Announcements

- **Project #2 has been posted**
- **Term Paper assignment will be posted soon**
- **First exam is on Monday 10/9 (ten days from now).**

JavaScript Overview

- **HTML/CSS is not really "programming"**
- **JavaScript is a "scripting" (lightweight programming) language.**
- **JavaScript programs can be embedded into web pages**
- **Web pages become "dynamic" (interactive)**
  - **Perform calculations**
  - **Respond to user's actions**
  - **Modify behavior based on circumstances**
  - **Repeat tasks/outputs**
  - **Communicate with server**
- **JavaScript is not Java**

Where does JavaScript Code Go?

Ways to include JavaScript in your webpage:

1. **As an attribute to a particular element:**

```
<p onClick="…">Click this paragraph!</p>
```

2. **On its own:**

```
<script>
    …
</script>
```

3. **In a separate file (ending with .js)**

```
<script src="myJavaScriptFile.js"></script>
```

Statements

- **A Javascript program is a series of *statements***
- **Statements are individual instructions**
- **Statements are executed one-by one from the top down**

Punctuation, Etc.

We recommend:

- **End each statement with a semi-colon**
  - **JavaScript does not require it**
- **Use spacing (indenting, blank lines, etc.) to make code more readable**
  - **JavaScript ignores these extra spaces**

Strings

A *string* is a sequence of characters (symbols)

Use quotes (“ “) or single quotes (‘ ‘):

- **“This is a string”**
- **‘This is a string’**

Use + to *concatenate* (join) strings:

- **“This” + “ is “ + “a string”**

Output

Simple Pop-Up Box:

```
alert("Put message here!");
```

Replace the content of an element:

```
myIdentifier.innerHTML = "something";
```

Append to the content of an element:

```
myIdentifier.innerHTML += "more stuff";
```

Example: Output.html

Try this one in different browsers!

Arithmetic Operators

- **+**
- **–**
- *****
- **/**

All work as expected!

Variables

- **Memory location with a name.**
- **Used to store a value**
- **Values can be numbers, strings, or objects**
- **Always use "var" to create a *local* variable**
- **Use *assigment operator* (=) to assign a value to a variable.**

```
var x;
x = 77;

var y = 22;
var z = 18.95;
var s = "Hello there.";
```

Example: VariablesOutput.html

Names of Variables

Rules

- **May use letters, digits, underscore '_'**
- **First character may not be a digit**
- **Avoid "reserved" words: `alert`, `var`, `innerHTML`, many others**

“Good Form” for Variables

Use “Camel Case”:

- **Lower-case letters**
- **First letter of subsequent words capitalized**
- **Examples of camel-case variable names:**
  - `temperature`
  - `userInput`
  - `numberOfPlayers`
  - `columnNumber`

Names of Variables

Choose meaningful names:

Good names	Bad names
temperature	temp
shoeSize	sSize
row	r
maximumWidth maxWidth	max maxW

Input

Use *prompt* function for basic dialog box:

```
var name = prompt("What is your name?");
```

Result will always be a string.

Example: Input.html

Type Conversions

Usually JavaScript automatically converts things:

```
string1 = "40";
string2 = "30";
product = string1 * string2;  // works
```

It doesn't always work out…

Example: NumberConversion1.html

Type Conversions

To convert from string to number:

```
"40"                // This is a String (text)
Number("40")        // This is the number 40
```

Always Use Number function when prompting for a numerical value. (We should fix the "Input" example!)

```
var size = Number(prompt("Enter size: "));
```

Example: NumberConversion2.html

More Math…

There are lots of built-in math functions.

Examples:

```
result = Math.abs(-7);        // result will be 7
result = Math.max(3, 23);     // result will be 23
result = Math.min(3, 23);     // result will be 3
result = Math.sqrt(16);       // result will be 4
result = Math.PI;             // 3.1415926….
result = Math.random();       // result will be a
                              // random value
                              // between 0 and 1
```

Comparisons and Boolean Variables

Comparison Operators

These work as you would expect:

```
x < y
x > y
x <= y
x >= y
```

These are "boolean expressions" (either true or false)

Equality Operators

Checking for equality is confusing

There are TWO equality operators:

`x == y` **true if x and y evaluate to the same value**

`x === y` **true if x and y are the same TYPE and also evaluate to the same value**

We strongly favor using === **(You'll see why later…)**

Example:

```
x = 5;
y = "5";
x == y   // this is true
x === y  // this is false
```

Example: Comparisons.html

Equality Operators

Two more operators:

`x != y`	**opposite of**	`x == y`
`x !== y`	**opposite of**	`x === y`

If-Statement

Behavior depends on a "condition"…

Simple Example:

```
x = Number(prompt("Enter first value: "));
y = Number(prompt("Enter second value: "));
if ( x < y ) {
     alert("THE FIRST VALUE WAS SMALLER");
}
alert("That was fun.");
```

Example: IfStatements.html

If-Else-Statement

Two possible paths – One or the other will happen…

Simple Example:

```
x = Number(prompt("Enter first value: "));
y = Number(prompt("Enter second value: "));
if ( x === y ) {
     alert("THE VALUES ARE THE SAME");
} else {
     alert("THE VALUES ARE DIFFERENT");
}
alert("That was fun.");
```

Example: IfElseStatements.html

Nested Conditionals

Conditional statements can be nested inside each other:

```
if (…) {
     if (…) {
     } else {
     }
} else {
     if (…) {
     }
}
```

Example: Nested.html

Comments

Two styles available:

```
/* Put comment here.  Comment
   Can span multiple lines… */

// Put single-line comment here
```