



WELCOME TO FRONT-END BOOTCAMP!



INTRODUCTION

JAMES YORK

james@grandcircus.co | @kroysemaj | kroysemaj

JAMES YORK

Work

- OnStar, Nexient, & DTE

Education

- Baker College of Allen Park
- Many expensive degrees which I am now paying off
:(

Community

- Detroit Craftsman Guild (co-organizer)
- Girl Develop It (Instructor, TA)
- Conference speaker (self.conference, Code Mash 2015)

TAX DEDUCTIONS





INTRODUCTIONS (NOW YOU)

- Who are you?
- Where are you from?
- What's your favorite movie?

COURSE EXPECTATIONS

What I expect from you...



GRAND
CIRCUS
DETROIT



GRAND
CIRCUS
DETROIT



BE PRESENT

Listen, soak up the information (there's a lot of it),
process it.

ASK QUESTIONS

I'm not big on formality. Feel free to ask any question that comes to mind. I'll also stop frequently for



questions.

SUPPORT EACH OTHER

I'm a big proponent of pair programming. Any in-class exercise or project work may be paired and certain labs *will* be paired or in groups. Larger final projects will be group projects.

GRAND
CIRCUS
DETROIT

HAVE FUN

See previous comment on my stance on formality and ceremony. Bored people learn nothing. I'll do what I can to make this fun as well as instructive.

MY GOALS FOR YOU

MY GOALS FOR YOU

Inculcate you with vital tools such as version control, specifically git and github.





GRAND
CIRCUS
DETROIT



MY GOALS FOR YOU

Baptise you in the fires of JavaScript.



GRAND
CIRCUS
DETROIT



GR
CIR



GRAND
CIRCUS
DETROIT



MY GOALS FOR YOU

Learn the basics of web and mobile architecture by seeing it in practice.

COURSE OUTLINE

(What are we doing here anyway?)

'THE PLAN'

- Week 1 - HTML / CSS
- Week 2 - JavaScript
- Week 3 - jQuery / AJAX / JSON
- Week 4 - Angular JS / TDD
- Week 5 - Node JS / Stylus / Gulp / Express JS / APIs / Heroku / Node Project
- Week 6 - Advanced JS
- Week 7 - Project
- Week 8 - Project / Demo Day

KEYS TO SUCCESS



KEYS TO SUCCESS

As of right now, your primary job description is:

FIGURE IT OUT

KEYS TO SUCCESS

Developers are essentially well-paid problem solvers.
If you come across something you're not sure about
you need to make an effort to solve the issue
yourself.

KEYS TO SUCCESS

some 'rules' for when you get stuck:

1. First, ask google
2. Second, ask a fellow bootcamper
3. Third, ask another fellow bootcamper
4. If you're still stumped ask me

KEYS TO SUCCESS

The rule about 'the rules'

That said, don't be a hero and spin your wheels for hours without progress. It profits no one. If you're genuinely stuck, ask for help and keep asking until you get it.



KEYS TO SUCCESS

Most importantly

PRACTICE!!

We won't have a lot of 'homework' for this class but any extra time you spend workin on this stuff will only deepen your understanding of the material. If you finish an assignment early, look for ways to go a step further or do some extra work on the Final Project.

GETTING STARTED

Note that some of this should already be done. This is mostly a checklist of things we will need for smooth(ish) sailing in this course.

GET CONNECTED

wifi: Madison

pw: Bro@dw@y#

GRAND
SIRIUS

GRAND
CIRCUS
DETROIT

TRELLO

Trello.com

Make yourself a Trello profile.

Trello is a web-based organizational tool. It can be used for lots of different things but we will be using it as our card wall for the final project for this class.

Once you have your Trello and GitHub user names, post them in the Slack channel.

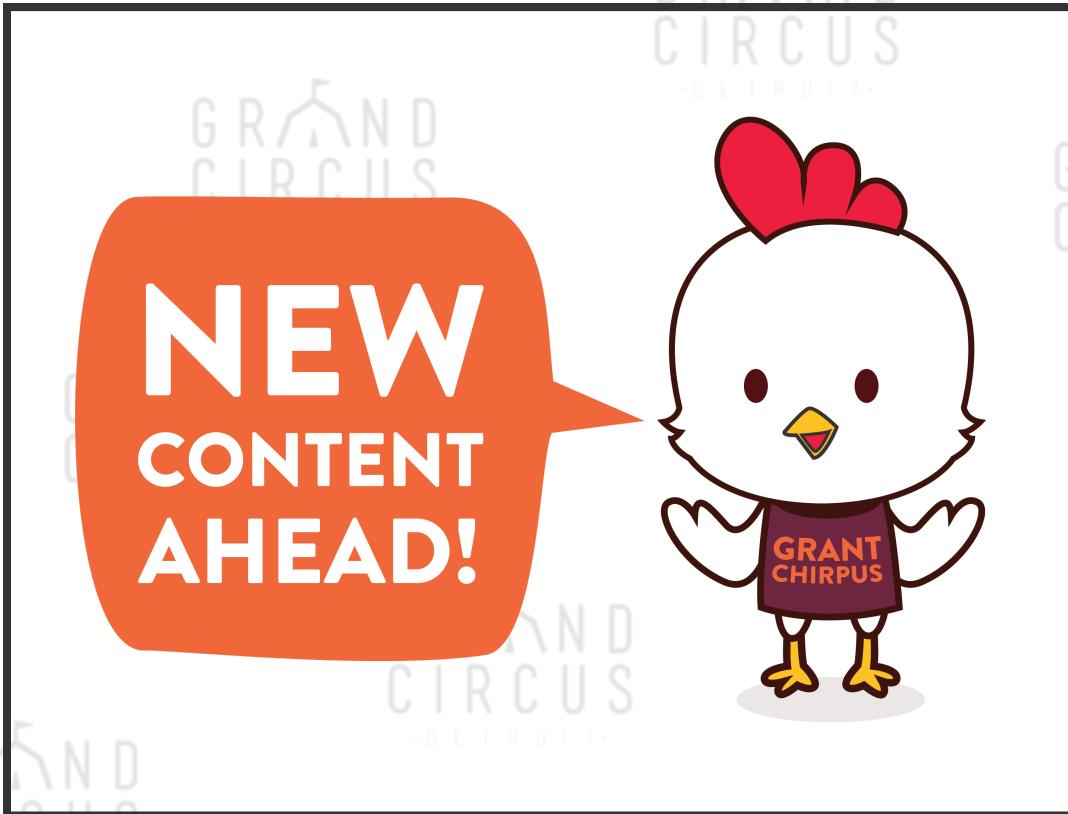
QUESTIONS?

CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT



GOALS FOR THIS UNIT

1. The Website as Communication
2. Introduction to HTML & CSS
3. HTML 101
4. **Exercise:** Build a static HTML site from scratch

ASSOCIATED READING

From HTML & CSS:

- Introduction: 2-11
- Chapter 1: 12-39
- Chapter 2: 40-61
- Chapter 3: 62-73
- Chapter 4: 74-93
- Chapter 5: 94-119
- Chapter 6: 126-143
- Chapter 8: 176-199

Every website on the internet uses HTML & CSS.





CIRCUS
DETROIT



CTR
DET



...most of them use JavaScript as well, in one form or another.

GRAND
CIRCUS
DETROIT

DEMO!

(right click in the browser and select **Inspect**
Element)

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS

GRAND
CIRCUS

The screenshot shows a browser's developer tools with the "Elements" tab selected. The left pane displays the DOM tree for a slide presentation. The right pane shows the "Styles" panel, which lists the CSS rules applied to the selected element. The main content area of the browser shows a slide with a purple background and white text.

```
<!DOCTYPE html>
<html lang="en">
  <head></head>
  <body style="transition: -webkit-transform 0.8s ease; -webkit-transition: -webkit-transform 0.8s ease;">
    <!-- Any section element inside of this container is displayed as a slide -->
    <div class="reveal convex center slide" role="application" data-transition-speed="default" data-background-transition="fade">
      <!-- END SLIDES DIV -->
      <div class="backgrounds"></div>
      <div class="progress" style="display: block;"></div>
      <aside class="controls" style="display: block;"></aside>
      <div class="slide-number"></div>
      <div class="pause-overlay"></div>
      <div id="aria-status-div" aria-live="polite" aria-atomic="true" style="position: absolute; height: 1px; width: 1px; overflow: hidden; clip: rect(1px 1px 1px 1px);">DEMO!
        (right click in the browser and select Inspect Element)
      </div>
      <::after
      </div>
      <script src="lib/js/head.min.js"></script>
      <script src="js/reveal.js"></script>
    </script>
    // Full list of configuration options available at:
    // https://github.com/hakimel/reveal.js#configuration
    Reveal.initialize({
      controls: true,
      progress: true,
      history: true,
      center: true,
      transition: 'slide', // none/fade/slide/convex/concave/zoom
      // Optional reveal.js plugins
      dependencies: [
        { src: 'lib/js/classList.js', condition: function() { return !document.body.classList; } },
        { src: 'plugin/markdown/markdown.js', condition: function() { return !!document.querySelector(' [data-markdown]'); } },
        { src: 'plugin/markdown/markdown.js', condition: function() { return !!document.querySelector(' [data-markdown]'); } },
        { src: 'plugin/highlight/highlight.js', condition: function() { return !!document.querySelector(' pre code' ); }, callback: function() { hljs.initHighlightOnLoad(); } },
        { src: 'plugin/zoom-js/zoom.js', async: true, condition: function() { return !!document.querySelector(' .reveal .zoom' ); } },
        { src: 'plugin/notes/notes.js', async: true }
      ]
    });

    </script>
    <script type="text/javascript" src="plugin/markdown/markdown.js"></script>
    <script type="text/javascript" src="plugin/markdown/markdown.js"></script>
    <script type="text/javascript" src="plugin/zoom-js/zoom.js"></script>
    <script type="text/javascript" src="plugin/notes/notes.js"></script>
  </body>
</html>
```

html body script

Styles Computed Event Listeners »

element.style { }
script { user agent stylesheet
display: none;
} Inherited from body
body { reveal.css:30
position: relative;
line-height: 1;
background-color: #ffff;
color: #0000;
}
html, body, .reveal div, reveal.css:11
.reveal span, .reveal applet, .reveal
object, .reveal iframe, .reveal h1, .reveal
h2, .reveal h3, .reveal h4, .reveal
h5, .reveal h6, .reveal p, .reveal blockquote,
.reveal pre, .reveal ul, .reveal ol,
.reveal acronym, .reveal address, .reveal
b, .reveal cite, .reveal code, .reveal
del, .reveal dfn, .reveal em, .reveal img,
.reveal ins, .reveal kbd, .reveal q,
.reveal s, .reveal samp, .reveal small,
.reveal strike, .reveal strong, .reveal
sub, .reveal sup, .reveal tt, .reveal var,
.reveal b, .reveal u, .reveal center,
.reveal dl, .reveal dt, .reveal dd, .reveal
ol, .reveal ul, .reveal li, .reveal
fieldset, .reveal form, .reveal label,
.reveal legend, .reveal table, .reveal
caption, .reveal tbody, .reveal tfoot,
.reveal thead, .reveal tr, .reveal th,
.reveal td, .reveal article, .reveal aside,
.reveal canvas, .reveal details, .reveal
embed, .reveal figcaption, .reveal figure,
.reveal footer, .reveal header, .reveal
hgroup, .reveal menu, .reveal nav, .reveal
output, .reveal ruby, .reveal section,
.reveal summary, .reveal time, .reveal
mark, .reveal audio, video {
margin: 0;
padding: 0;
border: 0;
font-size: 100%;
font: inherit;
vertical-align: baseline;
}
Inherited from html
html, body, .reveal div, reveal.css:11
.reveal span, .reveal applet, .reveal
object, .reveal iframe, .reveal h1, .reveal
h2, .reveal h3, .reveal h4, .reveal h5,
Find in Styles

LAK GRAND CIRCUS DETROIT



TERMINOLOGY

Some key terms or phrases that are used as a matter of course in the software industry. You may know some or all of these, or you may have heard the terms but be unclear about their actual meaning.

They are common jargon among developers

WEB DEVELOPMENT

Web development is a broad term for the work involved in developing a web site for the Internet. In industry parlance, 'web development' usually refers to the more code-related tasks such as programming JavaScript, coding HTML and CSS. It can even extend to tasks related to the back end server infrastructure such as creating web services and handling business logic for a company or product.

WEB DESIGN

The process of planning & structuring a website; specifically, the visual aspects and assets for the site.

Recently, this job description has also begun to include interaction design. That is, designing the user experience (UX), information architecture, and the flow of the application or site.



WEB SITE

A largely informational web page. While they may include dynamic elements and react to user inputs.

The general purpose of a web site is to provide information about a person, business, product, or service.

WEB APPLICATION

A more recent term to indicate a web site whose sole purpose is not just informational, but rather functional. Web applications have become robust enough to do everything from our taxes, manage our personal calendars, or even do standard desktop publishing tasks.

FRONT END

The visible and interactive parts of a website or application.

BACK END

The 'invisible' or inner functionality of a website or application. Examples include costly calculations, interacting with a database or making use of web service end points. While we're at it...

WEB SERVICE

A software function provided at a network address over the Web. The W3C defines a Web service generally as:

"A software system designed to support interoperable machine-to-machine interaction over a network."



DATABASE

A software system for storing data long term. This is also sometimes referred to as a 'persistance layer'.

VERSION CONTROL

A software tool for managing changes to a set of files, website, application or any collection of files and for reconciling the differences between those files when conflicts emerge.

APPLICATION PROGRAMMING INTERFACE (API)

An API is a set of routines, protocols, and tools for building software applications. An API expresses a software component in terms of its operations, inputs, outputs, and underlying types. An API defines functionalities that are independent of their respective implementations, *which allows definitions and implementations to vary without compromising each other.*



QUESTIONS?



GRAND
CIRCUS
DETROIT

IT'S ABOUT



GRAND
CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

HYPertext Markup Language

- Developed by Tim Berners-Lee
- Developed in 1989 at CERN
- Originally developed as a way to share documents



HTML STRUCTURE



```
<html>
  <head>
    <title>My First Website!</title>
  </head>
  <body>

    <p>Look at the amazing content of my awesome website.</p>

  </body>
</html>
```

-DETROIT-

GRAND
CIRCUS
-DETROIT-

-DETROIT-

GRAND
CIRCUS
-DETROIT-

CSS PRESENTATION

GRAND
CIRCUS
-DETROIT-



```
body {  
background-color: white;  
font-size: 18px;  
color: blue;  
}  
  
p {  
text-align: center;  
font-weight: bold;  
}  
  
img {  
border: 1px solid black;  
padding: 50px;  
}
```





GRAND
CIRCUS
DETROIT



GRAND
CIRCUS
DETROIT



HTML 101

GRAND
CIRCUS
DETROIT



GRAND
CIRCUS
DETROIT



GRAND
CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT



GRAND
CIRCUS
DETROIT



HTML

HYPertext Markup Language



HTML ELEMENTS

An element is an individual component of HTML



HTML TAGS

A tag marks the beginning and end of an element.

Tags are containers. They tell you something about the content between the opening & closing tags.



ANATOMY OF AN HTML ELEMENT

OPENING TAG

< p >

Character
Left-Angle Bracket (LESS-THAN SIGN) Right-Angle Bracket (GREATER-THAN SIGN)

CLOSING TAG

< /p >

Forward Slash Character
Left-Angle Bracket (LESS-THAN SIGN) Right-Angle Bracket (GREATER-THAN SIGN)

EXAMPLE

This is a paragraph

A paragraph element consists of an opening `<p>` tag and a closing `</p>` tag and the content between the tags.

DIFFERENT KINDS OF ELEMENTS

Elements can be either container elements (they hold content) or stand-alone elements, sometimes called self-closing elements.

DIFFERENT KINDS OF ELEMENTS

Paragraph elements are containers

```
<p>Hi, I contain content</p>
```

Image elements are stand-alone

```

```

ATTRIBUTES

1. Provide additional information about HTML elements
2. Attribute tags include class, id, style, language, and source
3. Attributes are positioned inside the opening tag, before the right bracket
4. Attributes are paired with values. Key / value pairs are an important concept in programming.
5. Selected from a pre-defined set of possible attributes depending on the element.

VALUES

1. Values are assigned to attributes
2. Values must be contained inside quotation marks

GRAND
CIRCUS
DETROITGRAND
CIRCUS
DETROITGRAND
CIRCUS
DETROIT

EXAMPLE ATTRIBUTES

```
<p id="trademark">An inline element</p>
```

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

EXAMPLE ATTRIBUTES

```
<div class="container">  
A bunch of stuff!  
</div>
```

DOCTYPE

The first element on every HTML page. It tells the browser to expect HTML and what version to use.

HTML 5

```
<!doctype html>
```

HTML 4

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01  
Transitional//EN"  
"http://www.w3.org/TR/html4/loose.dtd" >
```

HTML TAG

After the doctype, all page content must be contained in the `<html>` tags



CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

MAIN HTML TAGS

Tag	Description
<code>head</code>	contains the page title and metadata
<code>title</code>	is the name of your page
<code>body</code>	contains all of the visible content

NESTING

HTML elements 'nest' inside of one another. The element that opens first closes last.



CIRCUS
DETROIT



GRAND
CIRCUS
DETROIT



CIR
DET

CONTENT TAGS

COMMON CONTENT TAGS

Tag

Description

`div`

defacto container element

`p`

used for body copy

`h1 thru h6`

designating titles/subtitles

`ol`

create a numbered list

`ul`

create an unordered list

`li`

list elements

CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

LINKS

ANCHOR ELEMENT

Links to other sites on the web (or within your project) are created using this element.

```
<a href="http://facebook.com">Facebook</a>
```

LINK ELEMENT

Unlike the anchor element. The `<link>` specifies relationships between the current document and an external resource.

Most often this manifests as how CSS files are included with an HTML file.

```
<link src="main.css" rel='stylesheet' />
```

HTML COMMENTS

Like any other good coding language, HTML offers comments. They operate like comments in any other language. They are ignored by the browser engine.

```
<!-- Hello, I am a comment. -->
```

CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

GRAND

GRAND

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

TABLE ELEMENT

Tables are a way to represent complex information in a grid format. They are made of rows and columns.

Tables are compound elements (similar to lists) they are made up of several elements.

Element Description

`<table>` Table element

`<tr>` Table row

`<td>` Table cell

`<th>` Table header cell (optional)

TABLE EXAMPLE

```
<table>
  <tr>
    <th>Business</th>
    <th>Numbers</th>
    <th>Synergy</th>
  </tr>
  <tr>
    <td>Cell 1</td>
    <td>Cell 2</td>
    <td>Cell 3</td>
  </tr>
  <tr>
    <td>Cell 4</td>
    <td>Cell 5</td>
    <td>Cell 6</td>
  </tr>
</table>
```

Business Numbers Synergy

Cell 1	Cell 2	Cell 3
Cell 4	Cell 5	Cell 6

STYLED TABLES



Id	Name	Email	Investments	Action
261	Alfred Alan	aalan@gmail.com	Stocks	
227	Alison Smart	asmart@biztalk.com	Residential Property	
246	Ally Emery	allye@easymail.com	Stocks	
212	Andrew Phips	andyp@mycorp.com	Stocks	
218	Andy Mitchel	andym@hotmail.com	Stocks	
221	Ann Melan	ann_melan@iinet.com	Residential Property	
243	Ben Bessel	benb@hotmail.com	Stocks	
232	Bensen Romanolf	benr@albert.net	Bonds	
233	Brad Cole	bradc@hotmail.com	Stocks	
241	Catherine Benchman	cathb@hotmail.com	Stocks	

74 items found, displaying 1 to 10.
[First/Prev] [1](#), [2](#), [3](#), [4](#), [5](#), [6](#), [7](#), [8](#) [Next/Last]

Table caption					
Header	Header	Header	Header	Header	Header
sample data	sample data	sample data	sample data	sample data	sample data
sample data	sample data	sample data	sample data	sample data	sample data
sample data	sample data	sample data	sample data	sample data	sample data
sample data	sample data	sample data	sample data	sample data	sample data
sample data	sample data	sample data	sample data	sample data	sample data
footer	footer	footer	footer	footer	footer



A NOTE ON TABLES

You may be thinking it now or you may think later that tables would be a great way to position content for site...

-DETROIT-

-DETROIT-



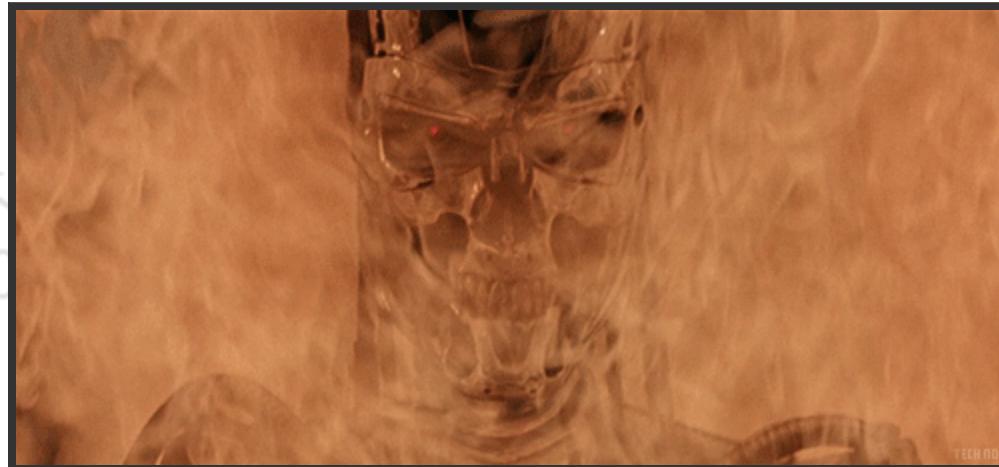
Don't do that. Just... don't. Trust me on this. It's a bad idea.



GRAND
CIRCUS

-DETROIT-

(srsly, don't)



GRAND
CIRCUS

-DETROIT-

GRAND

GRAND
CIRCUS

-DETROIT-

GR
CIR

-DET

GRAND
CIRCUS

-DETROIT-

WHAT'S WRONG WITH THIS CODE?

Look at the following examples and tell me what is wrong with the code.

GRAND
CIRCUS
DETROIT

```
<html>
  <head>
    <body>
      </head>

      </body>
    </html>
```

GRAND
CIRCUS

GRAND
CIRCUS
DETROIT

GR
CIR
DET

```
<html>
  <head>
    <title>The Best Site Evar!!
  </head>
  <body>

    <p>Check out this riveting content!</p>

  </body>
</html>
```

```
<p style=hotStuff>Check out this riveting content!</p>
```

-DETROIT-

GRAND CIRCUS DETROIT

- D E T R O I T -



DETROIT



FOLDER STRUCTURE

This stuff is not exciting but it's *important*.



RECAP

You should understand and be able to use:

- HTML elements
- Proper nesting
- HTML Comments
- Correct folder structure

GRAND
CIRCUS
DETROIT

LAB 1

BUILD ANOTHER WEBSITE

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS

SLIGHTLY MORE COMPLEX WEBSITE CONSTRUCTION

In a new project folder, create another basic site using only HTML from scratch.

Your site should contain the following elements:

- Basics: Doctype, HTML, Head, & Body
- One or more division tags for organization
- At least 3 headings of different size
- At least 5 paragraphs
- One of each list type (ordered and unordered)
- At least 3 links
- At least 2 images
- some HTML comments
- 1 table



-DETROIT-



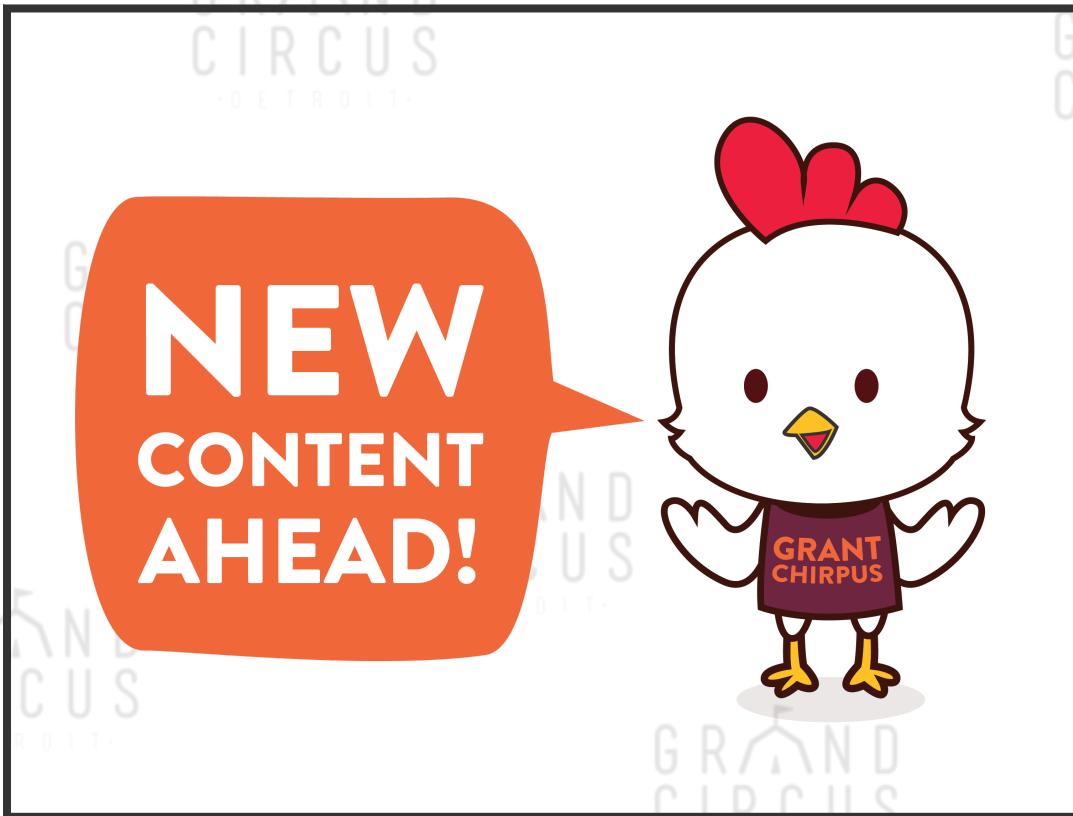
BONUS!

Use some combination of elements such as:

- A list of links
- An image that serves as a hyperlink



BETROIT



GR
CIR



CIRCUS
DETROIT



WELCOME BACK!



GRA
CIR

CTR
DET

CIRCUS
DETROIT

CIRCUS
DETROIT

CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

GR
CIR
DET

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS

QUESTIONS?

GOALS FOR THIS UNIT

1. Review HTML
2. CSS 101
3. Inline vs Block Elements
4. HTML + CSS

ASSOCIATED READING

From HTML & CSS:

- Chapter 10: 226-245
- Chapter 11: 246-253
- Chapter 12: 264-287
- Chapter 13: 300-329
- Chapter 14: 330-340
- Chapter 15: 358-376
- Chapter 16: 406-418

REVIEW

GRAND
CIRCUS
DETROIT

CSS 101

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT



CIRCUS
DETROIT



CASCADING STYLE SHEETS

(remember that 'cascading' part)

CSS IN THE WILD

```
.reveal.page .slides section {  
padding: 30px;  
min-height: 700px;  
-moz-box-sizing: border-box;  
box-sizing: border-box; }  
  
.reveal.page .slides section.past {  
z-index: 12; }  
  
.reveal.page .slides section:not(.stack):before {  
content: '';  
position: absolute;  
display: block;  
width: 100%;  
height: 100%;  
left: 0;  
top: 0;  
background: rgba(0, 0, 0, 0.1);  
-webkit-transform: translateZ(-20px);  
transform: translateZ(-20px); }
```

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

GR
CIR
DET

CSS RULES

Individual components of CSS are called rules

GRAND
CIRCUS
DETROIT

GR
CIR
DET

GRAND
CIRCUS

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS

CSS RULES

CSS rules are made up of two parts:

- One or more selectors
- One or more declarations
- The declaration must be inside curly braces that follows the selector

CSS SYNTAX

Anatomy of a CSS Rule

```
h1 { color: orange; text-align: center; }
```

Selector Property Value

Declaration = Property + Value



CSS COMMENTS

Just like HTML, CSS offers comments:

```
/* This is a CSS comment,  
it can be multi-line */
```



CSS DECLARATIONS

Declarations are made up of the property and value of the style you want to apply.

They can be grouped together so that more than one declaration may be applied to a selected element.

Declaration groups must be surrounded by curly brackets.

Declarations must end in a semicolon.

```
.selector {  
    background-color: red;  
    color: white;  
    border: 1px solid black;  
    border-radius: 5px;  
}
```

CSS SELECTOR

The selector instructs the browser to search the page for any HTML element that matches the given criteria. It applies any applicable declarations to that element.

CSS SELECTOR - ELEMENT

Elements can be selected by their element name. In this case, all elements of that element type will be selected and have the styles applied.

CSS SELECTOR - ELEMENT

```
p {  
  position: absolute;  
  top: 0px;  
  left: -100px;  
}
```

CSS SELECTOR - CLASS

Elements can be selected based on HTML attributes such as class. In this case all elements that have a matching class attribute will be selected.

CSS SELECTOR - CLASS

```
.timer {  
    position: absolute;  
    top: 0px;  
    left: -100px;  
}
```



CSS SELECTOR - ID

Elements can also be selected based on HTML attribute ID. In this case only one element would be selected, as HTML IDs are intended to be unique.





CSS SELECTOR - ID

```
#fluffy {
    position: absolute;
    top: 0px;
    left: -100px;
}
```

CSS SELECTOR - CHILD SELECTORS

Selectors can be combined to become more specific.
This example selects searches for any paragraph tag
that is nested inside a div tag.

```
div p {  
  position: absolute;  
  top: 0px;  
  left: -100px;  
}
```

CSS SELECTOR - MULTIPLE

In addition a set of declarations can be applied to more than one selector by listing a number of comma-separated selectors.

```
.timer, img, div p, #kitty {  
    position: absolute;  
    top: 0px;  
    left: -100px;  
}
```

POP QUIZ HOT SHOT

Look at the following examples and tell me which Elements (if any) would be returned by the following selectors.

GRAND
CIRCUS
DETROIT

POP QUIZ

```
p {  
/* blah blah */  
}
```



-DETROIT-



POP QUIZ

```
#fuzzy {  
/* blah blah */  
}
```





POP QUIZ

```
.blue {  
/* blah blah */  
}
```





POP QUIZ



```
.blue, #fuzzy {  
/* blah blah */  
}
```





POP QUIZ

```
div p {  
/* blah blah */  
}
```

DETROIT



DETROIT





POP QUIZ

```
body div {  
/* blah blah */  
}
```



The logo for Grand Circus Detroit, featuring a stylized 'G' and 'D' above the word 'CIRCUS' and 'DETROIT' stacked below it.

QUESTIONS?

GRAND
CIRCUS
DETROIT

DETROIT

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

CSS PROPERTIES

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS

GR
CIR

CSS PROPERTIES

There are literally hundreds of css properties that are available for use. We don't have time to go over more than just a few. We will go over a few of the most common. However, the best strategy is to google for styling options as you're working.

COMMON CSS PROPERTIES

Property

Description

`background-color`

background color for an element

`color`

color of the *text* in an element

`font-family`

typeface for text

`font-size`

size for text (px, %, em, pt)

`font-weight`

used to bold text (if possible)

`text-decoration`

used for underline (mostly)

`height`

specifies the height of an element

`width`

specifies the width of an element

COLOR IN CSS

Method	Syntax	Description
color name	<code>white</code>	a list of 140 predefined colors
hexidecimal	<code>#FF0000</code>	RGB values in hex 00 - FF (0 - 255)
RGB	<code>rgb(255, 0, 187)</code>	RGB values in decimal numbers (0 - 255)
RGBA	<code>rgba(255, 0, 187, 0.5)</code>	RGB values with an added alpha (opacity) value

FOLDER STRUCTURE

- Still not exciting
- Still important



PROJECT SETUP

Wherever you are going to keep your project work:

- Create a new folder called CSS-Exercise
- Inside that folder create two files called `index.html` & `main.css`



GRAND
CIRCUS
DETROIT



SET UP

LINKING HTML AND CSS FILES

Add this element to your HTML page's head

```
<link href="main.css" rel="stylesheet" type="text/css">
```

CSS UNITS

Method Syntax Description

em	1em	Scalable unit based on font size
----	-----	----------------------------------

pixels	16px	Fixed number of pixels
--------	------	------------------------

percent	120%	Percent value based on font size
---------	------	----------------------------------

points	16pts	Fixed unit based on number of points
--------	-------	--------------------------------------



CIRCUS
DETROIT



CTR
DET



CSS POSITIONING



GRA
CIR
-DET



GRAND
CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

BLOCK VS. INLINE ELEMENTS

BLOCK ELEMENTS

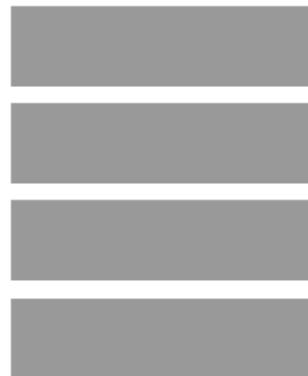
Block elements each appear on a new line of a web page, like paragraphs. Spatially, what is happening is that the block element takes up all of the horizontal space it can. It stretches to fill all the space to the left and right of the element within its parent container.

INLINE ELEMENTS

Inline elements are rendered without starting a new line. They appear side by side until reaching the edge of its parent container. Then it will start a new line.

BLOCK VS. INLINE ELEMENTS

BLOCK:



INLINE:



BLOCK VS. INLINE ELEMENTS

- *Inline element:* `<a> `
- *Block element:* `<div> <p> <table>`
...and just about everything else.



CIRCUS
DETROIT



GRAND
CIRCUS
DETROIT



THE BOX MODEL



GRAND
CIRCUS
DETROIT



GRAND
CIRCUS
DETROIT



GRAND

THE BOX MODEL

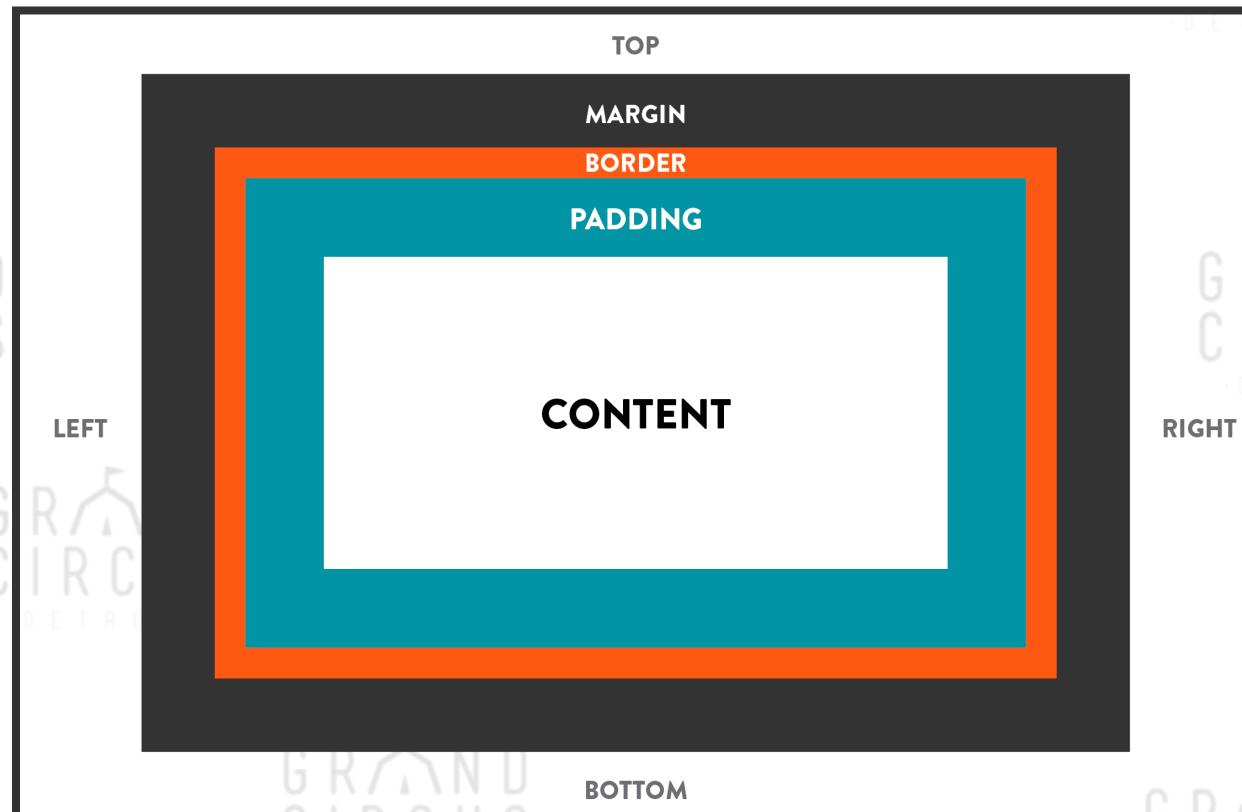
Every HTML element is in a box, regardless of its visible shape.

THE BOX MODEL

The total size of an element is a combination of the following:

- Content
- Padding
- Border
- Margin

THE BOX MODEL

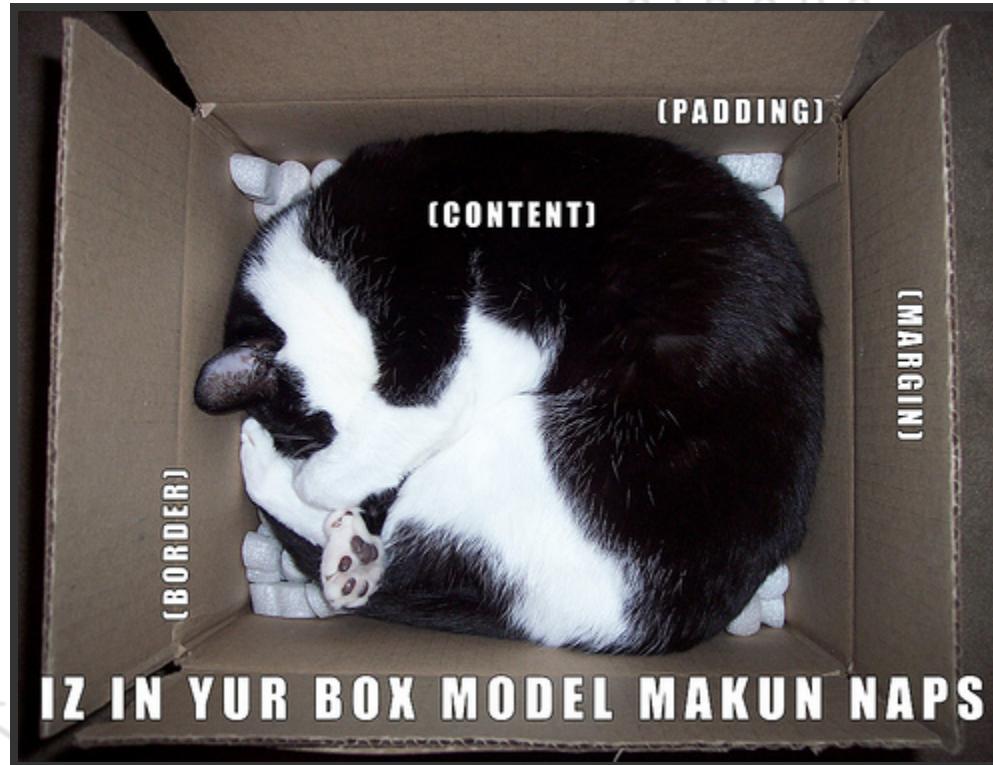


GRAND
CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

THE BOX MODEL



GRAND
CIRCUS
DETROIT

PADDING

PADDING

Padding is the space between the content within an element and the border. Padding can be specified in units of pixels (px), Ems (em), or points (pt) or percentage (%) of the containing element.

PADDING

Padding is a CSS property (just like color or font-family) and is defined using the same rules. The only difference is that there are a few different ways to write them.



GRAND
CIRCUS
DETROIT

PADDING

```
p {  
    padding: 10px; /* 10 px of padding on all sides */  
}  
p {  
    padding: 10px 5px;  
    /* 10 px on top/bottom, 5px on left/right */  
}  
p {  
    padding-top: 5px;  
    padding-right: 10px;  
    padding-bottom: 15px;  
    padding-left: 20px;  
}  
p {  
    padding: 5px 10px 15px 20px;  
    /* top, right, bottom, left */}
```



GRAND
CIRCUS
DETROIT



GRAND
CIRCUS
DETROIT



GRAND
CIRCUS
DETROIT





GRAND
CIRCUS
DETROIT



GRAND
CIRCUS
DETROIT



BORDER

GRAND
CIRCUS
DETROIT



GRAND
CIRCUS
DETROIT



GRAND
CIRCUS
DETROIT



GRAND
CIRCUS
DETROIT



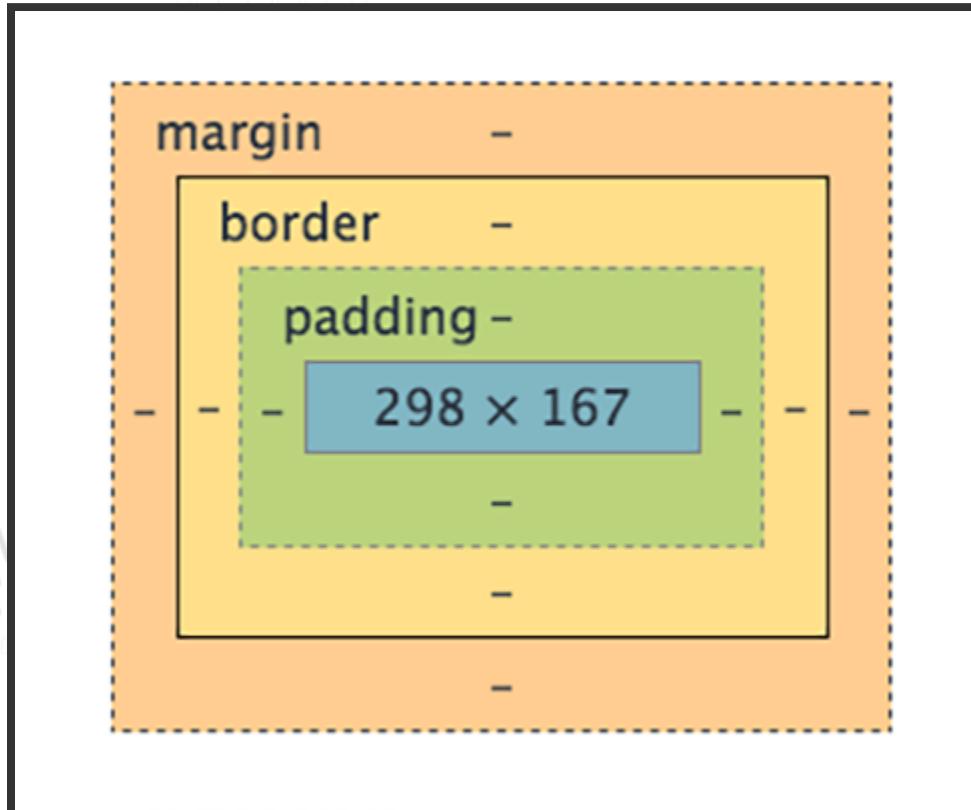
GRAND
CIRCUS
DETROIT



GRAND
CIRCUS
DETROIT



THE BOX MODEL



CIRCUS
DETROIT



BORDER

Border is the edge around the element. It has elements of thickness, style, and color.



BORDER

Like padding, border is a CSS property and is defined using CSS rules. Also like padding, there are a few different ways to specify a border.



BORDER



```
p { /* each property specified separately */
  border-width: 10px; /* thickness */
  border-style: solid;
  border-color: blue;
}
p { /* all properties specified in one rule */
  border: 1px dashed black;
}
p { /* Two separate styles on one element */
  border-top: 1px dotted #FF0000;
  border-bottom: 1px groove #FF0;
}
```



SPRING



BORDER STYLES

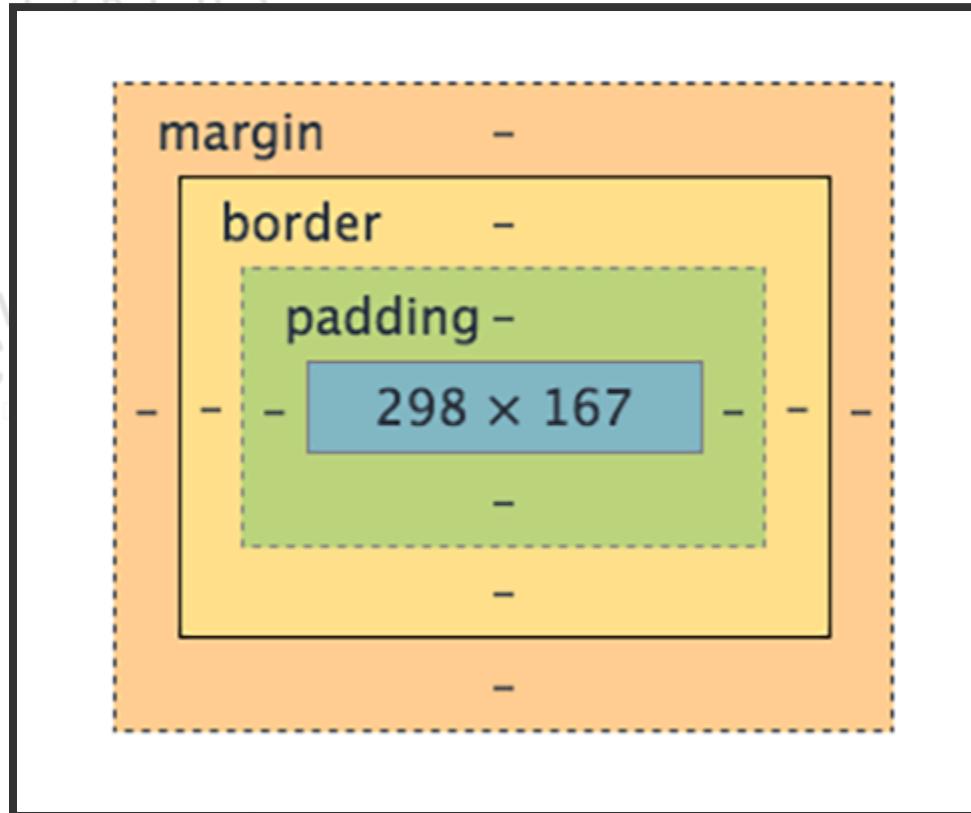
none hidden dotted dashed
solid double groove ridge
inset outset



MARGIN



THE BOX MODEL



MARGIN

Margins are the space outside the element that separates it from other elements. It is 'outside the fence' in terms of its relationship to the border.

MARGIN

Like padding & border is a CSS property and is defined using CSS rules. Much like padding, there are a few different ways to specify it.



MARGIN



```
p {  
margin: 10px; /* 10 pixels of margin on all sides */  
}  
p {  
margin: 10px 5px; /* 10 px on top/bottom, 5px on left/right */  
}  
p {  
margin-top: 5px;  
margin-right: 10px;  
margin-bottom: 15px;  
margin-left: 20px;  
}  
p {  
margin: 5px 10px 15px 20px; /* top, right, bottom, left */  
}
```



AUTO MARGIN

If margin is set to `auto` on an element that has a set width, it will take up as much space as possible. You can use this to center or left or right justify an element. This only works for horizontally centering.

AUTO MARGIN

```
p {  
margin: 0 auto;  
width: 300px;  
}  
p {  
margin-right: auto;  
margin-left: 10px;  
width: 25%;  
}
```



QUESTIONS?





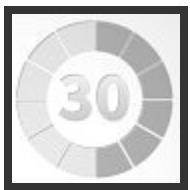
-DETROIT-



GR
CIR

EXERCISE

MESSING WITH THE BOX MODEL



BOX MODEL SHENANIGANS

- New Project (index.html, main.css)
- Create a basic HTML site
- Add 3 div elements with IDs and put a paragraph tag in each.
- Add some content
- Apply the following CSS properties
- Using the ID selectors, specify padding, border, and margin for each div
- try different border styles
- try setting some different widths and heights just to see how they're affected (if at all)

CIRCUS
DETROIT

CIRCUS
DETROIT

CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

CSS POSITIONING

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS

CSS POSITIONING

Every HTML element has a property called **position** which dictates how that element flows on a document. This property can be set to a number of different values, each of which behave slightly differently. This gives a web developer greater control over the design and overall look of a page.

CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

STATIC POSITIONING

GRAND
CIRCUS
DETROIT

STATIC POSITIONING

Static positioning is the default for all HTML elements. And it adheres to the previously discussed behavior of block and inline elements. Block elements flow from top to bottom and each element appears on a new line. Inline elements flow from left to right, wrapping as needed.



RELATIVE POSITIONING





CIRCUS
DETROIT



RELATIVE POSITIONING

Relative positioned elements appear in the normal flow of the document but can be offset by using the top, bottom, left and right properties.

GRAND
CIRCUS
DETROIT

CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

RELATIVE POSITIONING

Demo

GRAND
CIRCUS
DETROIT

CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

GR
CIR

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

ABSOLUTE POSITIONING

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

GR
CIR

GRAND

GRAND
CIRCUS
DETROIT

GRAND

ABSOLUTE POSITIONING

Absolutely positioned elements are removed from the normal flow of the document. They are not affected by and do not affect other elements in the flow, regardless of their position in the document.

ABSOLUTE POSITIONING

Absolute positioned elements appear relative to their parent container elements, using the top/bottom/right/left properties for offsets.

ABSOLUTE POSITIONING

You can think of absolutely positioned elements as being stuck onto a page with a magnet. Once placed, they don't move.

CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

ABSOLUTE POSITIONING

Demo

GRAND
CIRCUS
DETROIT

GRAND

GRAND
CIRCUS
DETROIT

GR
CIR

FIXED POSITIONING

Fixed positioned elements are very similar to absolutely positioned elements. Fixed positioned elements are also removed from the normal flow of the document. They are not affected by and do not affect other elements in the flow, regardless of their position in the document.



FIXED POSITIONING

Fixed positioned elements appear relative to the view port (the browser window), using the top/bottom/right/left properties for offsets.

FIXED POSITIONING

Fixed elements do not move when you scroll up or down the page. They seem fixed to the window.

GRAND
CIRCUS

-DETROIT-

FIXED POSITIONING

Demo

GRAND
CIRCUS
DETROIT

FLOAT

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS

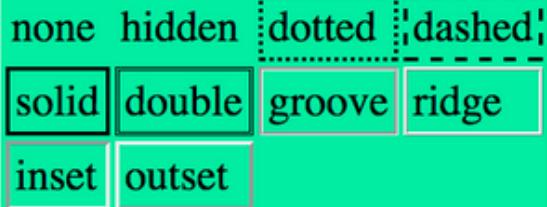
FLOAT

"Floating" takes an element in the normal flow and pushes it as far to the left or right of its parent element as possible. When an element is floated, other elements will wrap around it

FLOAT

- Always specify a width when floating an element
- Specify whether an element should float left or right
- If two or more elements are floated, they will 'stack up' on whichever side they are floated to.

FLOAT



Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum. Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.



CLEAR

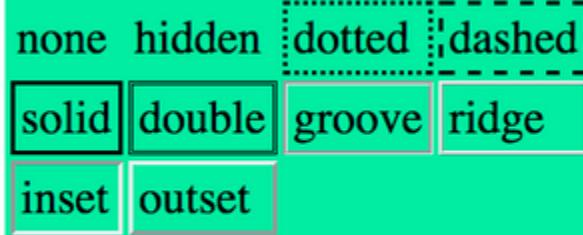




CLEAR

Clear specifies on which side of an element other elements cannot appear. You can clear left, right, or both.

FLOAT



Not cleared Item
Not cleared Item
Not cleared Item

Cleared left



Z-INDEX





Z-INDEX

When elements are moved out of the normal flow of content, they can overlap. Z-index can be used to define the order of overlapping elements. The element with the highest z-index goes on top.



GRAND
CIRCUS
DETROIT

GRAND
CIRCUS

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS

GRAND
CIRCUS
DETROIT

Z-INDEX

Demo

RECAP

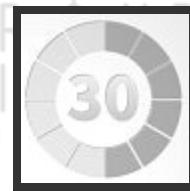
You should understand and be able to use:

- CSS Syntax
- Declarations
- Selectors
- Common Properties
- Correct Folder Structure
- Inline VS Block Elements
- Positioning Properties (Static, Relative, Absolute)
- Floats
- The Box Model



EXERCISE

CSS POSITIONING



CSS POSITIONING EXERCISE

- New Project (index.html, main.css)
- Create a basic HTML site
- Add 4 divs and add some content to each
- Float 2 of the divs so that they rest side-by-side
- In another of the divs, use relative positioning to offset some content inside the div
- In the last div, attach the content to the viewport using fixed positioning. Add enough height to the body tag to scroll up and down



CIRCUS
DETROIT



GRAND
CIRCUS
DETROIT



CTR
DET



GRAND
CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT



LAB 3

IMAGE GALLERY



GRAND
CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT



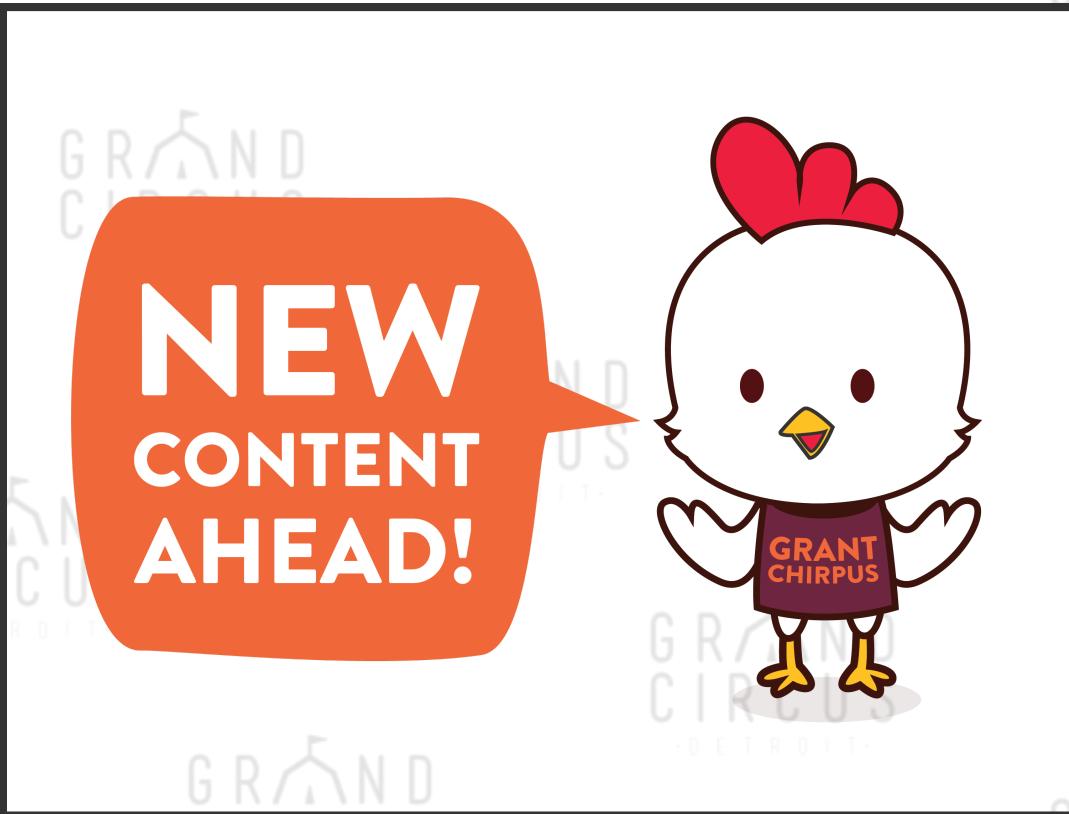
GRAND
CIRCUS



GR
CIR
DET

CSS LAB

- Make a website!
- Your website should include 3 rows of images, 3 images per row
- Each image should say something about you (hobby, vacation spot, favorite book, etc.)
- Each image should have an explanatory caption
- Each row of images should be contained in a div with padding and margin
- Each image should have a border, padding, and margin
- Your page should include a header and footer
- Use at least one absolutely positioned element
- Use at least one fixed positioned element



CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

WELCOME BACK!

GRAND
CIRCUS
DETROIT

GOALS FOR THIS UNIT

- Review
- HTML5 tags
- CSS psuedoselectors & psuedoclasses
- CSS Media Queries

ASSOCIATED READING

From HTML & CSS:

- Chapter 5: 120-125
- Chapter 7: 144-175
- Chapter 11: 254-263
- Chapter 12: 288-299
- Chapter 13: 319-329
- Chapter 14: 341-357
- Chapter 15: 377-405
- Chapter 16: 419-427
- Chapter 17: 428-451



CIRCUS
DETROIT



GRAND

GRAND
CIRCUS
DETROIT

CIRCUS
DETROIT

GRAND
CIRCUS

GRAND
CIRCUS

GRAND
CIRCUS

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

CIR
DET

HTML5

HTML5 is a new spec from the W3C that expands the language of HTML and adds a bunch of new features.

HTML5

We don't have time to go into all the details of all of HTML5's new features. Here are some highlights:

- New semantic tags
- Multimedia (new Audio and Video tags)
- Offline storage (local storage)
- 3d graphics support
- CSS3 animations and effects



HTML5

We will take a very brief look at a number of the HTML semantic tags that are available for use in your projects.

NEW HTML5 TAGS

class	Description
section	Container tag used for page organization
header	Container for introductory and navigational stuff
footer	Container for footer content (site map, internal links, etc.)
nav	Container for a major block of navigation links
audio	Multimedia tag for playing audio files
video	Multimedia tag for playing video files
canvas	element can be used to draw graphics via JavaScript



- # RESOURCES
- [HTML5 Please](#)
 - [HTML5 Rocks](#)
 - [CanIUse Compatibility Tables](#)



CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

QUESTIONS

GRAND
CIRCUS
DETROIT

CSS PSUEDO-CLASSES AND PSUEDO- SELECTORS

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

CSS PSUEDOCASSES

Psuedo-classes are used to add special effects to some selectors.

PSUEDO-CLASS SYNTAX

```
selector:pseudo-class { property:value }
```

```
selector.class:pseudo-class { property:value }
```

ANCHOR PSUEDO-CLASSES

```
a:link { color: aliceblue; } /* unvisited link */
a:visited { color: darkblue; } /* visited link */
a:hover { color: lightblue; } /* mouse over link */
a:active { color: yellow; } /* mouse click link */
```

OTHER PSUEDO-CLASSES

class

Description

`:first-child`

Any element which is the first child of its parent

`:first-letter`

First letter of an element's content

`:first-line`

First line of an element's content

`:last-child`

Any element which is the last child of its parent

`:only-child`

Any element which is the only child of its parent

`:empty`

Any element which has no content

`:nth-child`

Any element which is the nth child of its parent

NTH-CHILD SYNTAX

`:nth-child` takes a mathematic express to determine which children to select.

```
element:nth-child(an + b) { property:value; }
```

NTH-CHILD EXAMPLES

These expressions can be as complex as you want.

Feel free to mess around with it.

```
div:nth-child(3) { color:red; }
```

```
p:nth-child(4n) { color:blue; }
```

```
ul:nth-child:(2n + 5) { color:yellow; }
```

NTH-CHILD EXAMPLES

Realistically though, in practice `nth-child` is usually only used to alternate styles for large groups of data like table rows, etc.

```
element:nth-child(even) { property:value; }
```

```
element:nth-child(odd) { property:value; }
```



GRAND
CIRCUS
DETROIT



GRAND
CIRCUS
DETROIT



GR
CIR
DET

@FONT-FACE

@FONT-FACE

The @font-face property allows you to use non-default fonts on your web sites



DETROIT



@FONT-FACE USAGE

```
@font-face {  
  font-family: 'myCoolFont';  
  src: url('path/to/myCoolFont.ttf');  
}  
  
div.demo {  
  font-family: 'myCoolFont';  
}
```



@FONT-FACE

1. Download your font
2. Place the font file in your web site
3. Create a @font-face CSS rule
4. Give your font an identifier (e.g., "myCoolFont")
5. Point the identifier to your font file(s)

@FONT-FACE

You can download fonts from external sources or reference them via the internet using services like Google Web Fonts.

QUESTIONS

CSS3 FEATURES

CSS3 VENDOR PREFIX

Not all browsers support all of the newer CSS3 properties, which means we sometimes need to create rules for certain browsers

BORDER-RADIUS USAGE

```
.example {  
-moz-border-radius: purple;  
-ms-border-radius: purple;  
-webkit-border-radius: purple;  
-o-border-radius: purple;  
border-radius: purple;  
}
```



CIRCUS
DETROIT



CIRCUS
DETROIT



VENDOR PREFIXES

Order matters! The non-prefixed property should always go last.

HANDY CSS3 PROPERTIES

There are a great many very handy CSS3 properties that you may need to use on your current and future projects. They can add a level of polish to your project that really sells it. That said, we dont have a time to go over even a very few of them in details so I'll just create a list that you can investigate further.

CSS3 PROPERTIES

Property	Description
<code>border-radius</code>	creates rounded corners or circular borders for elements
<code>rgba</code>	An alternate way to define color and alpha
<code>opacity</code>	Controls the transparency of an element
<code>box-shadow</code>	Adds a calculated shadow effect to an element
<code>text-shadow</code>	Adds a calculated shadow effect to an element
<code>transforms</code>	Adds the ability to transform HTML elements (rotation, scale, translate)
<code>animations</code>	Allows for one or more property changes to be applied as an animation

CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

RESPONSIVE DESIGN

GRAND
CIRCUS
DETROIT



RESPONSIVE DESIGN

Changes your website presentation in relation to the screen size

MEDIA QUERIES

This is done via Media Queries. Media queries are able to detect a bunch of things about the device with which you're viewing a site. The browser can use specifically developed CSS styles for those different screen sizes.



GRAND
CIRCUS
DETROIT



MEDIA QUERY EXAMPLE

```
@media (max-width: 600px) {  
  /* specific styles for this device size */  
}
```

CIRCUS
DETROIT



GRAND
CIRCUS
DETROIT



GRAND
CIRCUS
DETROIT



GR

THINGS TO DO WITH MEDIA QUERIES

- resize images / font
- collapse navigation
- remove some larger content (video / images)

GRAND
CIRCUS
DETROIT

CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

GR
CIR

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS

LAB 4

BRINGING IT ALL TOGETHER

RESPONSIVE LAB

- Make a website!
- Your website should include 4 rows of images, 4 images per row
- Each image should have a heading
- Each image should have an explanatory caption
- Each image should have a border, padding, and margin

RESPONSIVE LAB (CONT)

- Your page should include a header and footer
- Your images should reformat based on the size of the view-port
- Use at least 2 media queries
- Some styling should change based on the size of the view-port (ex: border color around images changes based on screen width)

CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

GR
CIR

DET

HTML FORMS

GRAND
CIRCUS
DETROIT

GR
CIR

DET

FORMS

Forms are used to send data back to a server from user inputs



GRAND
CIRCUS
DETROIT



GRAND
CIRCUS
DETROIT



GRAND
CIRCUS
DETROIT

FORM TAGS

```
<form action="something" method="get">  
    <!-- Form inputs and other html -->  
</form>
```

METHODS AND ACTIONS

Action is the URL you want the data to be sent to

Method is the HTTP methods used to transfer data

- GET
- POST
- PUT
- DELETE

CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

FORM TAGS

```
<input type="text" method="get" name="firstname" placeholder="Name?">
```

TYPES OF INPUTS

- Text (type='text')
- Radio Button (type='radio')
- Checkbox (type='checkbox')
- Password (type='password')
- Date (type='date')
- Time (type='time')
- Date & Time (type='datetime')
- Email (type='email')
- Submit
 - `<input type='submit' />`
 - `<button type='submit' />`

-DETROIT-

GRAND
CIRCUS
-DETROIT-

GRAND
CIRCUS
-DETROIT-

GRAND
CIRCUS
-DETROIT-

GRAND
CIRCUS
-DETROIT-

GR
CIR
-DET

GRAND
CIRCUS
-DETROIT-

DEMO

GRAND
CIRCUS
-DETROIT-

GRAND
CIRCUS
-DETROIT-

GRAND
CIRCUS
-DETROIT-

GRAND
CIRCUS
-DETROIT-

GR
CIR
-DET

GRAND
CIRCUS
-DETROIT-

GRAND
CIRCUS
-DETROIT-

GRAND
CIRCUS
-DETROIT-

GRAND
CIRCUS
-DETROIT-

GRAND

GR
CIR
-DET



BOOTSTRAP



BOOTSTRAP

HTML5 & CSS framework that combines

- Responsive Design
- Grid Positioning
- Iconography
- Mobile Templating
- Javascript plugins



ABOUT BOOTSTRAP

Bootstrap is a CSS framework that makes making halfway—decent—looking sites a snap!

It was created by Twitter and was made open source a few years ago

BOOTSTRAP GRIDS

The best of CSS positioning and table layouts

- 12 columns wide
- Infinite rows
- Responsive-ready out of the box
- Uses only CSS

ROWS & COLUMNS

GRAND

```
<div class="container">
  <div class="row">This is Row 1</div>
  <div class="row">
    <div class="col-md-6"> This is Row 2 Col 1 </div>
    <div class="col-md-6"> This is Row 2 Col 2 </div>
  </div>
</div>
```

GRAND
CIRCUS
DETROIT

DETROIT

GRAND
CIRCUS
DETROIT

DEMO

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS

GR
CIR

DET



WHERE TO GET IT

<http://www.getbootstrap.com> > Getting Started

Hit the "Download Bootstrap" button

HOW TO USE IT

1. Unzip the folder and rename it "bootstrap"
2. Save "bootstrap" in the folder containing your html and css files
3. Link to bootstrap in your index.html file

```
<link href="bootstrap/css/bootstrap.min.css" rel="stylesheet">  
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"></script>  
<script src="bootstrap/js/bootstrap.min.js"></script>
```

CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

```
<!-- Latest compiled and minified CSS -->
<link rel="stylesheet" href="//maxcdn.bootstrapcdn.com/bootstrap/3.2.0/css/bootstrap.min.css">

<!-- Optional theme -->
<link rel="stylesheet" href="//maxcdn.bootstrapcdn.com/bootstrap/3.2.0/css/bootstrap-theme.min.css">

<!-- Latest compiled and minified JavaScript -->
<script src="//maxcdn.bootstrapcdn.com/bootstrap/3.2.0/js/bootstrap.min.js"></script>
```

Copy

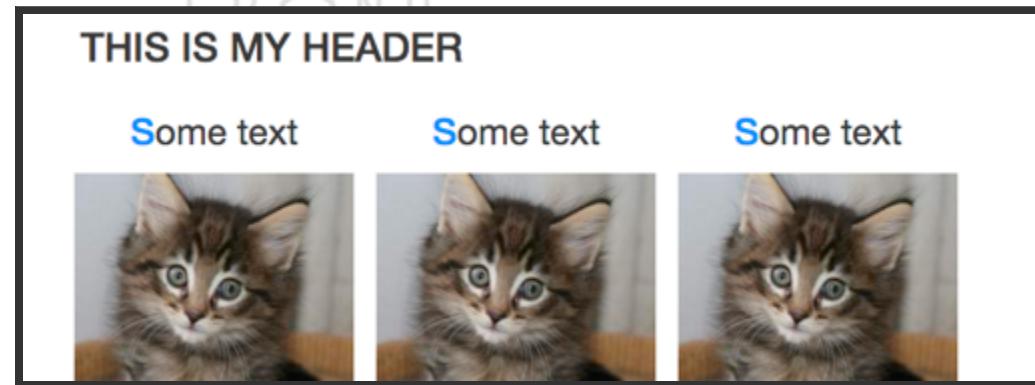
Like jQuery, Bootstrap can also be linked via CDN.

Just linking to the Bootstrap CDN will make a noticeable difference on your page

Before



After



CSS COMPONENTS

Bootstrap provides a bunch of CSS Components

- Grid layout
- Buttons
- Primary and Secondary headings
- Form elements
- Automatic table styling
- All the things basically

CSS COMPONENTS

Bootstrap provides great docs which demonstrate each CSS component and how to add them to your site.

There are so many things available in just the CSS portion of the Bootstrap. We don't have time to look at them all.



USING BOOTSTRAP

As said before, all of the bootstrap components are shown on their docs but for the most part it's as simple adding some classes to your html.

ex:

```
<a href="#" role="button" class="btn btn-primary">A button!</a>
```

RECAP

You should understand and be able to use:

- HTML5 semantic tags
- CSS Pseudo-selectors
- CSS3 Properties (vendor-prefixes, transforms, animations)
- HTML Forms
- Media Queries
- Bootstrap (CSS, Grids)

CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

LAB 5

BRINGING IT ALL TOGETHER

GRAND
CIRCUS
DETROIT



LAB 5

Create a blog page using Bootstrap Grid. The page should have the following:

Requirements

- (optional) Nav section on Left, 1/3rd page width
- (optional) Main section on Right, 2/3rd page width
- At least 1 faked article of blog content
- At least 1 floated image with text wrapping around it in the article
- Use bootstrap glyphicons
- An HTML form with at least 3 elements, one of which must be radio button group, check box group, date picker, or dropdown, i.e. `select`
- Footer with copyright, contact link, some other content
- Use CSS psuedo classes
- BONUS: Research and build in a Bootstrap JavaScript Plugin (ex. Carousel, Accordion, Collapse, Tooltip, Popover, or Modal)



GRAND
CIRCUS
DETROIT



GRAND
CIRCUS
DETROIT



GRAND
CIRCUS
DETROIT



GRAND
CIRCUS
DETROIT

JAVASCRIPT



CIRCUS
DETROIT



GRAND
CIRCUS
DETROIT



CTR
DET

WELCOME BACK!



GOALS FOR THIS UNIT

1. Review HTML & CSS
2. Intro to Programming
3. Conditional Logic



ASSOCIATED READING

From JavaScript & jQuery:

- Introduction: 1-10
- Chapter 1: 11-52
- Chapter 2: 53-74, 78-84

CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

REVIEW

GRAND
CIRCUS
DETROIT

DEMOS

GRAND
CIRCUS
DETROIT

CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

A BRIEF HISTORY OF JAVASCRIPT

HISTORY OF JAVASCRIPT

- Developed by Brendan Eich of Netscape in 1995
- First draft of the language was written in 10 days (!!)
- Originally called “Mocha,” then “LiveScript”
- Is not related to the Java language in any way
- “Script” refers to instructions that can be executed by a computer
- JavaScript was standardized in 1997
- As a scripting language it is interpreted, not compiled

HISTORY OF JAVASCRIPT

- 2005: AJAX, a method of asynchronous server communication using JavaScript, is developed
- 2006: jQuery, the most popular JavaScript library, debuts
- 2010: Node.js brings JavaScript to back-end development

BONUS KNOWLEDGE!

ASYNCHRONOUS

In this context, **asynchronous** means that our JavaScript is doing more than one thing at the same time. With AJAX, the rest of our page will keep working even as a section of it is updated with new information. More on that later!

OUTPUT

Output is any information provided by the program.
We will use it often when debugging our code.

OUTPUT

`console.log()` is the JS equivalent to a standard out.

This works in the browser and in NodeJS. It will print whatever is contained inside the parentheses to the console.



GRAND
CIRCUS
DETROIT

STANDARD OUT

GRAND
CIRCUS
DETROIT



GRAND
CIRCUS
DETROIT

```
console.log('Hello, World!');
```

GRAND
CIRCUS
DETROIT

produces:



GRAND
CIRCUS
DETROIT



Hello, World!

CIRCUS
DETROIT



GRAND
CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

GRAND



CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT



ALERTS

We can alert the user to something in a similar manner, using `alert()`.

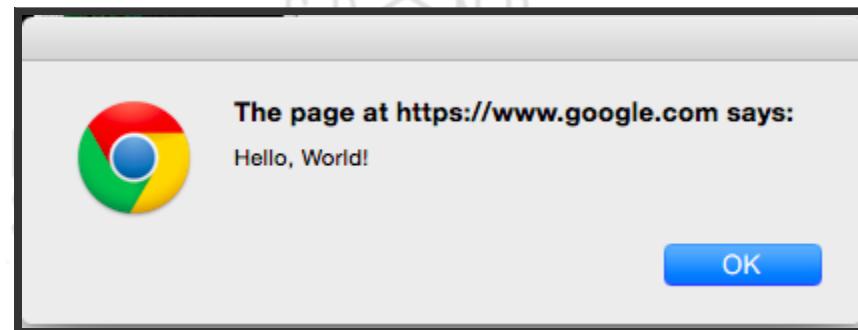
GRAND
CIRCUS
DETROIT

CIRCUS
DETROIT

ALERT

```
alert('Hello, World!');
```

produces:





GRAND
CIRCUS
DETROIT



GRAND
CIRCUS
DETROIT



GRAND
CIRCUS
DETROIT

COMMENTS

Just as in HTML & CSS, we can (and should) use comments in JavaScript.

```
```javascript
/* Multi-line comments look
like this. */

// Single-line comments look like this.
```



GRAND  
CIRCUS  
DETROIT



GRAND  
CIRCUS  
DETROIT



GRAND  
CIRCUS  
DETROIT





GRAND  
CIRCUS  
DETROIT



# VARIABLES



# VARIABLES

Think of variables as boxes that you put stuff inside of. Each box has a label (the variable's name), which gives you some idea of what it holds. JavaScript variables can contain lots of different types of information.

# VARIABLES

```
var numberOfHackerFellow = 15;
var firstName = "James";
var _lastName = 'York'; // single or double quotes for strings
var $isDetroitCool = true;
var anotherVar2 = 'hooray';
```

# DECLARATION

Before we can use a variable, we need to declare it using the `var` keyword.

```
// declare one or more variables individually
var school;
var student;

// declare multiple variables in one statement
var teacher,
subject,
period;
```

# INITIALIZATION

There's not too much we can do with an empty box.  
To give our variables some value, we need to initialize  
them.

...

```
school = "Grand Circus";
teacher = "James";
```

**Note:** these variables have been declared earlier in  
our code!

# ONE FELL SWOOP

We can declare *and* initialize our variables all at once  
(as we saw in the first "variables" slide).

```
var andreSays = "Anybody want a peanut?";
```



# NAMING IN JAVASCRIPT

1. Must begin with a letter, \_, or \$
2. Can contain letters, numbers, \_, and \$
3. Names are case-sensitive

# DATA TYPES

# DATA TYPES

- Number
- String
- Boolean
- Array
- Object
- Function

# NUMBER

The only numeric data type in JavaScript, it can be used to represent integers and floating-point numbers and has three symbolic values `+Infinity`, `-Infinity`, and `NaN` (Not a Number).

```
var myAge = 24;
var priceOfIceCream = 6.25;
```

**Note:** Be careful not to mix integers with floating-point numbers (which have decimals) when doing arithmetic! You will probably not like what happens.



# STRING

A string is a sequence of characters strung together to represent text. Think of a "Happy Birthday!" banner. Each letter is connected by a *string* to create a message humans can read.

```
```javascript
var myName = "Aisha";
var faveMovie = "The Princess Bride";
````
```



# BOOLEAN

Boolean values are either true or false. They are generally used in our programs' logic.

```
```javascript
var jsIsFun = true;
var detroitIsScary = false;
```

ARRAY

An array is basically a list of items, each item having what's called an index. The first item in an array has an index of 0 and each subsequent item's index increases by 1, e.g., the 42nd item in an array has an index of 41.

```
```javascript
var contentTeam = ["Aisha", "James", "Jeseekia", "Kim", "Xinrui"];
var randomStuff = [5, false, 847.3, "puppies", ["another", "array"]];
```

```

OBJECT

We will talk in much more detail about objects later in the course, but for now think of them as a collection of properties and values.

```
```javascript
var startup = {
 name: 'Grand Circus',
 city: 'Detroit',
 staffSize: 13
}..
```

# FUNCTION

We will also spend a lot of time talking about what we can do with functions further down the road.

Functions allow us to bundle up parts of our code that we want to use more than once.

```
```javascript
function sayHello() {
  console.log("WAAASSAAAAAPP?!?!?")
}
```

```

# DYNAMICALLY TYPED

Hold on to your butts, C coders



HOLD ON TO YOUR BUTTS

# JS DYNAMIC TYPING

```
var yourButts = "hold on";
var yourButts = 121;
var yourButts = true;
var yourButts = ['hold', 'onto', 'your', 'butts'];
var yourButts = {
 youMust: 'hold onto them',
 all: 'of this is legal JavaScript',
 we: 'are reassigning these values to the same variable handle'
};
var yourButts = function() {
 return 'mindblown.gif';
};
```

CIRCUS  
DETROIT

CIRCUS  
DETROIT

CIRCUS  
DETROIT

GRAND  
CIRCUS  
DETROIT

GRAND  
CIRCUS  
DETROIT

GRAND  
CIRCUS  
DETROIT

# TRUTHY & FALSEY VALUES

GRAND  
CIRCUS  
DETROIT

GRAND  
CIRCUS

GRAND  
CIRCUS

GRAND  
CIRCUS  
DETROITGRAND  
CIRCUS  
DETROITGRAND  
CIRCUS  
DETROIT

# WHAT'S IT MEAN?!

Any value in JavaScript can be treated as either true or false.

# FALSY

Falsy values are treated as if they are false.

- The actual value `false`
- The number 0
- NaN (Not a Number)
- Empty values
- A variable with no value assigned

# TRUTHY

Truthy values are treated as if they are true.

- The actual value `true`
- Nonzero numbers
- Nonempty strings
- Mathematical expressions

# ARITHMETIC

## Arithmetic operator

Addition

+

Subtraction

-

Multiplication

\*

Division

/

Modulus

%

Increment

++

Decrement

--

# ASSIGNMENT

## Arithmetic

## operator

---

standard assignment



plus equals



minus equals



assignment by multiplication



assignment by division



# STRING CONCATENATION

We can use the plus sign to combine strings.

```
var firstName = 'James ';
var lastName = 'York';
var fullName = firstName + lastName;
console.log(fullName); // > 'James York'

var name = 'j';
name += 'ames';
console.log(name); // > 'james'
```

GRAND  
CIRCUS  
DETROIT

CIRCUS  
DETROIT

GRAND  
CIRCUS  
DETROIT

GRAND  
CIRCUS  
DETROIT

GRAND  
CIRCUS  
DETROIT

GRAND  
CIRCUS  
DETROIT

GR  
CIR  
DET

GRAND  
CIRCUS  
DETROIT

GRAND  
CIRCUS  
DETROIT

GRAND  
CIRCUS  
DETROIT

# COMPARISON

GRAND  
CIRCUS  
DETROIT

# COMPARISON

## comparison

Equality

`==`

Inequality

`!=`

Greater than

`>`

Greater than or equal to

`>=`

Less than

`<`

Less than or equal to

`<=`

# DOUBLE EQUALS

- 'Shallow' equals ==
- 'Shallow' inequals !=

Performs a type coercion before checking equality.

# TYPE COERCION

```
true == "true" // > false
true === "true" // > false
"1" == 1 // > true
"1" === 1 // > false
"1" != 1 // > false
```

Don't use double equals. I'm only showing them because you may see them in the code bases you work on.

CIRCUS  
DETROIT

GRAND  
CIRCUS  
DETROIT

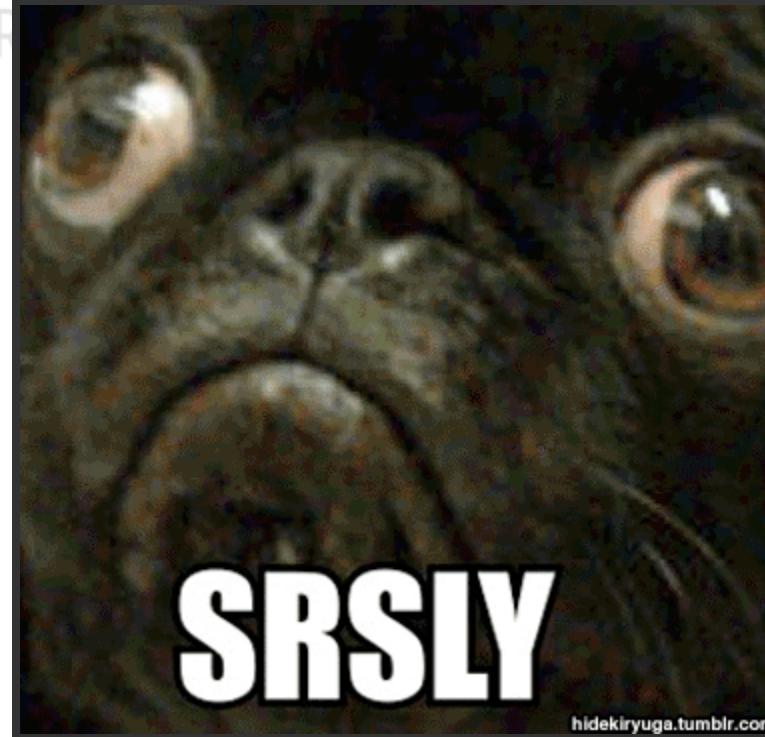
GRAND  
CIRCUS  
DETROIT

GRAND  
CIRCUS  
DETROIT

# SERIOUSLY. DON'T USE THEM.

GRAND  
CIRCUS  
DETROIT

GRAND  
CIRCUS  
DETROIT



GRAND  
CIRCUS  
DETROIT

GRAND

GRAND  
CIRCUS  
DETROIT

GRAND  
CIRCUS  
DETROIT

GRAND  
CIRCUS  
DETROIT

CIRCUS  
DETROIT

GRAND  
CIRCUS  
DETROIT

NEW  
CONTENT  
AHEAD!



GRAND  
CIRCUS  
DETROIT

GRAND  
CIRCUS  
DETROIT