Lecture Notes 3
Introduction to JavaScript

Anand Tripathi

CSci 4131

Internet Programming

Significance of JavaScript

- It’s client (browser) side technology.
- JavaScript code contained in an HTML document is executed by the browser when the document is loaded and also when the user clicks on certain objects in the document.
- It can be used for presentation control such as animations, dialog windows, new pop-up windows.
- For performing validity checks before form submission.
  - With HTML5 many of the common checks are built into the new form input elements.
- HTML contents can be dynamically generated when code is executed by the browser.
  - A web document can be dynamically modified at the client side.

JavaScript

1. This language was developed by Netscape. Originally it was called LiveScript.
2. It has nothing to do with Java.
3. The capabilities of Java and JavaScript are somewhat complementary.
4. Both languages allows downloading and execution of code from a website.
5. JavaScript allows manipulation and control of browsers window and other functions. Such a creating a new display window, loading a document, or responding to mouse clicks or movements.
6. It lacks graphics and networking capabilities of Java.

JavaScript

1. Chapters 6-12 from Deitel's book
   - Chapter 6 - Introduction
   - Chapter 7 – Control statements – if ..else, while
   - Chapter 8 – Control statements
   - Chapter 9 - JavaScript functions
   - Chapter 10 – JavaScript Arrays
   - Chapter 11 – JavaScript Objects
   - Chapter 12 – Document Object Model
2. Detailed Reference Book: JavaScript The Definitive Guide – by David Flanagan
3. JQuery interface

Documentation and Reference

- Please see the mozilla.org page for documentation and reference
- https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference

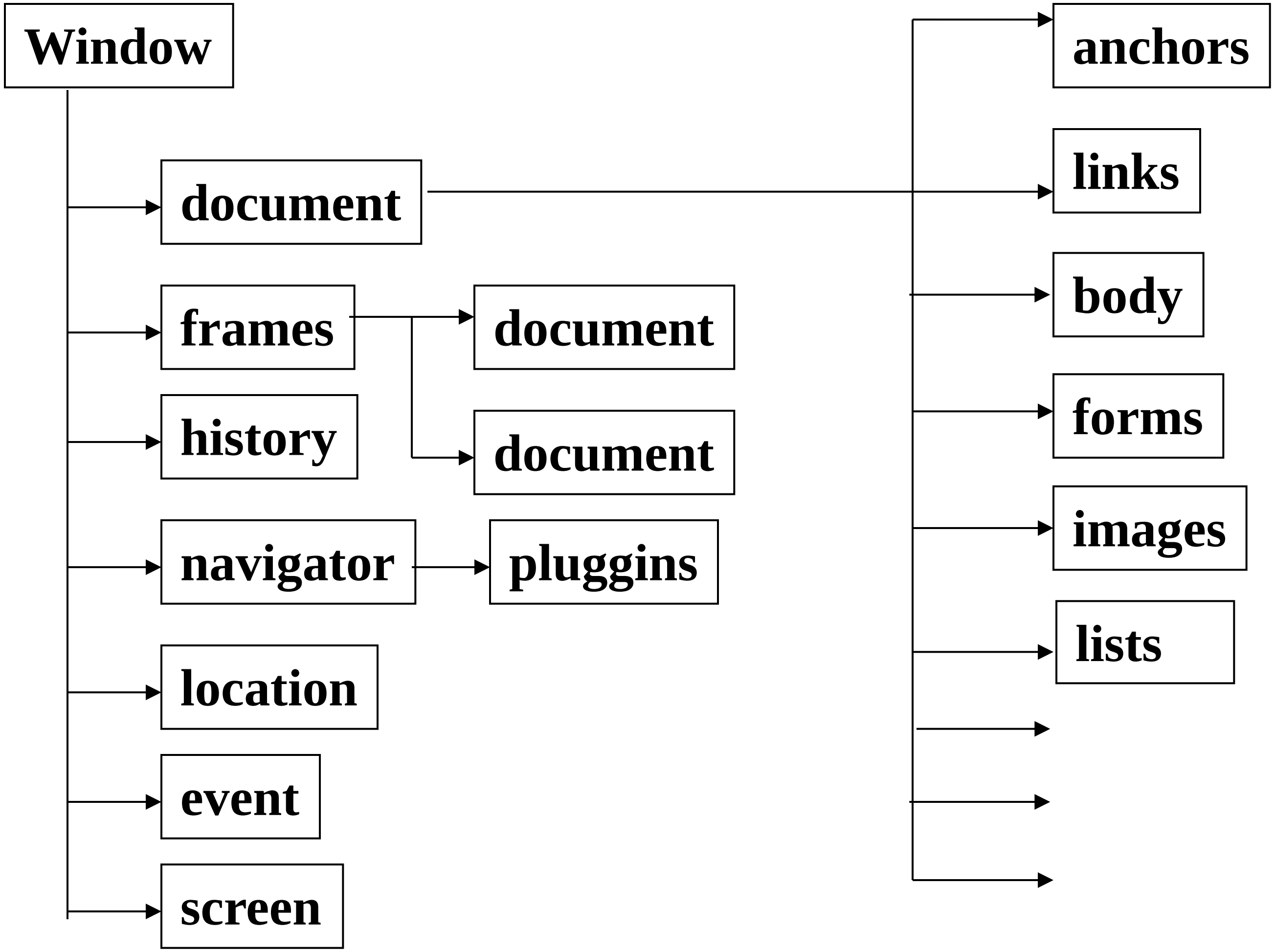
Important Aspects to Learn

- Core Language
- Document Object Model (DOM)
  - Earlier HTML based DOM and later XML based DOM
- Event handling mechanisms

JavaScript Language

- Core JavaScript
  - Defines the general and basic language concepts independent of web applications
    - This defines the basic aspects of the language such as its data model, object model, control statements.
    - Chapters 1-11 of Flanagan’s book (Part I of the book)
- Client-side JavaScript
  - Specific features that are used for programming browser and document control functions.
    - Chapters 12-22 of the book (Part 2)

Dynamic HTML: Object Model



Core and Client-Side JavaScript

Reference Book: JavaScript The Definitive Guide – by David Flanagan

Important concepts to study here:

- Data model and primitives data types (Chapter 3)
- Functions (Chapter 7)
- Object concepts and prototype based object model (Chapter 8)
- Arrays (Chapter 9)

Client-side JavaScript

- Window and Frame objects and its properties (Chapter 13)
- Document object – Forms, Image, Links (Chapter 14)
- Naming convention for accessing windows, frames, and document elements.
- Events and event handlers (Chapter 15)
- XML based Document Object Model (Chapter 17)
- Client-side security (Chapter 21)

CLIENT-SIDE JAVASCRIPT

- This allows you to control your browser's behavior.
- One can control its various objects such as
  - Window,   frames
    - Create new windows (pop-up)
  - elements of the document displayed in a frame
    - Text, Forms , Buttons, Input fields, Images
  - Chapters 13-20 of Flanagan’s Book
- Load a new document in a window/frame.
- A naming/referencing mechanism that allows a JavaScript code running in one frame to refer to other frame/window objects or the documents displayed in those frames.
- It also supports an event based model for taking actions on user's actions with the mouse.

JavaScript use in a web document

1. In a JavaScript code, we can refer to elements of a Form which is defined in the HTML body part.
2. New values can be assigned to the elements of an HTML Form.
3. Functions can defined in JavaScript. A function can be called from the HTML body.
4. New windows can be created (pop-up windows) and closed.
5. We can execute some JavaScript code when certain mouse events occur. For example: "onclick", "mouseover".
6. Timer events can be defined to reschedule the execution of some JavaScript code.

Topics

- Introduction to JavaScript with a set of simple examples
- Basic data types – boolean, integer, string, float
- Dynamically generating HTML document contents
- Alerts and Prompts
- Object model and arrays
- Document object model – accessing elements of a document.
- Controlling and managing new display windows

Example 1

Click here to load this program execution

Example from Deitel’s book 6.2 (edition 4)

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset = “utf-8” />
    <title> A First Program in JavaScript  </title>
    <script type = "text/javascript">
        document.writeln(
        "<h1>Welcome to JavaScript Programming!</h1>" );
    </script>
  </head>
  <body>
  </body>
</html>
```

Example 1

Note the following in this example:

- JavaScript code is embedded in the head of the document.
- It is included inside <script> element.
- The body of the loaded document is empty.
- The “document” object refers to the HTML document being displayed by the browser.
- The JavaScript code dynamically generates contents of this document object using “write”.
  - It writes “Welcome to JavaScript Programming!” and encloses it within <h1> tags.

Document Object

- Associated with each browser window or frame is a Document object.
- Document object represents the document being displayed in the window.
- By invoking the “write” method of the Document object we can create new dynamic content for the object being displayed in the browser window.
- Using the Document object we can create dialog boxes to interact with the user.

Example 2

Click here to load this program execution

Example 6.2 from Deitel’s book

```
<html xmlns = "http://www.w3.org/1999/xhtml">
  <head>
    <title>Printing a Line with Multiple Statements</title>
    <script type = "text/javascript">
      document.write( "<h1 style = 'color: magenta'>" );
      document.write( "Welcome to JavaScript " +
        "Programming!</h1>" );
   </script>
  </head>  <body></body>
</html>
```

A JavaScript Example

```
<html xmlns = "http://www.w3.org/1999/xhtml">
<head>
<title>   JavaScript Object Access Test     </title>
<script type=“text/JavaScript">
    document.write( " <font color=blue> Hello World  <br />");
    document.write( "Looks like this JavaScript stuff is going to be
    fun!!! <br />");
</script>
</head>
<body>
     <font color=“red”> This is the HTML body. </font>
</body>
</body>
```

See the above program's execution.

Window Object Methods

Confirm, Prompt, and Alert

These are dialog boxes dynamically created to get use input:

Alert - Used for notifying the user

Confirm - Used for getting a Yes (OK) or No (Cancel) inputs from the user

Prompt - Used for getting some text input from the user

Example 4 - use of “prompt”

Click here to load this program execution

Example 6.5 from Deitel’s book

```
<html>
   <head>
      <meta charset = "utf-8">
      <title>Using Prompt and Alert Boxes</title>
      <script>
         var name; // string entered by the user
         // read the name from the prompt box as a string
         name = window.prompt( "Please enter your name" );
         document.writeln( "<h1>Hello " + name + ", welcome to
                                   JavaScript programming!</h1>" );
      </script>
</head>
<body>  </body>
</html>
```

Confirm, Prompt, and Alert

```
<html>
<body>
<script language="JavaScript">
    if ( window.confirm( 'Do you want to proceed?') ) {
        document.write( "That is good!! <br/> " );
    } else {
    document.write( "It seems you do no want to go further, <br/> but I will
    ignore you.<br/>" ); }
     var name = window.prompt( "Please tell me your name!", "Type your
    name    here!" );
     document.write( "Your name is " + name );
     window.alert( "This shows an alert!!" );
 </script>
</body> </html>
```

See this example in action

Computing Factorials

```
<html xmlns = "http://www.w3.org/1999/xhtml">
<script type=“text/JavaScript">
 //  See page 30 of the Flanagan’s JavaScript Book
  document.write( "<h2> Table of Factorials </h2>" );
  for ( i = 1, fact = 1; i < 10 ; i++ ) {
     fact = fact * i;
     document.write( "Factorial of " + i + " is = " + fact);
     document.write( "<br />" ); }
</script>
<body>   <font color=red> This is the HTML body. </font>
</body> </html>
```

See the above program's execution.

Accessing Form Elements

```
<head>
<script type=“text/JavaScript">
var done = 0;
function Blink( val ) {
    if (done!=0) {
       if ( (done % 2) == 0 ) {
            document.myForm.info.value = val ;
            setTimeout( "Blink('CSci 4131')", 1000 );
            done++;
       }  else {
            document.myForm.info.value = val ;
            setTimeout( "Blink('Internet Programming')", 1000 );
            done++; }
        }
} </script> </head>
```

```
<body>
     <form name="myForm“  >
         <input type=text    name="info"   value="CSci 4131">
          <br />
          <input type=button value="Show Course Number"
                  onClick="alert('CSci4131')"  />
           <br />
           <input type=button value="Blink Course Title"
                  onClick="done = 1; Blink('Internet Programming' )"  />
            <br />
            <input type=button value="Stop Blink“
                   onClick="done = 0" />
           <br />
       </form>
</body>
```

See the above program's execution.

Mouse Over/Out Events

```
<html >
<body>
<h3> <font color=red>
This example shows the onMouseOver and onMouseOut events.
<br /> Just slide the mouse cursor over or out of the circle image.
</font> </h3>
<img src="lotus-1.jpg"   name="flower"
          onMouseOver ="document.flower.src = 'lotus-5.jpg'"
          onMouseOut  ="document.flower.src = 'lotus-1.jpg'"
          width="400" height="300">

</body>
</html>
```

See the above program's execution.

Fade-out/Fade-in of Image

Change the opacity property associated with the style of an image.

```
<html >
<body>
<h3> <font color=red>
This example shows the onMouseOver and onMouseOut events.
<br /> Just slide the mouse cursor over or out of the circle image.
</font> </h3>
<img src="lotus-1.jpg"   name="flower“
          onMouseOver ="document.flower.style.opacity = '0.5'"
          onMouseOut  ="document.flower.style.opacity = '1.0' "
width="400" height="300">

</body>
</html>
```

See the above program's execution.

An Example Loading a URL

See the above program's execution.

```
<head>
<script type="text/JavaScript">
    function gotoURL() {
    document.location.href = document.myForm.urllink.value ;  }
</script>
</head>
<body>
<form name="myForm">
     <input   type ="text"    size="90"   name="urllink" />
    <input type="button"     value="Go to this URL"
          onClick = "gotoURL()" />  <br />
     <input type=button  value="Go to YAHOO"
          onClick = " document.location.href = 'http://www.yahoo.com' " />
     <br />
</form>
</body>
```

Core JavaScript LITERAL DATA:

Examples:

```
12,
1.2,
"Hello World",
'hi'
true,
false,
null
```

VARIABLE NAMES consist of characters: a-z A-Z 0-9 _ and $

Statements put on separate lines need not be terminated with ;

(Do not follow this practice.)

```
a = 3
b = 5
```

is equivalent to: a = 3; b = 5;

PRIMITIVE DATA TYPES

JavaScript does not enforce any type checking.

A variable can represent any type of data.

var x = 10; x = “ten”;

Numbers

No distinction between integers and floating points.

Numbers are represented in IEEE 8-byte standard.

Octal numbers start with 0, such as 0377.

Hexadecimal numbers start with 0x or 0X followed by a sequence of hex digits.

Strings

Represent sequence of text characters:
var greeting = "Hello world!";
Strings can be represented with double-quotes or single-quotes.
Example:
"I don't like this kind of use of strings."
‘ He said - "Never mind.“ '

Boolean

toString Function

To print a number, convert it to string using "toString" number.

```
var x = 25;
var y = x.toString();
```

y will have decimal presentation of number stored in x.

```
var z = x.toString(16);
```

z will have hex representation of the number in x.

String Operations

String concatenation

message = "Hello " + "World"

Length

message.**length**

Indexing within a string - Character at position i

var c = message.**charA**t(i)
First character's index is 0.

Substring

sub = message.**substrin**g(1, 4)
Start index is 1 and last index is 4.

indexOf

i = message.**indexOf**(‘g');
Position of the first occurrence of ‘g'.

upperCase

s.**toUpperCase**() converts all lowercase characters in s to uppercase.

Setting Properties of Strings

Click here to See this example

```
var anchorText = "This is an anchor";
var fixedText = "This is monospaced text";
var linkText = "Click here to go to  CS Department Homepage”;
var strikeText = "This is strike out text";
var subText = "subscript";
var supText = "superscript";

document.writeln( anchorText.anchor( "top" ) );
document.writeln( "<br />" + fixedText.fixed() );
document.writeln( "<br />" + strikeText.strike() );
document.writeln( "<br />This is text with a " + subText.sub() );
document.writeln( "<br />This is text with a " + supText.sup() );
document.writeln( "<br />" + linkText.link( “http://www.cs.umn.edu" ) );
```

Math Object

Math object provides many useful functions such as:

- sin (x)
- cos(x)
- abs(x)
- log(x)
- floor(x)
- random() returns a random number between 0 and 1
- exp(x) for e^x

All mathematical functions are available as the properties of a system defined object called "Math".

For example:

```
sine_of_x = Math.sin(x) ;
root = Math.sqrt(x);
```

"Number" object stores various properties such as:

Number.MAX_VALUE which is the largest value
Number.MIN_VALUE which is the smallest value
Number.NaN which represents a special value "not a number"

null

represents a special "no value"

undefined

means that the variable is either not declared or no value has been assigned to it yet.

Example 5

Click here to load this program execution

Example 6.6 from Deitel’s book

```
<script type = "text/javascript">
    var   firstNumber; // first string entered by user
    var   secondNumber; // second string entered by user
    var number1; // first number to add
    var number2; // second number to add
    var sum; // sum of number1 and number2
    // read in first number from user as a string
    firstNumber = window.prompt( "Enter first integer" );
    // read in second number from user as a string
    secondNumber = window.prompt( "Enter second integer" );
    // convert numbers from strings to integers
    number1 = parseInt( firstNumber );
    number2 = parseInt( secondNumber );
    sum = number1 + number2; // add the numbers and display the results
    document.writeln( "<h1>The sum is " + sum + "</h1>" ;
</script>
```

Date/Time Object

Click here to See this example

```
 var current = new Date();
document.writeln( "<h1>String representations and valueOf</h1>" );
document.writeln( "toString: " + current.toString() +  "<br />toLocaleString: "
   + current.toLocaleString() + "<br />toUTCString: " + current.toUTCString()
   +  "<br />valueOf: " + current.valueOf() );

document.writeln ( "<h1>Get methods for local time zone</h1>" );
document.writeln( "getDate: " + current.getDate() +
          "<br />getDay: " + current.getDay() +
          "<br />getMonth: " + current.getMonth() +
          "<br />getFullYear: " + current.getFullYear() +
          "<br />getTime: " + current.getTime() +
          "<br />getHours: " + current.getHours() +
          "<br />getMinutes: " + current.getMinutes() +
          "<br />getSeconds: " + current.getSeconds() +
          "<br />getMilliseconds: " + current.getMilliseconds() +
          "<br />getTimezoneOffset: " + current.getTimezoneOffset() );
```

Date/Time Object

Click here to See this example

```
document.writeln( "<h1>Specifying arguments for a new Date</h1>" );
var anotherDate = new Date( 2007, 2, 18, 1, 5, 0, 0 );
document.writeln( "Date: " + anotherDate );

document.writeln( "<h1>Set methods for local time zone</h1>" );
 anotherDate.setDate( 31 );
 anotherDate.setMonth( 11 );
 anotherDate.setFullYear( 2007 );
 anotherDate.setHours( 23 );
 anotherDate.setMinutes( 59 );
 anotherDate.setSeconds( 59 );
 document.writeln( "Modified date: " + anotherDate );
```

Example 6

Click here to load this program execution

```
<html >
   <head>
      <title>Using Relational Operators</title>
      <script   type = "text/javascript">
         var name; // string entered by the user
         var now = new Date(); // current date and time
         var hour = now.getHours(); // current hour (0-23)

         // read the name from the prompt box as a string
         name = window.prompt( "Please enter your name" );

         // determine whether it is morning
         if ( hour < 12 )
            document.write( "<h1>Good Morning, " );
```

```
// determine whether the time is PM
      if ( hour >= 12 )
      {
        // convert to a 12-hour clock
        hour = hour - 12;

        // determine whether it is before 6 PM
        if ( hour < 6 )
          document.write( "<h1>Good Afternoon, " );

        // determine whether it is after 6 PM
        if ( hour >= 6 )
          document.write( "<h1>Good Evening, " );
      } // end if

      document.writeln( name +
        ", welcome to JavaScript programming!</h1>" );
 </script>     </head>
  <body>
    <p>Click Refresh (or Reload) to run this script again.</p>
  </body>     </html>
```

Example of Control Statements

Click here to load this program for execution

```
<html >
   <head>
      <title>Solution 8.3</title>
      <script type = "text/javascript">
       // part a
         var x = 1;
         while ( x <= 10 )
         {           document.writeln( x );
                       x ++;
         }
         document.writeln( "<br /> <br />" );
         // part b
         var y;
         for ( y = 1; y != 10; y++ )
            document.writeln( ( y / 10 ) + " " );
            document.writeln( "<br /> <br />" );
```

```
// part c
      var n = 1;
      switch ( n )
      {
      case 1:
        document.writeln( "The number is 1" );
        break;
      case 2:
        document.writeln( "The number is 2" );
        break;
      default:
        document.writeln( "The number is not 1 or 2" );
        break;
      }
      document.writeln( "<br /> <br />" );

      // part d
      while ( n <= 10 )
        document.writeln( n++ );
   </script>     </head><body></body>
</html>
```

Example 7: Dynamic Generation of HTML

Click here to load this program for execution

```
head>
    <title>Counter-Controlled Repetition</title>
    <script type = "text/javascript">
     var counter = 1; // initialization

      while ( counter <= 7) // repetition condition
      {
        document.writeln( "<p style = \"font-size: " +
          counter + "ex\"> XHTML font size " + counter +
          "ex</p>" );
        ++counter; // increment
      } //end while
   </script>
  </head>
<body></body>
```