2. JavaScript Objects and Methods

Objects and Properties

- Aggregation of real world data types
- An object has properties and methods
    - Constructors for creating objects at run time
    - Accessing object properties
    - Accessing object methods
- Have states, behavior, and identity
- Instances of classes
- Examples of real-world objects:
    - Lamp: wattage, onofftstate, turnOn(), turnOff(), isOn(), isOff()
    - Person: lastName, firstName, height, weight, hairColor, taxID,
    - Book: title, author, publisher, isbnNo, pages, turnToPage(), turnToNextPage()
    - WebPage: title, bgColor, links, open(), close(), write()

JavaScript Predefined Objects

- window
- document (contained in a window)
- form[ ] (contained in a document)
- frames (children of a window)
- images (placed in a document)
- Array
- String
- Math (public utility objects)
- Functions (user defined functions)

JavaScript Client-Side Object Hierarchy

The Current Window (main object)

- Various window objects: *self*, ***window***, *parent*, *top*, *frame*
- Navigator object: **navigator**
  - Array of plug-in objects (version 1.1): plugins[ ]
  - Array of MIME type objects(version 1.1): mimeTypes[ ]
- Arrays of Window objects: frame[ ]
- Location object: window.**location**
- History object: window.**history**
- Document object: **document**
  - Array of form objects: forms[ ]
    - Array of HTML form element objects: elements[ ]
      - Button
      - Checkbox
      - FileUpload (version 1.1)
      - Hidden
      - Password
      - Radio
      - Reset
      - Select
        - Array of option objects: options[ ]
      - Submit
      - Text
      - Textarea
  - Array of anchor objects; anchors[ ]

    Examples:
    - document.anchors[n]
    - document.anchors.length
  - Array of link objects: links[ ]
  - Array of image objects (version 1.1): images[ ]

- Array of applet objects (version 1.1): applets[ ]

  Examples:

  - document.applets[n]
  - document.appletName
- Array of embedded objects (version 1.1): embeds[ ]

Creating Objects

```
var obj = new Object();
var today = new Date();  // using constructor
var point = new Object();
point.x = 10;
point.y = -10;
var rectangle = {
                upperLeftPoint: {x:1, y:1},
                lowerRightPoint: {x:4, y:4}
                };
```

Global objects

- Object
- Navigator object
- Number object
- String object
- Screen object
- Event object
- RegExp object – pattern matching in strings

Object

- A primitive JavaScript object type
- The constructor -- Object()
- Methods:

eval()
toString()
valueOf()
handleEvent()
watch()
unwatch()

Navigator Object

- Getting the Name of the Browser Language
- Getting the Name of the Computer Platform
- Methods
  preference()
- Properties
  navigator.appCodeName
  navigator.appName
  navigator.appVersion
  navigator.language
  navigator.platform
  navigator.userAgent

Example 2-1: Browser detection: using the navigator.appName attribute to check if the browser is Netscape navigator. This example is saved as checkbrowser.html.

```
<html>
      <!-- checkbrowser.html -->
      <head>
            <title>Browser type</title>
      </head>
      <body>
      <h3>
      <script language="javascript" type="text/javascript">
            if (navigator.appName == "Netscape") {
            document.write("This is Netscape Navigator.")
            }
            else {
            document.write("Not running Netscape Navigator")
            }
      </script>
      </h3>
      </body>
</html>
```

Number Object

- Properties
  - MAX_VALUE
  - MIN_VALUE
  - NaN (not a number)
  - NEGATIVE_INFINITY
  - POSITIVE_INFINITY
  - prototype
- Methods
  - toString()
  - toString(radix)

document object

- Browser's document object (screen for displaying messages, control, graphics, etc)
- How to reference to the document object:
  - window.document
  - document
- It represents the HTML document currently displayed in the browser
- Has methods or functions
  - document.close() // Close an output stream
  - document.open(mimetype) // Begin a new document
  - write(value, ...) // Append data to a document screen
  - writeln(value, ...) // Append data and a new line to a document
- Has attributes or properties
  - document.bgColor // The document background color
  - document.alinkColor // The color of active links
  - document.vlinkColor // The color of visited links
  - document.linkColor // The color of unfollowed links
  - document.lastModified // The modification date of a document

- document.URL // The URL of the current document
- document.referer // The URL of the linked-from document
- document.title // The title of the current document
- document.cookie
- document.fgColor // the default text color
- document.form[ ]
- document.images[ ]

- Pass argument to a method within a pair of parentheses

document.write()

- write() method for writing text in the document screen
- No new line character is issued
- Examples:
  - document.write("<h1>ECET</h1>")
  - document.write("string for display")

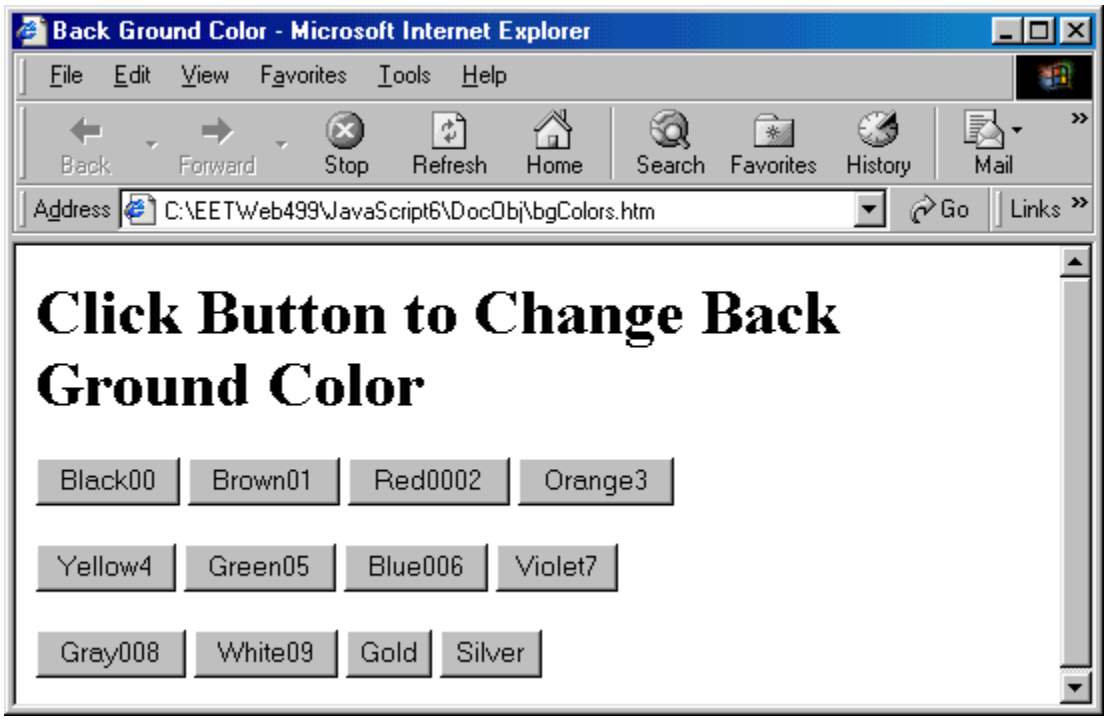
document.writeln()

- writeln() method for writing text in the document screen
- A new line character is added
- An example
  - document.writeln(" <font color=’red’ <h1>Department Welcomes You!</h1>")

Example 2-2: An example shows how to change background color properties through document object and its bgColor attributes. This example is saved as bgcolors.html.

```
<html>
<!-- bgcolors.html  -->
       <head>
              <title>Back Ground Color</title></head>
       <body>
              <h1>Click Button to Change Back Ground Color</h1>
       <form>
        <p>
       <input type="button" name="blcak" value="Black00"
OnClick='document.bgColor="black"'>
       <input type="button" name="brown" value="Brown01"
OnClick='document.bgColor="brown"'>
```

```
        <input type="button" name="red" value="Red0002" OnClick='document.bgColor="red"'>
        <input type="button" name="orange" value="Orange3"
OnClick='document.bgColor="orange"'>
        </P>
        <p>
        <input type="button" name="yellow" value="Yellow4"
OnClick='document.bgColor="yellow"'>
        <input type="button" name="green" value="Green05"
OnClick='document.bgColor="green"'>
        <input type="button" name="blue" value="Blue006"
OnClick='document.bgColor="blue"'>
        <input type="button" name="violet" value="Violet7"
OnClick='document.bgColor="violet"'>
        </p>
        <p>
        <input type="button" name="gray" value="Gray008"
OnClick='document.bgColor="gray"'>
        <input type="button" name="white" value="White09"
OnClick='document.bgColor="white"'>
        <input type="button" name="gold" value="Gold" OnClick='document.bgColor="gold"'>
        <input type="button" name="silver" value="Silver"
OnClick='document.bgColor="silver"'>
        </p>
        </form>
        </body>
</html>
</HTML>
```



window

- For creating another new window outside the current browser with appropriate configuration of window geometry (width, height), keyboard focus, scrolling, etc.
- Referencing to a window
  - self
  - window
  - window.frame[n]
- window object properties
  - self // the window itself
  - window // the window itself
  - document // reference to document object of the window
  - closed // true if window is closed
  - status // specify a line of message on the status bar
  - name // name of the window
  - location // URL of the window
  - history // the history of the window
  - length // the number of frames in the window
  - screen // information about the screen
  - frames[] // List of frames within a window
  - opener // Original window which opens a new window,
- window methods
  - alert(message) // Display message through a pop-up dialog box
  - prompt(messageString, defaultValue) // Receive input through a pop-up dialog box
  - confirm(question) // Ask for confirmation on a yes-no question
  - focus() // Receives keyboard focus
  - blur() // Relinquishes keyboard focus
  - open(url, name, features, replace) // open a new window
  - close() // close a browser window
  - scroll(xPt, yPt)
  - moveTo // Move the window
  - moveBy()
  - setInterval(FunctionName, interval_in_mSec) // Repeatedly execute the code at a specific time interval
  - clearInterval()
  - setTimeout(FunctionName, holdOffTime_in_mSec) // Defer execution of a function for a period of time before another invocation
  - clearTimeout()
- window event handlers
  - onLoad // invoked when browser completed the loading
  - onBlur // invoked when the window loses keyboard focus
  - onDragdrop // invoked when the user drops items in the window
  - onError // invoked when error occurs

- onFocus // invoked when window is given the focus
- onMove // invoked when window is moved
- onResize // invoked when window is resized
- onUnload // invoked when the browser leaves a page

window.alert()

- Call alert() method to create an alert dialog box
- The user must hit OK button to close the alert box
- window.alert("Thank you!\nTry Again!")

window.prompt()

- Call prompt() method to display a dialog control for receiving user input strings through the keyboard
- You can give the prompt message and setup a default display
- The received input will be assigned to a given variable
- parseInt() and parseFloat() may be needed if the input numbers are to be used for computation
- Examples:
  - firstNum = window.prompt("Enter the first integer number", "0");
  - secondNum = window.prompt("Enter the second integer number", "0");

window.open()

- Examples of opening a new window

newWindow1 = window.open(URL, "nameOfWindow", "toolbar=no, width=350, height=400, status=no, scrollbar=yes, resize=no, menubar=no");

newWindow2 = window.open('bulbon.gif', 'bulbWin', 'wdith=330, height=250, scrollbars=yes')

newWindow3 = window.open('book.html', 'bookWin', 'wdith=330, height=250, scrollbars=yes')

Example 2-3: Using window.prompt() method to receive user input and window.alert() method to display. This example is saved as constr.html.

```
<HTML>
<!-- constr.html -->
<HEAD>
<TITLE>Concatenating Strings</TITLE>
<script language="javascript">
<!--
  var FullName;

  FullName = prompt( "Enter a Name", "John Paul");

  alert("Hello " + FullName + ".");
-->
</SCRIPT>
</HEAD>
<BODY>
</BODY>
</HTML>
```