



ITEC 136

Business Programming Concepts

Week 1
Module 1: Overview of HTML
and Introduction to JavaScript

FRANKLIN UNIVERSITY

FOUNDED 1902

Agenda

- Course overview
- This week's expected outcomes
- This week's topics
- This week's homework
- Upcoming deadlines
- Questions and answers

Course Overview

- Course Outcomes
 - Explain the stages of the software lifecycle
 - Design solutions using top-down stepwise refinement
 - Implement solution algorithms using selection and repetition control structures

Conditional constructs

Looping constructs



Course Overview

- Course Outcomes
 - Translate algorithms into clearly documented and modularized programming language code
 - Document and Debug Code
 - Generate and execute test plans



Course Overview

- Books
 - Primary: JavaScript by Gosselin
 - Secondary: Learning JavaScript by Powers
- Additional
 - Key points on web site
 - Web, tutorials, etc.



Course Overview

- Why is this important?
 - Solid grounding in programming logic and design
 - Understanding of the software development lifecycle
 - Automation of tasks
 - Pre-requisite for other language-based coursework

Why JavaScript

- Ubiquitous on systems since 1996
- Syntax derived from Java/C++ but is neither!
- Dynamic language – change easily
- Key Component for Web 2.0 interfaces
- Can be embedded in other programs
- It can be fun!

JavaScript for Coding

- Why not language X or Y?
 - Not a specific language-focused course
 - Runs Within Internet Browser
 - Principles apply to most languages
 - Variables
 - Functions
 - Control structures
 - Testing

Course Overview


Wet, lather, rinse, repeat. Wet, lather, rinse, repeat...

- Course Structure
 - Lots of practice (“shampoo” method)
 - Reading
 - FranklinLive presentations
 - Homework Exercises
 - Lab Exercises
 - Two Exams
 - Final exam

Increasing difficulty and point value



Course Overview

- Tools you will need
 - Aptana Integrated Development Environment (IDE)
 - A standards-compliant web browser (Mozilla Firefox for use with Aptana)
 - Firebug
 - Your textbooks
 - Patience and experimentation! 



Other Useful Resources

- Safari Books Online
 - <http://www.safaribooksonline.com>
 - Also accessible through Franklin library online
- Google (or preferred search engine)
- Any number of web sites
 - <http://www.pageresource.com/> (tutorial)
 - <http://jennifermadden.com/> (tutorial)
 - <http://www.javascriptmall.com/learn/contents.htm> (tutorial)
 - <http://www.squarefree.com/shell/> (sandbox)
 - <http://www.webreference.com/> (reference tutorial)
 - <http://www.visibone.com/> (fact sheets \$)

VisiBone

- Information packed
- spiral bound
- Covers
 - HTML/XHTML tags
 - JavaScript
 - CSS and Styles
 - Fonts and Colors
 - Document Object Model
 - Regular Expressions



Week 1 Outcomes

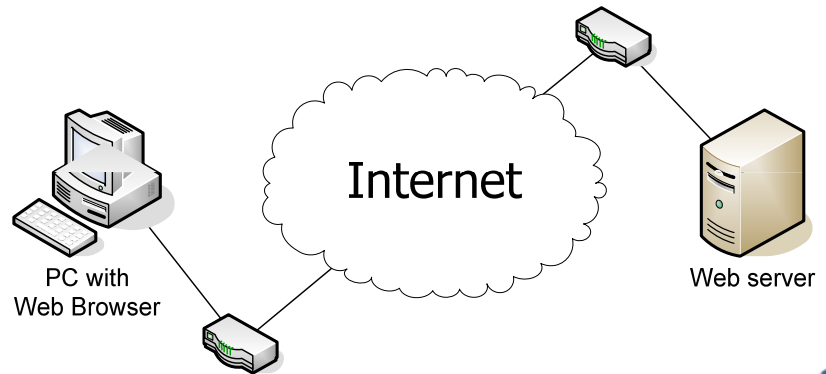
- Create and display simple syntactically correct HTML documents
- Describe the execution a JavaScript program in a web browser

Week 1 Preparation

- Have You Already?
 - Read Key Points on Course Web Site
 - Read Chapter 1 in Both Books
 - Reviewed the HTML Primer

HTML Primer

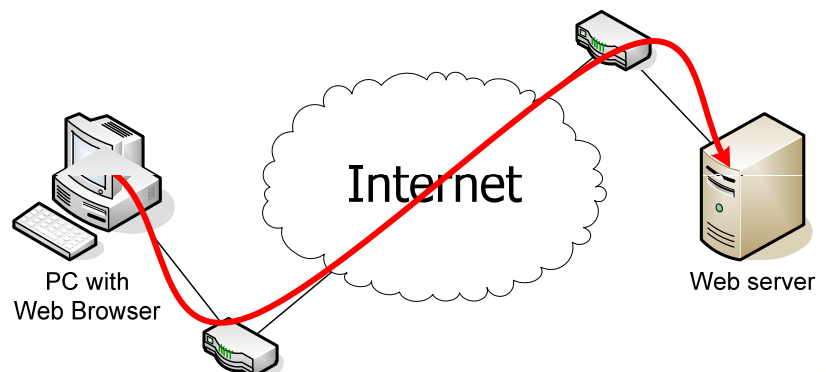
- How does the web work?
 - Request and response cycle
 - Text and binary transfer over HTTP



HTML Primer

- How does the web work?
 - Browser sends HTTP request as text

```
GET /article.pl?sid=06/12/19/2256259/ HTTP/1.1  
Host: it.slashdot.org
```



HTML Primer

- How does the web work?
 - Server sends back HTTP response

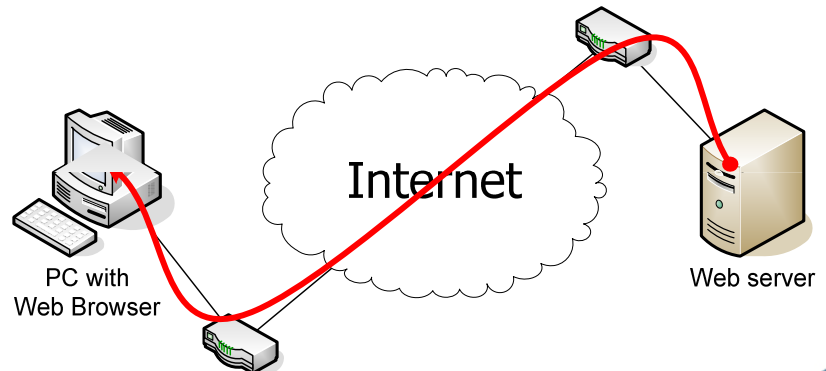
HTTP/1.1 200 OK

Date: Thu, 21 Dec 2006 21:31:30 GMT

Content-Type: text/html

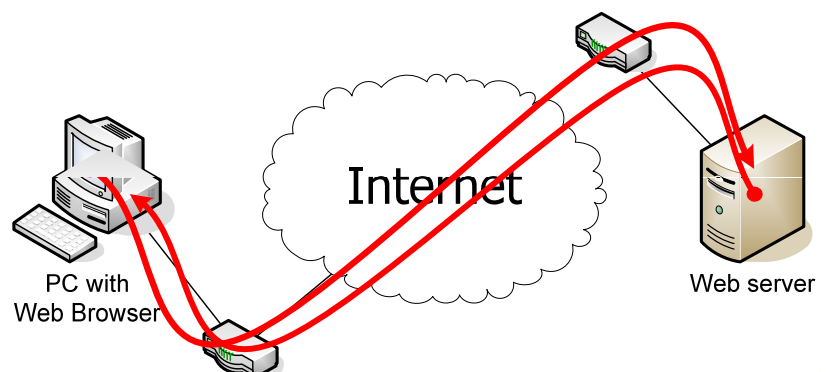
Content-Length: 1354

```
<html>
<body>
<h1>Should JavaScript
Get More Respect?</h1>
(more file contents)
.
.
</body>
</html>
```



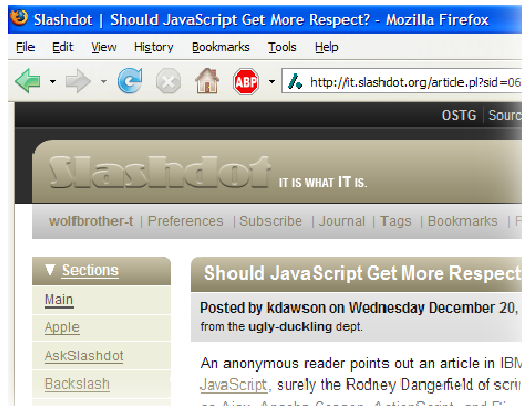
HTML Primer

- How does the web work?
 - Web browser parses document to identify and request other resources.



HTTP and the Web

- Web browser renders the text as a formatted document based on HTML



HTML Primer

- HTML: **H**yper**T**ext **M**arkup **L**anguage
- Use `<` and `>` to delimit an HTML element – usually called a tag
- Open & Close tags around info
 - `<x>info</x>`
- No element? Can use `/` in single tag, **must** have space prior to `/`
 - `<x />`

HTML Examples

- `<title>JavaScript 101</title>`
- ``
- ``
- `<head><title>Hello World</title></head>`

Two contrasting `` tags. First is becoming more popular

Can be nested. The `<title>` tag is inside the `<head>` tag



Full Document HTML

- Document structure

```
<!DOCTYPE html PUBLIC
"-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html>
  <head>
    <title>Should JavaScript Get More Respect?</title>
  </head>

  <body>
    <h1>Should JavaScript Get More Respect?</h1>
    <p>An anonymous reader points out an article...</p>

    <!-- (more file contents) -->
  </body>
</html>
```



HTML Primer

- Elements (tags)

Element	Within	Purpose
<html>		HTML content
<head>	<html>	Page metadata
<title>	<head>	Title of the page
<body>	<html>	Page data
<h1>	<body>	Heading level 1
<p>	<body>	Paragraph data

HTML Primer

- Elements (tags)

Element	Within	Purpose
<div>	<html>	Arbitrary page section
		Arbitrary inline section
	<html>	Ordered list
	<html>	Unordered list
		Numbered list item
		Bulleted list item

HTML Primer

- Elements (tags)

Element	Within	Purpose
<code><table></code>	<code><html></code>	Table
<code><tr></code>	<code><table></code>	Table row
<code><td></code>	<code><tr></code>	Table data (cell)
<code><dl></code>	<code><html></code>	Definition list
<code><dt></code>	<code><dl></code>	Definition term
<code><dd></code>	<code><dl></code>	Definition data

HTML Primer

- Elements (tags)

Element	Within	Purpose
<code>
</code>	<code><html></code>	Line break
<code><hr /></code>	<code><html></code>	Horizontal rule
<code></code>		Image (JPEG, GIF, PNG)
<code><form></code>	<code><html></code>	Form for user input
<code><input /></code>	<code><form></code>	User input field (text, button)
<code><script></code>		Executable code

HTML Entities

- Entities (special characters)

Code	Render
"	"
&	&
<	<
>	>
 	
©	©

Code	Render
®	®
™	™
£	£
€	€
“	“
”	”

A space, HTML typically ignores spaces and CRs in the HTML



HTML Primer

- Attributes
 - Parameters to elements
`Franklin`
 - Refines or further defines the tag

href: hyperlink reference to another page.



HTML Primer

- Attributes

- Parameters to elements

```
<div id="sidebar">As an aside...</div>
```

id: internal identifier used to programmatically access and alter the division.

HTML Primer

- Attributes

- Parameters to elements

```
<script type="text/javascript"  
  language="JavaScript"  
  src="employee.js"></script>
```

type: programming language the script uses.
language: default scripting language and version.
src: file from which to read.

HTML Primer

- Attributes

- Parameters to elements

```
<input type="text" name="last" id="last" />
```

Programming Tip

Set id= and name= to the same value.

Ex: id="txt" name="txt"

type: kind of input control.

name: field name sent to server.

HTML Primer

- Attributes

- Parameters to elements

```

```

src: location of image file.

Try it out!

- Create a “hobby” page
 - Specifications
 - Title
 - Picture
 - Paragraph description
 - Bulleted list with links to other hobby pages

Introduction to JavaScript

- Dynamic web pages
 - HTML provides static content
 - Client-side scripting provides
 - Dynamic content
 - Form validation
 - Interactivity
 - Special effects

Introduction to JavaScript

- Client-side scripting restrictions
 - No access to the file system
 - No network communication outside the originating domain
- JavaScript is case dependent

The <script> tag

index.html

```
1. <script type="text/javascript">
2.   //<![CDATA[
3.   document.writeln("Hello");
4.   //]]>
5. </script>
```

Alternative style as W3C validate
dislikes the CDATA method

index.html

```
1. <script type="text/javascript" language="JavaScript">
2. <!-- Hide script from old browsers.
3.   // Put code below.
4.   document.writeln("Hello");
5.   // Stop hiding script from old browsers. -->
6. </script>
```



Introduction to JavaScript

- Hello world program

index.html

```
1. <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML...>
2. <html>
3.   <head><title>Hello World</title></head>
4.   <body>
5.     <script type="text/javascript" language="JavaScript">
6.       <!-- Hide script from old browsers.
7.         document.write("<h1>Hello World</h1>");
8.       // Stop hiding script from old browsers. -->
9.     </script>
10.  </body>
11. </html>
```

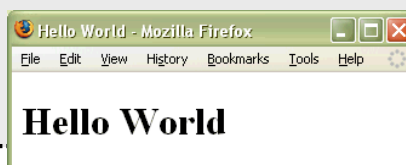


Introduction to JavaScript

- Hello world program

index.html

```
1. <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML...>
2. <html>
3.   <head><title>Hello World</title></head>
4.   <body>
5.     <script type="text/javascript" language="JavaScript">
6.       <!-- Hide script from old browsers.
7.         document.write("<h1>Hello World</h1>");
8.       // Stop hiding script from old browsers. -->
9.     </script>
10.  </body>
11. </html>
```



Introduction to JavaScript

- Comparing `writeln()` & `write()`

```
document.writeln("<h1>Hello World</h1>");
```

Appends a carriage return (in the HTML)

```
document.write("<h1>Hello World</h1>");
```

Does not append a carriage return.



Introduction to JavaScript

- JavaScript objects
 - Calling object methods:
`noun.verb(parameters)`

```
document.writeln("<h1>Hello World</h1>");
```



Introduction to JavaScript

- JavaScript objects
 - Calling object methods:
noun.verb(parameters)

```
document.writeln("<h1>Hello World</h1>");
```

Object (noun)



Introduction to JavaScript

- JavaScript objects
 - Calling object methods:
noun.verb(parameters)

Method (verb)

```
document.writeln("<h1>Hello World</h1>");
```



Introduction to JavaScript

- JavaScript objects
 - Calling object methods:
`noun.verb(parameters)`

```
document.writeln("<h1>Hello World</h1>");
```

Parameter (extra needed information)

Introduction to JavaScript

- JavaScript objects
 - Changing object properties:
`noun.property = new_value`

```
document.bgColor = "#AABBCC";
```

Introduction to JavaScript

- JavaScript objects
 - Changing object properties:
`noun.property = new_value`

```
document.bgColor = "#AABBCC";
```

Object (noun)

Introduction to JavaScript

- JavaScript objects
 - Changing object properties:
`noun.property = new_value`

```
document.bgColor = "#AABBCC";
```

Property (noun)

Introduction to JavaScript

- JavaScript objects
 - Changing object properties:
`noun.property = new_value`

```
document.bgColor = "#AABBCC";
```

New value for
property

Introduction to JavaScript

- Prompting for input

```
index.html
```

```
1. <script type="text/javascript">  
2.   <!-- Hide script from old browsers.  
3.   var name = prompt("What is your name?");  
4.   document.writeln("<h1>Hello "  
5.     + name + "</h1>");  
6.   // Stop hiding script from old browsers. -->  
7. </script>
```


Introduction to JavaScript

- Prompting for input

index.html

```
1. <script type="text/javascript">
2.     <!-- Hide script from old browsers.
3.     var name = prompt("What is your name?");
4.     document.writeln("<h1>Hello "
5.         + name + "</h1>");
6.     // Stop hiding script from old browsers. -->
7. </script>
```

Introduction to JavaScript

- Prompting for input

index.html

```
1. <script type="text/javascript">
2.     <!-- Hide script from old browsers.
3.     var name = prompt("What is your name?");
4.     document.writeln("<h1>Hello "
5.         + name + "</h1>");
6.     // Stop hiding script from old browsers. -->
7. </script>
```

The variable, name, temporarily holds the user input

Introduction to JavaScript

- Prompting for input

index.html

```
1. <script type="text/javascript">
2.   <!-- Hide script from old browsers.
3.     var name = prompt("What is your name?");
4.     document.writeln("<h1>Hello "
5.       + name + "</h1>");
6.     // Stop hiding script from old browsers. -->
7. </script>
```

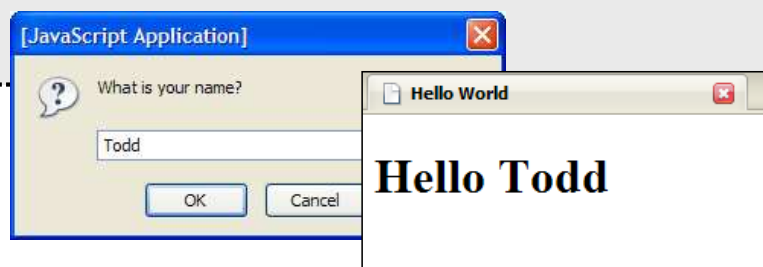
Whatever content the user typed is substituted for the variable.

Introduction to JavaScript

- Prompting for input

index.html

```
1. <script type="text/javascript">
2.   <!-- Hide script from old browsers.
3.     var name = prompt("What is your name?");
4.     document.writeln("<h1>Hello "
5.       + name + "</h1>");
6.     // Stop hiding script from old browsers. -->
7. </script>
```



Introduction to JavaScript

• External JavaScript files

index.html

1. `<script type="text/javascript" src="hello.js">`
2. `</script>`

hello.js

1. `var name = prompt("What is your name?");`
2. `document.writeln("<h1>Hello " + name + "</h1>");`

Introduction to JavaScript

• External JavaScript files

index.html

1. `<script type="text/javascript" src="hello.js">`
2. `</script>`

hello.js

1. `var name = prompt("What is your name?");`
2. `document.writeln("<h1>Hello " + name + "</h1>");`

Why use separate files?

- Keeps HTML (presentation) separated from JavaScript (programming logic)
- Allows many HTML pages to reuse a set of scripts
- Browsers can cache the files separately
- Browsers that don't understand JavaScript ignore it rather than render

JavaScript Placement

- Where should `<script>` be put?
 - In the `<head>` section
 - To load external scripts
 - To define new functions
 - In the `<body>` section
 - To execute as or after the page loads

JavaScript Summary

- JavaScript is case dependent
- Use `<script>` for inline and external JavaScript code
- Hide script from older browsers by using HTML comments or the CDATA method around JavaScript code blocks
- Separate code into separate HTML and JS files

Try it out!

- Edit your "hobby" page to include
 - A script in the body that prompts the user for their name, and then renders a custom welcome message using their name on your page.

Self Quiz

- What is the difference between a JavaScript *method* and a JavaScript *property*?
- How are HTML tags that have no matching closing tag written? Give an example.

Self Quiz

- What is one advantage of placing your JavaScript in a separate file versus writing it all in the HTML document?
- What does a DTD (Document Type Definition) do?
- What is the difference between an HTML *attribute* and an HTML *tag*?



ITEC 136

Business Programming Concepts

Debugging Tools

FRANKLIN UNIVERSITY


FOUNDED 1902

Debugging Tools

- Old school approach
 - Logging: debugging statements placed strategically in program code

```
function log(div, message) {  
    document.getElementById(div).innerHTML  
        += message + "<br />";  
}  
// then later...  
if (debug == true) {  
    log("debug", "Got to here");  
}
```

Debuggers

- **Debugger:** Programs that allow you to examine the state of another running program 
- Built into the IDE
 - Stop your program at a particular point (*breakpoint*)
 - Inspect the contents of a variable (*inspect* or *watch*)
 - Step through a program as it executes (*step into*, *step over*, *step out*)

Typical Debugging Session

- Set a breakpoint in your code just prior to where:
 - You think a problem is occurring
 - The Debugger has shown an error
- Run the program in debug mode, Debugger will suspend program at breakpoint

Typical Debugging Session

- Examine variables at breakpoint to determine what may have gone wrong. Use the watch or inspect features
- Step forward through the program line-by-line to examine how objects or variables change

Firebug Tutorial & Video

- <http://getfirebug.com/>
- Links in Key Points (14.2)
 - Firebug Tutorial #1
 - <http://alternateidea.com/blog/articles/2006/05/12/an-in-depth-look-at-the-future-of-javascript-debugging-with-firebug>
 - <http://alternateidea.com/blog/articles/2006/05/12/an-in-depth-look-at-the-future-of-javascript-debugging-with-firebug>
- Video Screencasts – Excellent!
 - <http://www.digitalmediaminute.com/screencast/firebug-js/>
 - <http://files.jnewland.com/firebug.mov>

Upcoming Deadlines

- Homework 1 due Jan 12
- Readings for next week
 - Chapter 2 in *JavaScript*
 - Chapter 2 and part of chapter 3 in *Learning JavaScript*
 - Module 2 Key Points - Software Life Cycle

General Q & A

- Questions?
- Comments?
- Concerns?