

# Week12

## Intro to JavaScript

# The Course

Markup (XHTML, HTML, HTML5) - structure

Style (CSS) - style

Functionality (JavaScript CSS3 / HTML5) - function

Functionality (JavaScript frameworks)

# JavaScript

- an object-oriented scripting language
- builds interactions between web page content, the state of the browser, and the actions of the reader
- makes web pages interact with users and respond to what they do
- works in all major browsers

Firefox

Chrome

Opera

Safari

Internet Explorer

# JavaScript

- Created by Netscape in 1995
- JavaScript is NOT Java
- Netscape and Microsoft worked to standardize JavaScript through ECMA International as ECMAScript
- JavaScript is a powerful object-based scripting language with support for proper software engineering techniques
- JavaScript is most commonly seen in use on the Web, but is used in many other places

# JavaScript

- Servers
- Rich web client libraries
- HTML5
- Databases
- JavaScript-based languages (ex. jQuery and jQuery Mobile)

# JavaScript

- gives HTML designers a programming tool
- can react to events
- can read and write HTML elements
- can be used to validate data
- can be used to detect the visitor's browser
- can be used to create cookies

# JavaScript

## Internal JavaScript

`<head>` or `<body>` sections of the HTML document

## External JavaScript

External JavaScript files have the file extension `.js`

External JavaScript files often contain code to be used on several different web pages

# JavaScript

JavaScript provides *programmatic* access to virtually all aspects of a page:

CSS properties

Markup content

Forms, communication to Server

Add functionality



# JavaScript References

W3schools

<http://www.w3schools.com/js/default.asp>

JavaScript Guide

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

# JavaScript in HTML Document

```
<script language="JavaScript">
```

```
...
```

```
</script>
```

# JavaScript example: monday.html

```
<html>
<head>
<title>Javascript basics</title>
</head>
<body>
<p>
<form>
<input value="Press" type="button" onClick="alert('Happy Monday')">
</form> </p>
<p>
<script language="JavaScript"> document.write("Updated:");
document.write(document.lastModified);
</script> </p>
</body>
</html>
```

# JavaScript Language Features

Data types

Constants

Variables

Expressions

Statements

Operators

Statements: conditional, loops

Functions

Methods

Events

# JavaScript example: hello.html

```
<html>
<head>
<title>hello</title>
<script>
<!-- hide script from old browsers
document.write("Hello class!");
// end hiding script from old browsers -->
</script>
</head>
<body>
</body>
</html>
```

# JavaScript Variables and values

```
var first=window.prompt("Please enter your name:", "");  
document.write("Welcome to the world of JavaScript, " + first + "<br>");
```

`var first=...` declares a variable and sets the value to user input  
(assignment statement)  
;  
statement terminator

JavaScript is case sensitive

A declared variable is local

Reserved words cannot be used as variable names

# JavaScript example: welcome.html

```
<html>
<head>
<title>welcome</title>
<script language="JavaScript">
<!--
var first=window.prompt("Please enter your name:", "");
document.write("Welcome to the world of JavaScript, " + first + "<br>");
//-->
</script>
</head>
<body>
</body>
</html>
```

# JavaScript Data Types

- Numbers     0.44
- Strings             `document.write ("greeting"+mj);`  
in quotations ( ' or ")  
`<input value="Press" type="button" onClick="alert('HELLO')">`
- Null "empty"
- String literals  
`alert("I am an alert box!! \n\t Man!");`  
when HTML is not in use, adds a new line and a tab
- Boolean values     true, false



# JavaScript example: window.html

```
<html>
<head>
<title>window</title>
<script>
</script>
</head>
<body>
<a href="#" onmousedown="window.open('http://
www.uic.edu','UIC','width=800,height=600')">open new window </a>
</body>
</html>
```

# JavaScript Events

click

change

submit

load

mouseover

mouseout

focus

blur

select

keydown

keyup

keypress

unload

mousedown

mousemove

mouseup

Reset

dblclick

# JavaScript Event Handlers

<code>onmousedown</code>	A mouse button is pressed
<code>onmouseout</code>	The mouse is moved off an element
<code>onmouseover</code>	The mouse is moved over an element
<code>onreset</code>	The reset button is clicked
<code>onresize</code>	A window or frame is resized
<code>onselect</code>	Text is selected
<code>onsubmit</code>	The submit button is clicked
<code>onunload</code>	The user exits the page
<code>setTimeout(), clearTimeout()</code>	timer is activated

# JavaScript Event Handlers

onabort	Loading of an image is interrupted
onblur	An element loses focus
onchange	The user changes the content of a field
onclick	Mouse clicks an object
ondblclick	Mouse double-clicks an object
onerror	An error occurs when loading a document or an image
onfocus	An element gets focus
onkeypress	A keyboard key is pressed or held down
onload	A page or an image is finished loading

# JavaScript example: rollovers.html

```
<html>
<head>
<title> rollovers </title>
<script language="JavaScript">
</script>
</head>
<body>
<a href="#"
onmouseover="document.sample.src='images/image2.jpg';
return false;"
onmouseout="document.sample.src='images/image1.jpg';
return false;" >

</a>
</body>
</html>
```

# JavaScript Arrays

```
var myPix = new Array("images/red.gif", "images/  
green.gif", "images/blue.gif")
```

myPix.length            gets value of all 3 elements

myPix[0]            contains image "images/red.gif"

["images/red.gif", "images/green.gif", "images/blue.gif"] Contains  
array

Arrays can contain different types of data document.images[0].src  
= pics [frame].src

# JavaScript Expressions

`i <= 10` conditional expression: true or false

String operation:

`"result is" + summary`

Statement:

```
timerID = setTimeout('alternate()', 800);  
;      statement terminator
```

# JavaScript Operators

## Assignment Operators

+	addition
x+=y	is the same as x=x+y
x++	same as x=x+1
-	Subtraction
*	Multiplication
/	Division
%	remainder



# JavaScript Operators

Comparison Operators, true or false

== is equal to

!= is not equal

5!=8 returns true

< less than

> greater than

>= greater than or equal

<= less than or equal

# JavaScript Operators

## Logical Operators

&&    AND

||    OR

!    NOT

# JavaScript Conditional Statements

```
if ( !Math.random ) // here you check existence of a function
{
document.write('<em> -- weather called off due to rain --</em>');
}
else if ( Math.floor((Math.random()*2)) == 0 )
{
document.write ("<b>It's just awful. </b>");
}
else
{
document.write ("<em>How wonderful it is!</em>");
}
```

# JavaScript example: random.html

```
<html>
<head>
<title>Random Script</title>
<script>
var myPix = new Array("images/red.gif", "images/green.gif", "images/
blue.gif")
var thisPic = 0;
function choosePic()
{
  if(document.images)
  {
    randomNum = Math.floor(Math.random()*myPix.length)
    document.image.src=myPix[randomNum]
  }
}
}
```

# JavaScript example: random.html

```
</script>  
</head>  
<body onLoad="choosePic()">  
  
</body>  
</html>
```

# JavaScript Loops – loop.html

```
function myFunction() {  
  var x="";  
  for (i=0;i<50;i++)      /* Increment i=i+1 or i++ */  
  {  
    x=x + "The number is " + i + "<br>";  
  }  
  document.getElementById("demo").innerHTML=x;  
}
```

<p>Click the button to loop through a block of as long as <em>i</em> is less than 50.</p>

<button onclick="myFunction()">Try it</button>

<p id="demo"></p>

# JavaScript Functions: functions.html

User defined

Predefined alert prompt

parseInt	converts variable into an integer
parseFloat	converts variable into a number
Math.sqrt	square root
Math.floor	rounding to the lower integer
Math.round	rounding

# JavaScript Functions: functions.html

```
<html>
<head>
<title> Functions: user defined </title>
<script type="text/javascript">
function disp_alert()
{
alert("I am an alert box!!")
}
</script>
</head>
```



# JavaScript Functions: functions.html

```
<body>  
<form>  
<input type="button" onclick="disp_alert()" value="Display  
alert box">  
</form>  
</body>  
</html>
```

# JavaScript Functions

Ex. `Functions_countdown.html`

# JavaScript Functions

- `close()`
  - Closes an output stream opened with the `document.open()` method, and displays the collected data
- `getElementById()`
  - Returns a reference to the first object with the specified id
- `getElementsByName()`
  - Returns a collection of objects with the specified name
- `getElementsByTagName()`
  - Returns a collection of objects with the specified tagname
- `open()`
  - Opens a stream to collect the output from any `document.write()` method
- `document.write()`
  - Writes HTML expressions or JavaScript code to a document

# JavaScript Functions

Object-oriented:

Instead of writing procedural programs, write class libraries to encapsulate behaviors

DOM is not a collection of dumb elements but a hierarchy of types

Styles are properties of objects

Complete OO code with error handling, instance methods, static methods and type hierarchies

Versatile use of functions

A large number of object-oriented libraries

Used to create User Interfaces

# JavaScript example: bgcolor.html

```
<html>
<head>
<title>bgcolor_change</title>
<SCRIPT LANGUAGE="JavaScript">
<!-- Begin
function newbg(thecolor)
{
document.bgColor=thecolor;
}
// End -->
</script>
</head>
```

## JavaScript example: bgcolor.html

```
<body textcolor="black" link="black" alink="black">  
<center>  
  <a href="#" onmousedown="newbg('olive');"> olive</a><br />  
  <a href="#" onmousedown="newbg('blue');"> blue</a><br />  
  <a href="#" onmousedown="newbg('Beige');"> beige</a><br />  
  <a href="#" onmousedown="newbg('yellow');">yellow</a><br />  
</center>  
</body>  
</html>
```

# JavaScript example: bgcolor.html

## Exercise

Add two background colors of your choice to this example

# JavaScript: exercise

## Random Background color

Create a script that assigns a random background color to the HTML document (changes background color randomly)

when the user **clicks** on the

[“Random Background Color”](#) link.