{ } INTRO TO WEB DESIGN

Introduction to Web Design

{ } INTRO TO WEB DESIGN

What Websites are made of

Front-end

- What users can "see"
- Ex. Buttons, Images, Text

Back-end

- What users can't "see"
- Ex. Algorithms behind Logins, Search, etc.

{ } INTRO TO WEB DESIGN

HTML

- Structure
- = Skeleton

CSS

- Design
- = Clothes

Javascript

- Function
- = Muscle

{ } INTRO TO WEB DESIGN

HTML provides the structure of a page

A well-written HTML document should be "understandable" without a stylesheet.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
        "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>
    <meta http-equiv="content-type" content="text/html; charset=utf-8" />
    <link rel="stylesheet" type="text/css" href="assets/style.css" />
    <script type="text/javascript" src="assets/jquery-1.8.3.min.js" charset="utf-8"></script>
    <script src="assets/jquery-ui-1.9.2.min.js"></script>
    <script type="text/javascript" src="assets/script.js" charset="utf-8"></script>
    <title>THE WEB DESIGN WORKSHOP</title>
<body>
    <div id="menu">
        <div id="menu_icon">
            <img class="img_scaled" src="assets/icons/logo.svg" />
        </div>
        <div id="menu_strip_container">
            <div id="menu_strip">
                <a href="index.html"><div class="menu_item" style="margin-left: 35px;">About</div></a>
                <a href="lectures.html"><div class="menu_item">Lectures</div></a>
                <a href="gradebook.html"><div class="menu_item">Gradebook</div></a>
                <a href="showcase.html"><div class="menu_item">Showcase</div></a>
                <div class="clear"></div>
            </div>
        </div>
    </div>
    <div id="cover">
        <div id="cover_bg"><img class="img_scaled" src="assets/thumbnails/cover.png" /></div>
        <div id="cover_hero_container">
            <div id="cover_hero">
                <div id="cover_hero_title">The Web Design Workshop</div>
                <div id="cover_hero_subtitle">DeCal Fall 2013</div>
            </div>
        </div>
        <div id="cover_overlay"></div>
    </div>
    <div id="content">
        <div id="content_title_container">
            <div id="content_title">About Class</div>
```

CSS provides the design of a page

Just several lines of "code" is enough to create "design" – shapes, color, buttons, you name it.

```
body {  /*the outer layer of the site and global style attributes*/
font-family: "Museo Sans", "Helvetica Neue", Helvetica, Arial, sans-serif;
margin: 0px;
}

a {
color: inherit;
text-decoration: none;
}

.clear {
clear: both;
}

.img_scaled {
width: 100%;
}

.img_scaled_wh {
width: 100%;
height: 100%;
}

.img_scaled_h {
height: 100%;
}

#logo {
height: 50px;
float: left;
}
#header_container {
position: absolute;
height: 80px;
width: 100%;
z-index: 100;
}
#header {
```

Search for people, places and things

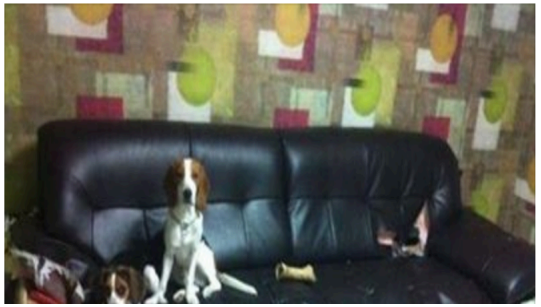
Home Shawn

Tuhina Das was tagged in **Sooln Yoon**'s photos. — with **Kin Yi Tsui** and **8 others** in **Hong Kong, Hong Kong**.

37 minutes ago ·

Eunyoung Lee shared 아웃겨ㅋㅋㅋ's photo.

내가 집에 동물을 안키우는 이유



OnStartups

Do you like startups? Click the like button below and get free tips and insights on startups.

Like · Minsu Daniel Kim and Zhibai Han like OnStartups.

Bethsarim Van Koetsveld

Build your list & drive sales - Get your click provoking website header today! I can help.

Like · 115 people like Bethsarim van Koetsveld.

Organize for Action
barackobama.com

Join millions at OFA and support President Obama's national agenda. Are you in?

Builders all tied up?

Are you stuck? Unable to upgrade your army? Move faster in Dojo Mojo. Play now, free!

Facebook © 2013

English (US) · Privacy · Terms · Cookies · More

Tuhina Das was tagged in Sooln Yoon's photo. — with Xiao Li in Hong Kong, Hong Kong.

Michael Kang added a new photo to the album Mobile Uploads.

James Chi likes Brenda Yee's post in Free & For Sale.

Jian Chang likes Patrick Vith's status.

Makai Cartman likes Breaking Bad's photo.

Do Yeob David Hyun commented on ELO HELL's photo: "TIME TO FIST"

James Chi

Jiny Her

Louis J. Kang

Andy He

Kathy Khuu

Jeff Zhan

Hyun-Ho Hayden Jung

Eddie Lee

Brennan Lee

Maritza Gonzalez Tellez

Patrick Ou

MORE FRIENDS (26)

Alex Wang

Search

Facebook

- *0 Requests*

Friend Requests

Find Friends · Settings

Friend Requests

See All

0 Messages

Messages;

Send a New Message

Inbox(3)Other(12)

- 
  
  Team Ego
  You sent a sticker.
  11:02pm
- 
  Elmer Almeida
  Native: Eclipse
  9:46pm
- 
  Rachit Nanda
  Rachit sent a photo.

{ } INTRO TO WEB DESIGN

JS provides the function of a page

What will happen to the website with Javascript disabled?

```
$(document).ready(function(){

    $('.about_instructors_item').hover(function(){
        $(this).children('.about_instructors_cover').animate({'margin-top': '-400px'});
    }, function(){
        $(this).children('.about_instructors_cover').animate({'margin-top': '0px'});
    });

    $(window).scroll(function(){
        if ($(window).scrollTop() <= 0){
            $('#menu_strip').css({'margin-left': '0px'});
        } else {
            $('#menu_strip').css({'margin-left': '-400px'});
        }
    });

    $('#menu').hover(function(){
        $('#menu_strip').css({'margin-left': '0px'});
    }, function(){
        if ($(window).scrollTop() > 0){
            $('#menu_strip').css({'margin-left': '-400px'});
        }
    });

    $('.showcase_nav_item').hover(function(){
        if ($(this).attr('id') == 'showcase_wow') {
            $('#showcase_nav_item_active').css({'left': '0%'});
        } else {
            $('#showcase_nav_item_active').css({'left': '50%'});
        }
        return false;
    }, function(){
        if ($(this).hasClass('showcase_nav_item_toggled')){
            return false;
        }
        else if ($(this).attr('id') == 'showcase_wow') {
            $('#showcase_nav_item_active').css({'left': '50%'});
        } else {
```

{ } INTRO TO WEB DESIGN

How do I make a Website?

Web Browser	Code Editor	Image Editor (Optional)
• Chrome • Firefox • Safari	• Coda 2 • Sublime Text • Dreamweaver	• Adobe Photoshop

{ } INTRO TO WEB DESIGN

UI Design Workflow

HTML	• Code up the structure in a code editor
CSS	• Stylize the website using the Web Inspector
JS	• Add the missing functionality in the code editor

{ } INTRO TO WEB DESIGN

HTML Structure

{ } INTRO TO WEB DESIGN

Tags

- Tags start with a left bracket < and end with a right bracket >
- This is a paragraph tag: <*p*>
- There are *opening* tags and *closing* tags. Closing tags start with </ instead
  - Example: </*p*>
  - Together, a set of tags is called an **element**
  - <*p*></*p*> is a paragraph element
- You can add *content* between a opening and closing tag
  - Example: <*p*>*Hello Class!*</*p*>

{ } INTRO TO WEB DESIGN

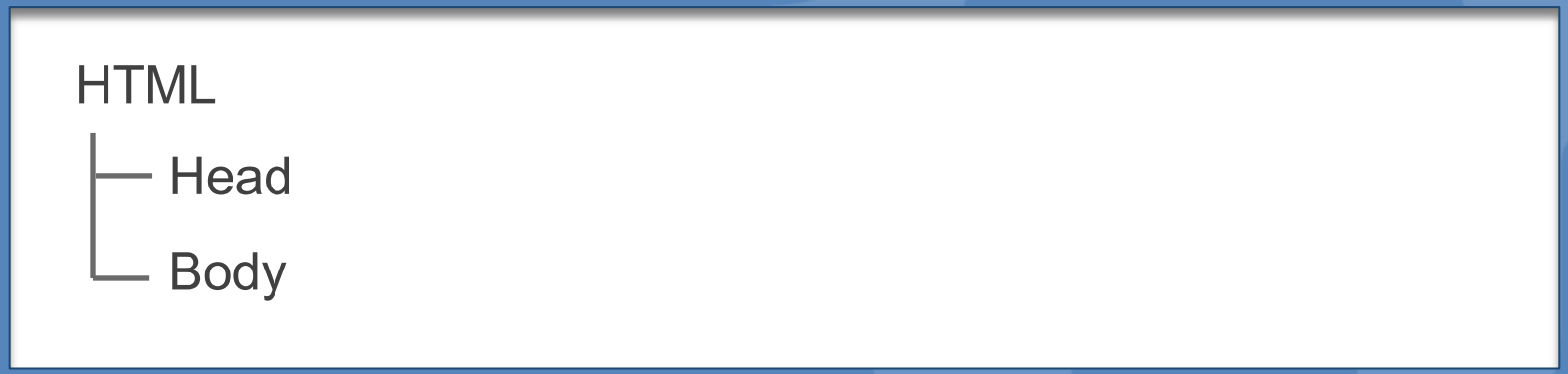
Tags

- **Opening tags must have a closing tag**
  - Not acceptable: Hello!
  - Close it: Hello!
- There are *exceptions* for special tags
  - Example: <*br*>
    - This is a *line break* tag
  - Also: <*link type="text/css" rel="stylesheet" href="style.css"*>
    - This is a *link* tag to your css file (we will go over this later)

{ } INTRO TO WEB DESIGN

Basic HTML Structure

- All webpages have **3 core elements** to them:
  - HTML tag
  - Head tag
  - Body tag
- The *head* and *body* tags are within the HTML tags



Basic HTML Structure

- All HTML pages must start with the *<html></html>* tags
  - This tells the browser to render meaningful HTML code
- Inside the HTML tags, put *<head></head>* tags

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

- Notice we indented the *<head></head>* tags. This is for readability purposes (you could put everything in one line too, but we indent to make it easier for us to read)

{ } INTRO TO WEB DESIGN

Basic HTML Structure - Header

- What goes in *<head></head>* ?
  - Title of page, links to css & javascript files, search engine keywords and description, website info, etc.
  - Everything **not rendered** on the page (the content)
- **Title**
  - *<title></title> tags*
  - Defines the title displayed on browser windows/tabs

{ } INTRO TO WEB DESIGN

Basic HTML Structure - Header

```
<html>
    <head>
        <title>First Webpage</title>
    </head>
</html>
```

{ } INTRO TO WEB DESIGN

Basic HTML Structure - Body

- What goes in *<body></body>* ?
  - Content for your page
  - This is what the **viewer sees**
- **Heading Tags**
  - *<h1></h1>, <h2></h2>, …, <h6></h6>*
  - 6 default heading sizes. 1 is the largest, 6 is the smallest
- **Paragraph Tags**
  - * *
  - Adds some space above and below your paragaph

{ } INTRO TO WEB DESIGN

Basic HTML Structure - Body

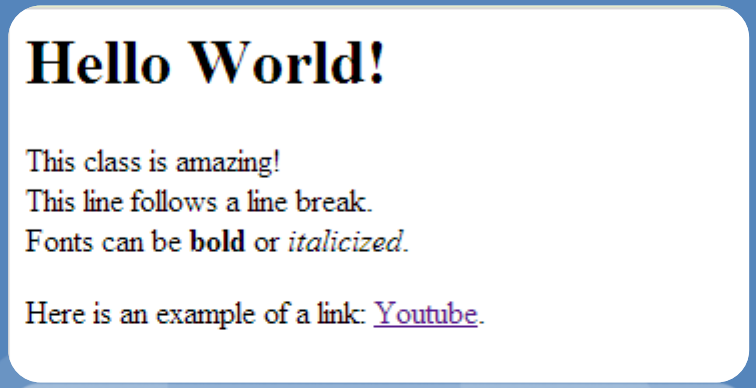
- Other Useful Tags
  - <*br*> for a line break (jumps to next line for a text)
  - <*strong*>…</*strong*> for bold text
  - <*em*>…</*em*> for italicized text
  - <*hr*> for a single horizontal line

{ } INTRO TO WEB DESIGN

Images & Links

{ } INTRO TO WEB DESIGN

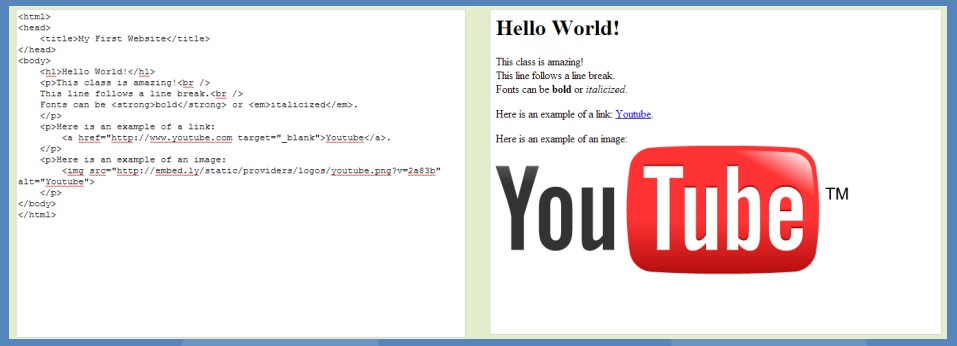
Images & Links



- **Links**
  - *<a>…* tags
  - Also called **anchor tags**
  - Example:
    - *Youtube*
  - *href* is an example of an element **attribute**
    - Attributes are followed by = and "…", with something in "…"
    - *href* needs a url inside the "…". Can be relative or absolute
  - Between the *<a>…* tags is the text displayed on the browser

{ } INTRO TO WEB DESIGN

Images & Links



- **Images**
  - *<img>* tag
  - Image tags rely a lot on HTML attributes (like *href* in anchor tags)
  - *src* attribute
    - Defines the image source (either relative or absolute path)
    - Example: *<img src="/path/to/filename.png">*
  - *height & width* attributes
    - *<img src="/path/to/filename.png" height="100" width="100">*
    - Note: Number is in # of pixels
  - *alt* attribute
    - Example: *<img ... alt="yo!">*
    - Displayed if image is unavailable, or is read if using screen reader

{ } INTRO TO WEB DESIGN

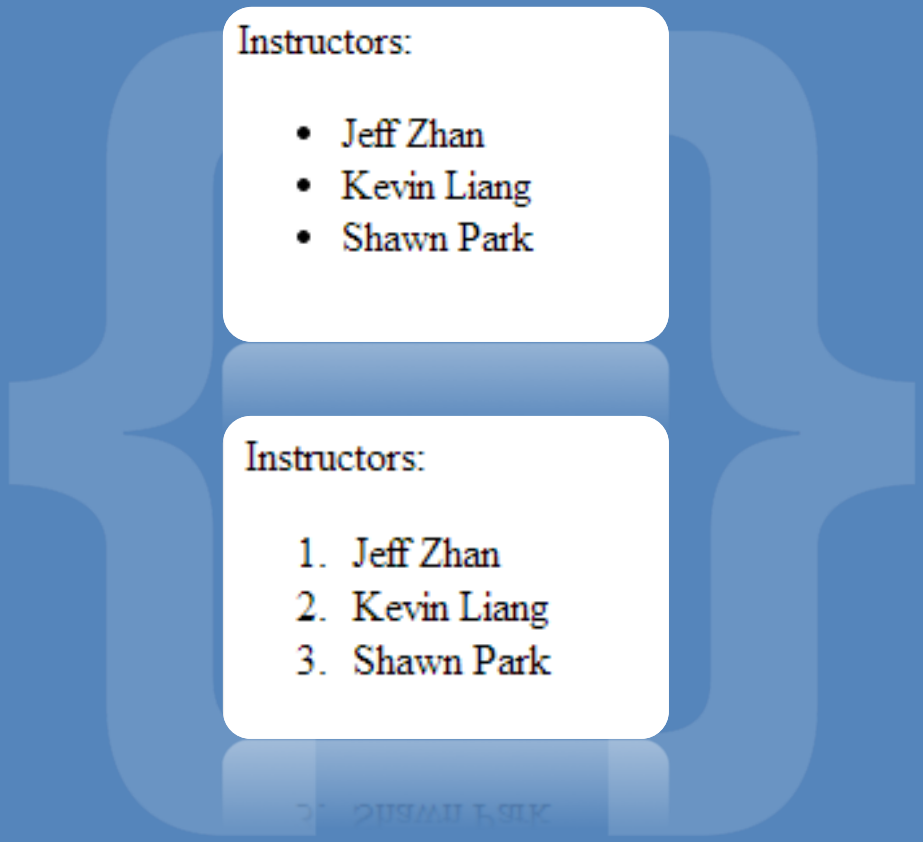
Lists

- **Unordered Lists** (bullets)

```
<ul>
    <li>Jeff Zhan</li>
    <li>Kevin Liang</li>
    <li>Shawn Park</li>
</ul>
```

- **Ordered Lists** (numbers)

```
<ol>
    <li>Jeff Zhan</li>
    <li>Kevin Liang</li>
    <li>Shawn Park</li>
</ol>
```



{ } INTRO TO WEB DESIGN

Lists

```
<html>
<head>
    <title>My First Website</title>
</head>
<body>
    <h1>Hello World!</h1>
    <p>This class is amazing!<br>
    This line follows a line break.<br>
    Fonts can be <strong>bold</strong> or <em>italicized</em>.
    </p>
    <p>Here is an example of a link:
        <a href="http://www.youtube.com target=" _blank"="">Youtube</a>.
    </p>
    <p>Here is an example of an image:
        <img src="http://embed.ly/static/providers/logos/youtube.png?v=2a83b"
alt="Youtube" width="100">
    </p>
    <hr>
    Instructors:
    <ul>
        <li>Jeff Zhan</li>
        <li>Kevin Liang</li>
        <li>Shawn Park</li>
    </ul>
</body>
</html>
```

Hello World!

This class is amazing!
This line follows a line break.
Fonts can be **bold** or *italicized*.

Here is an example of a link: Youtube.

Here is an example of an image: YouTube

Instructors:

- Jeff Zhan
- Kevin Liang
- Shawn Park

{ } INTRO TO WEB DESIGN

Hands-on Session 1
My first website

{ } INTRO TO WEB DESIGN

Divs & Spans

{ } INTRO TO WEB DESIGN

Divs and Spans

- * * tags are the primary tags we will use from now on
  - They are essentially tags with no special properties (like * or <h1>*)
  - You customize them with CSS
  - They are stacked vertically, meaning you cannot have two divs side by side unless you alter the CSS
- ** tags are used from time to time
  - Like a div tag, they have no special properties
  - You customize them with CSS
  - They are stacked **horizontally**, meaning you can have two spans side by side

{ } INTRO TO WEB DESIGN

IDs and Classes

{ } INTRO TO WEB DESIGN

IDs and Classes

- IDs and Classes are both **attributes** to HTML elements
- They look like this: * * or *<p class="intro">*
- **IDs**
  - *Unique* to a single HTML file (no other IDs on page can have the same ID)
- **Classes**
  - There can be one or more of the same class on a given page
- IDs and Classes are important concepts for CSS. This is how CSS will select certain elements and style them

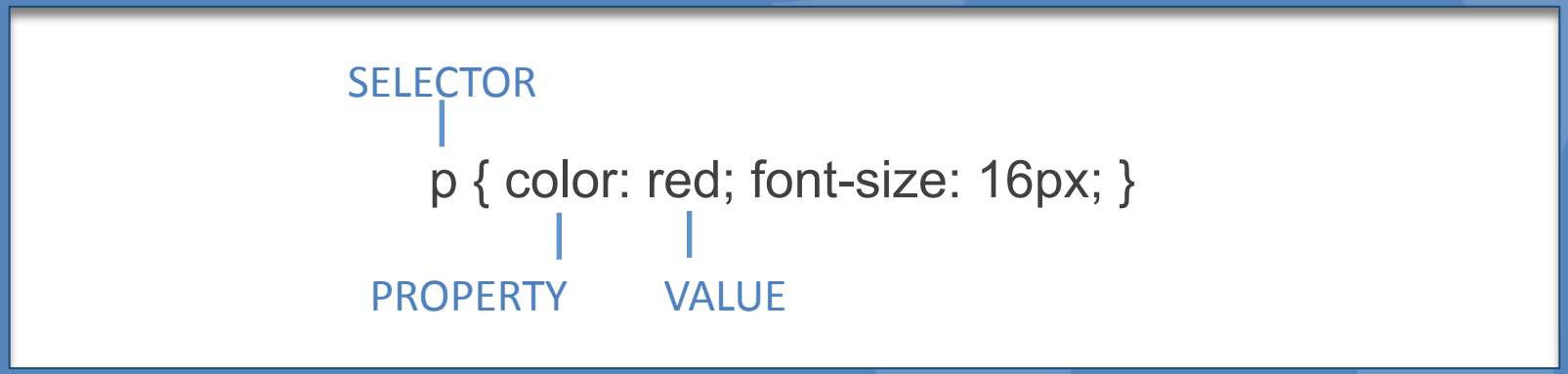
{ } INTRO TO WEB DESIGN

Intro to CSS

{ } INTRO TO WEB DESIGN

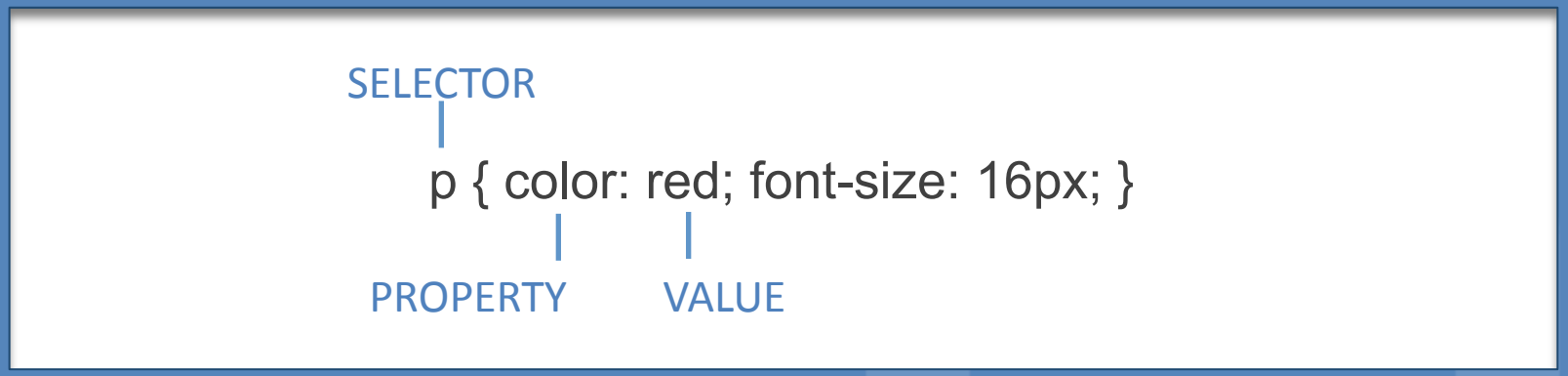
Intro to CSS

- CSS stands for **Cascading Style Sheets**
- They go in another file with an extension of .css
  - Link to the CSS file from your HTML file (in the *<head></head>* tags)
- Basic Structure of CSS:



{ } INTRO TO WEB DESIGN

Intro to CSS



- A “selector” can be the HTML element name, its ID name or Class name
- It is followed by braces **{ }**, with CSS property-value pairs in between
- **Properties** define a CSS style
  - They are followed by a colon **:** and a value
  - Close property-value pairs with a **semicolon ;** (important!!)

{ } INTRO TO WEB DESIGN

Intro to CSS

- Sample CSS Documents

```
@font-face {
    font-family: 'akashiregular';
    src: url('../fonts/akashi-webfont.eot');
    src: url('../fonts/akashi-webfont.eot?#iefi
         url('../fonts/akashi-webfont.woff') fo
         url('../fonts/akashi-webfont.ttf') for
         url('../fonts/akashi-webfont.svg#akash
    font-weight: normal;
    font-style: normal;
}

body {
    background: #2e2e2e url(../images/bg-main.j
    color: #d6f1ff;
    font-family: Verdana, Helvetica, sans-serif
    font-size: 20px;
    text-align: center;
}

p {
```

untitled | style.css

```
.entry-meta {
  margin-bottom: 15px;
  -webkit-font-smoothing: antialiased;
  -moz-osx-font-smoothing: grayscale;
}
.entry-meta ul {
  padding: 0;
  list-style-type: none;
}
.entry-meta li {
  float: left;
  margin-right: 12px;
}
.entry-meta a {
  color: #999999;
}
```

{ } INTRO TO WEB DESIGN

Intro to CSS

- Selecting elements to style is simple
- Use either the HTML element name, its class or its id
- * Hello World! *
  - Select this element with **p**
  - Style **p**: p { color: red; }
- What if you have **multiple** paragraphs, but want to style just the first one?
- You can use IDs:
  - * Hello World! My name is Bob. *
  - Select the ID *intro:*
    - #intro { color: red; }

{ } INTRO TO WEB DESIGN

Intro to CSS

- Select elements by ID with a pound symbol **#**
  - Example: *#intro { font-size: 16px; }*
- Select elements by Class with a dot symbol **.**
  - Example: *.items { line-height: 26px; }*
- How would you select “Second Paragraph”? What about both divs?

```
<div id=“first” class=“paragraph”>
        First Paragraph
</div>
<div id=“second” class=“paragraph”>
        Second Paragraph
</div>
```

- Answer 1: #second { … }
- Answer 2: .paragraph { … }

{ } INTRO TO WEB DESIGN

Linking to CSS

{ } INTRO TO WEB DESIGN

Linking to CSS

- 3 ways to style CSS
  - Inline CSS
  - Internal Stylesheet
  - External Stylesheet
- We will only cover the **External Stylesheet**, the most commonly used and *best* way to utilize CSS

{ } INTRO TO WEB DESIGN

Linking to CSS

- In your HTML file's *<head></head>* tags, include the following line

```
<link rel="stylesheet" type="text/css" href="/path/to/style.css">
```

- It is a lot to take in! Thankfully, you can copy this line for all your HTML pages and just change the href attribute to any .css file
- Just know this is a *<link>* tag, linking your HTML page to your CSS file

```
2733        font-weight: 300;
2734    }
2735    .user_popover_info_item_caption {
2736        font-size: 10px;
2737        font-weight: 700;
2738        color: #757c78;
2739        margin-top: 1px;
2740    }
2741    .user_popover_follow_icon {
2742        width: 12px;
2743        float: left;
2744    }
2745    .user_popover_follow_caption {
2746        float: left;
2747        color: white;
2748        font-size: 12px;
2749        font-weight: 700;
2750        line-height: 12px;
2751        margin-left: 8px;
2752    }
2753    .user_popover_follow {
2754        padding: 8px 13px;
2755        background: rgba(0, 0, 0, 0.5);
2756        float: right;
2757        border-radius: 2px;
2758        margin-top: 15px;
2759    }
2760
2761    /* Auxiliary */
2762
2763    .nomargin {
2764    margin: 0px;
2765    }
2766    .noborder {
2767    border: none;
2768    }
```

Do I really have to manually type out 2000 lines of CSS?

{ } INTRO TO WEB DESIGN

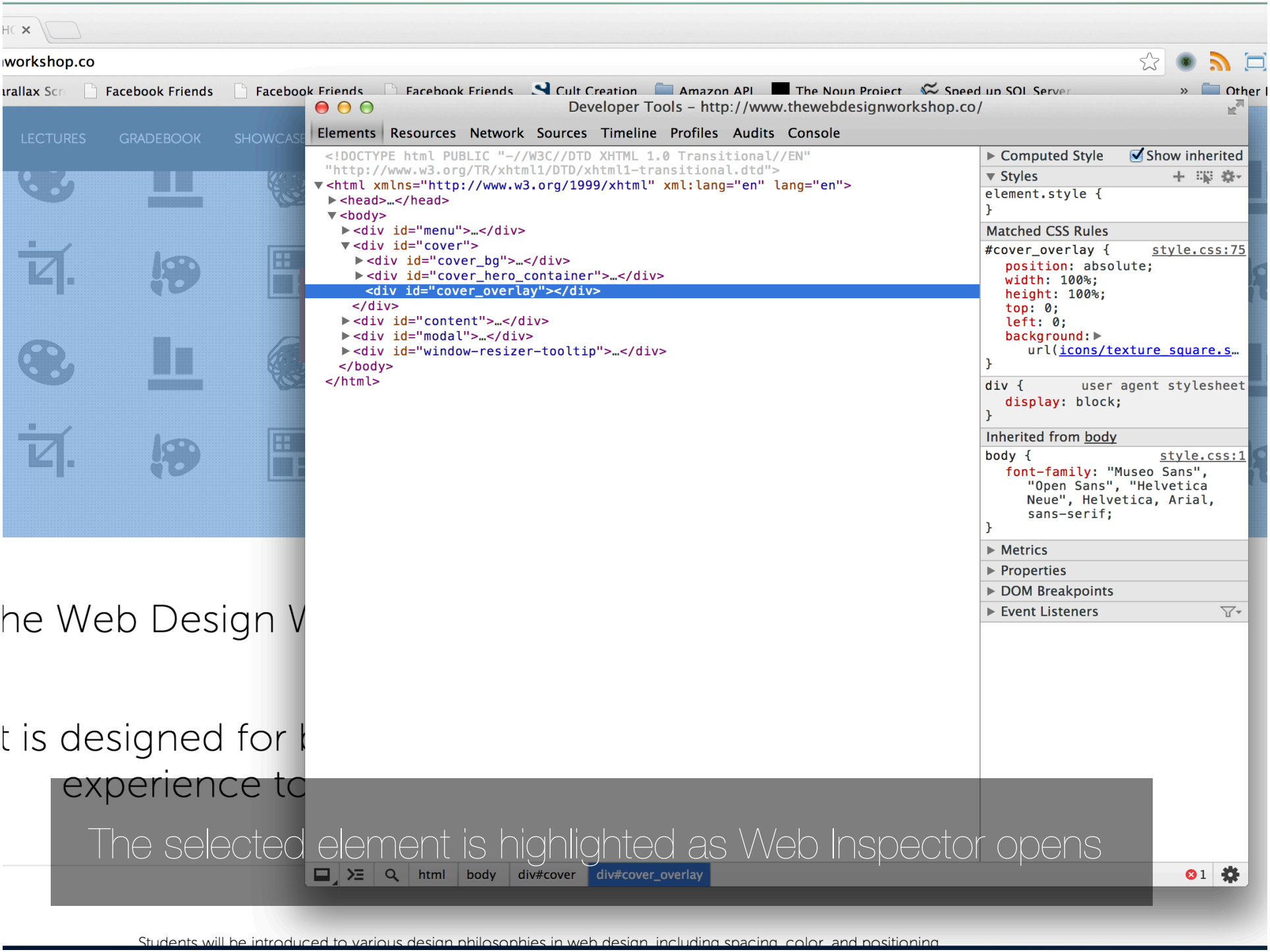
Web Inspector to the rescue!

- Allows you to test your designs in-browser
  - You can write out CSS rules and see how it looks live
  - No need to save a file, reload, etc.
- Allows you to learn from other people's work
- Other functionalities (for front-end) include:
  - Autocomplete for CSS rules
  - Color Picker
  - Browser Compatibility Test
  - JS Debugger
  - 3D View (Firefox)

The Web Design Workshop is a 2 Unit DeCal in Berkeley.

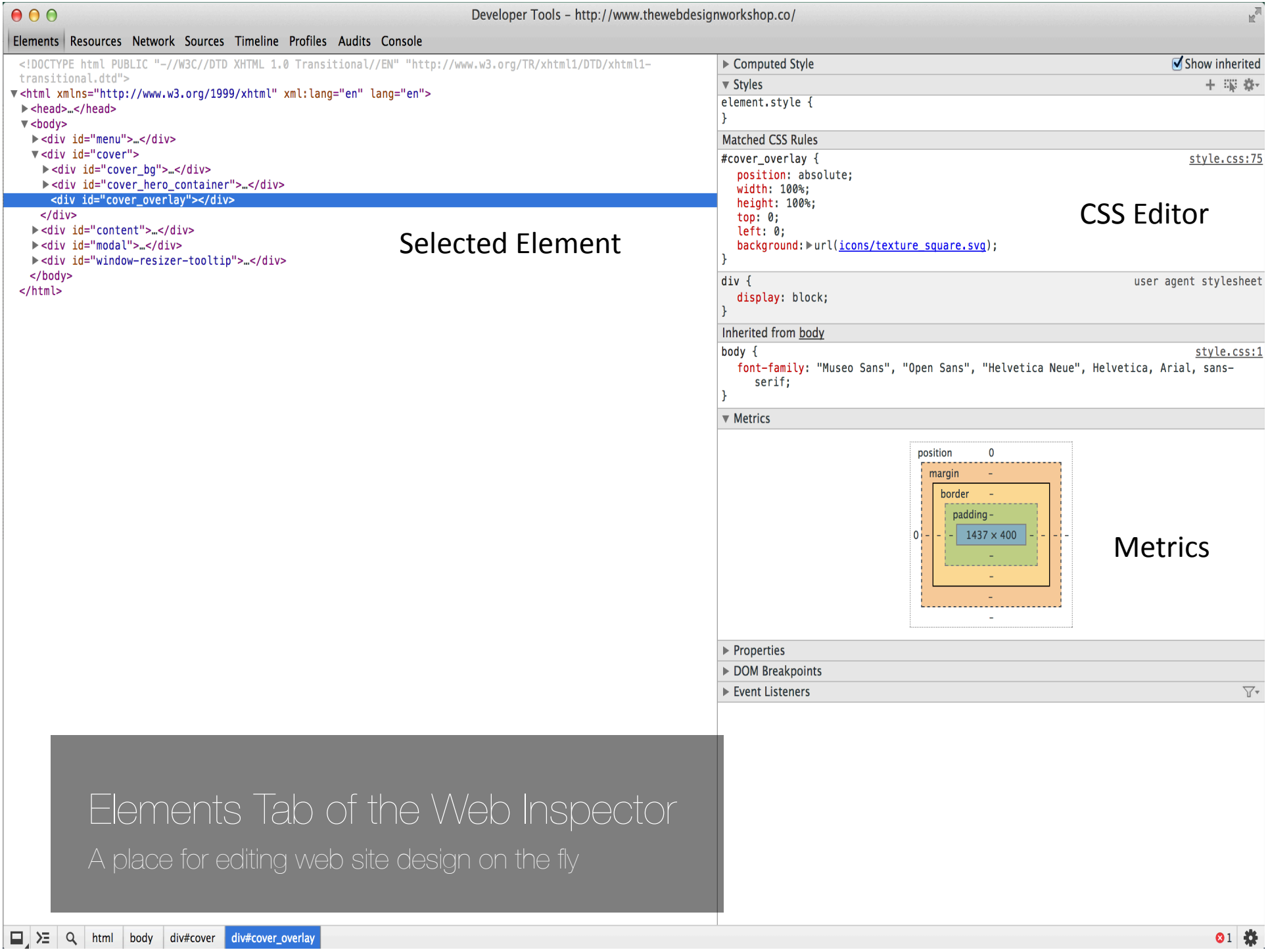
How to open up Web Inspector

It is designed for both students with or without


workshop.co
Facebook Friends
Facebook Friends
Facebook Friends
Cult Creation
Amazon API
The Noun Project
Speed up SQL Server
Other
LECTURES
GRADEBOOK
SHOWCASE
Developer Tools - http://www.thewebdesignworkshop.co/
Elements
Resources
Network
Sources
Timeline
Profiles
Audits
Console
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>…</head>
<body>
…

…
…

…
…
…
</body>
</html>
Computed Style
Show inherited
Styles
element.style {
}
Matched CSS Rules
#cover_overlay {
style.css:75
position: absolute;
width: 100%;
height: 100%;
top: 0;
left: 0;
background:
url(icons/texture_square.s…
}
div {
user agent stylesheet
display: block;
}
Inherited from body
body {
style.css:1
font-family: "Museo Sans", "Open Sans", "Helvetica Neue", Helvetica, Arial, sans-serif;
}
Metrics
Properties
DOM Breakpoints
Event Listeners
html
body
div#cover
div#cover_overlay
1
he Web Design
t is designed for
experience to
The selected element is highlighted as Web Inspector opens
Students will be introduced to various design philosophies in web design, including spacing, color, and positioning


Developer Tools - http://www.thewebdesignworkshop.co/
Elements Resources Network Sources Timeline Profiles Audits Console
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>…</head>
<body>
…

…
…

…
…
…
</body>
</html>
Selected Element
Computed Style
Show inherited
Styles
element.style {
}
Matched CSS Rules
#cover_overlay {
style.css:75
position: absolute;
width: 100%;
height: 100%;
top: 0;
left: 0;
background: url(icons/texture_square.svg);
}
CSS Editor
div {
user agent stylesheet
display: block;
}
Inherited from body
body {
style.css:1
font-family: "Museo Sans", "Open Sans", "Helvetica Neue", Helvetica, Arial, sans-serif;
}
Metrics
position
0
margin
border
padding
1437 x 400
Metrics
Properties
DOM Breakpoints
Event Listeners
Elements Tab of the Web Inspector
A place for editing web site design on the fly
html
body
div#cover
div#cover_overlay
1

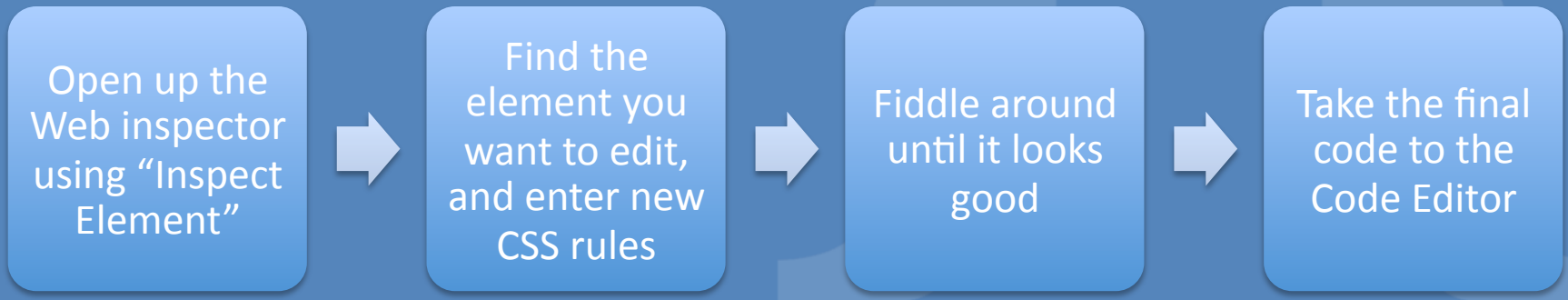
{ } INTRO TO WEB DESIGN

You will spend as much time on the Web Inspector as you will on the code editor

Try out the code on the Web Inspector before you write it on the Code Editor – It will save you tons of time

{ } INTRO TO WEB DESIGN

Web Inspector Workflow



{ } INTRO TO WEB DESIGN

A Few Caveats about Web Inspector

- Changes you made in web inspector are NOT saved automatically
- Must make sure changes are saved by:
  - Going to Resources Tab -> Find your CSS file and copy paste all the code to your CSS file (Modification)
  - Going to Resources Tab -> Find inspector-style.css and copy paste all the code to your CSS file (Addition)

{ } INTRO TO WEB DESIGN

Hands-on Session 2
Styling my first website

{ } INTRO TO WEB DESIGN

CSS Properties

{ } INTRO TO WEB DESIGN

CSS Properties

- There are **a lot** of properties! We will teach you the most essential
- **color**
  - Changes the color of your text
  - Value is either a default color value (red, blue, etc.), RGB or HEX value (we will go over this next week)
- **background-color**
  - Changes the background color of your HTML element
  - Value is either a default color value (red, blue, etc.), RGB or HEX value (we will go over this next week)

{ } INTRO TO WEB DESIGN

CSS Properties

- **text-align**
  - Values: left, center, right, justify
- **text-decoration**
  - Values: underline, overline, line-through
- **font-size**
  - Value in px (ex: 16px or 24px)
  - Other ways to define font too (but we won't go over them)
- **font-weight**
  - Values: normal, bold, bolder, or lighter
  - Or use values from 100, 200, …, to 900
    - 400 = normal, 700 = bold

{ } INTRO TO WEB DESIGN

CSS Properties

Sample:

```
1   body {
2       color: #333;
3       text-align: center;
4   }
5   a {
6       text-decoration: none;
7   }
8   #box {
9       background-color: #339ccd;
10      color: white;
11  }
12
```

{ } INTRO TO WEB DESIGN

CSS pseudo-classes

{ } INTRO TO WEB DESIGN

CSS pseudo-classes

- Adds effects to your selectors
- Syntax:
  - *.selector:pseudo-class { …css styles … }*
  - *Ex: a:visited { … }, #box1:hover { … }*
- Common pseudo-classes:
  - *:visited* = Links you've visited
  - *:active* = Links when you click on them (happens in milliseconds)
  - *:hover* = When mouse is over element
  - *:focus* = When typing in the current <input> tag (we go over <input> tags later on)
- Remember *:hover*! Most likely you will use this the most ;)

{ } INTRO TO WEB DESIGN

CSS pseudo-classes

:hover

{ } INTRO TO WEB DESIGN

CSS pseudo-classes

:active (happens briefly)

Click this Link

Click this Link

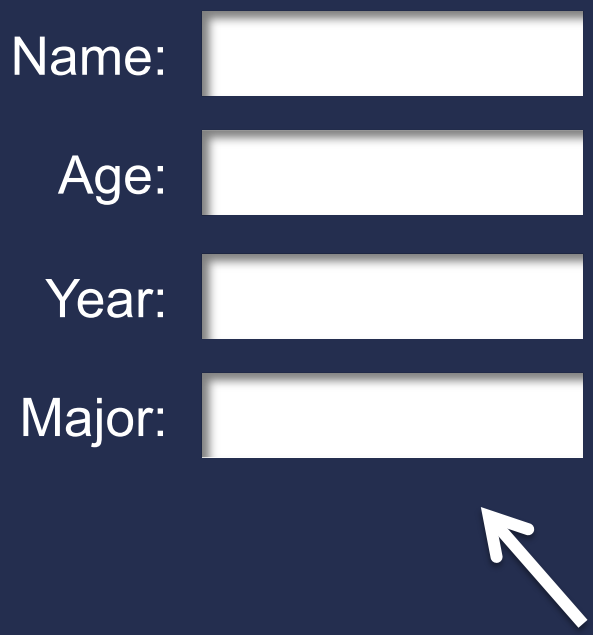
:visited

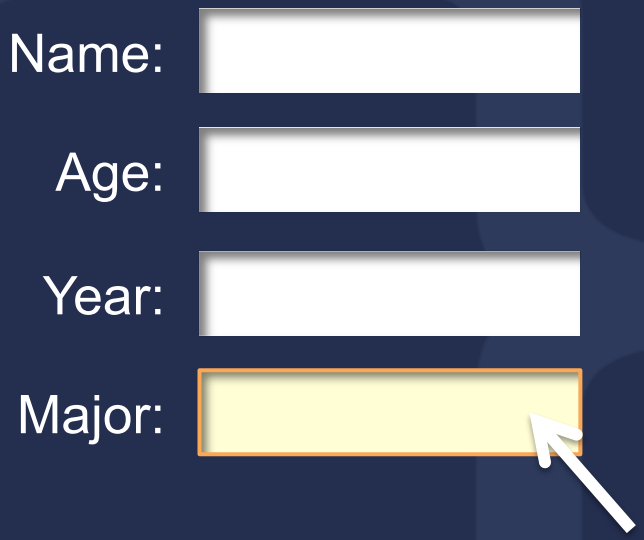
Click this Link

{ } INTRO TO WEB DESIGN

CSS pseudo-classes

:focus





{ } INTRO TO WEB DESIGN

CSS Properties – Font Family

- Before we talk about font-family, let's go over some basics
- There are **2 types** of fonts: **Serif** and **Sans-serif**

{ } INTRO TO WEB DESIGN

CSS Properties – Font Family

- On the web, we call commonly used fonts "**web-safe"**, meaning all browsers can render the fonts without trouble
- Here is a list of web-safe "**serif"** fonts:

Times New Roman, Times New Roman:
abcdefghijklmnopqrstuvwxyz / 1234567890

Copperplate Gothic Light
ABCDEFGHIJKLMNOPQRSTUVWXYZ / 1234567890

Georgia, Georgia; Georgia:
abcdefghijklmnopqrstuvwxyz / 1234567890

{ } INTRO TO WEB DESIGN

CSS Properties – Font Family

- Here is a list of web-safe "**sans-serif"** fonts:

Lucida Sans Unicode, Lucida Sans Unicode;
abcdefghijklmnopqrstuvwxyz / 1234567890

Gill Sans, Gill Sans; Gill Sans:
abcdefghijklmnopqrstuvwxyz / 1234567890

Verdana, Verdana; Verdana:
abcdefghijklmnopqrstuvwxyz / 1234567890

Arial, Arial, Arial, Arial:
abcdefghijklmnopqrstuvwxyz / 1234567890

{ } INTRO TO WEB DESIGN

CSS Properties – Font Family

- Let's see the CSS property for using a font-family
- **font-family**
  - Example: *font-family: Arial;*
- Usually there are more than one font-family value
  - Example: *font-family: Helvetica, Arial, Verdana, sans-serif;*
  - This means the font-family *Helvetica* will be used
  - If *Helvetica* is not supported by the browser, the next font is used
    - These are called **fallback** fonts
    - The last fallback font in this example is sans-serif (what the browser chooses as the default sans-serif)
- If no fallback fonts or no font defined, default serif font used

{ } INTRO TO WEB DESIGN

CSS Properties – Font Family

```
1.  body {  /*the outer layer of the site and global style attributes*/
2.          font-family: "Museo Sans", "Open Sans", "Helvetica Neue", Helvetica, Arial, sans-serif;
3.          margin: 0px;
4.  }
5.  a {
6.          color: inherit;
7.          text-decoration: none;
8.          outline: none;
9.  }
10. .img_scaled {
11.         width: 100%;
12. }
13. #logo {
14.         height: 50px;
15.         float: left;
16. }
17. #header_container {
18.         position: absolute;
19.         height: 80px;
20.         width: 100%;
21.         z-index: 100;
22. }
```

{ } INTRO TO WEB DESIGN

CSS Properties – Font Family

- Lastly, there is border
- **border**
  - Takes in 3 values, space-separated
  - 1st value: border-width in px (pixels)
  - 2nd value: border-style
  - 3rd value: color
- Example: *border: 1px solid #999;*
  - Thin solid gray border around your element

{ } INTRO TO WEB DESIGN

CSS Box Model

{ } INTRO TO WEB DESIGN

CSS Box Model

- The **CSS Box Model** is the standard way of structuring your elements and web pages. It allows us to add *space* and *borders* to an element

{ } INTRO TO WEB DESIGN

CSS Box Model

Facebook makes use of the Box Model everywhere

{ } INTRO TO WEB DESIGN

CSS Box Model

- 4 Key components
  - Content
  - Padding
  - Border
  - Margin
- **Content** is everything inside an element
  - Example: In <*p*>*Hello World*</*p*>, “Hello World” is the content
- **Border** is the line that surrounds an element
- **Padding** is the space between the content and border
- **Margin** is the space from the border and nearby elements

{ } INTRO TO WEB DESIGN

CSS Box Model

- We went over border last week
- **border**
  - Takes 3 values: width, style, color
  - Example: *border: 1px dashed blue;*
- Padding and Margin can take in 1, 2, 3 or 4 values
- **1 Value:**
  - *padding: 10px;* = 10 pixels of space all around, from content to border
- ***2 Values:***
  - *margin: 5px 10px;* = 5 pixels of space above and below border, 10 pixels of space to the left and right of border
  - Corresponds to top & bottom, left & right

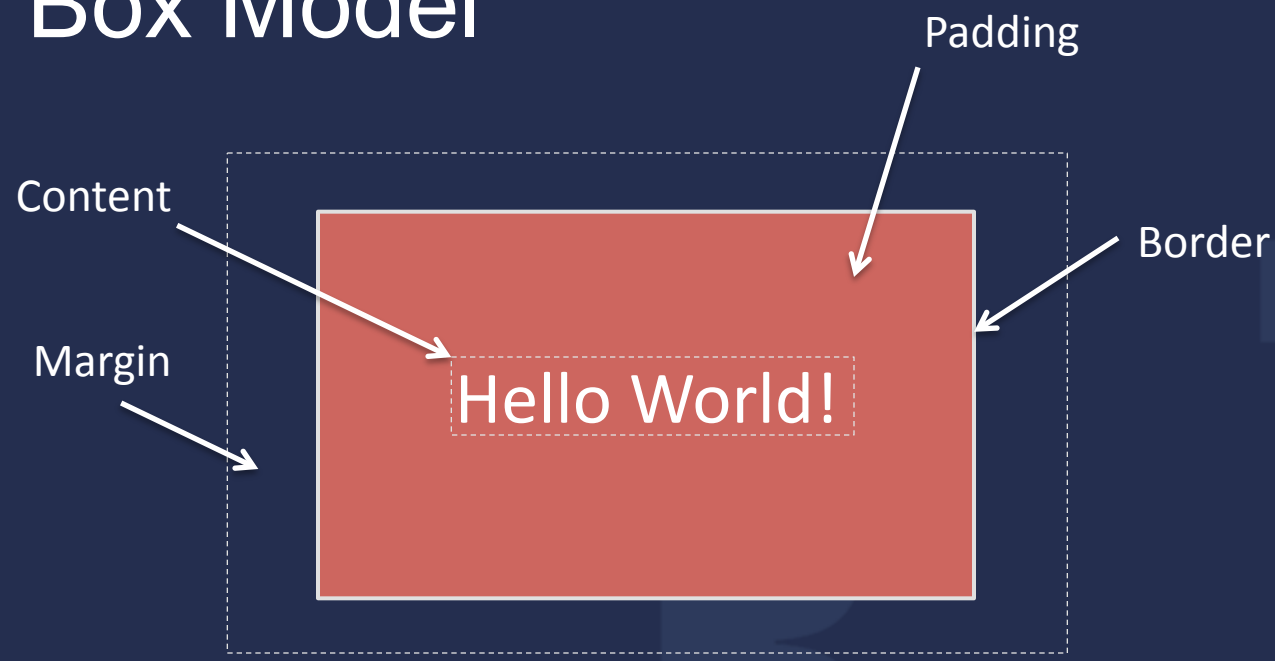
{ } INTRO TO WEB DESIGN

CSS Box Model

- **3 Values:**
  - *padding: 10px 5px 20px; = Corresponds to top, left & right, bottom*
- **4 Values:**
  - *margin: 10px 5px 20px 15px; = Corresponds to top, right, bottom, left (clockwise)*

{ } INTRO TO WEB DESIGN

CSS Box Model



- In this example, a possible style for the element is:
  - *padding: 15px;*
  - *border: 2px solid #ccc; /* light-gray */*
  - *margin: 5px 10px;*

{ } INTRO TO WEB DESIGN

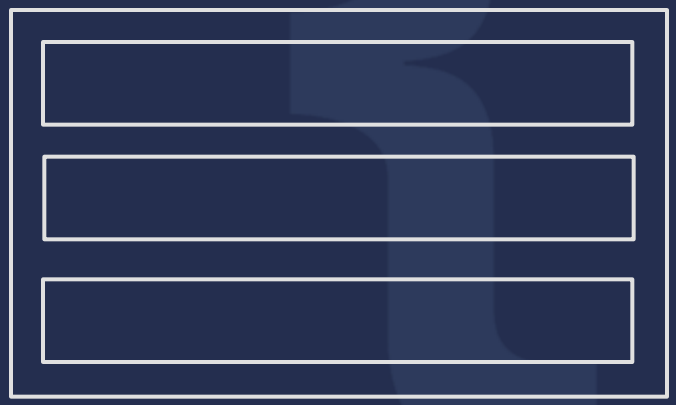
CSS Box Model

- Also know that if you add a *background-color* to your image, it will color in the **content** and the **padding** portions of the element
- *Background-color* will not affect the **margin**!!!

{ } INTRO TO WEB DESIGN

CSS Box Model

- Currently we have divs that stack vertically. What does this mean?
- **Divs** are, by default, *blocks*
  - They take up the entire row
  - By default, you **cannot** stack divs side by side
  - They look like this:



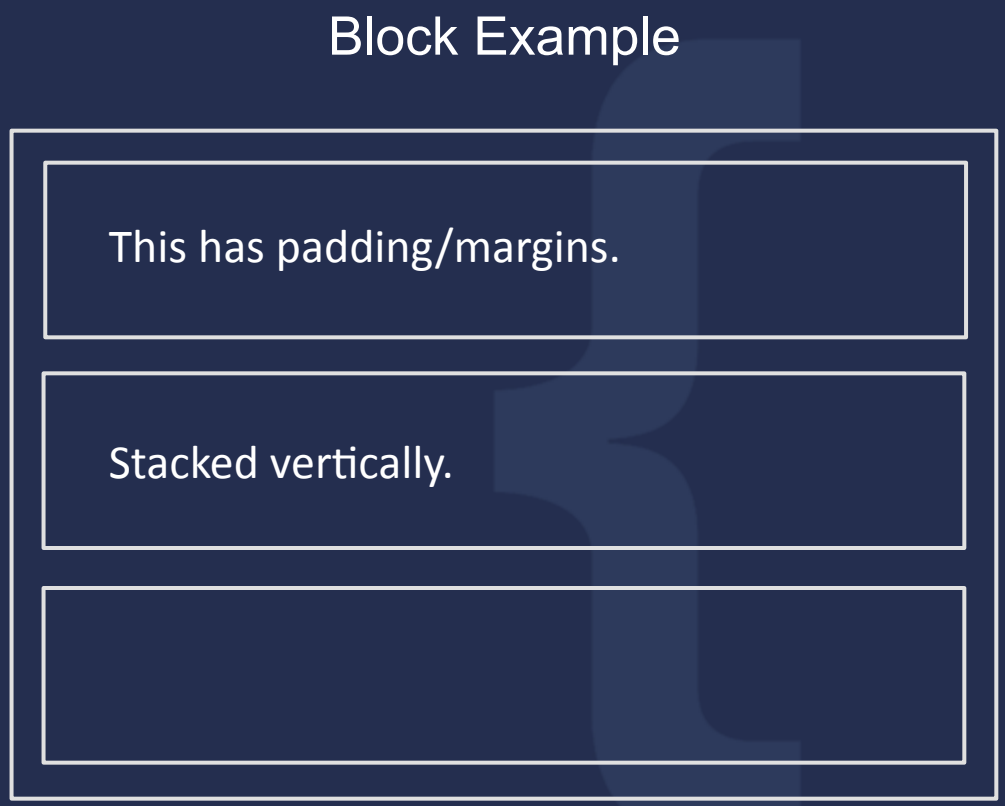
{ } INTRO TO WEB DESIGN

CSS Box Model

- In order to stack horizontally, you must alter the CSS style for *display*
- **display**
  - There are 3 important values for display: *block, inline-block, inline*
- **block**
  - “block” displays respect all margins, paddings, height & width
  - Except, they have an auto-line break after it (next element goes directly below it)
- **inline-block**
  - Exactly like a block **but** it has no line break after it
- **inline**
  - Can have left & right margins/paddings, but *no top & bottom* ones. Also has no height/width. Allows elements next to them

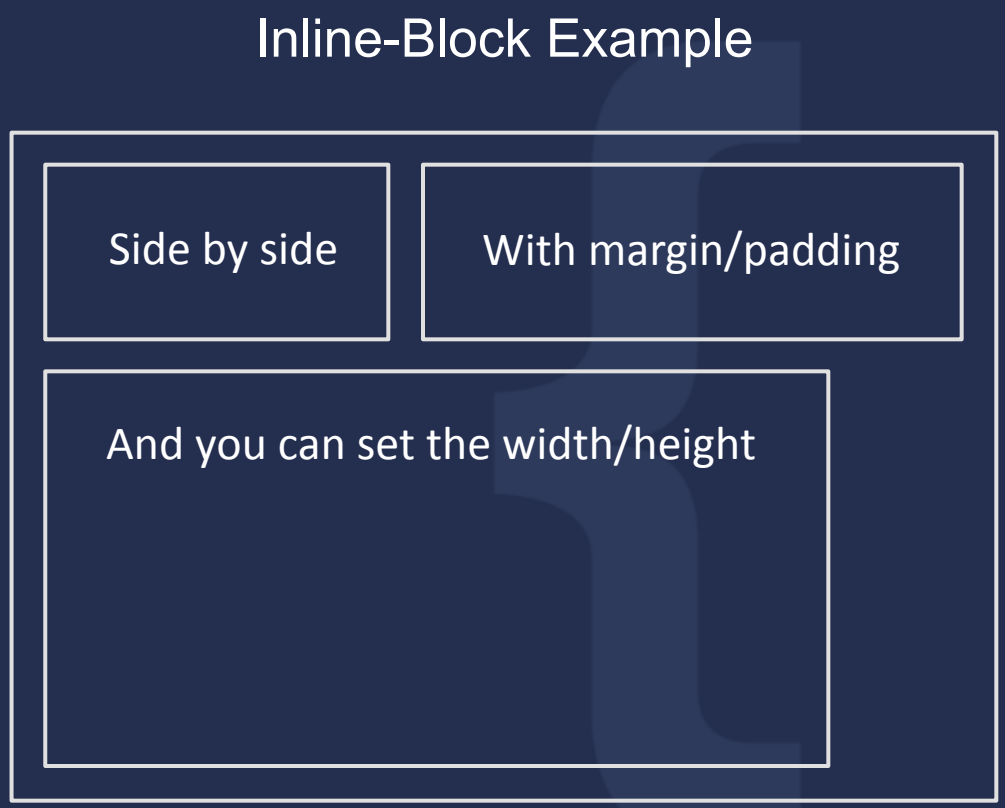
{ } INTRO TO WEB DESIGN

CSS Box Model



{ } INTRO TO WEB DESIGN

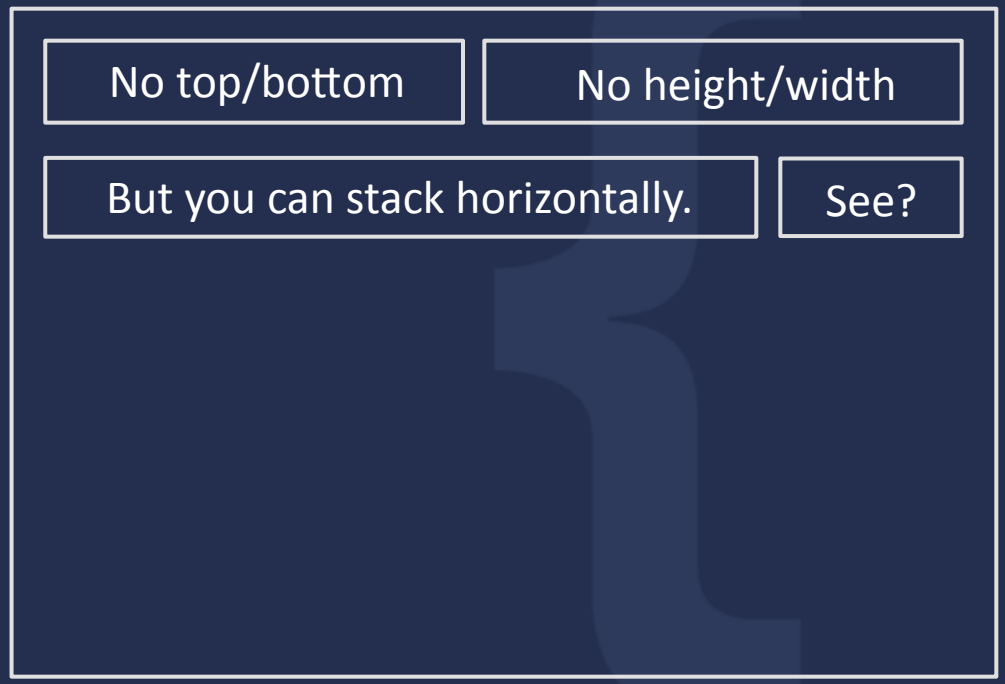
CSS Box Model



{ } INTRO TO WEB DESIGN

CSS Box Model

Inline Example (essentially a tag)



{ } INTRO TO WEB DESIGN

CSS Box Model - Examples

Inline Example (essentially a tag)

The quick **brown fox** jumps over the lazy dog.

```
<div>
    The quick <span id="bold">brown fox</span> jumps over the lazy dog.
</div>
```

{ } INTRO TO WEB DESIGN

CSS Box Model - Examples

What happens if instead of , you use ?

```
<div>
    The quick <div id="bold">brown fox</div> jumps over the lazy dog.
</div>
```

The quick
brown fox
jumps over the lazy dog.

{ } INTRO TO WEB DESIGN

CSS Box Model - Examples

Now if add the CSS properties to id="bold":
display: inline-block & padding: 0 40px

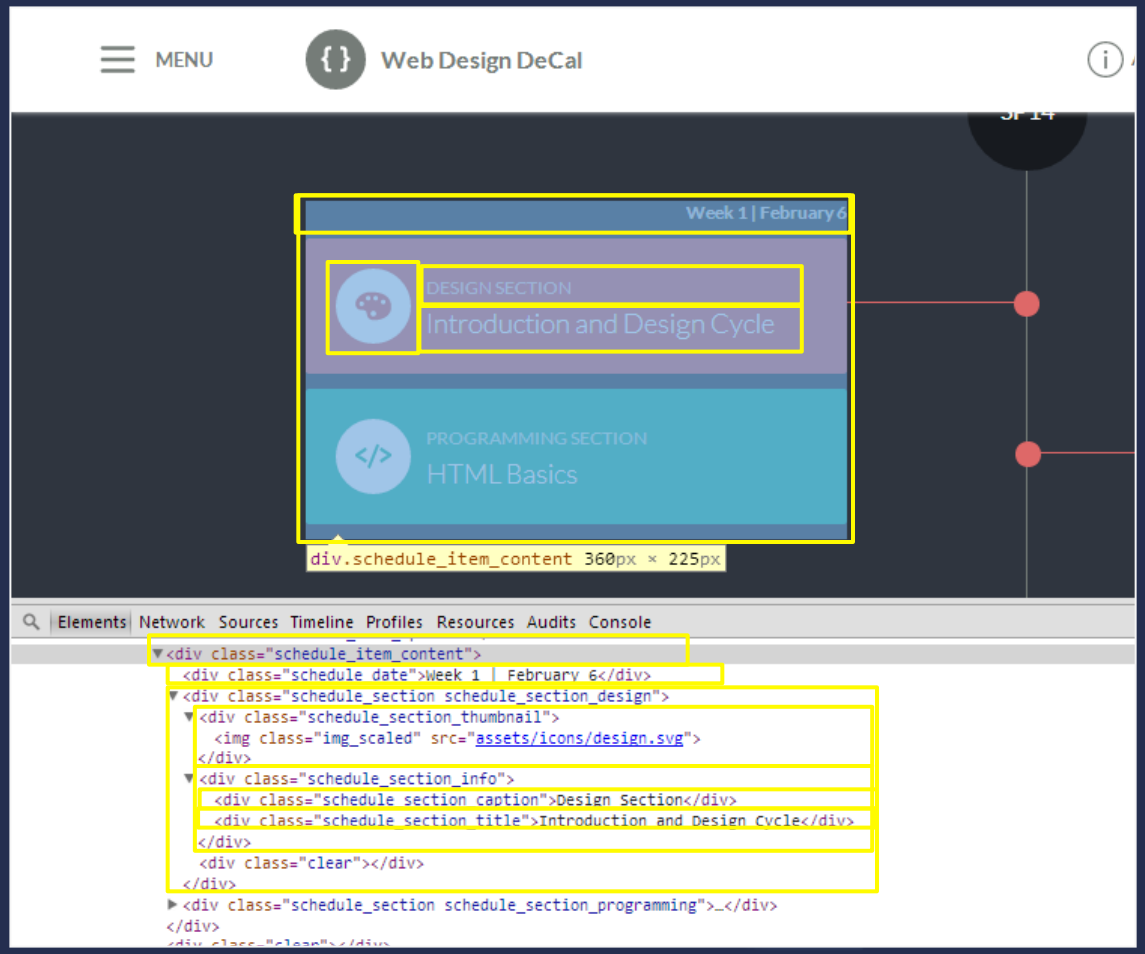
```
<div>
    The quick <div id="bold">brown fox</div> jumps over the lazy dog.
</div>
```

The quick **brown fox** jumps over the lazy dog.

You can add width, padding, and margins to your elements! Amazing!

{ } INTRO TO WEB DESIGN

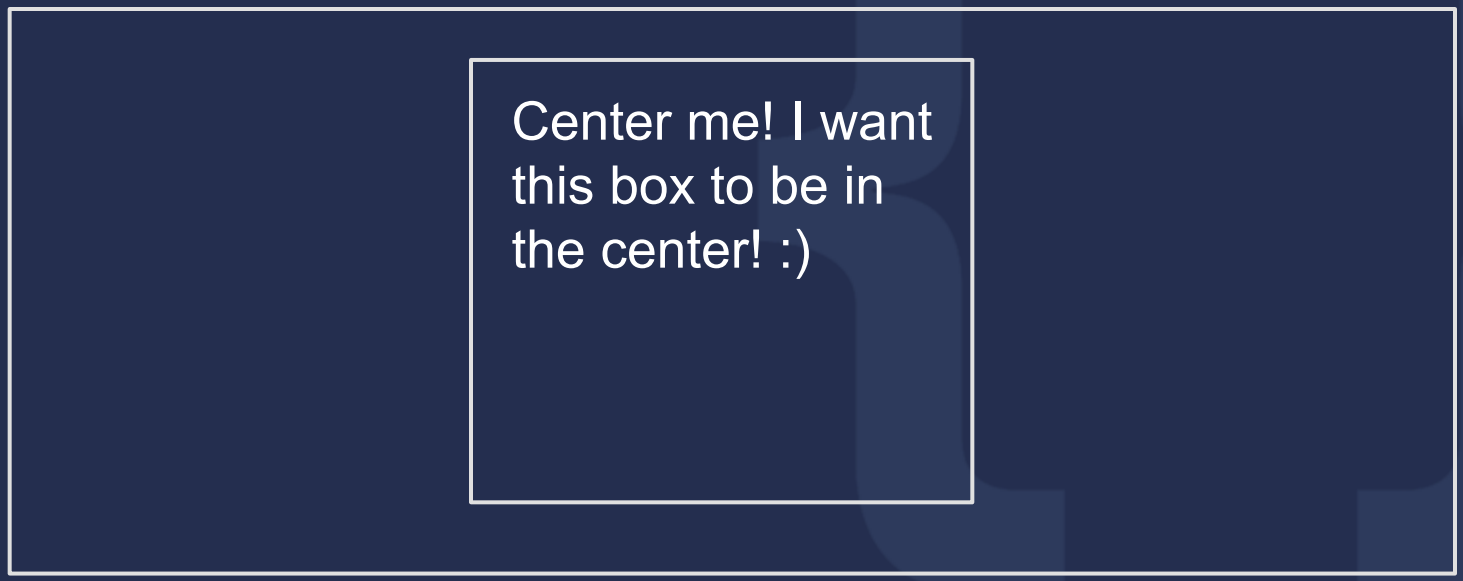
CSS Box Model



{ } INTRO TO WEB DESIGN

Centering a Div

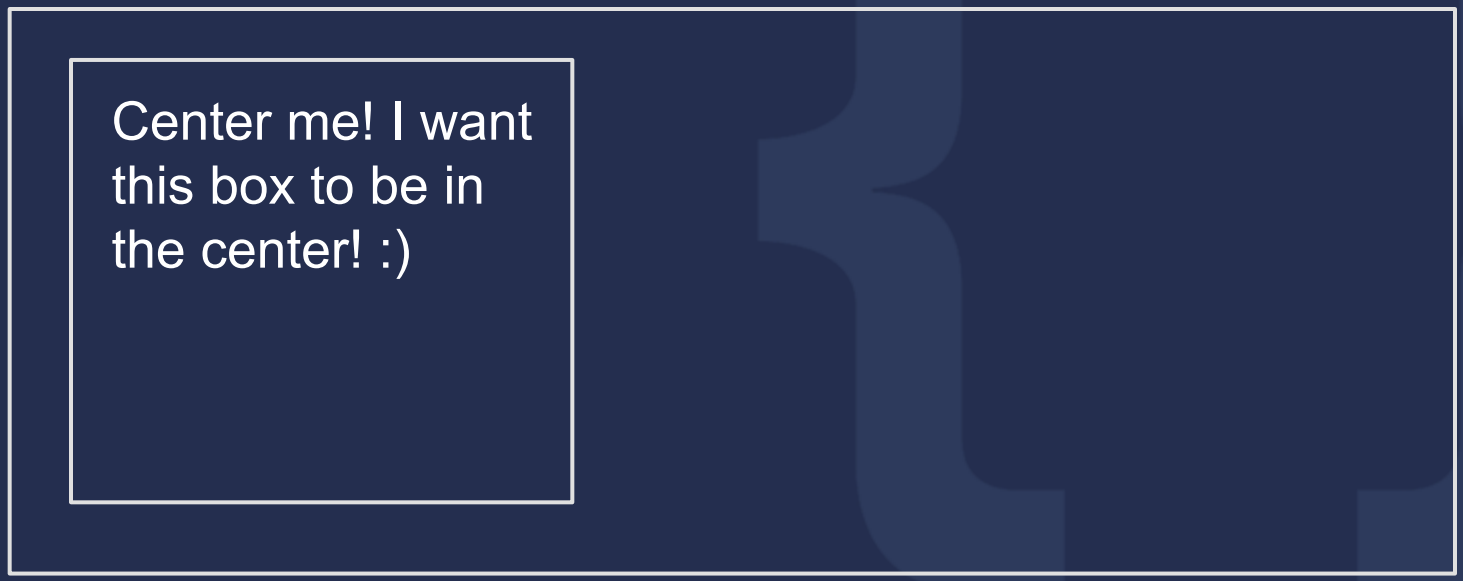
- **Text or Div Element**
  - Centering text is easy, use *text-align: center*
  - What if you want to center a that has a certain width?



{ } INTRO TO WEB DESIGN

Centering a Div

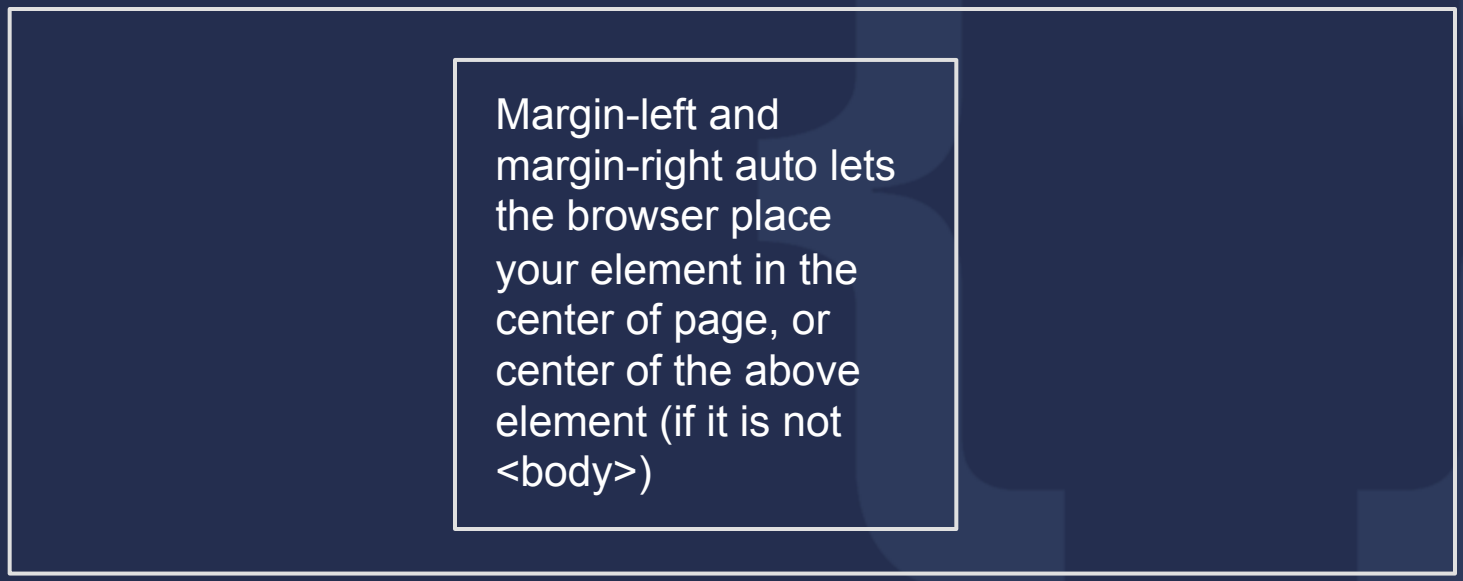
- **Text or Div Element**
  - By default, if you give your a width (say 200px), it will automatically be on the left



{ } INTRO TO WEB DESIGN

Centering a Div

- **Text or Div Element**
  - To center a , add 2 CSS properties to the element:
    - *margin-left: auto; margin-right: auto;*



{ } INTRO TO WEB DESIGN

Centering an Image

- **Image**
  - What about centering an image? (*<img> tags)*
  - Again, by default it is on the left

{ } INTRO TO WEB DESIGN

Centering an Image

- **Image**
  - To center an image, like text, add *margin-left/right to auto*
  - Also, specifically add *display: block*

{ } INTRO TO WEB DESIGN

Centering an Image

- **Image**
  - Remember that elements by default are *display: block*
  - <img> tags by default are not *display: block*, so we must type in *display: block* manually, to tell the browser that the image takes up the entire row

{ } INTRO TO WEB DESIGN

CSS Color

{ } INTRO TO WEB DESIGN

CSS Color

- Color is important for font colors, background colors, borders, etc.!
- **Color** on the web can be represented in 3 ways:
  - A default color value
  - Hex Value
  - RGB Value
- What do these mean?
- **Default Color**
  - 16 pre-defined CSS colors
  - Example: *red, blue, black, white, maroon, etc.*
  - Too limited may not be what you want!

{ } INTRO TO WEB DESIGN

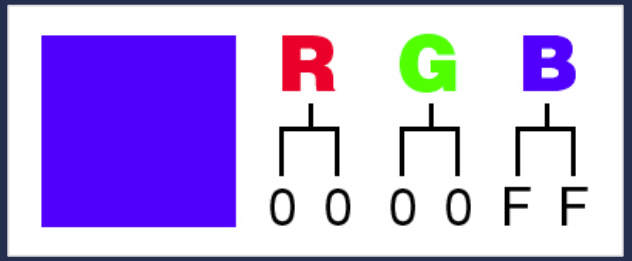
CSS Color

- **Hex Color** (Hexadecimal)
  - *Millions* of ways to define a color!
  - Syntax: Pound sign **#** followed by 6 digits/characters from 0 to 9 and A to F
  - Example: *#FF0000* (red), *#339CCD (light blue), #888888 (gray)*

0-9 & A-F = 16 possible values
16*16*16*16*16*16 = *16 Million Ways To Define 1 Color!*

{ } INTRO TO WEB DESIGN

CSS Color - Hex



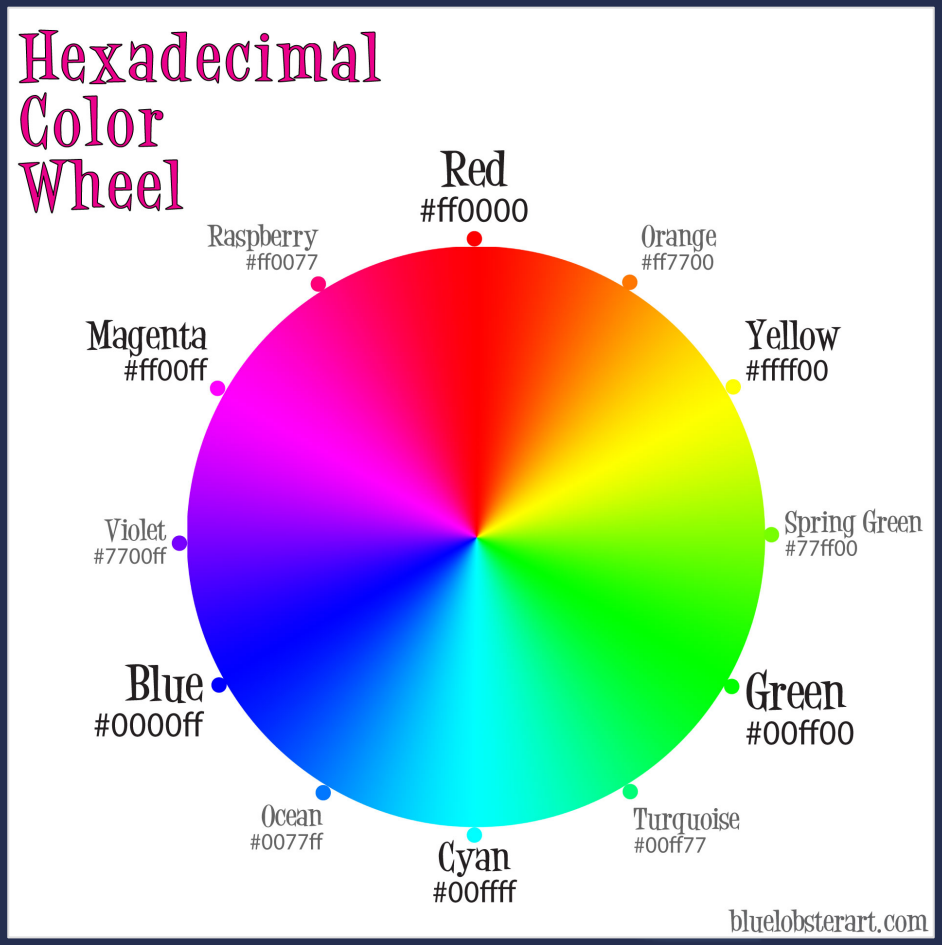
- Hex values are 6 digits, or 3 bytes
- Each byte is 2 digits and represents a “color”
- **Red** corresponds to the 1st byte, **Green** to the 2nd, **Blue** to the 3rd
- A “0” indicates *no color*. Increasing the value to 1, 2, etc. increases the color. An “F” indicates *full color (lightest)*
  - Example: *#0000FF*
  - Equivalent to “no reds”, “no greens”, “full blues” = Pure Blue
- What is “Purple”?
- **Answer:** #FF00FF! It is a mix of pure red and pure blue. Other shades may include #AA00AA or #330033.

{ } INTRO TO WEB DESIGN

CSS Color - Hex

- Screens and monitors are black by default
- When you have #000000 (no color), this means you get **black**
- When you have #FFFFFF (all colors), you get **white**
- Thus, some red on a black surface = dark red: *#330000*
- Another fun tip! If a color's byte has 2 repeating digits (88, FF, 00), and all 3 bytes have repeating digits, you can do a **shorthand** by using the single digit
  - Example: *#FF0000 → #F00*
  - *#CC88DD → #C8D*
  - *#000000 → #000*

CSS Color - Hex



{ } INTRO TO WEB DESIGN

CSS Color - RGB

- **RGB colors**
  - Syntax: *rgb(255, 0, 0)*
  - *rgb(…)* takes in 3 values: red, green, blue
  - Similar to Hex. 0 in rgb is 00 in hex, 255 in rgb is FF in hex
  - Examples:
    - *rgb(255, 0, 0) vs. #FF0000*
    - *rgb(51, 156, 205) vs #339CCD*
- Which to use? RGB or Hex?
  - Hex is great, supported by all browsers
  - Hex is used excessively in Photoshop, Illustrator, etc.
  - Easy to memorize over digits

{ } INTRO TO WEB DESIGN

Hands-on Activity

{ } INTRO TO WEB DESIGN

CSS Positioning

- 4 types of positioning in CSS:
  - position: static
  - position: relative
  - position: absolute
  - position: fixed
- By default, all your divs and elements are *position: static*

{ } INTRO TO WEB DESIGN

position: static

- Default position, follows the natural flow of the page
- **Cannot** use offsets: top, left, bottom, right

{ } INTRO TO WEB DESIGN

position: static

- Use position: static (default behavior) to stack elements vertically

{ } INTRO TO WEB DESIGN

position: relative

- Offsets element using position values (top, left, right, bottom)

Example: Set top to -10px, left to -10px

{ } INTRO TO WEB DESIGN

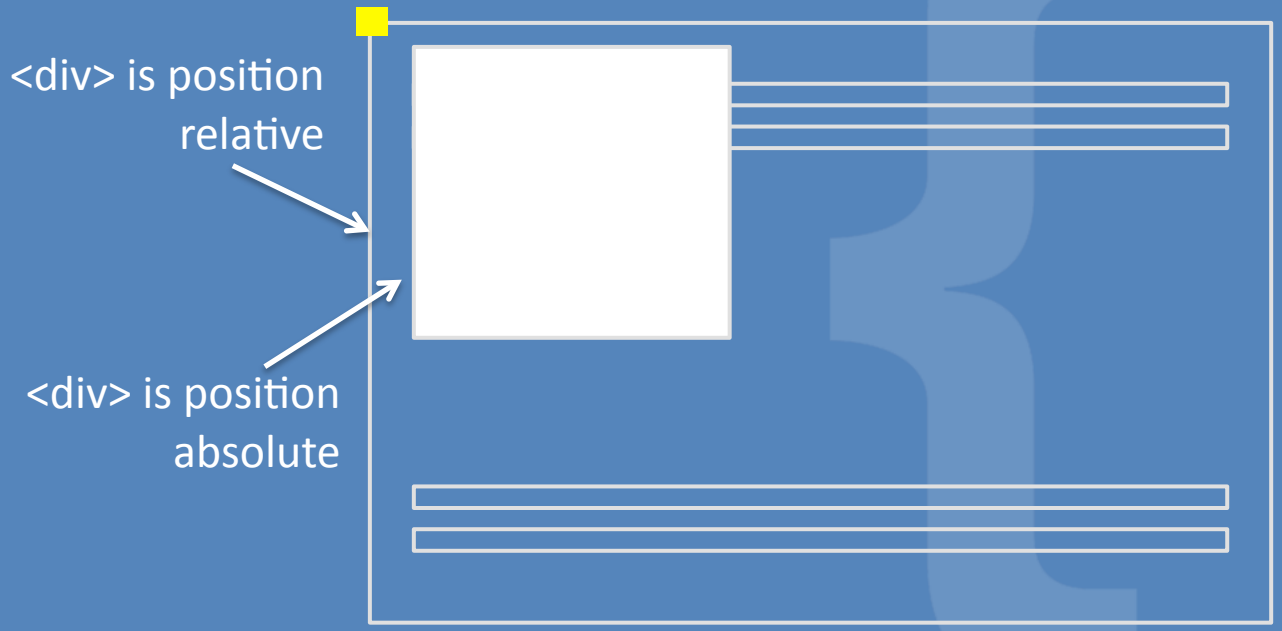
position: relative

- Use position: relative to provide a reference point for elements inside this div that will be absolutely positioned.

Example: Set top to -10px, left to -10px

position: absolute

- Element is moved relative to first non-static parent (default: *top: 0, left: 0*)



Example: Set left to 10px, top to 5px

{ } INTRO TO WEB DESIGN

position: absolute

- Element is moved relative to first non-static parent element

```
<div id="container">
        <div id="content">
                Hello World!
        </div>
</div>
```

```
#container { position: relative; }
#content {
        position: absolute;
        top: 20px;
        left: 20px;
}
```

{ } INTRO TO WEB DESIGN

position: absolute

- If no *non-static* parent element found, move relative to <body>

```
<body>
        <div id="content">
                Hello World!
        </div>
<body>
```

```
#content {
        position: absolute;
        top: 20px;
        left: 20px;
}
```

{ } INTRO TO WEB DESIGN

position: absolute

- Example of *div* relative to *div* relative to *div* (*div-ception!!)*

```
<div id="container">
        <div id="content">
                <div id="inner-content">Hello World!</div>
        </div>
</div>
```

```
#container { position: relative; }
#content { position: absolute; top: 20px; left: 20px;}
#inner-content { position: absolute; top: 30px; left: 10px; }
```

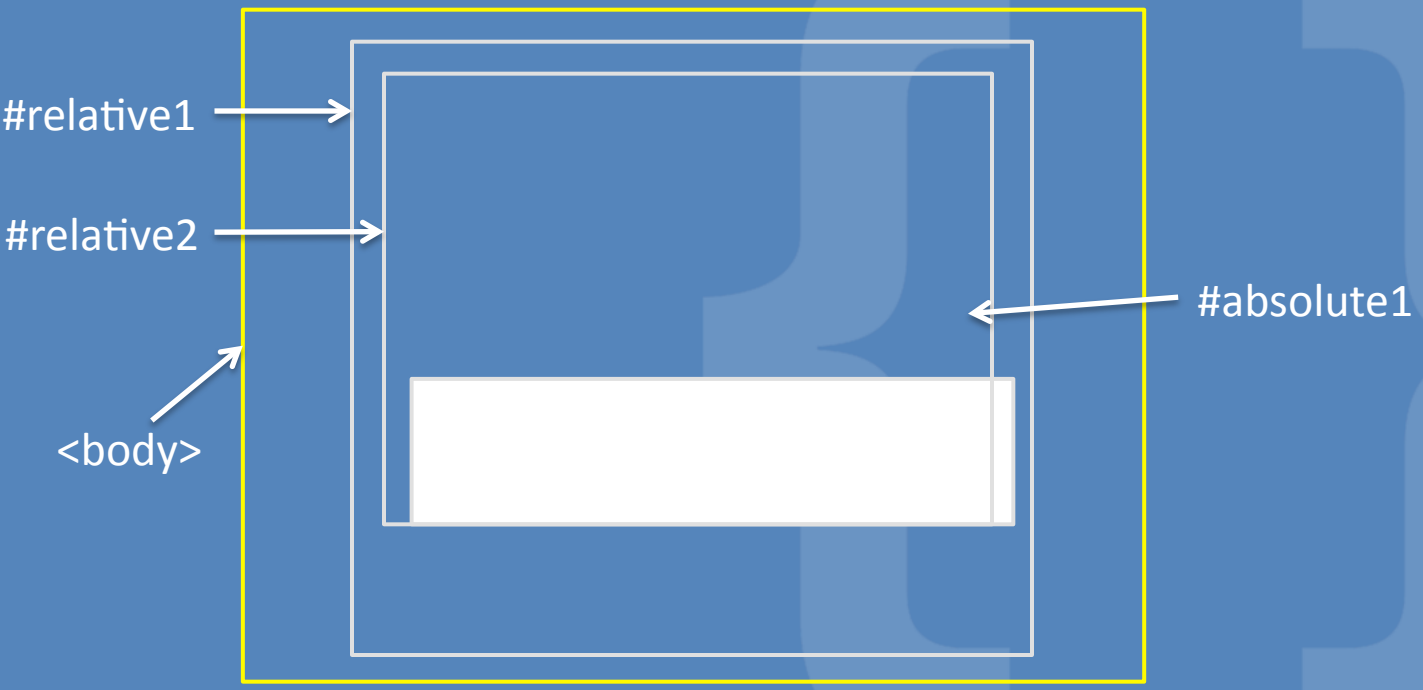
#inner-content will be **relative** to #content

So in total, it is 50px from top of #container, 30px left of #container

position: absolute

```
<body>
   <div id="relative1">
      <div id="relative2">
         <div id="absolute1"> ….
```

What are possible top, right, bottom, left values for #absolute1?



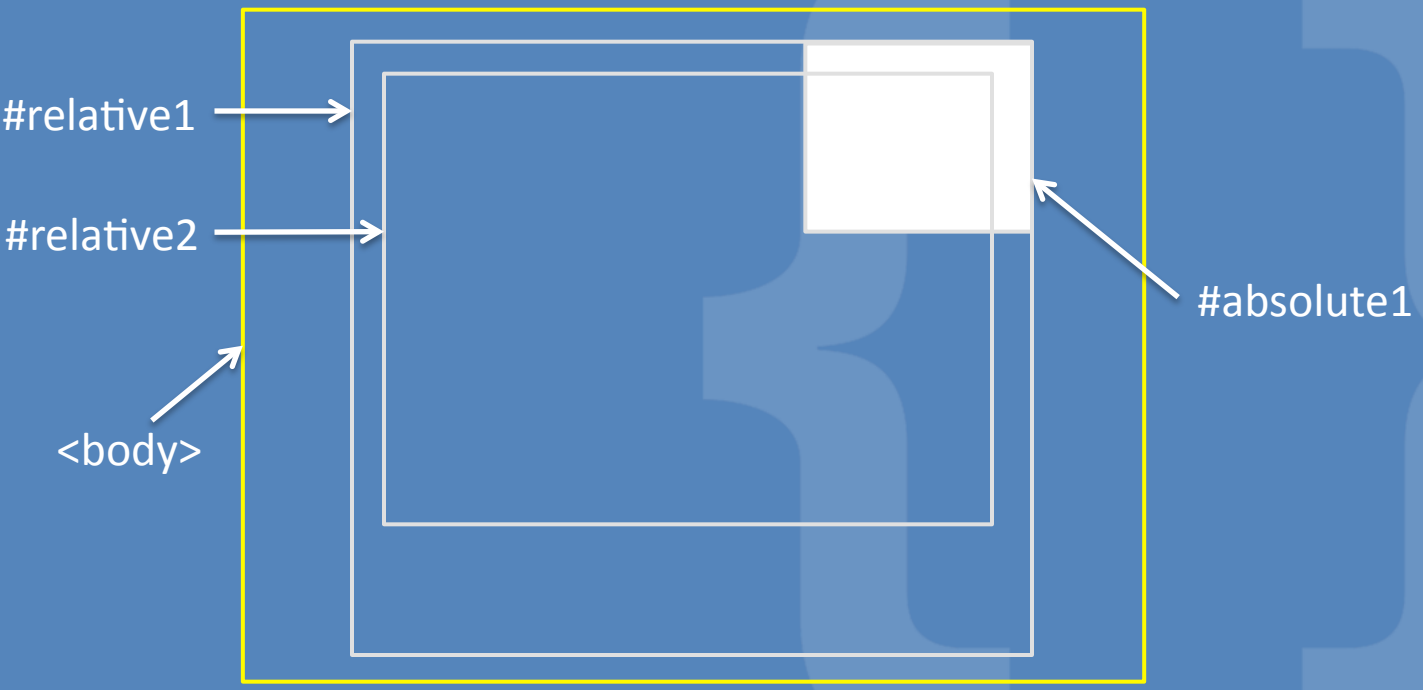
#absolute1 is moved relative to #relative2.
So… bottom: 0, left: 5px (example)

{ } INTRO TO WEB DESIGN

position: absolute

```
<body>
   <div id="relative1">
      <div id="absolute1">
         <div id="relative2"> ….
```

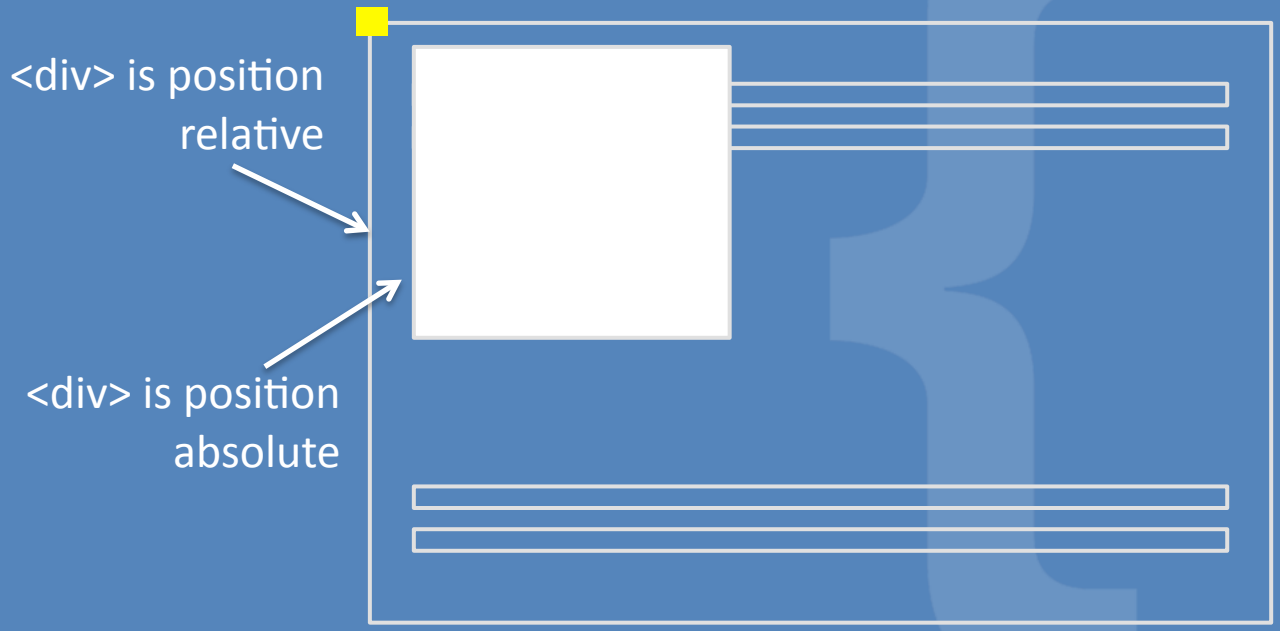
What are possible top, right, bottom, left values for #absolute1?



#absolute1 is moved relative to #relative1.
So… top: 0, right: 0 (don't even need to type px)

position: absolute

- Use position: absolute to manually place an element to a desired location, or to place elements on top of each other.



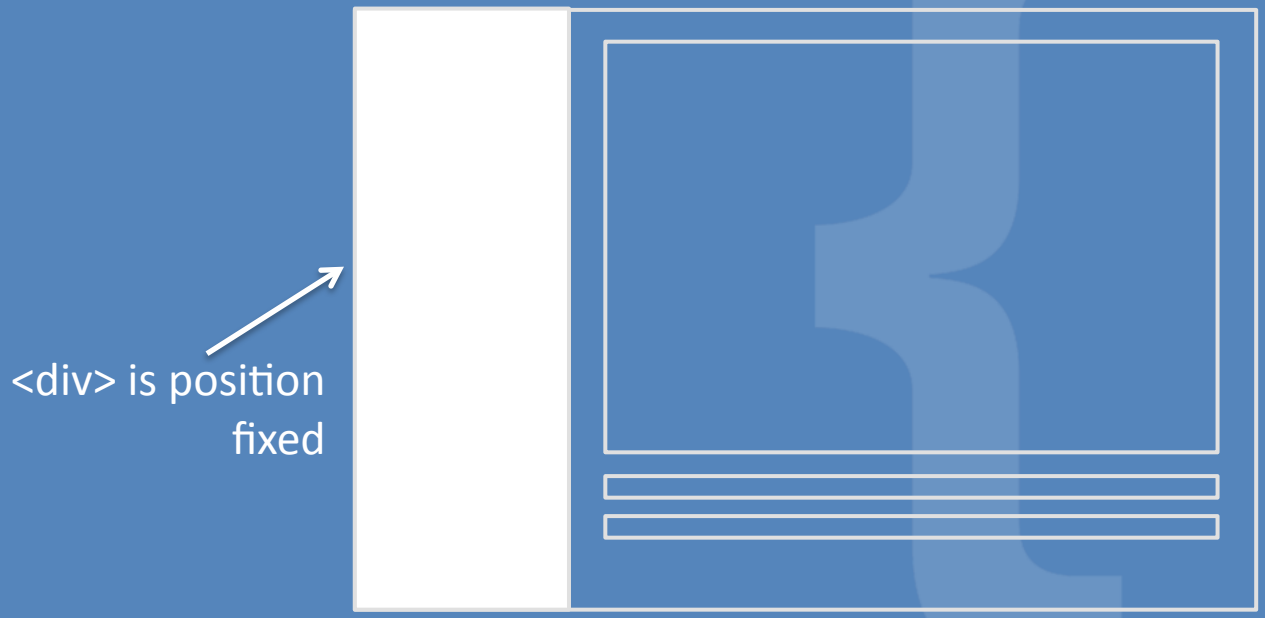
Example: Set left to 10px, top to 5px

{ } INTRO TO WEB DESIGN

Hands-on Session 3
Positioning Haven I

position: fixed

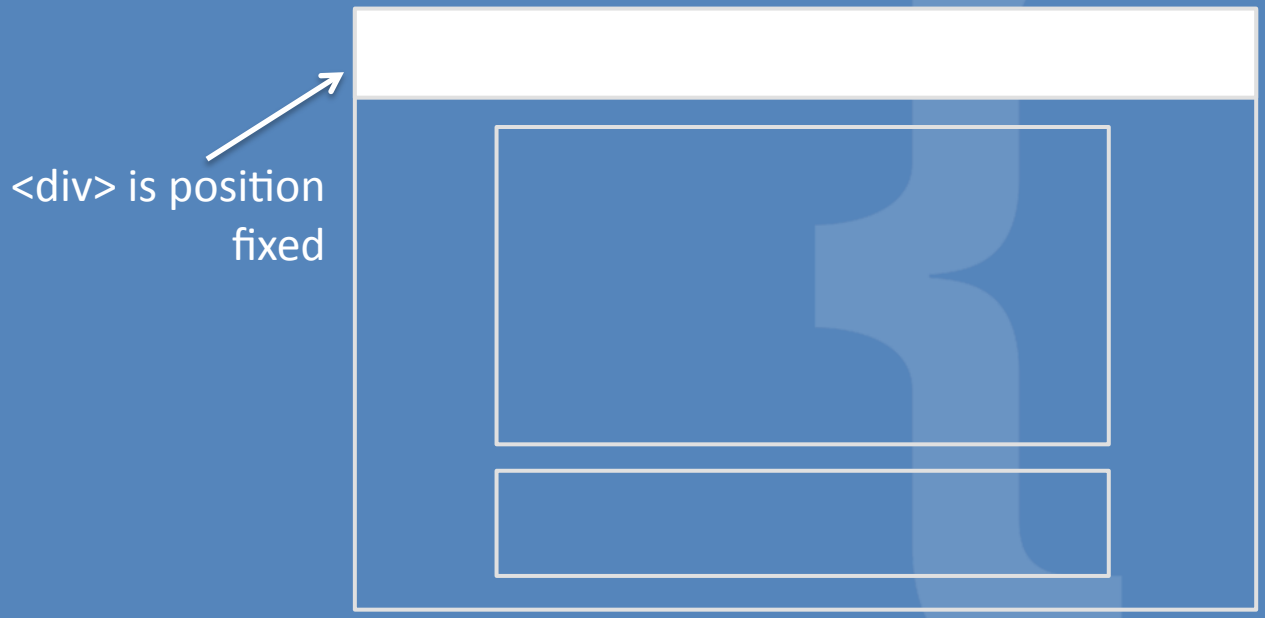
- Element stays on the screen, even when scrolling. Positioned relative to browser window



Example: Set left to 10px, top to 5px

position: fixed

- Element stays on the screen, even when scrolling. Positioned **relative to browser window**



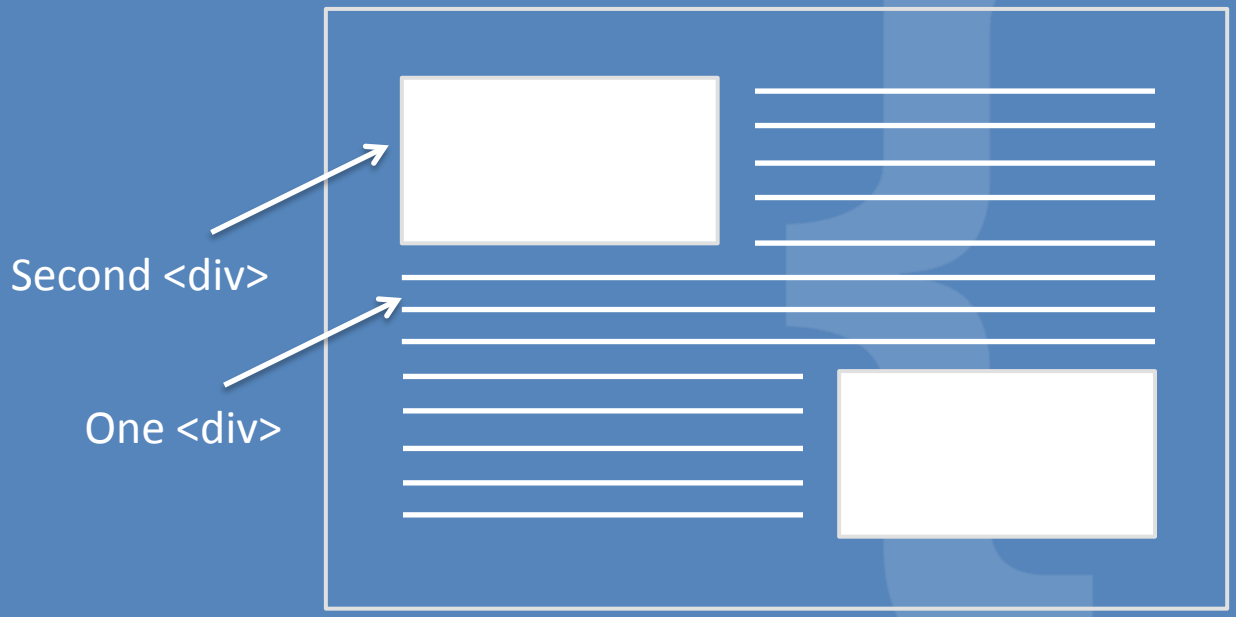
{ } INTRO TO WEB DESIGN

Floats

{ } INTRO TO WEB DESIGN

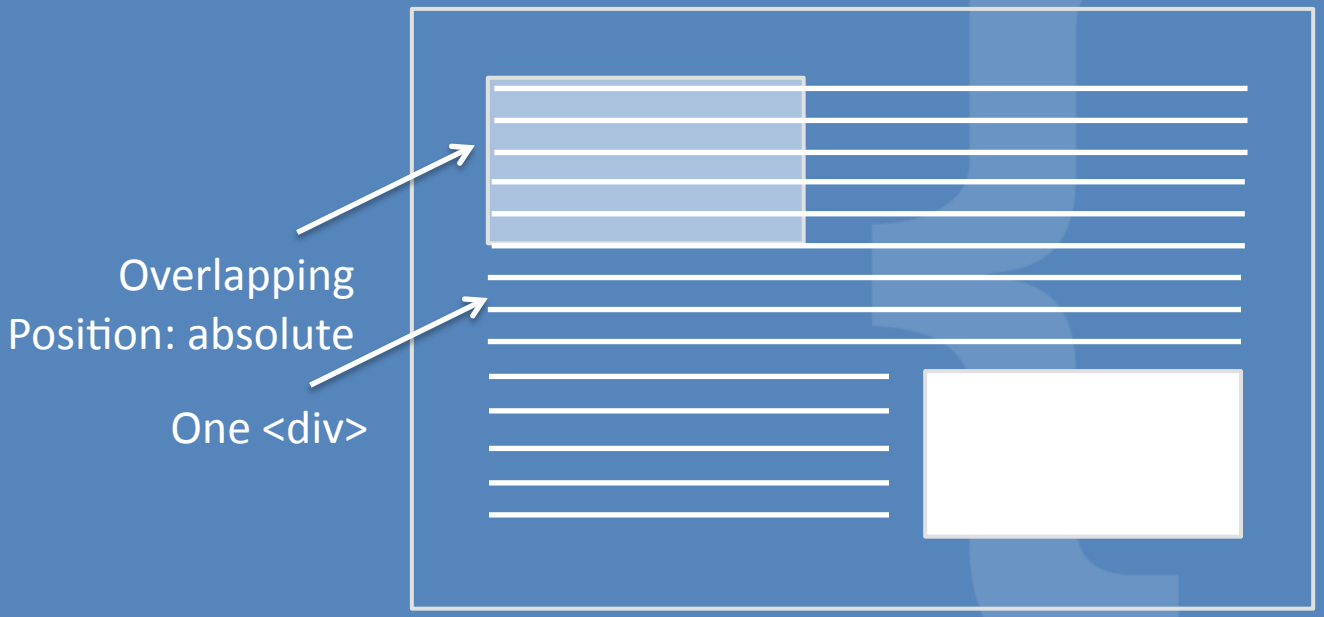
Float

- What is a *float*?
  - They allow your elements to be part of *the flow* on the page



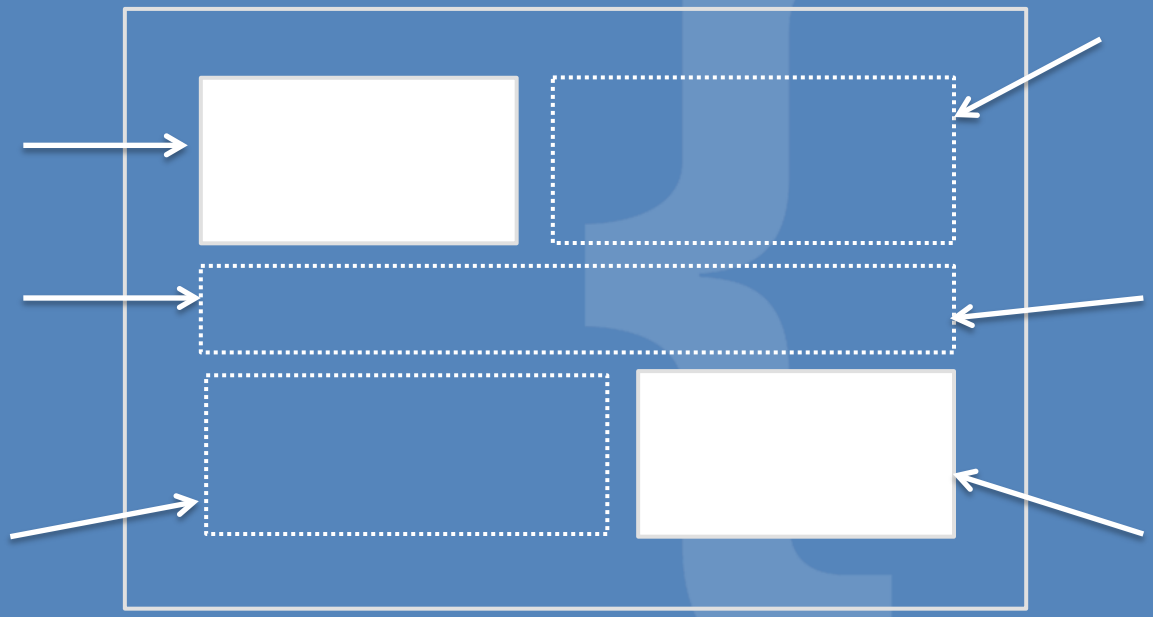
Float

- If you made one div *position: absolute*…



Float

- If you were to recreate the page with <*div*> elements set to *display: inline-block…*



Not efficient!

{ } INTRO TO WEB DESIGN

Float

- *float*
  - 2 properties
    - *float: left*
    - *float: right*
- Even if an element is a <*div*> or has *display: block*, float will essentially bypass these properties and act like *display: inline-block*, but automatically floating left or floating right
- The following element will be next to the floated <*div*>

{ } INTRO TO WEB DESIGN

Float

No floats, just

{ } INTRO TO WEB DESIGN

Float

First box, *float: left*

{ } INTRO TO WEB DESIGN

Float

One Use Case: Menus

- Home
- About
- Experiences
- Contact

{ } INTRO TO WEB DESIGN

Float

Floating Menu Items (each with set *width*)

Home | About | Experiences | Contact

Notice no gaps!

If you use *display: inline-block* for these 4 elements:

Home | About | Experiences | Contact

display: inline-block produces gaps on the sides

{ } INTRO TO WEB DESIGN

Float

If you use *display: inline-block* for these 4 elements:

Home | About | Experiences | Contact

display: inline-block produces gaps on the sides

- Work-around:
  - Add *margin-right: -4px*
  - *Or…*

Awkward! Use margin-right: -4px, or floats!

```
<div>Home</div
>About</div
>Experiences</div
>Contact</div>
```

```
<div>Home</div><!--
--><div>About</div><!--
--><div>Experiences</div><!--
--><div>Contact</div>
```

{ } INTRO TO WEB DESIGN

CSS Clear

- *clear*
  - *clear: both*
    - Individually, *clear: left* and *clear: right*
- If one element floats left, the following element will appear beside it
  - Float has a tendency to continue stacking the next element – we usually don't want this
  - Use clear: both after each series of horizontally stacked element via float.
- Use *clear: both* for the following element to **break** the float, so it appears below the floating element

{ } INTRO TO WEB DESIGN

CSS Clear

First box, *float: left*

{ } INTRO TO WEB DESIGN

CSS Clear

Second box, using *clear: both*

{ } INTRO TO WEB DESIGN

CSS Clear

- Tip: Create a class called .clear that you can use throughout your website!

```
.clear {clear: both}
```

{ } INTRO TO WEB DESIGN

Horizontal stacking

Home	About	Experiences	Contact

- Give float: left to each of the horizontally stacked element
- Add clear: both element at the very end
- Think of clear:both as a period mark!

```
<div class="menu-item">Home</div>
<div class="menu-item">About<div>
<div class="menu-item">Experiences</div>
<div class="menu-item">Contact</div>
<div class="clear"></div>
```

{ } INTRO TO WEB DESIGN

Hands-on Session 4
Positioning Haven II

{ } INTRO TO WEB DESIGN

CSS vertical centering

- Unfortunately, vertical centering in CSS is not intuitive
- You must carefully use position: absolute to make this work
  - Position: absolute
  - Top: 50%
  - Margin-top: - ((height of an element) / 2) px
- Since CSS cannot compute by itself, you must calculate the negative margin on your own. (or via javascript – do it only if you have dynamic elements)

{ } INTRO TO WEB DESIGN

Divide and Conquer

- Think about your website into individual components, and design them separately in order.
- Remember:
  - Horizontal stacking – Floats
  - Vertical stacking – Does automatically
  - On top of each other – position absolute
- Each individual component will be its own div
- Master this and you can take “any design” into code!

{ } INTRO TO WEB DESIGN

What should you design first?

STREAM ACTIVITY COLLECTIONS SOUNDBOX

Superman: Man of Steel

EXPLORE

Profile Settings
Statistics
Security & Privacy
Logout

Fabio Basile
8:34 AM
Seeking Contributions


Game of Thrones
S03 E05

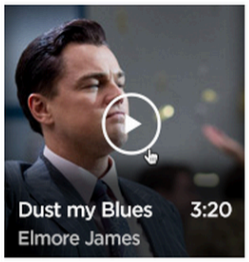

The Hangover 2
2011

This is the End
2013


2Guns
2013

COLLECT
GALLERY 8

Charlie Waite
7:15 AM



Wolf of Wall Street
2013


Breaking Bad
S01 E03


Elysium
2013


Ice Soldiers
2013

COLLECT
GALLERY 12

- 1.35

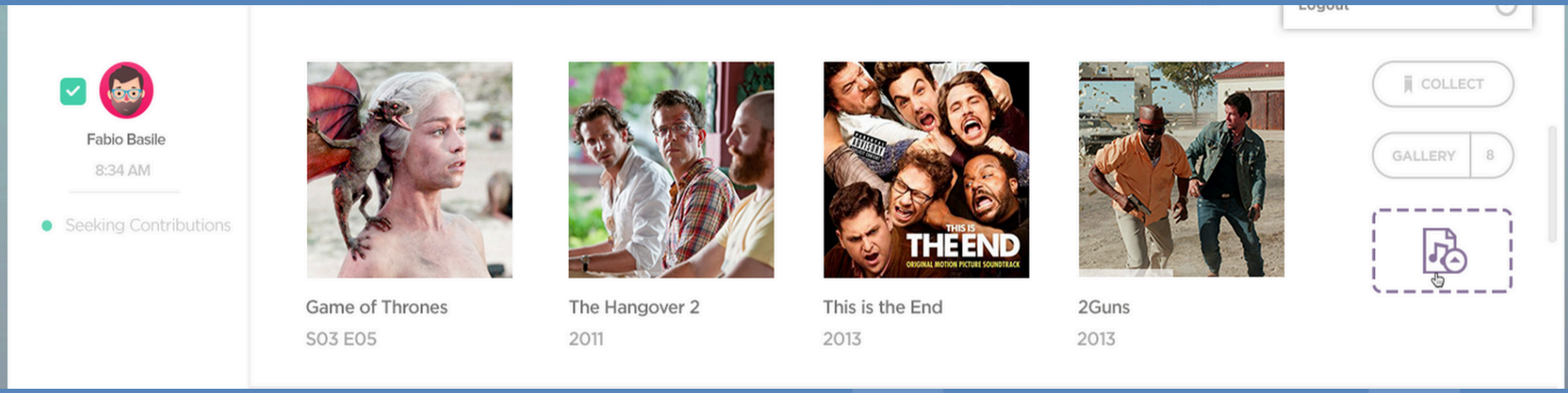
{ } INTRO TO WEB DESIGN

- Structure
  - Logo
  - Navigation
    - Stream | Activity | Collections | Soundbox
  - Search
    - Search box | User | Add

- Implementation
  - Horizontal => Floats

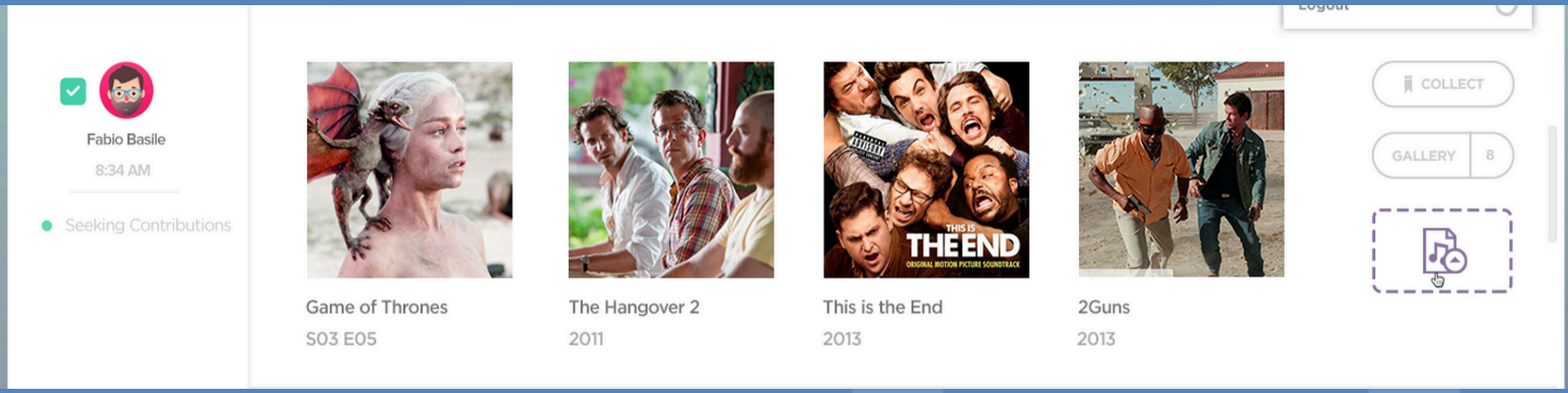
{ } INTRO TO WEB DESIGN

- Structure
  - Cover
  - Content
    - Title
    - Explore
    - Some dots
- Implementation
  - Cover (position: absolute)
  - Content (position: absolute)
    - Align vertically using divs

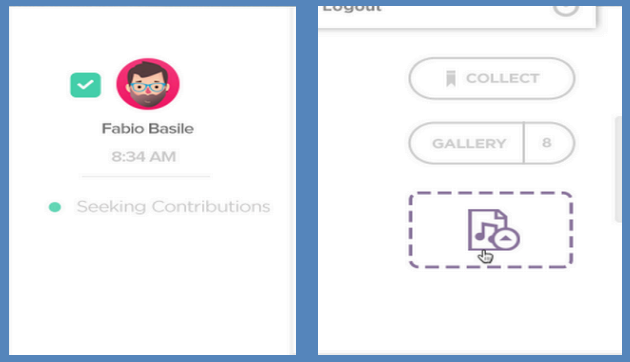


- 3 Parts: Sidebar, Gallery, Toolbar
- A bit complicated if you think about making this responsive (viewable in multiple screen sizes)

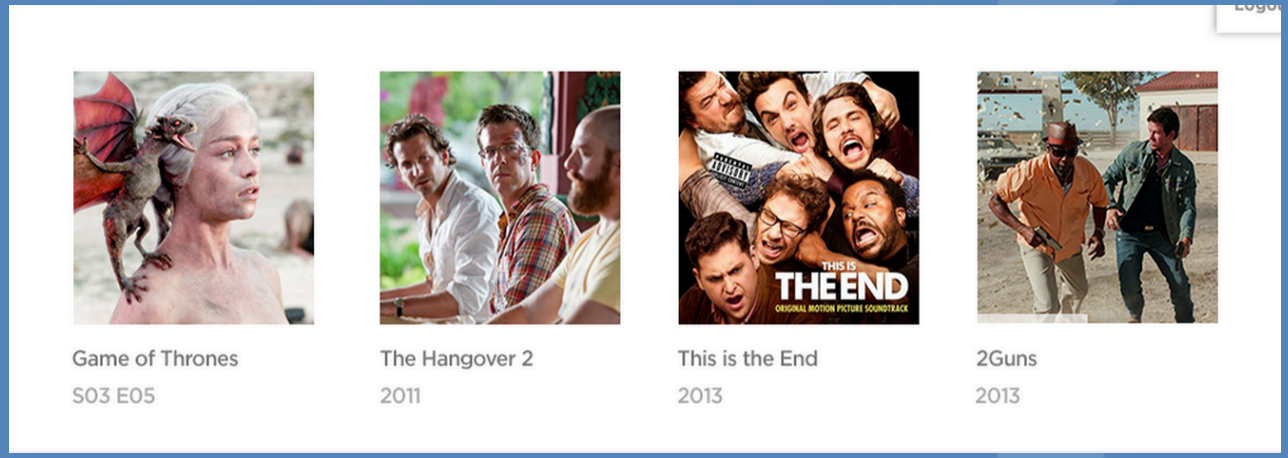
- Assume the size is fixed for this example only:
  - Horizontal stack => Floats



- Sidebar
  - Thumbnail
  - Name
  - Date
  - Contribution
- Content (4X)
  - Thumbnail
  - Title
  - Episode / Release Date
- Toolbox
  - Collect
  - Gallery
  - Some music thing



- Simple – just stack things vertically using divs



- Horizontal stack => Floats

{ } INTRO TO WEB DESIGN

- 3 Parts: Controls, Progress Bar, Options
  - Horizontal stack => Floats

- Controls (Horizontal stack => Floats)
  - Repeat
  - Play
  - Shuffle

- Progress bar

- Options (Horizontal stack => Floats)
  - Like
  - Dislike
  - Volume

{ } INTRO TO WEB DESIGN

Hands-on Session 5
Positioning Haven III

{ } INTRO TO WEB DESIGN

Intro to JavaScript

{ } INTRO TO WEB DESIGN

Intro to JavaScript

- JavaScript is the scripting language of the web
- **Not** the same as Java!
- Like CSS, you can select elements and *manipulate* their actions
  - Hide/Unhide menus
  - Photo Slider (**carousel**)
  - Popup modals (like Facebook's Photo Viewer)
  - Form Validation
  - One-Scroll Homepage

{ } INTRO TO WEB DESIGN

Intro to JavaScript

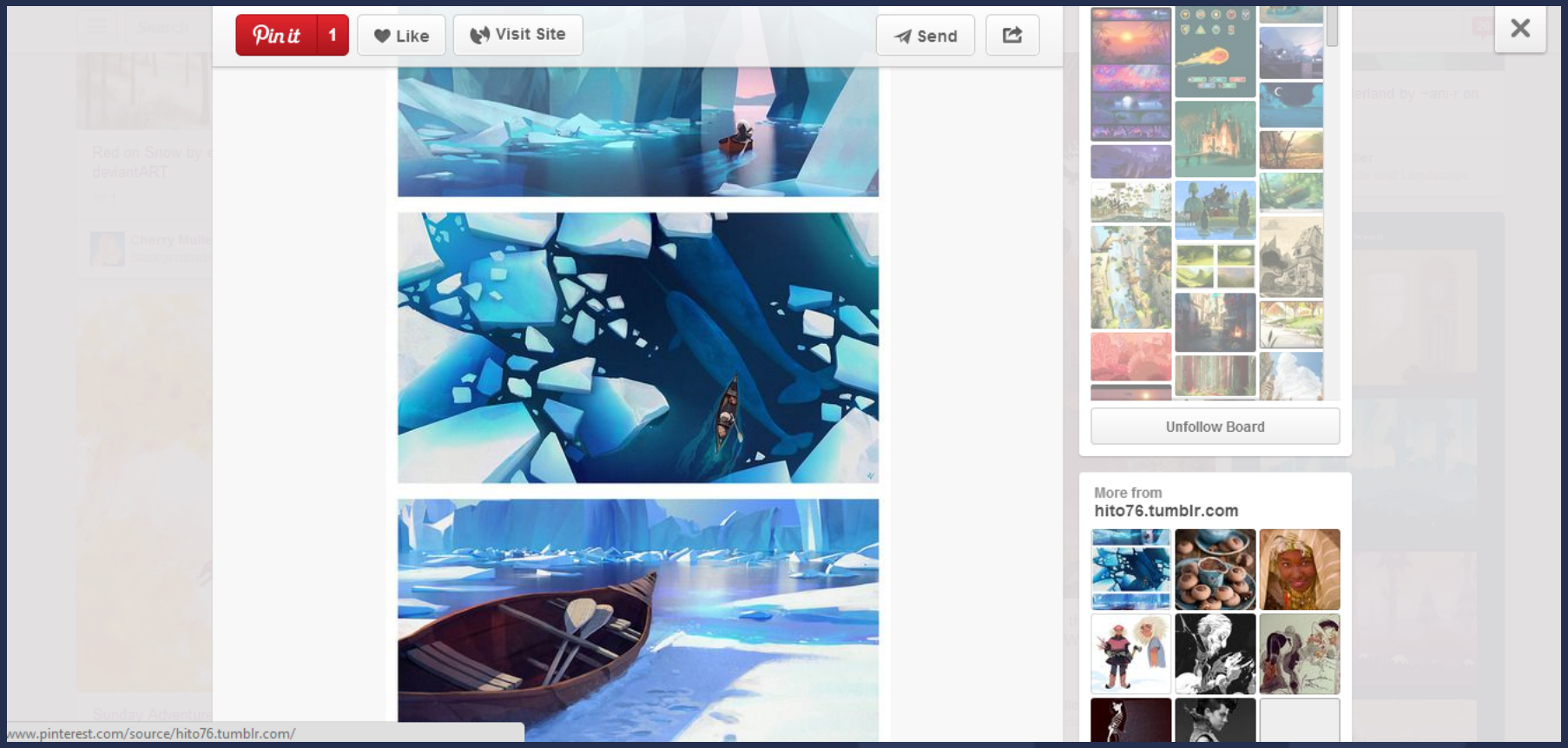
Sample JavaScript:

```
1 // Syntax highlighting
2 function printNumber()
3 {
4   var number = 1234;
5   var x;
6   document.write("The number is " + number);
7   for (var i = 0; i <= number; i++)
8   {
9     x++;
10    x--;
11    x += 1.0;
12  }
13  i += @; // illegal character
14 }
15 body.onLoad = printNumber;
```

```
Accordian.prototype = {
    init: function( opts ) {
        this.defaults = {
            speed: 200,
            closeAll: true
        };

        this.options = $.extend( this.defaults, opts || {} );
        this.build();
    },

    build: function() {
        var self = this;

        $(function() {
            self.triggers = $( '.accordian-trigger' );
            //self.containers = $( '.accordian-container' );
            self.events();
        });
    },
```

{ } INTRO TO WEB DESIGN

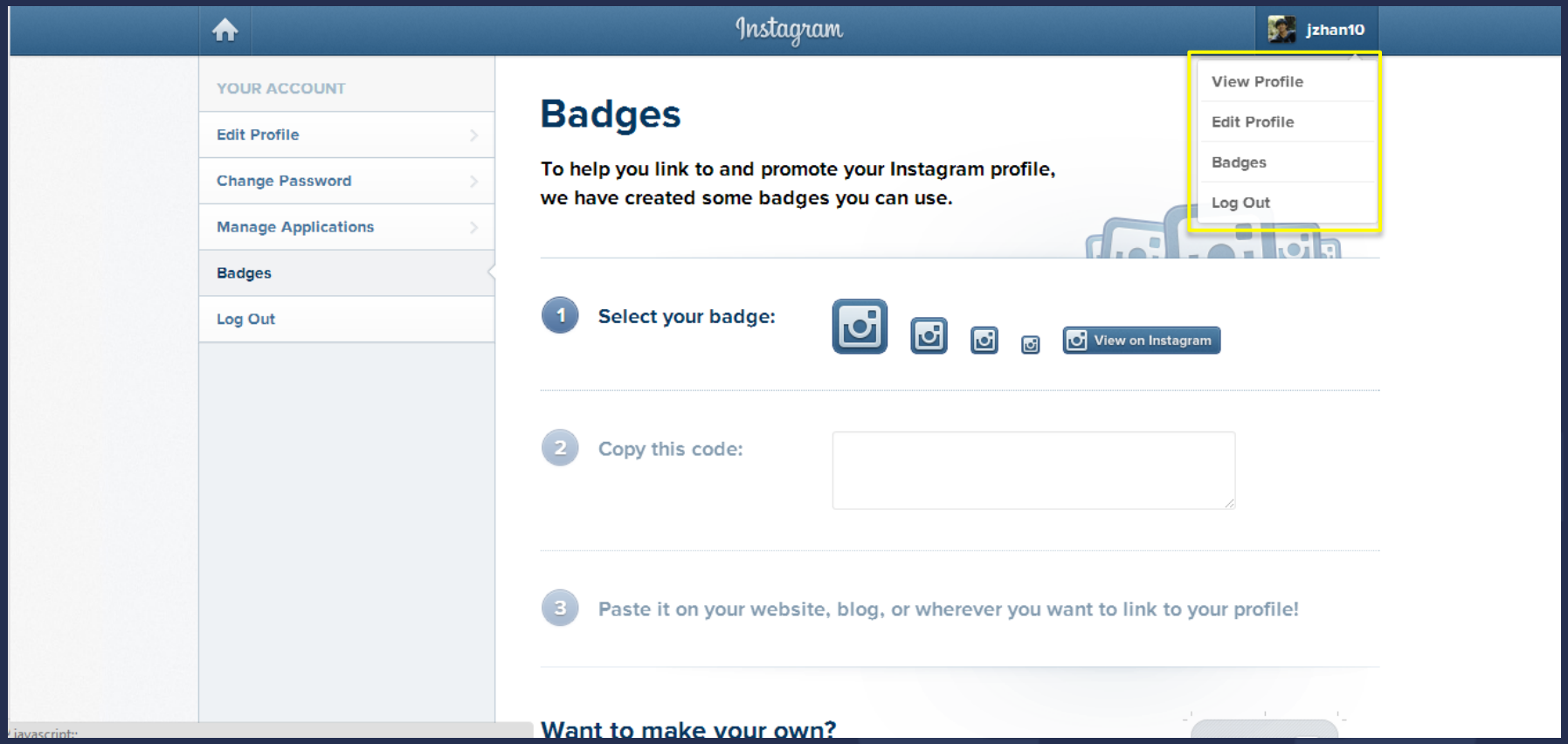
Intro to JavaScript - Examples



Pinterest - Modals

{ } INTRO TO WEB DESIGN

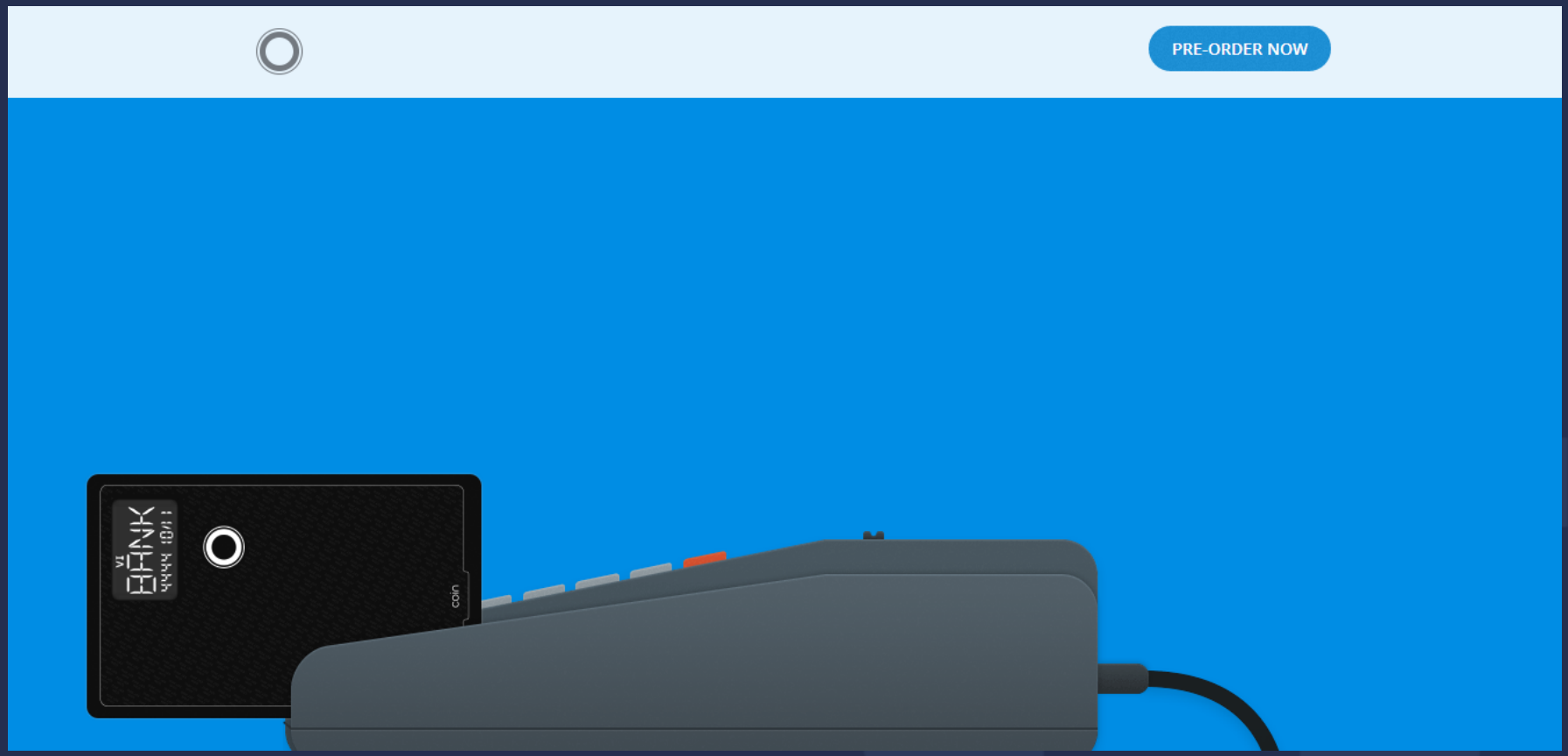
Intro to JavaScript - Examples



Instagram - Toggling Menus

{ } INTRO TO WEB DESIGN

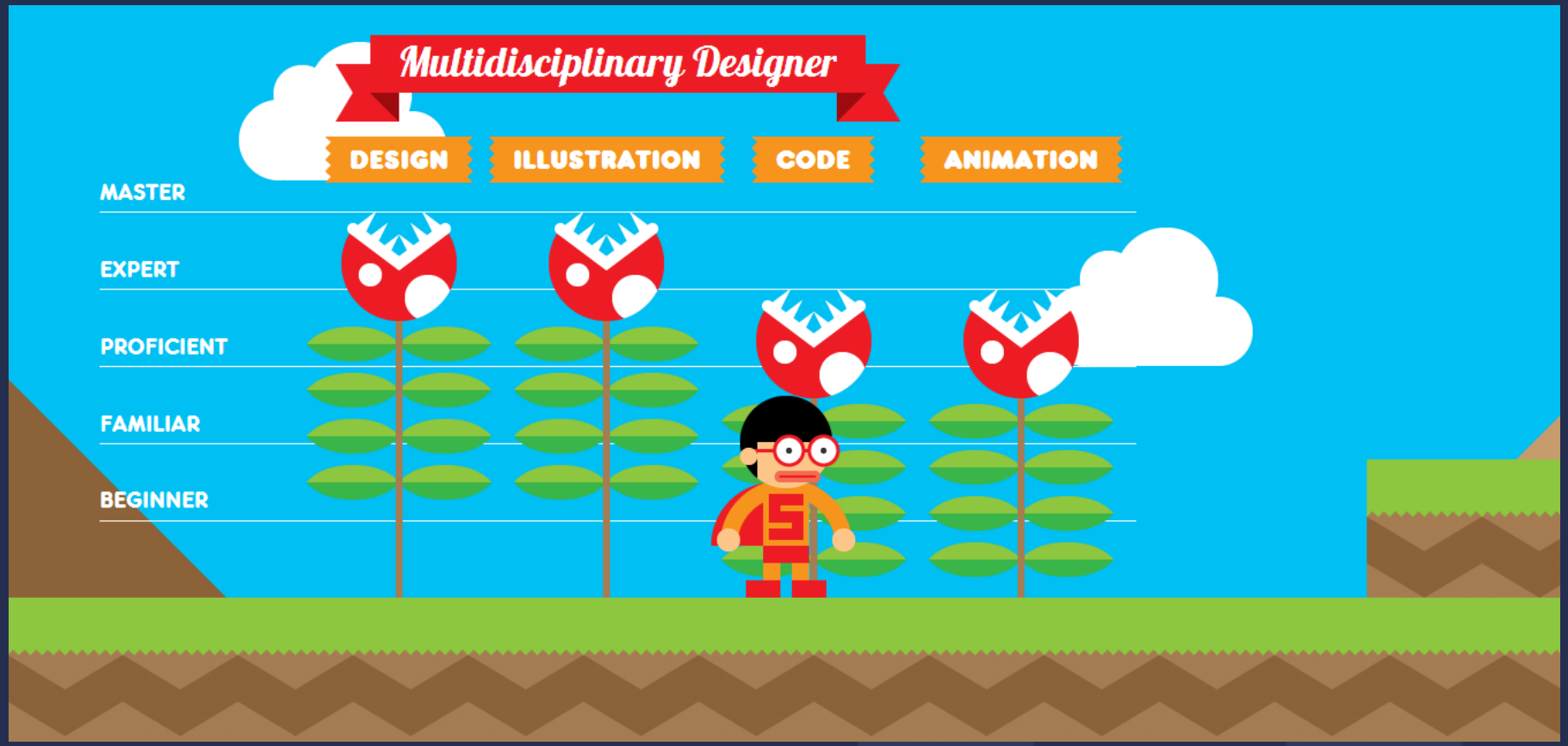
Intro to JavaScript - Examples



Coin – JavaScript animations while scrolling

{ } INTRO TO WEB DESIGN

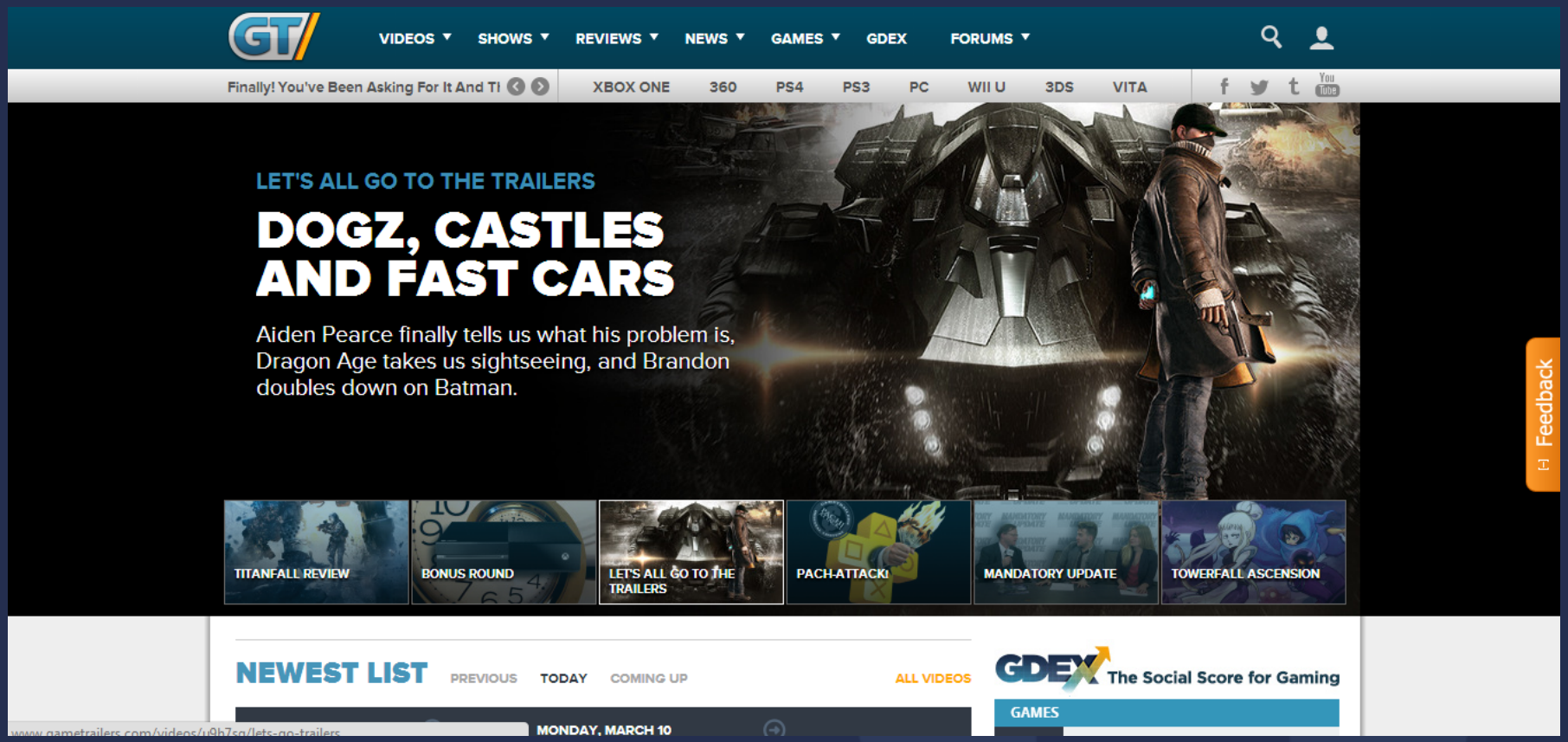
Intro to JavaScript - Examples



Interactive Resume *(http://www.rleonardi.com/interactive-resume/)*

{ } INTRO TO WEB DESIGN

Intro to JavaScript - Examples

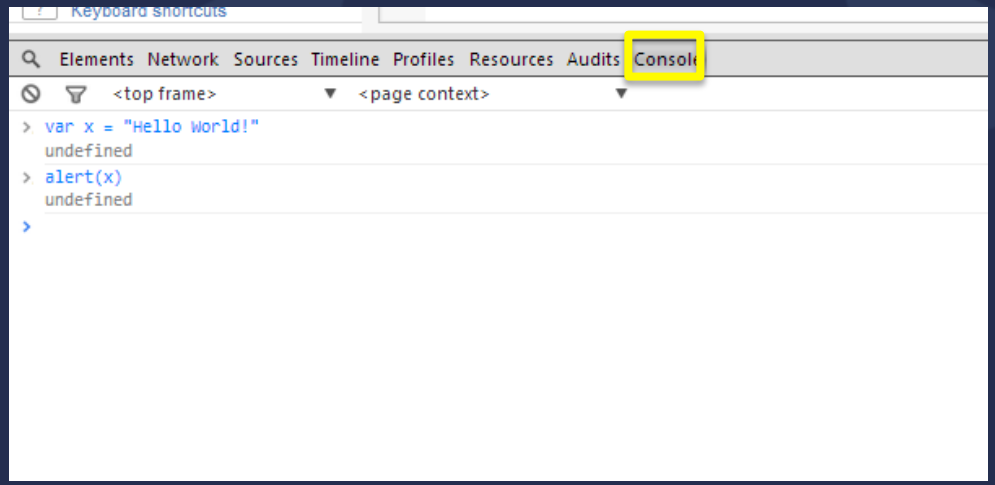


GameTrailers – Photo Slider

{ } INTRO TO WEB DESIGN

Intro to JavaScript

- How to you use Javascript?
- Like CSS, you link to a separate *.js* file that contains your file
  - *<script type="text/javascript" src="home.js"></script>*
- You can also **test** JavaScript in Chrome's **console** (Inspector Element window) and on JSFiddle.net
- Let's start coding!



{ } INTRO TO WEB DESIGN

Basic Syntax

- Variables
  - They take in numbers, text, booleans (true/false) and functions

```
var x = 8;
var y = “John”
var z = ‘Smith’
var a = true;
```

Text (in computer science, we call them “strings”) can be surrounded by single or double quotes

- Concatenation
  - You can add numbers or combine “strings”

```
var a = x + x;
var b = y + “ “ + z;
var c = 5 + “ five”;
```

“a” now stores 16. “b” stores John Smith.

Note: only use “var” to initialize (if the variable name has never been used before)

{ } INTRO TO WEB DESIGN

Basic Syntax

- Shorthand increment/decrement

```
var x = 1;

x = x + 1;
// x is now 2
x += 1;
// x is now 3
x++;
// x is now 4
x--;
// x is now 3
```

To increment by 1, there are 3 ways:

1) x = x+1
2) x += 1
3) x++

Same goes for decrementing

{ } INTRO TO WEB DESIGN

Conditionals

- If… Else…
  - If something do this, else do that

```
If(x > 9000) {
    alert(“Over 9000!”);
} else {
    console.log(“Weak”);
}
```

alert(…) is a JavaScript function that creates a popup.
console.log(…) outputs whatever is inside onto the console (e.g. Chrome’s console).
Both useful for debugging.

- Plain If, and If…Else If… Else If…

```
if(z < 9000) {
    alert(“Not Saiyan”);
}
```

```
if(z == 1000) {
        …
} else if (z == 2000) {
        …
```

{ } INTRO TO WEB DESIGN

Conditionals

- Comparisons
  - <, >, >=, <=
  - == and === (3 equal signs checks value **and** type)

```
var x = 3;
x == 3 // true
x === “3” // false
x === 3 // true
```

- And/Or - &&/||

```
if((x==3) && (y==5)) {
      …
```

```
if(x || y) {
      ….
```

{ } INTRO TO WEB DESIGN

Conditionals

- Not Equal - !=, !== (not same type too)

```
if(x != y) {
        ...
}
```

- Remember, you can use ( ) parentheses to separate your logic. (*e.g. if((x != 3) && (y != 4)))*

{ } INTRO TO WEB DESIGN

Arrays

- Variables can also store more than 1 value

```
var x = new Array("Porsche", "BMW", "Ferrari");

                or

var x = ["Porsche", "BMW", "Ferrari"];

                or

var x = new Array();
x[0] = "Porsche";
x[1] = "BMW";
x[2] = "Ferrari";
```

{ } INTRO TO WEB DESIGN

Arrays

- Access an array with […]
  - First value is always the array position. Positions start at 0

```
var x = [“Porsche”, “BMW”, “Ferrari”];
alert(x[0]); // Will alert “Porsche”
```

- Set values for an array

```
x[2] = “Tesla”;
```

- You can get an array’s length with arrayname.length

```
x.length // returns 3. However, max position is 2, since we start at 0
```

{ } INTRO TO WEB DESIGN

Loops

- For Loop
  - Loop through items/arrays
  - 3 components
    - *for(var i=0; i < 10; i++)*
  - *1st value initializes variable. i++ increments i by 1 after loop is complete. Loop while i < 10, and stop when condition is not met.*

```
var cars = [“Porsche”, “BMW”, “Tesla”];
for(var i = 0; i < cars.length; i++) {
        alert(cars[i]);
}
// alerts each value in cars, one by one
```

{ } INTRO TO WEB DESIGN

Loops

- While Loop
  - Loop through items/arrays
  - 1 component
    - *while(i < 30) { … }*
  - Initialize (var i = 0) before while loop, increment (i++) inside

```
var i = 0;
while(i < 30){
        alert(i);
        i++;
}
// alerts values 0 to 29
```

Warning!

If you don't increment, or don't fail the conditional statement at some point, you end up in a never-ending loop! Not good!

{ } INTRO TO WEB DESIGN

Loops

- For vs. While
  - For loops are more compact
  - But while loops are fast to type

```
for(var i=0; i<100; i++) {
        alert(i);
}
```

```
var i = 0;
while(i < 100) {
        alert(i);
        i++
}
```

Output: 0, 1, 2, … , 98, 99

{ } INTRO TO WEB DESIGN

Functions

- **Functions** are a set of JavaScript code, that can perform some action, return a result, or do computation
- They take in 0 or more **arguments** (values to be passed in)
  - Arguments are by default type **var**, so *arr* can be any **var**

```
function printOddValues(arr) {
        var odd = true;
        for(var i=0; i<arr.length; i++) {
                if(odd){
                        console.log(arr[i]+" ");
                        odd = false;
                } else {
                        odd = true;
                }
        }
}
```

Once you have a function, you can call it!

```
var arr = [1, 2, 3, 4, 5, 6, 7];
printOddValues(arr);
```

Output: 1 3 5 7

```
var arr = ["OH", "CA", "PA", "NY"];
printOddValues(arr);
```

Output: OH PA

{ } INTRO TO WEB DESIGN

Functions

- **Functions** can return values, and you can save them in a variable (for future use)

```
function calculateExp(value, times) {
        if(times < 0) {
                return -1;
        } else if(times == 0) {
                return 1;
        } else {
                var count = 1, total = 0;
                while(count <= times) {
                        total += value;
                }
                return value;
        }
}
```

Let's call this function!

```
var x = calculateExp(5, 3);
alert(x);
```

Output: 125

```
var x = calculateTimes(10, -999);
if(x != -1){
        alert("No neg nums please");
}
```

Output: "No neg nums please"

{ } INTRO TO WEB DESIGN

JavaScript

- Functions and variables are cool… but can we do more with JavaScript?
- **jQuery,** a JavaScript library, comes in handy!
- jQuery allows you to access HTML elements and do all sort of things!
  - Click events
  - Keyboard presses
  - Scrolling animations
  - Modal popups
  - Toggling menus
  - Smoother hovers and animations
  - Etc.

{ } INTRO TO WEB DESIGN

Intro to jQuery

{ } INTRO TO WEB DESIGN

jQuery

- Functions and variables are cool… but can we do more with JavaScript?
- **jQuery,** a JavaScript library, comes in handy!
- jQuery allows you to access HTML elements and do all sort of things!
  - Click events
  - Keyboard presses
  - Scrolling animations
  - Modal popups
  - Toggling menus
  - Smoother hovers and animations
  - Etc.

{ } INTRO TO WEB DESIGN

jQuery

- How do we add jQuery to our page?
- Either insert the first element below, or download the jQuery .js file and link to it
  - **Important!** Add this before any links to your own .js files!

```
<script src="http://code.jquery.com/jquery-1.11.0.min.js"></script>
```

Above: Link to an online copy of jQuery

```
<script src="assets/js/jquery-1.11.0.min.js"></script>
```

Above: Link to a local copy of jQuery

{ } INTRO TO WEB DESIGN

jQuery
Selectors & Accessing Content

{ } INTRO TO WEB DESIGN

jQuery - Selectors

- To use jQuery, you have to access HTML elements first
- Syntax
  - **$(selector)** – The $ tells the browser this is jQuery
  - Replace *selector* with the CSS selector in quotes (single or double)

```
$('body')
$('.box')
$('#container .box')
$('a')
```

Like in CSS, we must select elements before we can apply actions to them

- Once you select an element, you can access different values, properties, or functions to the element

{ } INTRO TO WEB DESIGN

jQuery – Accessing Content

- 3 common ways to access element content
  - **.text()** – Grabs all text inside an element
  - **.html()** – Grabs all HTML inside an element
  - **.val()** – Grabs the value attribute or an element (useful for inputs)

```
<div id="box">
    <div class="title">
        Hello World!
    </div>
</div>
```

$('#box').text() → Hello World!

$('#box').html() →
Hello World!

- You could test the above using **console.log($('#box').text())** or **alert($('#box').html());**

{ } INTRO TO WEB DESIGN

jQuery – Accessing Content

- .val()
  - Useful to get search results, form input, etc.

<input type="text" id="name">

$('#name").val() → Jeff

Jeff

{ } INTRO TO WEB DESIGN

jQuery – Setting Content

- Likewise, you can set content

```
$('#box').html('<p>New Title</p>');

$('#email').val('sample@example.com');
```

{ } INTRO TO WEB DESIGN

jQuery

- Accessing content, or setting content is useful!
- But how so? When **should** we access an element's value? Or HTML? Or text?
- When using **Events (actions user does)**
  - *E.g. Click a button to replace the HTML on a page*
  - *Hit enter to get User's Login information*
  - *Scroll down page and change CSS*

{ } INTRO TO WEB DESIGN

jQuery Events

{ } INTRO TO WEB DESIGN

jQuery – Event Listeners

- **Event Listeners** are always on the page, “listening” for an event to occur (and react to these events)
- Events include:
  - Clicking
  - Key presses
  - Scrolling
  - Hovering (smoother than CSS, more powerful)
  - Tons more

{ } INTRO TO WEB DESIGN

jQuery – Event Listeners

- Event Listeners are *attached* to jQuery selected elements
  - *E.g. $('#box').click(function() { … });*
- Syntax:
  - .click(function(){ … });
  - .hover(function(){ … });
  - .scroll(function(){ … });
- Always pass a **function** to an event listener! (*event handlers*)

```
$('#box').click(function(){
        alert("You are so cool.");
});
```

Browser binds **click** event to **#box**. It continues to "listen" as user is on page, and when user clicks on #box, a popup will appear!

{ } INTRO TO WEB DESIGN

jQuery – Event Listeners

- More examples

```
$('#box').hover(function(){
        alert($(this).html());
});
```

Inside your event listener function, you can refer to the current element (say **#box**) with simply **$(this)**.

```
$('.box').hover(function(){
        alert($(this).html());
});
```

This hover now applies to all class **box**. **$(this)** only refers to the element you hovered on, not to all elements with class box!

```
$('.box').click(function(){
        $('.box2').hide();
});
```

Don't have to use **$(this)!**
Here, we click on any element with class **.box**, and you hide all elements with class .**box2**

{ } INTRO TO WEB DESIGN

jQuery Effects

{ } INTRO TO WEB DESIGN

jQuery – Effects

- You have seen **ways to access content**
  - *.html(), .text(), .value()*
- You have seen **events**
  - *.click(), .hover(), .scroll()*
- Let's talk about **effects**
  - *Hiding elements*
  - *Showing elements*
  - *Fade In, Fade Out*
  - *Sliding up, Sliding Down*

{ } INTRO TO WEB DESIGN

jQuery – Effects

- .show(), .hide()
  - Extremely useful and common
  - Essentially just adds/removes the CSS property **display: none** to an element

```
$('#menu-show').click(function(){
        $('#menu').show();
});

$('#menu-hide').click(function(){
        $('#menu').hide();
});
```

{ } INTRO TO WEB DESIGN

jQuery – Effects

- .fadeIn(*time), .*fadeOut(*time)*
  - Like show() and hide(), but gradually fading
  - Time takes in milliseconds. 1000 is a second

```
$('#menu-show').click(function(){
        $('#menu').fadeIn(500);
});

$('#menu-hide').click(function(){
        $('#menu').fadeOut(500);
});
```

{ } INTRO TO WEB DESIGN

jQuery – Effects

- .slideUp(*time)*, .slideDown(*time)*
  - Like fadeIn() and fadeOut(), but height changes over time
  - Again, time in milliseconds

```
$('#menu-show').click(function(){
        $('#menu').slideDown(500);
});

$('#menu-hide').click(function(){
        $('#menu').slideUp(500);
});
```

{ } INTRO TO WEB DESIGN

jQuery – Effects

- .css*(style, value)*
  - Change the CSS
  - *Ex: $('body').css('color', 'red');*

```
$('#menu').click(function(){
        $(this).css('background', '#333');
});
```

{ } INTRO TO WEB DESIGN

jQuery Initialization

{ } INTRO TO WEB DESIGN

jQuery - Initializing

- Initializing jQuery
  - So you added the **<script … ></script>** for jQuery
  - Now, before we can do anything, you must let the browser know you are using jQuery in your JavaScript files
  - Start your .js files with:

```
$(document).ready(function(){

        … your javascript here …

});
```

$ - Calls the jQuery library
(document) – Selects the entire HTML page
.ready – When *all* the HTML is loaded…
function(){ … } – …do everything inside this function

{ } INTRO TO WEB DESIGN

jQuery

Your HTML

```html
<html>
<head>
    <title>Web Design Decal</title>
    <script src="http://code.jquery.com/jquery-1.11.0.min.js"></script>
    <script src="assets/js/site.js"></script>
</head>
<body>

</body>
</html>
```

Your Linked JavaScript File (assets/js/site.js)

```javascript
$(document).ready(function(){
    /* Your jQuery/JavaScript here */

    $('body').click(function(){
        alert('I am the body.');
    });

});
```

{ } INTRO TO WEB DESIGN

Hands-On Activity