

ITMD-362 WEEK 7

February 20, 2018



TONIGHT'S AGENDA

- Font Stacks
- Call to action Button
- JS
- Demo



FONT STACKS



- Dan's Tools: cssfontstack.com
- Order of font stack:
 - (1) Ideal Font, (2) System Fonts, (3) Generic Font Family



BORDERS



HOW MANY BORDERS DO YOU NEED?

HomePage2Page3Page4

MY TITLE

"Lorem ipsum dolor sit amet, consectetur adipiscing 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

My Form

Answer My Question

Answer My Question



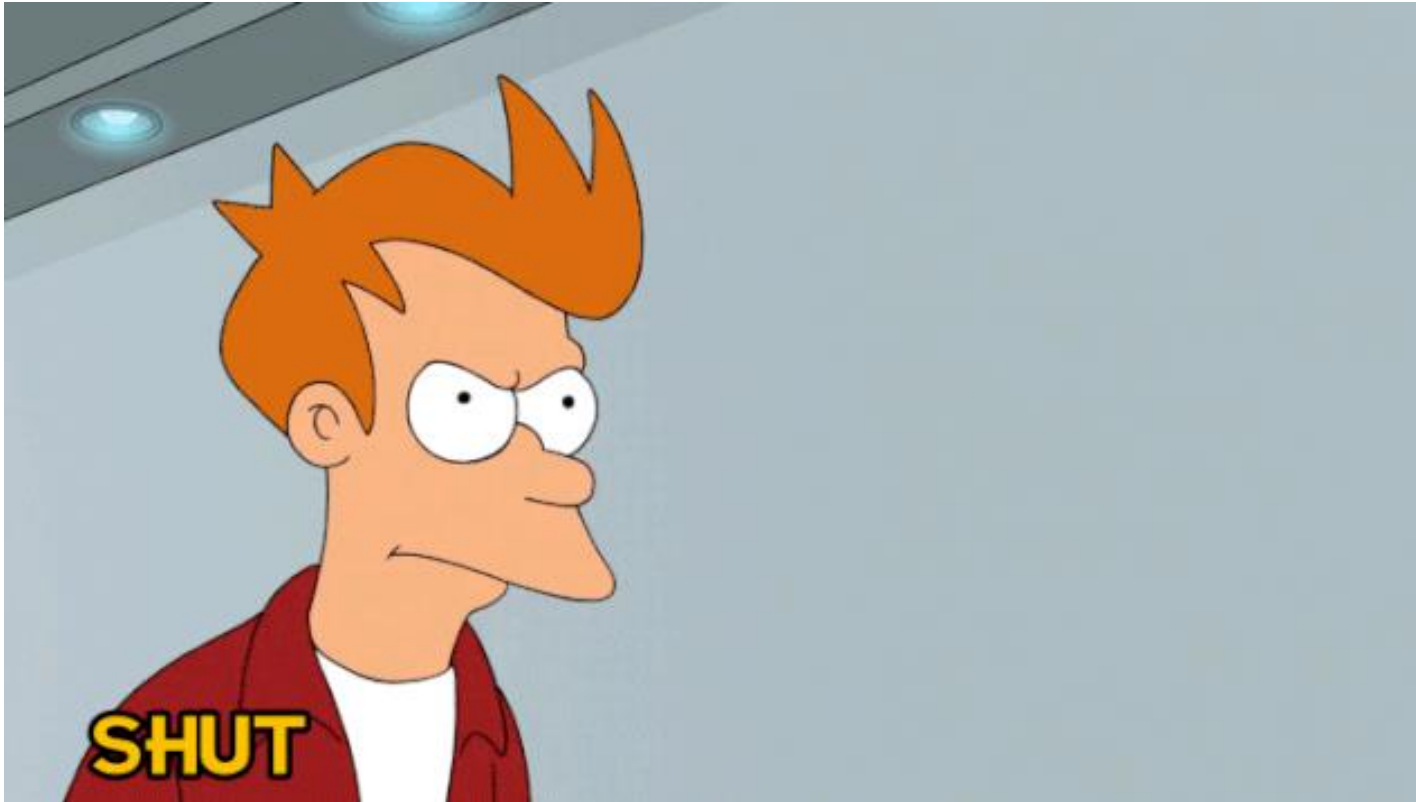
LESS IS MORE. Ludwig Mies van der Rohe



CALL TO ACTION BUTTON



WHAT IS A CALL TO ACTION?



TAKE CONTROL!



Task specific buttons

Create Account

Subscribe Now

Send Message

Register Free



Vague button

Submit



YOUR CTA BUTTON SHOULD:

- Stand on its own (no supporting text; no competing buttons)
- Be clear and specific
- Action-Oriented
- Describe a compelling action
- Provide rationale for a visitor to complete your offer
- Communicate what will happen when they click
- Be familiar
- Remove uncertain
- Impossible to miss (contrasting colors; obvious button)



25 BEST WORDS TO USE: WISHPOND

- Start
- Stop
- Build
- Join
- Learn
- Discover
- Sick
- Trouble
- Worried
- Confused
- You/Your
- Me/My
- Want
- Need
- Free
- Save
- [actual Number]
- Try
- Why
- Get
- Find
- Today
- Before
- Ends



JAVASCRIPT



USEFUL LINKS

- Mozilla JavaScript Guide
 - <https://developer.mozilla.org/en-US/docs/Web/JavaScript>
- Node.js for Command Line JavaScript Intro
 - <http://javascript.cs.lmu.edu/notes/commandlinejs/>
- JavaScript and Basic Programming Introduction Reading
 - <http://eloquentjavascript.net/>



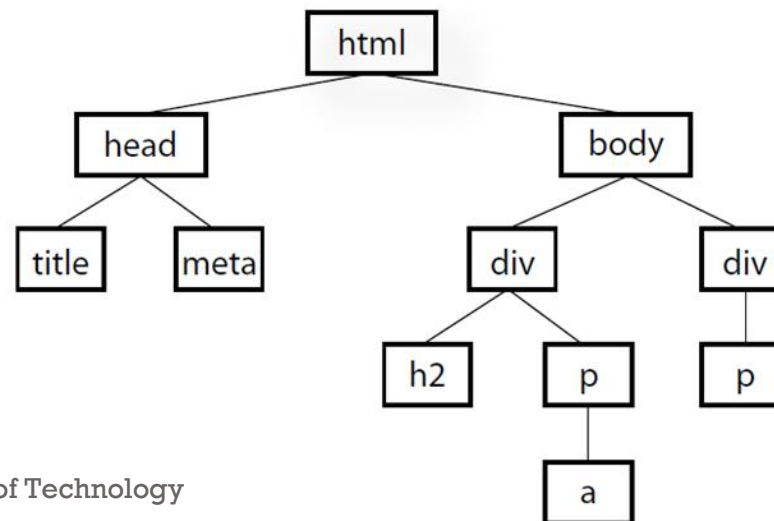
DOM

Document Object Model

The node tree

A simple way to think of the DOM is in terms of the document tree (Figure 20-1). You saw documents diagrammed in this way when you were learning about CSS selectors.

```
<html>
<head>
  <title>Document title</title>
  <meta charset="utf-8">
</head>
<body>
  <div>
    <h2>Subhead</h2>
    <p>Paragraph text with a <a href="foo.html">link</a> here.</p>
  </div>
  <div>
    <p>More text here.</p>
  </div>
</body>
</html>
```



JAVASCRIPT DOM

- API allowing accessing to the DOM elements
- Most common by id
 - `var a = document.getElementById("elementid");`
- Can also access by class, tag, selector
- Use the `object.getAttribute("src");` method to get a attribute's value from an object
- Set of methods to manipulate DOM objects.
- Examples:
 - Can use `object.innerHTML` to set the HTML contents of an element.
 - Can use `object.value` to get the contents of a form control



ACCESSING DOM NODES

- By element name:
 - `getElementsByTagName("p")`
- By id attribute value:
 - `getElementById("lead-photo")`
- By class attribute value:
 - `getElementsByClassName("headlines")`
- By selector:
 - `querySelectorAll(".headlines p")`
 - Similar to jquery

REVIEW: BASICS

- Embedded Scripts
 - Use script tags: `<script> JS Here </script>`
 - DO NOT USE EXCEPT FOR TESTING

- External Scripts
 - Use script tag with src attribute:
 - `<script src="myscript.js"></script>`

- Inside `<script>` is just code
 - `var number = 5;`
 - `var string = "five";`
 - `var boolean = true;`
 - `var foo = [5, "five", "5"];`
 - `typeof = NaN`

REVIEW:

COMPARISON OPERATORS

==	Is equal to
!=	Is not equal to
===	Is identical to (equal to and of the same data type)
!==	Is not identical to
>	Is greater than
>=	Is greater than or equal to
<	Is less than
<=	Is less than or equal to

REVIEW:

MATHEMATICAL OPERATORS

- Mathematical operators are used to perform math on numeric objects
 - Addition + (plus operator is also used to concatenate strings)
 - Subtraction -
 - Multiplication *
 - Division /
 - Modulus (division remainder) %
 - Increment ++
 - Decrement --
 - Add to self and reassign +=
 - `var car = 5; car += 2; car is now 7`

REVIEW: CUSTOM FUNCTIONS

- `function name(type) {`
 code
 }
- `function foo() {`
- `alert("Our function just ran!");`
 }
- `foo();` //actually runs function

REVIEW: IF/ELSE

- Conditional statements
 - if statements
 - else statements
 - else if statements
- `if (condition) {`
 run this block
- `} else if (condition) {`
 run this block
- `} else {`
 }

REVIEW: NATIVE FUNCTIONS

There are hundreds of predefined functions built into JavaScript, including:

`alert()`, `confirm()`, and `prompt()`

These functions trigger browser-level dialog boxes.

`Date()`

Returns the current date and time.

`parseInt("123")`

This function will, among other things, take a string data type containing numbers and turn it into a number data type. The string is passed to the function as an argument.

`setTimeout(functionName, 5000)`

Will execute a function after a delay. The function is specified in the first argument, and the delay is specified in milliseconds in the second (in the example, 5,000 milliseconds equals 5 seconds).

JAVASCRIPT LOOPS

- Loops
 - **for** { //loops through a block a specific # of times }
 - **while** { //loops through a block while condition true }
 - **do** { }
while { //loops through block once then repeats as long as a condition is true }
 - **for** { }
in { //loops through objects in an array or properties of an object, be careful with this one can be error prone }
- Basic Loop Syntax
 - *while (condition) {
code block to be executed }*

LOOPS CONTINUED

- Example 1

- `for` (*initialize the variable; test the condition; alter the value*)
- { loop code here }

- Example 2

- `for (var i = 0; i <= 2; i++) {`
- `alert(i);}`

- Example 3

- `do {`
 `text += "The number is " + i;`
 `i++;`
 `}`
 `while (i < 10);`

OBJECTS

- Document Object: (HTML)
- Browser Object

Property/method	Description
event	Represents the state of an event
history	Contains the URLs the user has visited within a browser window
location	Gives read/write access to the URI in the address bar
status	Sets or returns the text in the status bar of the window
alert()	Displays an alert box with a specified message and an OK button
close()	Closes the current window
confirm()	Displays a dialog box with a specified message and an OK and a Cancel button
focus()	Sets focus on the current window

EVENT OBJECTS

Event handler	Event description
onblur	An element loses focus
onchange	The content of a form field changes
onclick	The mouse clicks an object
onerror	An error occurs when the document or an image loads
onfocus	An element gets focus
onkeydown	A key on the keyboard is pressed
onkeypress	A key on the keyboard is pressed or held down
onkeyup	A key on the keyboard is released
onload	A page or an image is finished loading
onmousedown	A mouse button is pressed
onmousemove	The mouse is moved
onmouseout	The mouse is moved off an element
onmouseover	The mouse is moved over an element
onmouseup	A mouse button is released
onsubmit	The submit button is clicked in a form

APPLYING EVENT HANDLERS

- How to use addEventListener in <script>
- `window.addEventListener("click", myFunction);`
- *or as anonyms function:*
- `window.addEventListener("click", function() {`
- *function goes here*
- `});`

JAVASCRIPT EVENT HANDLING

- Two other bad methods for Even Handling
 - As attribute on HTML element:
 - `<body onclick="myFunction();">`
 - Only single binds
 - As a method attached to a DOM object:
 - `window.onclick = myFunction;`
 - Only single binds; second bind will override first

EVENT HANDLING EXAMPLE

- Check out the MDN Developer [Link](#)
- Example:
- ```
function addEventHandler(elem,eventType,handler) {
 if (elem.addEventListener) {
 elem.addEventListener (eventType,handler,false);
 } else if (elem.attachEvent) {
 elem.attachEvent ('on' + eventType,handler);
 }
}
```

# JavaScript CONTINUED



# 1 ARRAY VS. MULTIPLE VARIABLES.

⊘ | top ▼ | Filter

---

SW registered

```
> var email = {user1: "email1", user2: "email2", user3: "email3"}
< undefined

> email.user2
< "email2"

> |
```

⊘ | top

---

SW registered

```
> var user1 = "email1"
< undefined

> var user2 = "email2"
< undefined

> var user3 = "email3"
< undefined

> var user4 = "email4"
< undefined

> var user5 = "email5"
< undefined

> user1
< "email1"
```





# ADDING TO ARRAY

```
⊘ | top ▼ | Filter De
SW registered
> var email = {user1: "email1", user2: "email2", user3: "email3"}
< undefined
> email.user2
< "email2"
> email.user4 = "email4"
< "email4"
> email
< ► {user1: "email1", user2: "email2", user3: "email3", user4: "email4"}
> |
```



# REVIEW: CUSTOM FUNCTIONS

```
function name(type) {
 code
}
```

```
function foo() {
 alert("Our function just ran!");
}
```

```
foo(); //actually runs function
```

# REVIEW: IF/ELSE

- Conditional statements
  - if statements
  - else statements
  - else if statements

```
if (condition) {
 run this block
} else if (condition) {
 run this block
} else {
 }
}
```

# REVIEW: NATIVE FUNCTIONS

There are hundreds of predefined functions built into JavaScript, including:

`alert()`, `confirm()`, and `prompt()`

These functions trigger browser-level dialog boxes.

`Date()`

Returns the current date and time.

`parseInt("123")`

This function will, among other things, take a string data type containing numbers and turn it into a number data type. The string is passed to the function as an argument.

`setTimeout(functionName, 5000)`

Will execute a function after a delay. The function is specified in the first argument, and the delay is specified in milliseconds in the second (in the example, 5,000 milliseconds equals 5 seconds).

# JAVASCRIPT LOOPS

- Loops
  - **for** { //loops through a block a specific # of times }
  - **while** { //loops through a block while condition true }
  - **do** { }  
  **while** { //loops through block once then repeats as long as a condition is true }
  - **for** { }  
  **in** { //loops through objects in an array or properties of an object, be careful with this one can be error prone }
- Basic Loop Syntax:

```
while (condition) {
 code block to be executed }
```

# LOOPS CONTINUED

- Example 1

**for** (*initialize the variable; test the condition; alter the value*)

```
{ loop code here }
```

- Example 2

```
for (var i = 0; i <= 2; i++) {
```

```
 alert(i);}
```

- Example 3

```
do {
 text += "The number is " + i;
 i++;
}
while (i < 10);
```

# OBJECTS

- Document Object: (HTML)
- Browser Object

| Property/method | Description                                                                  |
|-----------------|------------------------------------------------------------------------------|
| event           | Represents the state of an event                                             |
| history         | Contains the URLs the user has visited within a browser window               |
| location        | Gives read/write access to the URI in the address bar                        |
| status          | Sets or returns the text in the status bar of the window                     |
| alert()         | Displays an alert box with a specified message and an OK button              |
| close()         | Closes the current window                                                    |
| confirm()       | Displays a dialog box with a specified message and an OK and a Cancel button |
| focus()         | Sets focus on the current window                                             |

# EVENT OBJECTS

| Event handler | Event description                                   |
|---------------|-----------------------------------------------------|
| onblur        | An element loses focus                              |
| onchange      | The content of a form field changes                 |
| onclick       | The mouse clicks an object                          |
| onerror       | An error occurs when the document or an image loads |
| onfocus       | An element gets focus                               |
| onkeydown     | A key on the keyboard is pressed                    |
| onkeypress    | A key on the keyboard is pressed or held down       |
| onkeyup       | A key on the keyboard is released                   |
| onload        | A page or an image is finished loading              |
| onmousedown   | A mouse button is pressed                           |
| onmousemove   | The mouse is moved                                  |
| onmouseout    | The mouse is moved off an element                   |
| onmouseover   | The mouse is moved over an element                  |
| onmouseup     | A mouse button is released                          |
| onsubmit      | The submit button is clicked in a form              |



# APPLYING EVENT HANDLERS

- How to use `addEventListener` in `<script>`
- `window.addEventListener("click", myFunction);`
- *or as anonyms function:*
- `window.addEventListener("click", function() {`
- *function goes here*
- `});`

# JAVASCRIPT EVENT HANDLING

- Two other bad methods for Event Handling
  - As attribute on HTML element:
    - `<body onclick="myFunction();">`
    - Only single binds
  - As a method attached to a DOM object:
    - `window.onclick = myFunction;`
    - Only single binds; second bind will override first

# EVENT HANDLING EXAMPLE

- Check out the MDN Developer [Link](#)
- Example:
- ```
function addEventHandler(elem,eventType,handler) {  
  if (elem.addEventListener) {  
    elem.addEventListener (eventType,handler,false);  
  } else if (elem.attachEvent) {  
    elem.attachEvent ('on' + eventType,handler);  
  }  
}
```

JS Validators



WHICH ONE SHOULD I USE?

- Google Search it...
 - Validators are not uniform and some throw odd errs.
 - Some annoyingly demand double quotation marks
 - Some annoyingly demand 4 space tabbing.
-
- I am only judging you on your JS working
 - e.g. Producing the behavior you were coding for



class="nojs"



<html lang="en" class="nojs">

- Sniffing browsers for JS
- Basically like css media queries, but for JS
- If JavaScript is turned off:
 - Rendering won't replace .nojs with .js or .hasjs
 - So you can use .nojs to style your site differently when JS doesn't work.
- Also done by Modernizer JS Library
 - Object "Modernizer" is created
 - Example: Modernizer.canvas is created if browser support canvas API



Little more JS



JAVASCRIPT

//if alt-box is clicked, check the age verification box

```
$('#alt-box').on('click', function() {  
    $('#age-verification').prop('checked',true);
```

//if alt-box is clicked, add big checkmark to alt-box

```
$('#alt-box').html('<p>&#10004;</p>');  
});
```



DEMO



JAVASCRIPT

//our function

```
jQuery(function($) {  
  })
```

//allows for css class to replace user blocked js

```
$('#html').removeClass('nojs');  
$('#html').addClass('hasjs');
```



// When focus on #email find #input-email label, and addClass 'active'

```
$('#email').on('focus', function() {  
    $('#input-email label').addClass('active');  
});
```

// When not focused, or (if) email text is 0, remove the active class

```
$('#email').on('blur', function() {  
    if($('#email').val().length == 0) {  
        $('#input-email label').removeClass('active');  
    }  
});
```

