JavaScript (Arrays)

- Image you need to keep track of the scores of students in a class and you have 50 students
- Declaring and handling 50 variables is not an easy task
- Arrays come to the rescue
- Array – Collection of values that can be treated as a unit and where you can handle each individual entry (similar to an egg carton)
- You can visualize an array as a set of variables one after another
- There are several ways to define arrays.

```
var scienceScores = new Array();   // Creates an empty array
var mathScores = new Array(3);     // Creates an array with 3 entries
var englishScores = [77, 88, 65];  // Creates an array with 3 entries
                                   // having the specified values
```

JavaScript (Arrays)

- To access elements of an array
  - Use the [ ] operator
  - Integer Index values **starting at zero** will represent each element
- Accessing array elements

```
mathScores[0] = 70; // Assigning 70 to the first array element
mathScores[1] = 80; // Assigning 80 to the second array element
var total = mathScores[0] + mathScores[1]; // reading the first
                                           // and second elements
```

JavaScript (Arrays)

- The array length property defines how many elements the array contains
- **Example 1** (See Example1.html)
- Several functions are associated with arrays.
  - sort() – sorts elements of an array
  - reverse() – reverse elements of an array
  - join() – converts elements of an array to string and concatenates them
  - Others
- **Example 2** (See Example2.html)
  - Computes letter grades and prints students in alphabetical order

JavaScript (Objects)

- Property – Entity with a name and a value
- In JavaScript an object is a composite type which aggregates several properties
- Object properties work like variables and fundamentally are variables. You can assign values to them and read values from them
- A property value can be any data type we have seen, including objects

JavaScript (Objects)

- An example of an object is the **navigator** object which provides information about the web browser
- Among the **navigator** object properties you can find
  - appName – web browser name
  - appCodeName code name for the browser
- You can access properties by specifying the object name, a period, and the property name or by specifying the property name using array-like syntax

```
navigator.appName
navigator["appName"]
```

- Notice that you can read of modify properties using the above notation
- **Example 3** (See Example3.html)

JavaScript (Frequently Used Objects)

- **window** – Represents the window displaying the document (either the web browser window or a frame within a window). Some attributes of this object are:
  - closed – boolean value set to true if the window has been closed
  - status – text that appears in the browser's status line
- **navigator** – Object we described earlier
- **document** – Refers to the document object associated with the window. This object represents the HTML document displayed in the window
- **location** - Refers to the location object associated with the window. This object represents the URL of the document being displayed
- **Example 4** (See Example4.html)
  - This example relies on the for in construct that allow us to go through the properties for an object

JavaScript (Example)

- By making use of functions, object properties and forms we can now define an html document with JavaScript that generates customized multiplication tables
- **Example 5** (See Example5.html)
  - Notice how we can pass properties of elements of the form to the multiplicationTable function

JavaScript (Window Object methods)

- A method is a function that is invoked through an object.
- The window object has several interesting methods
  - window.open() – creates and opens a window
  - window.print() – equivalent to clicking on the browser's Print button
  - window.back() – equivalent to clicking on the browser's Back button
  - window.forward() – equivalent to clicking on the browser's Forward button

JavaScript (Other methods)

- The location object has several interesting methods
  - location.reload() – reloads the document being displayed
  - location.replace(url) – replaces the displayed document with the one associated with the url

JavaScript (Examples on the Web

- Let's take a look at some JavaScript available on the web
- **Set 1**

  http://www.bodo.com/javacool.htm
- **Set 2**

  http://www.crowes.f9.co.uk/Javascript/index.htm