Chapter 4: JavaScript

Note

- All example files used in this lecture are available at

https://swe.umbc.edu/~zzaidi1/is448/chapter4/

Javascript

- a lightweight programming language (scripting)
- used to make web pages interactive
  - insert dynamic text into HTML (eg: user name)
  - react to events (eg: page load user click)
  - get information about a user's computer (eg: browser type)
  - perform calculations on user's computer (eg: form validation)
- a web standard (but not supported identically by some browsers)
- not related to Java other than by name and some syntactic similarities

JavaScript in XHTML

- Directly embedded (avoid this style)

```
<script type="text/javascript">
            <!--
              ...Javascript here...
            -->
</script>
```

- Indirect reference (good style)

```
<script type="text/javascript" src="tst_number.js"></script>
```

  - Preferred approach
  - <script> tag placed in XHTML page's head
  - Script code is stored in a separate .js file

Javascript in XHTML

- JavaScript code can be added to a web page in several ways

- In the HTML page's *body* tag
  - Directly within <script> tags
  - Or, link to an external .js script file

- In the HTML page's *head* tag
  - Directly within <script> tags
  - Or, link to an external .js script file

JavaScript in HTML body

- Example: See hello_in_body.html

```
<html><head></head>
<body>
    ...
            <script type="text/javascript">

              JavaScript code

            </script>
    ...
</body></html>
```

JavaScript in HTML head

- Javascript is loaded before rest of page loads
- Example: See hello_in_head.html

```
<html><head>
    ...
    <script type="text/javascript">

       JavaScript code

    </script>
    ...
</head>
<body>………</body></html>
```

Linking to a JavaScript File

- Can be placed in page's head or body
- Script is stored in a .js file
- Example: See hello_externalfile.html, hello_external.js

Syntax

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

Example

```
<script src="example.js" type="text/javascript”></script>
```

Overview of Javascript syntax

- Statements can be terminated with a semicolon
- document.write prints specified text to browser window

```
document.write("message");
```

- Variables explicitly declared using var keyword or implicitly declared by assignment
- Variable names are case sensitive
- Comments: // and /* .. */

```
//this is a small program
var pi=3.14;
var username = "Connie";
```

Statement Syntax

- Statements can be terminated with a semicolon
  - However, the interpreter will insert the semicolon if missing at the end of a line and the statement seems to be complete

Screen output: Dynamic text

```
document.write(“message”);
```

- Prints specified text to browser window
- Can be used to display HTML
- Argument can be a literal string in quotes or a variable
- Example: see hello.html

```
document.write(“The result is: “, result, “<br />”);
```

Variables

- Variables are explicitly declared using var keyword
  - Variable names are case sensitive
- Implicitly declared through assignment (give it a value and it exists!)
- Data type is not specified in variable declaration, but Javascript does have types

Syntax

```
var name = value;
```

Examples

```
var pi="3.14";
stop_flag = true;
var username = "Connie";
```

General Syntactic Characteristics

- Identifiers, i.e., variable names
  - Can start with $, _, letter
  - Can continue with $, _, letter or digit
  - Case sensitive
    - FRIZZY, fRIZZY, frizZY- distinct names
- Reserved words
- Comments
  - //
  - /* … */

4.6 Loop Statements

- Loop statements in JavaScript are similar to those in C/C++/Java
- While

```
while  (control expression) {
    statements;
}
```

- For

```
for (initial expression; control expression; increment expression){
    statements;
}
```

- do/while

```
do  {
Statements
} while (control expression);
```

- Example: See date.js and date.html

Control structure: for

- for loop
- Syntax:

```
for (initialization; condition; update) {
    statements;
}
```

- Example: see squared.html

```
for (var i = 0; i < 10; i++) {
    document.write("<p>" + i + " squared = " + (i * i) + "</p>");
}
```

Lab

- Modify hello_in_body.html to print “Hello Web Programmer” 100 times to the screen
- Make sure each occurrence of “Hello Web Programmer” is printed on a new line

4.6 if-else Statements

- The if and if-else are similar to that in other programming languages, especially C/C++/Java
- See ifstmt_example.html, ifstmt_example.js

```
var a = 2; var b = 3;
if  (a > b)
{
   document.write(“a is greater than b <br />”);
}
else
{
   document.write(“b is greater than a”);
}
```

switch statement

- Control expression in switch can be evaluated to a number, string or a Boolean value
- Case labels can be numbers, strings or Booleans
- Example: See switch_stmt_example.html, switch_stmt_example.js

4.6 switch Statement example

```
var bordersize = 3;
switch (bordersize) {
case “1”:
        document.write(“<table border = ‘1’ >”);
        break;
case “2”:
        document.write(“<table border = ‘2’ >”);
        break;

case “3”:
        document.write(“<table border = ‘3’ >”);
        break;
default:
        document.write(“invalid border size”);
        break;
}
```

Data types

- Common types: Number, Boolean, String, Null, Undefined
- Dynamic, weakly typed language
- Values are converted between types automatically as needed
- Can find out variable's type by calling **typeOf**

Number type

- Integers and real numbers are the same type
  - No *int* or *double* type in Javascript

```
var classStrength = 25;
var medianScore = 18;
```

- Converting a String into a Number

```
var num1 = parseInt(“12.33hello”);
var num2 = parseFloat(“12.33hello”);
var num3 = parseInt(“blah”);
```

  - parseInt(“12.33hello”) returns 12
  - parseFloat(“12.33hello”) returns 12.33
  - parseInt(“blah”) returns NaN (not a number)

Numeric Operators

- Standard arithmetic
  - `+   *   -   /   % --    ++`
- Comparison operators
  - `> < <= >= && || ! == != === !==`
    - `==` just checks value `(“5.0” == 5 is true)`
    - `===` also checks type `(“5” === 5 is false)`
- Similar rules of precedence and associativity as in Java
- Many operators auto-convert: "2" * 3 is 6,   5 < "7" is true

Strings and String Catenation

- A string literal is enclosed in double quotes or single quotes
- The operator + is used for string catenation
- In many cases, other types are automatically converted to string

```
var firstName = “George”;
var lastName = “Clooney”;
var fullName = firstName + lastName;
```

String Property

- One property: length
  - Note to Java programmers, this is not a method!
- Character positions in strings begin at index 0
- Example: see strings.js, string_operations.html

```
var str = “George”;
var len = str.length;
```

len is set to number of characters in str

String Methods

Method	**Parameters**	**Result**
charAt	A number	Returns the character in the String object that is at the specified position
indexOf	One-character string	Returns the position in the String object of the parameter
substring	Two numbers	Returns the substring of the String object from the first parameter position to the second
toLowerCase	None	Converts any uppercase letters in the string to lowercase
toUpperCase	None	Converts any lowercase letters in the string to uppercase

String Methods Usage

- What is the output of these statements?

```
var str=“George”;
var charLocation = str.charAt(2);
var charIndex = str.indexOf(‘r’);
var small = str.substring(2, 4);
var newString = str.toLowerCase()
```

The Date Object

- A Date object represents a *time stamp*, that is, a point in time
- A Date object is created with the new operator
  - var now= new Date();
  - This creates a Date object for the time at which it was created
- Example: see date.html, date.js

The Date Object: Methods

Method	**Returns**
toLocaleString	A string of the Date information
getDate	The day of the month
getMonth	The month of the year, as a number in the range of 0 to 11
getDay	The day of the week, as a number in the range of 0 to 6
getFullYear	The year
getTime	The number of milliseconds since January 1, 1970
getHours	The number of the hour, as a number in the range of 0 to 23
getMinutes	The number of the minute, as a number in the range of 0 to 59
getSeconds	The number of the second, as a number in the range of 0 to 59
getMilliseconds	The number of the millisecond, as a number in the range of 0 to 999

Window and Document objects

- The Window object represents the window in which the document containing the script is being displayed
- The Document object represents the document being displayed using DOM

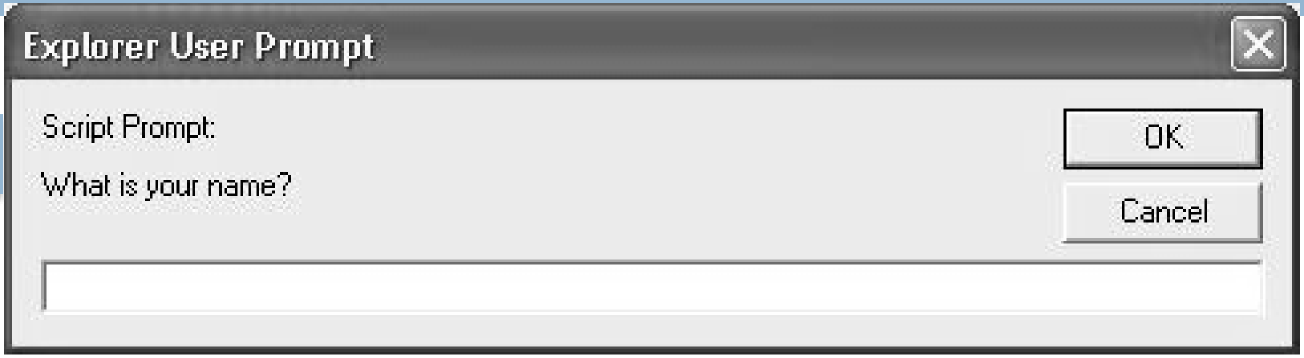
Window and Document objects

- Window has two properties
  - window property refers to the Window object itself
  - document property refers to the Document object

- The Window object is the default object for JavaScript, so properties and methods of the Window object may be used without qualifying with the class name
  - i.e., you need not say window.document.write, you can simply say document.write when you want to write to the current document

Popup boxes

prompt

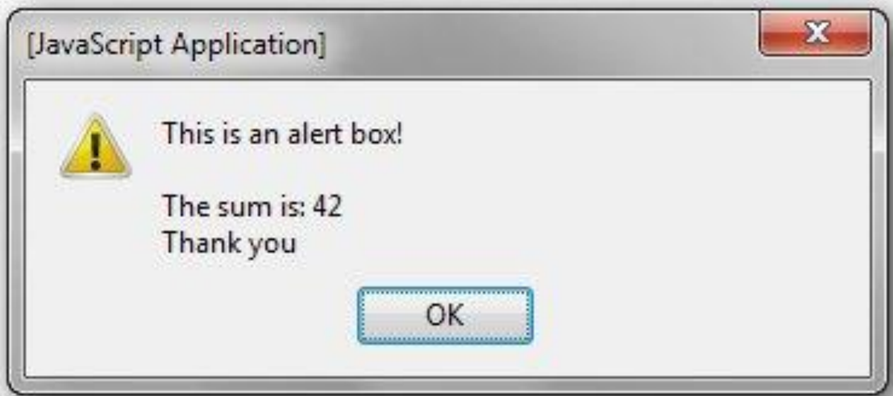

Explorer User Prompt

Script Prompt:

What is your name?

```
alert("message"); // message
var answer = confirm("message"); // returns true or false
var answer = prompt("message"); // returns user input string
```

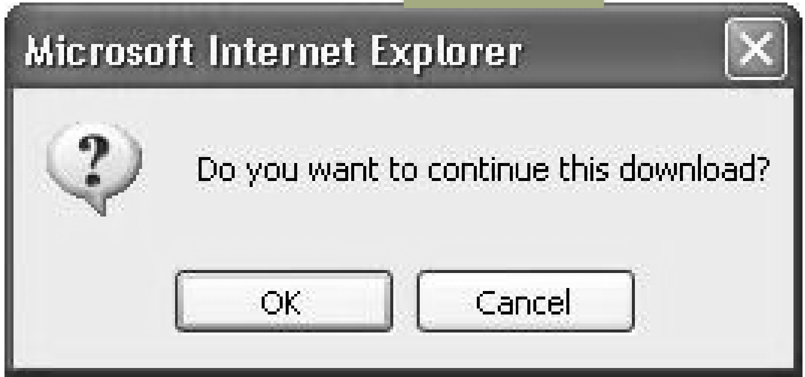
alert


[JavaScript Application]

This is an alert box!

The sum is: 42
Thank you

confirm

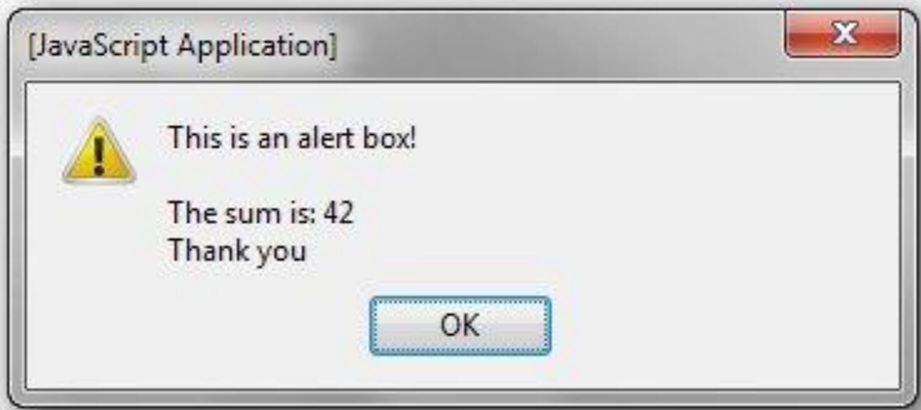

Microsoft Internet Explorer

Do you want to continue this download?

The alert Method

- alert is a method of the Window object. It opens a dialog box with a message, and the OK button
- The output of the alert is *not* XHTML, so use new lines (\n) rather than <br/> to insert new lines in your message

```
var sum = 42;
alert("The sum is:" + sum + "\n" + "Thank you");
```



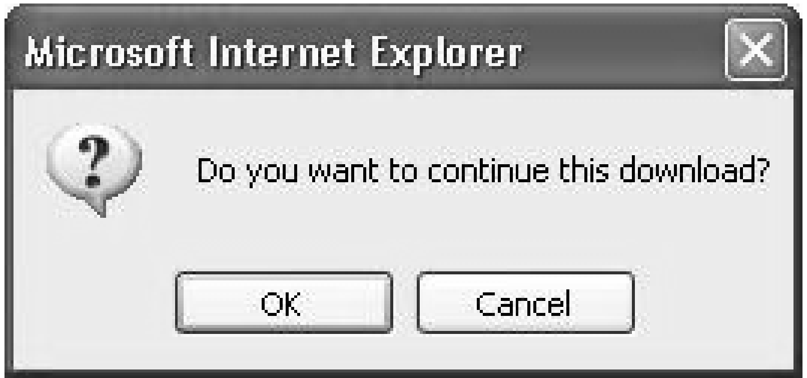
This is an alert box!

The sum is: 42
Thank you

The confirm Method

- confirm is a method of the Window object
- The confirm methods displays a message provided as a parameter
  - The confirm dialog has two buttons: OK and Cancel
- If the user presses OK, confirm returns a boolean value of true
- If the user presses Cancel, false is returned

```
var question =
   confirm("Do you want to
continue this download?");
```

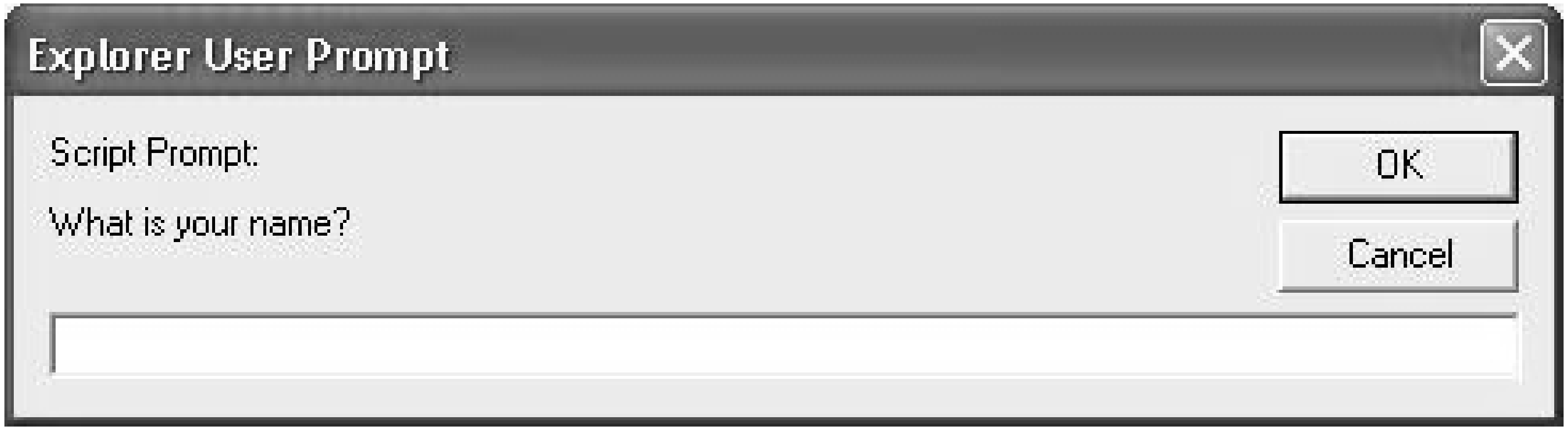

Microsoft Internet Explorer

Do you want to continue this download?

The prompt Method

- prompt is a method of the Window object
- This method displays its string argument in a dialog box
  - A second argument provides a default content for the user entry area. In this example, default content is the empty string
- The dialog box has an area for the user to enter text
- The method returns a String with the text entered by the user

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



Script Prompt:

What is your name?

Lab

- Download the html file, sum.html
- To this file, add the Javascript code to
  - prompt the user to enter two numbers
  - read in the two numbers from the user
  - compute their product and print the product to the screen
  - add them and print the sum to the screen
  - also print current date to the screen