MIS3690 WEB TECHNOLOGIES

BABSON COLLEGE

TOIM DIVISION

UNDERSTANDING VARIABLES
MANIPULATING IMAGES
MANIPULATING STYLES

AGENDA

- Understanding variables (Clear and Present Danger)
  - How to DECLARE (or create) a variable
  - How to assign values
  - How to operate with variables
- Back to DOM
  - Understanding the Object
  - Understanding methods (built-in) functions
  - Understanding how to manipulate properties through variables
- Context
  - Using ALERT – a built-in function
  - Manipulating image properties (source, width, height etc.)

VARIABLE IN JAVASCRIPT

- A variable is a temporary holding place for keeping web page elements, their properties, or values.
- We create (or *declare*) variables in JavaScript using
  - `var myForm;`
  - `var userChoice;`
- A variable is valid only within the function where it is created (there are some exceptions – for later…)
- Typically, variables make it easy to write functions.
- You can assign values to variables:
  - `myColor="red";`
- You can name a variable anyway you want – just do not use "reserved" words (e.g., don't name a variable as "form" or "element" or "color")

JAVASCRIPT VARIABLES

- Variable's value can change.
- Example:

```
var x;
x=10;
//Some JS statements
x=20;
```

- Variable can be created and given a value in one step.
- Example:

```
var x=10;
```

OPERATING ON VARIABLES

- + Addition
  - Example:
```
var x = y + 2;
```
- - Subtraction
  - Example:
```
x = x - 3; y = x - 3;
```
- * Multiplication
  - Example:
```
y = x * 17 + y;
```
- / Division
  - Example:
```
y = x / 10 + y;
```
- ++ increment by 1
  - Example:
```
x = ++y;    x = y++;
```
- -- decrement by 1
  - Example:
```
x = --y;    x = y--;
```
- Test them: http://www.w3schools.com/js/js_operators.asp

KEY THINGS TO REMEMBER ABOUT JAVASCRIPT

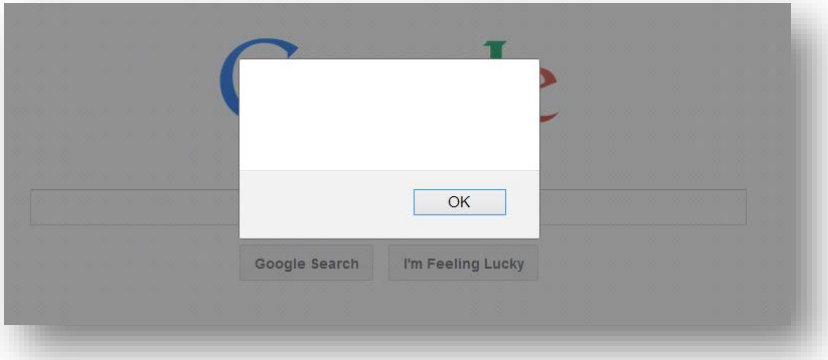
- You must end each JavaScript statement with a semicolon (;) or a new line (or both)
- Each statement is either
  - A JavaScript command (we will learn about these)
  - Or a JavaScript function (we will learn this too!)

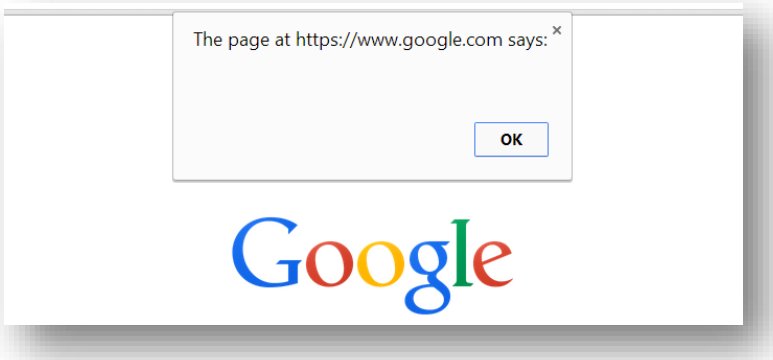
JAVASCRIPT FUNCTIONS

- Remember: A function is a set of instructions to the browser to do something
- You can create your own functions
  - We'll learn the basics
  - Like the one we just did
- Or, you can find functions and "copy" them
  - Some websites providing free functions
    - http://www.java-scripts.net/
    - http://www.javascriptkit.com/cutpastejava.shtml
  - By viewing the source of web pages that have JavaScript on them
  - if you don't understand the function, please do not simply copy and paste.
- There are pre-defined functions already available

PRE-DEFINED FUNCTIONS

- Pre-Defined Function Example:
  - `alert()` – Pops up an alert box like that shown below





  - `alert (x)` – where `x` is a variable – notice no quotes around x
    - Will show the alert box with the VALUE that is held in the variable x
    - Example: `alert("My name is Zhi");`
      - Will show the alert with the string "My name is Zhi" displayed as a message.

WRITING YOUR OWN FUNCTIONS

- We will typically place all functions in the `<head>` section
- Functions must appear between the two `<script>` tags

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

            and

</script>
```

FORMAT OF FUNCTIONS

```
function function_name(argument list)
{
    Lots of JavaScript statements
}
```

- The argument list is required. It can be
  - Null or empty – just the parenthesis like `()`
  - A single argument, as for the alert function example
    - `alert (x)` or `alert ("My name is Zhi.")`
  - Several arguments separated by commas

ARITHMETIC MANIPULATION OF VARIABLES

- `var x;`

  `x=10;`

  - Declare a variable `x`. Assign the value of `10` to variable `x`
- `var x = 10;`
  - Declare a variable and assign it a value of `10` (both declaration and assignment in the same one step)
- `x = x + 10;`
  - Add `10` to the value that is in variable `x` and store the result in `x`
- `x = x*5;`
- `x = x/5;`
- You can do all this when `x` stores a numeric value

VARIABLES CAN ALSO STORE STRINGS

- `x = "My name is Zhi";` (note the quote)
  - Strings are always placed within quotes
- `y = "Li";`
- `x = x + y;`
  - What will the result of this addition be?
- When you have a variable that has a alphanumeric (**string**) value, when you use the "+" sign to "add" another value, the "+" will concatenate the string with the value. If you use a variable, it will concatenate the string with the value in the variable.
- If the variable has a numeric value, the "+" will perform a regular add (mathematical) operation.

AS YOU CAN SEE…

- When you use + with strings, you "concatenate" strings – i.e., splice strings together!!
- When you use + with numbers, it works the way you expect it to – add numbers together!!

CS13-INCLASS.HTM

- Use CS13-InClass.htm with the image in it.
- We will try and write a function that enlarge the image when you mouse-over the image
- Action: define a variable, get the strings of the image into the variable, use the image properties
- NEW: JavaScript arithmetic functions: `parseInt()`

USING VARIABLES TO MANIPULATE AN IMAGE

- As we said earlier, a variable is a temporary holding device to manipulate objects in JavaScript.
- When we write a function to manipulate the height or width of an image, we need to use the built-in function called `parseInt()`.
- `parseInt()`
  - A built-in function that converts a `string` (or alphanumerical) into an integer.
  - To use a math operator (`*`), you need numeric values.
- When using data from a web page, all values are captured as `string`s (or alphanumerical).

EXTENDING CS13-INCLASS.HTM

- See if you can write the function to enlarge the image (by 50%) when the user moves over on the image.
- The event is called `onmouseover`
- Can you define the element?
- How do we write the "action" part?

EXTENDING CS13-INCLASS.HTM

- Write another function that will return the image to normal size when you mouse-out of it
- The event is called `onmouseout`
- The element is the image.
- The action is the complement of what we did to enlarge it.
- You need to write a new function (call it whatever you want).

COMPLETING CS13-INCLASS.HTM

- Using the same functions to work with multiple images.
- Use of "`this.id`" parameter
  - It passes the id of the element within which the call to the function is made
  - For example:

```
<img id="one" onclick="enlargeMe(this.id)"……>
<img id="two" onclick="enlargeMe(this.id)"……>
```

  - If you click on the first image, the `enlargeMe` function is called and `this.id` will have the value of "`one`", the id of the element within which the function is called.
  - If you click on image "`two`", `this.id` will have the value of "two"!
- Save CS13-InClass.htm, update low.htm and commit/push to GitHub.

NEXT CLASS - LOOK AT CONDITIONAL STATEMENTS

- Different ways to write the IF statement
- Syntactic requirements for the IF statement
- Fun things that you can do if you know the if statement.