IS 242 Web Application Development I

Lecture 8: Introduction to JavaScript

Outlines of today's lecture

- Continue our last lecture
- Introduction to JavaScript
- How to to use it
- JavaScript variables and operators

You already know we use HTML, or Hypertext Markup Language, to specify all the **content** of your pages along with their **structure**, like paragraphs, headings and sections.

And you already know that we use CSS, or Cascading Style Sheets, to specify how the HTML is presented...the colors, fonts, borders, margins, and the layout of your page. CSS gives you **style**, and it does it in a way that is separate from the structure of the page.

So let's introduce JavaScript, HTML & CSS's computational cousin. JavaScript lets you create **behavior** in your web pages. Need to react when a user clicks on your "On Sale for the next 30 seconds!" button? Double check your user's form input on the fly? Grab some tweets from Twitter and display them? Or how about play a game? Look to JavaScript. JavaScript gives you a way to add programming to your page so that you can compute, react, draw, communicate, alert, alter, update, change, and we could go on... anything dynamic, that's JavaScript in action.

From *Head first HTML and CSS*, by Robson, E., & Freeman, E., 2012, O'Reilly Media, Inc

Introduction

- With **HTML & CSS** what you're doing is largely declarative—for instance, you're declaring, say, that some text is a paragraph or that all elements in the "sale" class should be colored red.

- With JavaScript you're adding behavior/interactivity to the page, and to do that you need to describe computation. You need to be able to describe things like, "compute the user's score by summing up all the correct answers" or "do this action ten times" or even "go off and get my latest tweet, and put it in this page."

- JavaScript is usually embedded directly into HTML pages

- JavaScript is an interpreted language (means that scripts execute without compilation)

What is JavaScript?

- JavaScript is one of the programming languages of the Web.

- The majority of modern websites use JavaScript.

- All modern web browsers include JavaScript **interpreters**.

- JavaScript is part of the **triad** of technologies that all Web developers must learn: **HTML to specify the content of web pages, CSS to specify the presentation of web pages, and JavaScript to specify the behavior of web pages.**

Are Java and JavaScript the Same?

NO!

- Java and JavaScript are two completely different languages in both concept and design!

- Java (developed by Sun Microsystems) is a powerful and much more complex programming language - in the same category as C and C++.

What it can do?!

- **JavaScript Can Change HTML Content**

One of many HTML methods is document.getElementById().

```
<!DOCTYPE html>
<html>
<body>

<h1>What Can JavaScript Do?</h1>

<p id="demo">JavaScript can change HTML content.</p>

<button type="button" onclick='document.getElementById("demo").innerHTML = "Hello
JavaScript!"'>Click Me!</button>

</body>
</html>
```

What it can do?!

- **JavaScript Can Change HTML Attributes**

This example changes an HTML image, by changing the src attribute of an <img> tag:

```
<!DOCTYPE html>
<html>
<body>

<h1>What Can JavaScript Do?</h1>

<p>JavaScript can change HTML attributes.</p>

<p>In this case JavaScript changes the src (source) attribute of an image.</p>

<button onclick="document.getElementById('myImage').src='pic_bulbon.gif'">Turn on
the light</button>

<img id="myImage" src="pic_bulboff.gif" style="width:100px">

<button onclick="document.getElementById('myImage').src='pic_bulboff.gif'">Turn
off the light</button>

</body>
</html>
```

What it can do?!

- **JavaScript Can Change HTML Styles (CSS)**

Changing the style of an HTML element, is a variant of changing an HTML attribute:

```
<!DOCTYPE html>
<html>
<body>

<h1>What Can JavaScript Do?</h1>

<p id="demo">JavaScript can change the style of an HTML element.</p>

<button type="button"
onclick="document.getElementById('demo').style.fontSize='35px'">Click
Me!</button>


</body>
</html>
```

What it can do?!

- JavaScript Can Validate Data

JavaScript is often used to validate input

```
<!DOCTYPE html>
<html>
<head>
<script>
function validateForm() {
    var x = document.forms["myForm"]["fname"].value;
    if (x == "") {
        alert("Name must be filled out");
        return false;
    }
}
</script>
</head>
<body>

<form name="myForm" action="/action_page_post.php"
onsubmit="return validateForm()" method="post">
Name: <input type="text" name="fname">
<input type="submit" value="Submit">
</form>

</body>
</html>
```

JavaScript Program

- A computer program is a **list of "instructions"** to be "executed" by the computer.

- In a programming language, these program instructions are called **statements**.

- JavaScript statements are separated by semicolon.

JavaScript Statement

- JavaScript is a **sequence of statements** to be executed by the browser.

- Unlike HTML, JavaScript is case **sensitive** therefore watch your capitalization closely when you write JavaScript statements

- This JavaScript statement tells the browser to write to the web page:

```
document.write("Hello !!");
```

- The semicolon is **optional** (according to the JavaScript standard [Using semicolons makes it possible to write multiple statements on one line].

How to insert JavaScript

- Inside <head> and/or <body>

  - The HTML **<script>** tag is used to insert a JavaScript into an HTML page.
  - You can place any number of scripts in an HTML document.
  - Scripts can be placed in the <body>, or in the <head> section of an HTML page, or in both.
  - Keeping all code in one place, is always a good habit.

- Example

```
<html>
    <body>

     <h1>My First Web Page</h1>
     <script type="text/javascript">
     //  ... some JavaScript code ...
     document.write("Hello," +" today is "+Date());
     </script>

   </body>
</html>
```

How to insert JavaScript (Cont.)

- Scripts can also be placed in **external files.**
- External scripts are practical when the same code is used in many different web pages.
- JavaScript files have the file extension **.js.**
- To use an external script, put the name of the script file in **the src (source) attribute** of the <script> tag:

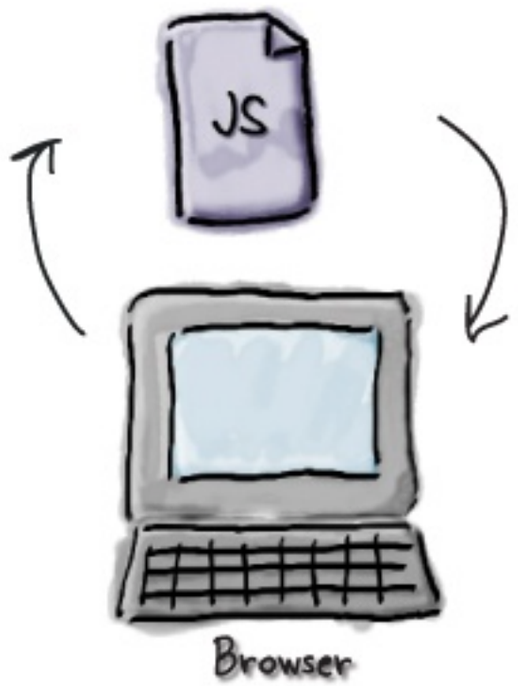
- Example

```
<!DOCTYPE html>
  <html>
    <body>
      <script src="myScript.js"></script>
    </body>
</html>
```

Example

```
<html>
  <body>
    <script type="text/javascript">
    document.write("<h1>here is heading 1</h1>");
    document.write("<p>here is paragraph</p>");
    document.write("<p>here is another paragraph</p>");
    </script>
  </body>
</html>
```

```
<html>
<head>
<title>Icecream</title>
<script>
 var x = 49;
</script>
<body>
<h1>Icecream Flavors</h1>
<h2><em>49 flavors</em></h2>
<p>All your favorite
flavors!</p>
</body>
</html>
```



Writing

1

You create your page just like you always do, with HTML content and CSS style. And you also include JavaScript in your page. As you'll see, just like HTML and CSS, you can put everything together in one file, or you can place JavaScript in its own file, to be included in your page.

Loading

2

Point your browser to your page, just as you always do. When the browser sees code, it starts parsing it immediately, getting ready to execute it. Note that like HTML and CSS, if the browser sees errors in your code, it will do its best to keep moving and reading more JavaScript, HTML and CSS. The last thing it wants to do is not be able to give the user a page to see.

Executing

3

The browser starts executing your code as soon as it encounters it in your page, and continues executing it for the lifetime of your page. Unlike early versions of JavaScript, today's JavaScript is a powerhouse, using advanced compilation techniques to execute your code at nearly the same speed as many native programming languages.

JavaScript Comments

- Single line comments start with //.
- Multi line comments start with /* and end with */.

Example

```
<script type="text/javascript">
  /*
  The code below will write
  one heading and two paragraphs
  */
  document.write("<h1>This is a heading</h1>");
  document.write("<p>This is a paragraph.</p>");
  document.write("<p>This is another paragraph.</p>");
</script>
```

JavaScript Variables

- In a programming language, **variables** are used to store data values.

- JavaScript uses the **var** keyword to define variables.
  var x;
- An equal sign is used to assign values to variables.
  var x=5;

- Rules for JavaScript variable names:

  - Variable names are **case sensitive** (y and Y are two different variables)
  - Variable names must begin with a letter, an underscore (_) or a dollar sign ($).
  - Subsequent characters can be letters, digits, underscores, or dollar signs

Local and Global Variables

- **Local JavaScript Variables**
  - A variable declared within a JavaScript function becomes LOCAL and can **only** be accessed within that function. (the variable has local scope).
  - Local variables are destroyed when you exit the function.

- **Global JavaScript Variables**
  - Variables declared outside a function become GLOBAL, and all scripts and functions on the web page can access it.
  - Global variables are destroyed when you close the page.

Local and Global Example

```
<html>
  <head>
        <script type="text/javascript">
             function displaymessage()
                {
                  var x=3;    //local variable
                  alert("Local x = " + x + " , Global y="+y);
                 }
         </script>
  </head>
  <body>
        <script type="text/javascript">
            var y=5;    //global variable
            alert(" Global y = " + y);
         </script>
         <form>
               <input type="button" value="Click me! " onclick="displaymessage()" />
         </form>
   </body>
</html>
```

JavaScript Arithmetic Operators

Given that y=5, the table below explains the arithmetic operators:

Operator	Description	Example	Result	
+	Addition	x=y+2	x=7	y=5
-	Subtraction	x=y-2	x=3	y=5
*	Multiplication	x=y*2	x=10	y=5
/	Division	x=y/2	x=2.5	y=5
%	Modulus (division remainder)	x=y%2	x=1	y=5
++	Increment	x=++y	x=6	y=6
		x=y++	x=5	y=6
--	Decrement	x=--y	x=4	y=4
		x=y--	x=5	y=4

JavaScript Assignment Operators

- Assignment operators are used to assign values to JavaScript variables.

- Given that x=10 and y=5, the table below explains the assignment operators:

Operator	Example	Same As	Result
=	x=y		x=5
+=	x+=y	x=x+y	x=15
-=	x-=y	x=x-y	x=5
*=	x*=y	x=x*y	x=50
/=	x/=y	x=x/y	x=2
%=	x%=y	x=x%y	x=0

The + Operator

The + operator can also be used to add string variables or text values together.

```
txt1="What a very";
txt2="nice day";
txt3=txt1+txt2;
txt4=txt1+" "+txt2;
```

After the execution of the statements above, the variable txt3 contains "What a verynice day".

After the execution of the statements above, the variable txt4 contains: "What a very nice day"

Comparison Operators

- Comparison operators are used in logical statements to determine equality or difference between variables or values.
- Given that x=5, the table below explains the comparison operators:

Operator	Description	Example
==	is equal to	x==8 is false x==5 is true
===	is exactly equal to (value and type)	x===5 is true x==="5" is false
!=	is not equal	x!=8 is true
>	is greater than	x>8 is false
<	is less than	x<8 is true
>=	is greater than or equal to	x>=8 is false
<=	is less than or equal to	x<=8 is true

Logical Operators

Given that x=6 and y=3, the table below explains the logical operators:

Operator	Description	Example
&&	and	(x < 10 && y > 1) is true
||	or	(x==5 || y==5) is false
!	not	!(x==y) is true

Conditional Operator

JavaScript also contains a conditional operator that assigns a value to a variable based on some condition.

Syntax

```
variablename=(condition)?value1:value2;
```

References

- www.w3schools.com
- Duckett, J. (2011). *HTML and CSS: Design and Build Websites*. John Wiley & Sons.
- Deitel & Deitel (2011). *Internet and World Wide Web How to Program, 5th Edition, Harvey & Paul Deitel & Associates.*