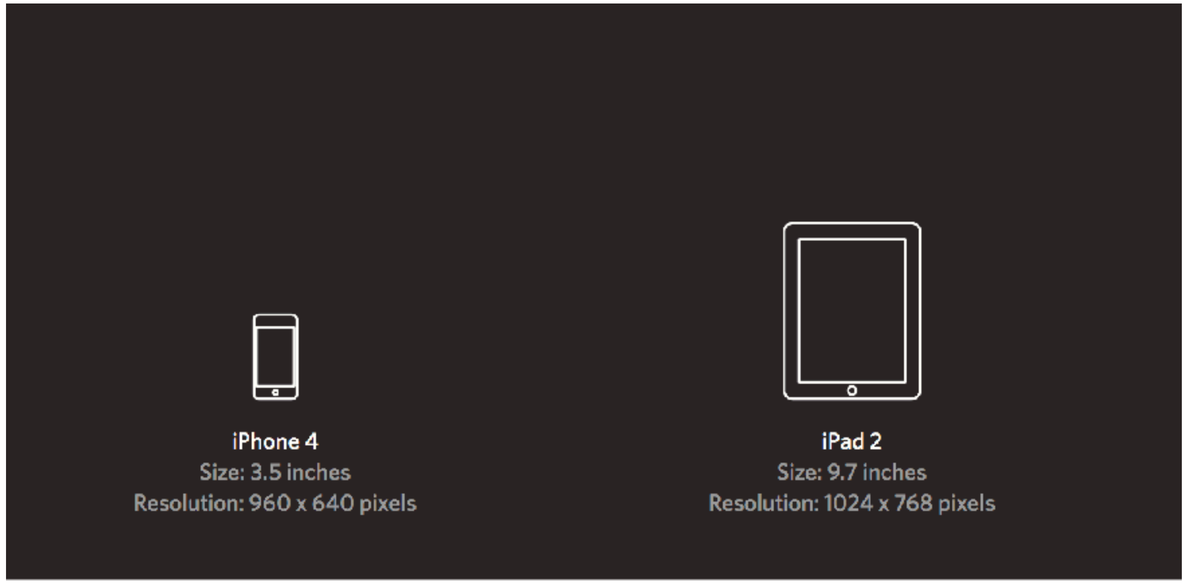
- JavaScript
- jQuery
- Bootstrap
- Web hosting solutions
- Basic server side recognition

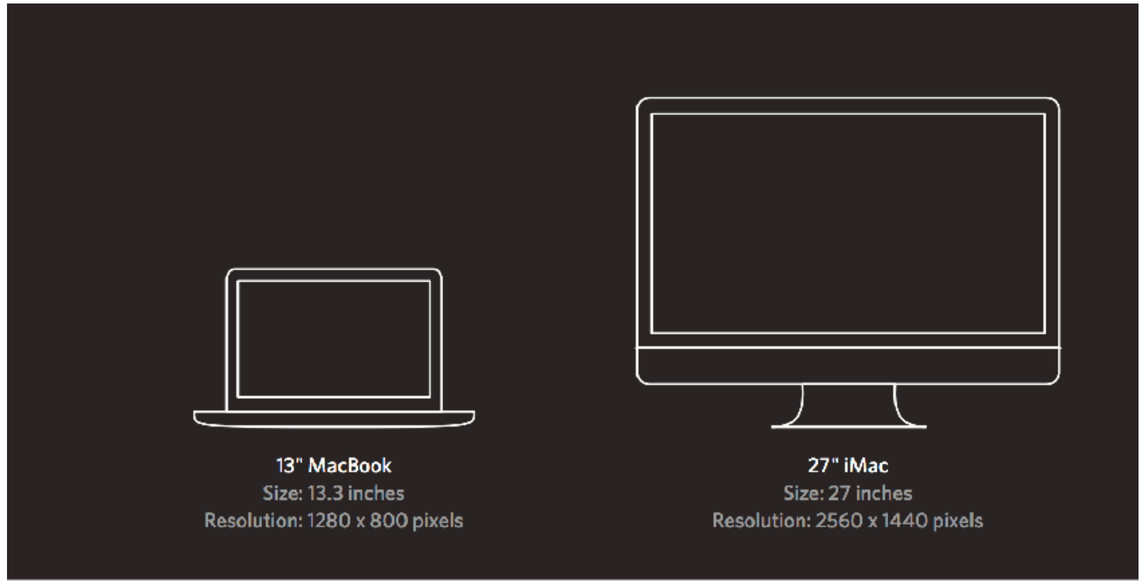
...and a bunch of other stuff if you ever want to work in web development...

Intro to Responsive Design

What is responsive design?

- It makes your website look good on all the devices
- It can be done using only HTML and CSS, without JavaScript





What is responsive design?

- It adapts information
  to fit any device

Desktop

Tablet

Phone

The Viewport

- The viewport is the user's visible area of a web page (remember the “above the fold”?)
- You can set it in the <meta> tag in the <head> section of an HTML document
  - *<meta>* gives the browser instructions on how to control the page's dimensions and scaling
  - *width=device-width* sets the width of the page to follow the screen-width of the device (varies)
  - *initial-scale=1.0* sets the initial zoom level when the page is first loaded by the browser

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

Example 1

Example 2

Grids

- We can also build responsive grids
- A responsive grid-view often has 12 columns, and has a total width of 100%, and will shrink and expand as you resize the browser window
- Notice using of “%” is important, so it’s not a fixed layout

Building a responsive grid

1. border-sizing should be set to border-box
2. Entire width of the grid equals to 100%
3. One column equals to 8.33% (100 divided by 12)
4. All columns should float left
5. All rows should be wrapped in a

Example 1

Example 2

Media Query

1. Uses the @media rule to include a block of CSS properties only if a certain condition is true
2. For example, change the background color to blue if the screen is smaller than 400px:

```
@media only screen and (max-width: 400px) {
  body {
    background-color: lightblue;
  }
}
```

Example

Adding breakpoints with media queries

1. Using media queries we can add breakpoints when the width of columns should change (for example, become 100% when browser width becomes small)
2. You can add as many breakpoints as you want (if you want to add tablets, etc.) - then you'd have to add multiple classes to the same HTML element, so it knows what to do at each breakpoint

```
@media only screen and (max-width: 768px) {
   [class*="col-"] {
      width: 100%;
   }
}
```

Example

Mobile first

1. Mobile first means designing for mobile devices first and only later for larger screens

```
@media only screen and (min-width: 768px) {
    .col-1 {width: 8.33%;}
    .col-2 {width: 16.66%;}
    .col-3 {width: 25%;}
    .col-4 {width: 33.33%;}
    .col-5 {width: 41.66%;}
    .col-6 {width: 50%;}
    .col-7 {width: 58.33%;}
    .col-8 {width: 66.66%;}
    .col-9 {width: 75%;}
    .col-10 {width: 83.33%;}
    .col-11 {width: 91.66%;}
    .col-12 {width: 100%;}
    }
```

Example

Bootstrap

Bootstrap

- Bootstrap is the main piece you should aim to get to with your web development
- It's also just a library like jQuery or 960.gs, but it uses jQuery/JavaScript and CSS to make websites work for mobile without major issues
- You can do everything Bootstrap does yourself, but Bootstrap makes it easier to think about responsive and mobile first sites
- You need to have the jQuery code linked in your website
- Their website is pretty good about simple templates
- You'll need to start using meta tags, and making sure all CSS and JS files are linked to appropriately!!!

To use Bootstrap

- You can download it from getbootstrap.com
- Or link it from CDN (must include jquery as well): https://cdnjs.com/libraries/

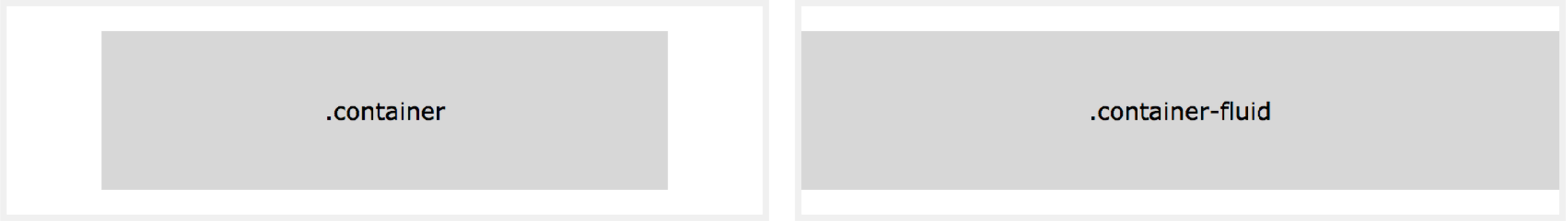
```
<!-- Latest compiled and minified CSS -->
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/
bootstrap/3.3.7/css/bootstrap.min.css">

<!-- jQuery library -->
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/
jquery.min.js"></script>

<!-- Latest compiled JavaScript -->
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/
bootstrap.min.js"></script>
```

Creating Bootstrap website

- Do not forget !doctype at the top of HTML page
- Also lang="en" in the opening <html> tag & charset="utf-8" in the <meta> tag in <head> section
- Set a viewport:
  - `<meta name="viewport" content="width=device-width, initial-scale=1">`

Choosing Bootstrap container

- You have two container classes to choose from:
  - .container - a fixed width container
  - .container-fluid - full width container that spans the entire width of the viewport



Creating Bootstrap grid system

- Bootstrap’s grid system is responsive and will rearrange automatically depending on a screen size
- Grid classes:
  - `xs` - for phones, screens less than 768 px
  - `sm` - for tables, >= 768px
  - `md` - for small laptops, >= 992px
  - `lg` - for laptops and desktops, >= 1200px
- Classes can be combined to create more dynamic and flexible layouts
- Columns in the same row should be wrapped by with a class “row”
- Columns in the same row should add up to 12

Bootstrap grid example

Homework assignment

Homework

By Wednesday, November 8, 6pm

1. **Go through responsive design tutorials:**
    1. Codecademy class: https://www.codecademy.com/learn/learn-responsive-design
    2. Bootstrap documentation:
        1. http://getbootstrap.com/docs/4.0/getting-started/introduction/
        2. http://getbootstrap.com/docs/4.0/examples/grid/
        3. https://www.w3schools.com/bootstrap/
    3. Bootstrap tutorial:
        1. https://www.youtube.com/watch?v=e5VQampkqdg
2. **Create your first website using responsive Bootstrap grid:**
    - Make sure libraries are linked properly
    - Do not forget to add columns to separate under class="row"
    - You can add any content (text or images), any column number, whatever you want, just make sure it makes sense
    - Upload your code to a new folder on Github and post two links on #general channel on Slack:
        - Link to your project's repository (folder)
        - Working link to the Github page (yourusername.github.io/pathtofolder)