## ITMD-461 CLASS 12 APRIL 7, 2015

## TONIGHT'S AGENDA

- JavaScript Introduction
- Readings & Assignments
  - Lab 6
  - Quiz 2

#### 17MD-461

## JAVASCRIPT INTRODUCTION



## JAVASCRIPT INTRODUCTION

JavaScript is the behavioral layer of our web pages. HTML is structural, CSS is presentational

Can access all the elements, attributes and text on a web page using the DOM (Document Object Model)

Can test for browsers features and capabilities

Modify elements and css properties to show, hide, and change element appearance

Makes AJAX interactions possible

Historically support between different browsers has sometimes been mixed.

• Some browser implementations support some features and some use different names or syntax for a given feature.

## JAVASCRIPT INTRODUCTION

- Not Related to Java Programming Language
- Originally named LiveScript and created by Brendan Eich at Netscape in 1995. Later renamed JavaScript for marketing reasons because of popularity of Java Language at the time.
- Standardized by ECMA technically ECMAScript
  - Current latest stable version is ECMAScript 5 JS 1.8.5
  - Working on ES6 Draft, use it now with a complier like https://babeljs.io
- Lightweight Object-oriented scripting language
  - Procedural, object-oriented (prototype-based), and functional style
- Dynamic Language
  - Doesn't need to be compiled to machine code
  - Loosely typed Don't need to declare variable types
- Read and interpreted on the fly

# JAVASCRIPT<br/>INTRODUCTION

Mozilla JavaScript Guide

https://developer.mozilla.org/en-US/docs/Web/JavaScript

Wikipedia JavaScript Entry

http://en.wikipedia.org/wiki/JavaScript

Node.js for Command Line JavaScript Intro

http://javascript.cs.lmu.edu/notes/commandlinejs/

JavaScript and Basic Programming Introduction Reading

http://eloquentjavascript.net/

## JAVASCRIPT INTRODUCTION

**Embedded Scripts and External Scripts** 

Embedded Scripts

Use script tags <script> JS Here </script>

External Scripts

- Use script tag with src attribute <script src="myscript.js"></script>
- Script tag must be empty inside

Can be placed anywhere on the page

Most commonly in head section or at the bottom of the body before the closing body tag

Type attribute is required in < html4 but not html5

<script type="text/javascript"></script>

There is a new async attribute in html5, async="async" & defer="defer"

- Async executes as page parsing. Defer executes script when page finishes parsing.
- When either is not present (default), executes immediately then finishes parsing page

## JAVASCRIPT INTRODUCTION

- JavaScript is case-sensitive "foo" not equal "Foo"
- Made up of statements which should end with a semicolon.
- Contains reserved words you can not use. Search for a list of JavaScript reserved words for details.
  - https://developer.mozilla.org/en-US/docs/JavaScript/Reference/Reserved\_Words
- Comments can be single or multi line
  - Single Line two slashes // This is a comment
  - Multi Line similar to css /\* This is a comment \*/

#### 17MD-461

# LANGUAGE FEATURES AND SYNTAX

## JAVASCRIPT LANGUAGE

### Comprised mainly of:

- Variables
- Statements
- BIOCKS
- Functions
- Operators
- Comparison
- Conditional Statements
- Looping
- Objects
- events

# JAVASCRIPT VARIABLES

#### Variables hold values or objects

- Declare with var keyword var foo;
- Set value with single = sign var foo = 5;
- Names are case sensitive
- Names must begin with a letter or the underscore
- Can be a set of very basic data types (p 466)
- No special characters in name (! . , / \ + \* =)
- Has functional scope not block scope
- If a variable is declared in a function without var keyword it's global

### Array – grouping/list of objects

- Arrays are defined with new Array() or [ ]
- Zero indexed so first element is arrayname [0]

## JAVASCRIPT STATEMENTS

- Statements are commands to the browser that are executed in order
  - There are some built in statements/functions in our browsers.
  - A few real basic ones are alert(), confirm(), prompt()
- Statements can be grouped together in blocks with the curly brackets { }
  - Usually blocks are used when defining functions or using conditionals
  - JavaScript does not use block scope like most programming languages. It has function scope.

## JAVASCRIPT FUNCTIONS

### Functions (p 473)

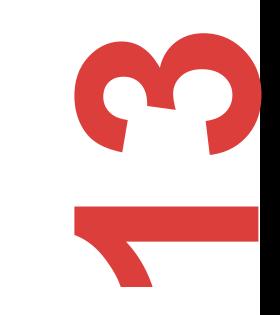
 Named blocks of code that can be called and executed by events or other code

```
function funcname (var1, var2, ...) {
code block (may make use of parameters)
}
```

 The return statement will stop executing the function and return the value

```
function addnum(n1, n2) {
return n1 + n2;
}
```

JavaScript has many built in functions



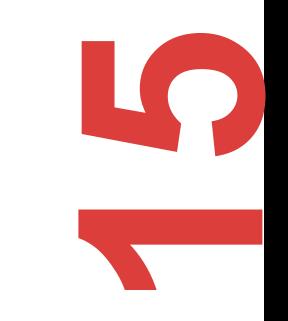
## JAVASCRIPT MATHEMATICAL OPERATORS

Mathematical operators are used to perform math on numeric objects (see page 469)

- 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

# JAVASCRIPT COMPARISON OPERATORS

- Comparisons are used to compare the value of two objects and return true or false
  - List of comparison operators on page 468
- Comparison Operators
  - == Is equal to
  - != Is not equal to
  - === Is identical to (equal to and 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</p>
- alert(5 > 1); // Will alert "true"



### JAVASCRIPT OBJECTS

- Objects All items except core data types in JavaScript are objects including functions
- Objects are basically a custom data type
- No class system in JavaScript like in other programming languages. Uses Prototypes instead.
- The browser is the window object the html page is the document object
- Objects are composed of properties and methods
  - Properties are basically variables
  - Methods are basically functions
- Access and objects property obj.propertyName
  - or obj["propertyName"]
- Execute an object method obj.methodName()



## JAVASCRIPT OBJECTS

#### JavaScript Objects Creation and Use

- Created by a function with new keyword
  - var obj = new Object();
- Created with an object literal
  - var obj = {};
     var obj = { key: value, key2: value2 };
     Key needs to be a string with no spaces
     var obj = { color: "red", quantity: 5, instock: true };
- Access or set properties with dot notation
  - obj.color = "blue"; sets color of obj to blue
  - obj.quantity; would be equal to 5
  - Can also set or execute methods this way
  - You can also access properties with the array like syntax of obj["color"]
    - Useful when you need the property value to come from another variable

### JAVASCRIPT OBJECTS

### JavaScript Object Literal format

An object literal is a comma separated list of name value pairs wrapped in curly braces.

```
var myObject = {
    stringProp: 'some string',
    numProp: 2,
    booleanProp: false
};
```

Value can be any JavaScript Datatype including a function or other object.

# JAVASCRIPT CONDITIONAL STATEMENTS

- Conditional statements (pg. 470)
  - if statements
  - else statements
  - else if statements

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



### JAVASCRIPT LOOPS

- Loops (pg. 471)
  - 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.
- For Loop Syntax

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

# JAVASCRIPT EVENT HANDLING

#### Event Handling

Three Methods (see page 478)

- As attribute on HTML element
- As a method attached to a DOM object
- Using the add event handler method of a object
  - object.addEventListener("click", myFunction);
  - IE < 9 needs to use a different format attachEvent
  - object.attachEvent('onclick', modifyText);

```
https://developer.mozilla.org/en-US/docs/DOM/element.addEventListener
function addEventHandler(elem,eventType,handler) {
  if (elem.addEventListener) {
    elem.addEventListener (eventType,handler,false);
} else if (elem.attachEvent) {
    elem.attachEvent ('on' + eventType,handler);
}}
```

### JAVASCRIPT DOM

- Document Object Model (DOM)
- Object representation of a HTML or XML Document
- All elements are represented by objects
- DOM is an API that can be used in many languages
- JavaScript uses DOM scripting to modify the elements on a page
- DOM is a collection of nodes in a tree (see 486)
- Also provides standard methods to traverse the DOM, access elements and modify elements



### JAVASCRIPT DOM

- Accessing the DOM elements
- Use methods of the document object (see p 487)
- 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. Some of the most common are listed on page 489



### JAVASCRIPT DOM

JavaScript & DOM Reference

http://reference.sitepoint.com/javascript/domcore

http://www.javascriptkit.com/domref/elementproperties.shtml

https://developer.mozilla.org/en-US/docs/DOM/element

https://developer.mozilla.org/en/docs/JavaScript

# JAVASCRIPT OBJECT REVIEW

### JavaScript Object Literal format

An object literal is a comma separated list of name value pairs wrapped in curly braces.

```
var myObject = {
    stringProp: 'some string',
    numProp: 2,
    booleanProp: false
};
```

Value can be any JavaScript Datatype including a function or other object.

#### 17MD-461

# READINGS AND ASSIGNMENTS

# READING & ASSIGNMENTS

#### 

- Read all JavaScript Chapters. Ch 19-20
- Look through additional JavaScript Books I uploaded to Blackboard. They will give you more details regarding jQuery.

### Assignment

- Lab 6 Due April 11 11:59pm Chicago time
  - Quiz 2 Due April 11 11:59pm Chicago time