CS120

Introduction to Web Site Development

Lecture 7 - JavaScript

Gabor Kondas
California State University, Los Angeles
(Based on slides by John Hurley)

JavaScript

- Programming language that executes while you are browsing
- Sent to the browser as code and interpreted by browser
- JavaScript can modify the contents of a web page when the page is loaded or in response to a user action
- JavaScript is the dominant **client-side** scripting language of the world wide web

JavaScript

JavaScript is a misnomer

- Syntax similarities to Java are due to common legacy of Algol/C family languages and superficial adoptions from Java
- Java was the hot new thing when JS came out, and Sun was one of the early endorsers of JS. The name was pasted on although the internal working of JS are radically different from those of Java

JavaScript examples

Button

Image

CSS

Input Validation

Internal JavaScript

JavaScript can be written inside an HTML file using the script element

Start and end tags are required

Example:

```
<head>
<script type="text/javascript">
  // your JavaScript code goes here
</script>
</head>
```

Internal JavaScript

- The SCRIPT element can go inside the HEAD
  - Script is run before BODY is loaded
- The SCRIPT element can go inside the BODY
  - Script is run as BODY is being loaded

External Scripts

- Include a script that is saved in a separate file, much like using an external CSS
- Place a link in the head section:

```
<script type = “text/JavaScript” src = “scripts/myscript.js” />
```

Linking JS Examples

External

Internal

Statements

- A computer program executes statements
- A statement is the smallest unit of execution in a computer program
- A statement should always be terminated by a semicolon in JavaScript
- There are many types of statements
  - assignment
  - return
  - if / else
  - etc.

The Window Object

- Every browser has a window object that you can access through your JavaScript
- Window Functions:
  - window.alert(message);
  - window.open(URL);
  - window.showModalDialog(URL);
  - window.prompt(message, defaultValue);
  - window.confirm(message);

The Window Object

Notice that:

- The object name, window, comes first
- Next is the period, which is called the member access operator
- Next is the name of the function
- Next is the left parenthesis
- Next is zero or more comma-separated arguments
- Next is the right parenthesis
- And finally, there is the semicolon

Alerts

window.alert displays a simple dialog box

- `window.alert(message);`
- `window.alert(x);`

Message is a JavaScript string that you would like to have shown in an alert box

Prompts

- Window.prompt displays a simple dialog box with a text input box, that allows the user to enter text
- This function is used for user input.

```
var value = window.prompt(message, default-value);
```

  - message is a JavaScript string that you would like to have shown in an prompt box
  - default-value is a JavaScript string that you would like to have shown in the input box
  - value is the JavaScript string that was in the input box when the user closed the prompt dialog box by pressing OK; if the user pressed CANCEL, the keyword null is returned

Confirmation

- window.confirm displays a simple dialog box with a message and two button, OK and CANCEL
- This function is used for user input.
  - `var value = window.confirm(message);`
  - message is a JavaScript string that you would like to have shown in the confirm box
  - value is the JavaScript boolean value that is true if the user pressed the OK button, and false if the user pressed the CANCEL button

JavaScript Data Types

There are three basic data types that we need to understand form the outset: numbers, strings, and Booleans

- Numbers
  - 1, 2, 3, 1.0, 2.0, 3.0, -1.0, -2.0
  - Interpreter will distinguish correctly between integers and floating point numbers
- Strings
  - "This is a string of characters."
- Boolean
  - true, false

More Info

JavaScript Variables

- In an HTML page, sometimes we need to maintain the state of something
- To do this we can using a named variable
- The name can be any alpha-numeric character that does not start with a number
- To declare a variable, use the **var** keyword

More Info

Syntax:

```
var variable-name = initial-value;
```

Assignment Statements

- Assignments give a value to a variable
- Each one of the four statements below is an assignment statement

```
var myCarValue = 1000;
myCarValue = 2000;
var total;
total = 10 + 14 – 3;
x = 10;
```

Variables Example

Example:

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

 var x      = 3;
 var y      = 1.0;
 var str    = "This is a string.";
 var b1     = true;
 var b2     = false;

</script>
```

Expressions

- An expression is sequence of numbers, variable names, values, function calls, and operators that computes a single value
  - 3 + 4, is an expression that computes 7
  - ((7 – 2)*4), is an expression that computes 20
- There are many different types of expressions
  - Mathematical
  - Boolean
  - Logical

Mathematical Expressions

- Addition: x + y
- Subtraction: x – y
- Multiplication: x * y
- Division: x / y
- Remainder (Modulus): x % y

Modulus

% finds the remainder in an integer division problem:

1 % 1 is 0

5 % 1 is 0

5 % 5 is 0

5 % 3 is 2

5 % 2 is 1

6 % 2 is 0

9 % 2 is 1

Boolean Expressions

A Boolean expression is an expression that evaluates to either true or false

- Less Than: x < y
- Greater Than: x > y
- Less Than or Equal: x <= y
- Greater Than or Equal: x >= y
- Equal: x == y
- Not Equal: x != y

if / else statements

- We use the if / else statement in conjunction with Boolean and logical expressions to execute branches of statements
- The syntax is:
  - ```
    if(boolean-or-logical-expression)
        single-statement;
    else
        single-statement;
    ```
- Example:
  - ```
    if(x < 10) y = 3;
    else
    ```

if / else statements

If multiple statements are needed, use curly brackets to label a block (a set of statements).

The syntax is:

```
if(boolean-or-logical-expression){
   statement1;
   statement2;
   statement3;
}
else{
   statement1;
   statement2;
   statement3;
}
```

Document Functions

`document.write(str)`

- A function that allows you append HTML code to your document while it is being loaded
- If the output contains html tags, the browser will render the output using them.
- CSS styles apply to markup generated this way

Quotes Nested In Output

- You sometimes need to nest quotes inside string quotes in document.write or other JavaScript statements
- **Choice a**: more intuitive but less rigorous
  - use single quotes for literal quotes (that is, ones you want to render as quotes)
  - document.write(" ");
- **Choice b**:
  - escape with backslash
  - document.write("<div class = \"main\">");
- document.write("She said \"They tried to make me go to home, but I said \"no, no,no\"\"");

Linefeed with document.write

- To skip to a new line with document.write, just print the line break element:
  - document.write(“
”);
  - \n will NOT work!

Logical Expressions

- A logical expression computes a value based on a truth table
- Logical OR: **x || y**
  - If x is true or y is true, then the result is true; otherwise false
- Logical AND: **x && y**
  - If both x and y are true, then the result is true; otherwise false
- Logical NOT: **!x**
  - If x is true, result is false; otherwise true

Loops

- Loops are used in programming to repeat actions
- Loop breaks when some condition is met, for example when x == 100 or cancelled == true
- JavaScript loop example:

More Info For | More Info While

```
var counter = 0;
while (counter < 100) {
document.write(counter + “<br />”);
counter++;    // increment counter by 1
}
document.write(“done”);
```

Loops

Nested Loop

```
var x = 0;
  var y = 0;
  while(x < 10){
      y = 0;
      while (y <10){
          document.write("x = " + x + "; y = " + y +"<br />");
          y++;
      }
      document.write(“<br />");
      x++;
  }
```

For Loop

```
for(var counter = 0; counter < 5; counter++) {
      window.alert(counter + "!");
}
```

- The first line declares and initializes a variable, defines the test, and sets the variable to increase by 1 at the end of each execution of the loop.
- The initialization only occurs in the first iteration of the loop.

For Loop

```
<script>
    for (var counter = 0; counter < 100; counter++){
        document.write("Always watch out for Godzilla: " +
counter + "<br />");
}
</script>
```

Do While loop

```
var choice;
        do{
            choice = window.prompt("Who would win? Enter G for
Godzilla or B for Bruce Lee");
            choice = choice.toUpperCase();
        } while (choice != 'G' && choice != 'B');

        var winner;
        if(choice == 'G') winner = "Godzilla";
        else if(choice == "B") winner = "Bruce Lee";
        document.write("user thinks " + winner + " would win");
```