Information Infrastructure II

INFO I211 – Spring 2014 – Sections 18530 & 22719

Lecture 21 – 2014.04.07 & 2014.04.08

Instructor:
Mitja Hmeljak,
http://mypage.iu.edu/~mitja
mitja@indiana.edu

Hello World in JavaScript

a JavaScript one-line program:

```
<html> <head><title>Javascript Hello World 01</title></head>
<body>

    <script language=javascript>

        document.write("<h1>Hello World, this is Javascript!</h1>")

    </script>

</body>
</html>
```

Hello World, this is Javascript!

Hello World (in JavaScript) part 2

JavaScript variables and arrays:

```
<html> <head><title>Javascript Hello World 02</title></head>
<body>

    <script language=javascript>
        var myArray = new Array(50)


        document.write("<h3>Hello World, this is a Javascript array:</h3>")

        document.write("<br>")

        for (var j = 0; j < 5; j = j+1) {
            myArray[j] = j * j;
            document.write("j is: " + j + ",    ")
            document.write("myArray["+j+"] is: " + myArray[j] + "<br>")
        }

        document.write("<br>")

    </script>

</body>
</html>
```

Hello World, this is a Javascript array:

j is: 0, myArray[0] is: 0
j is: 1, myArray[1] is: 1
j is: 2, myArray[2] is: 4
j is: 3, myArray[3] is: 9
j is: 4, myArray[4] is: 16

JavaScript input/output

JavaScript code typically executes inside a web browser.

Therefore, most the input/output and user interaction to and from a JavaScript program will be handled within a web browser's page.

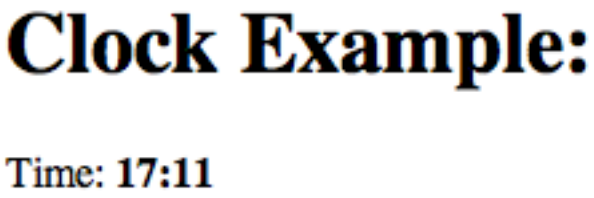
Input: the user will provide input using HTML <form> elements, buttons, clicking on links, etc.

Output: a JavaScript program's output is displayed typically as part of a web page, as HTML rendered by the web browser.

JavaScript output on a web page

Since the JavaScript program inside <script> tags is run *as the web browser renders the page*, the program can produce output that is calculated right at that moment:

```
<html>
    <body>
    <h1>Clock Example:</h1>
    <script language=javascript>
        var date = new Date()
        var hours = date.getHours()
        var min  = date.getMinutes()
        document.write("Time: <b>" +hours+":"+min+"</b>")
    </script>
</body>
</html>
```



JavaScript output... with *seconds* precision

Solution:

```
<html>
    <body>
    <h1>Clock Example:</h1>
    <script language=javascript>
        var date = new Date()
        var hours = date.getHours()
        var min  = date.getMinutes()
        var sec  = date.getSeconds()
        document.write("Time: <b>" +hours+":"+min+":"+sec+"</b>")
    </script>
</body>
</html>
```

Clock Example:

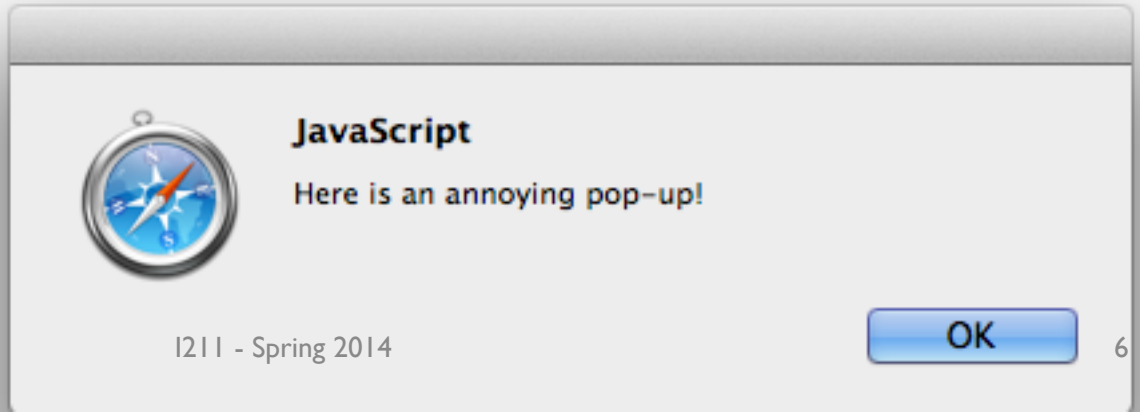
Time: **17:20:11**

JavaScript output as separate GUI widgets: an "Alert" Dialog Box!

A script can use *alert("a string")* function to pop up an "Alert" Dialog box.

```
<html>
    <head><title>Javascript Hello World 04</title></head>
    <body>
        <h3>Javascript Hello World 04: an <em>Alert</em> Dialog Box</h3>
            <script language=javascript>
                alert("Here is an annoying pop-up!")
            </script>
    </body>
</html>
```

Javascript Hello World 04: an *Alert* Dialog Box



JavaScript *input* on a web page

Let's define a JavaScript function:

```
<script language=javascript>
    function myFunc() {
      alert("You clicked it!")
    }
</script>
```

The syntax to call the myFunc() JavaScript function from HTML is an *extra parameter* to the HTML button *input* tag: onClick="FUNCTIONNAME();"

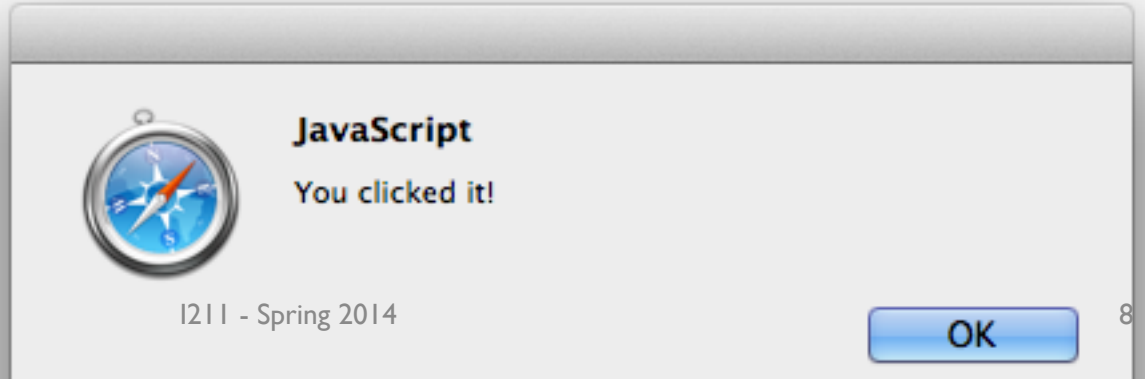
For example:

```
<input type=button value="Click me!" onClick="myFunc();" >
```

Clicking on the "Click me!" button will cause the JavaScript *Alert* Dialog Box to appear, as in the hello05.html example seen above.

JavaScript *input* on a web page

```
<html>
  <head><title>Javascript Hello World 05</title></head>
  <body>

    <script language=javascript>
      function myFunc() {
        alert("You clicked it!")
      }
    </script>

    <input type=button value="Click It!" onClick="myFunc();" >

  </body>
</html>
```



JavaScript output: modifying an already loaded webpage

The built-in document.write() JavaScript function can be used to add text to a webpage as it is being loaded, as seen in previous example.

But it's useful to be able to modify a web page's contents *after* it has been loaded. In that case, an HTML *named area* tag can be used. That name area's content can subsequently be changed from JavaScript.

For example, the HTML code:

```
<div id="txt">Plain Text</div>
```

generates a tag *named area* with the name "txt".

The HTML content inside the tag *named area* can then be changed from its initial value (i.e. the string "Plain Text" above) by using JavaScript code as follows:

```
var myDIV = document.getElementById('txt')
myDIV.innerHTML = "<b>New Text!</b>"
```

Putting it all together: onClick in HTML + a JavaScript function

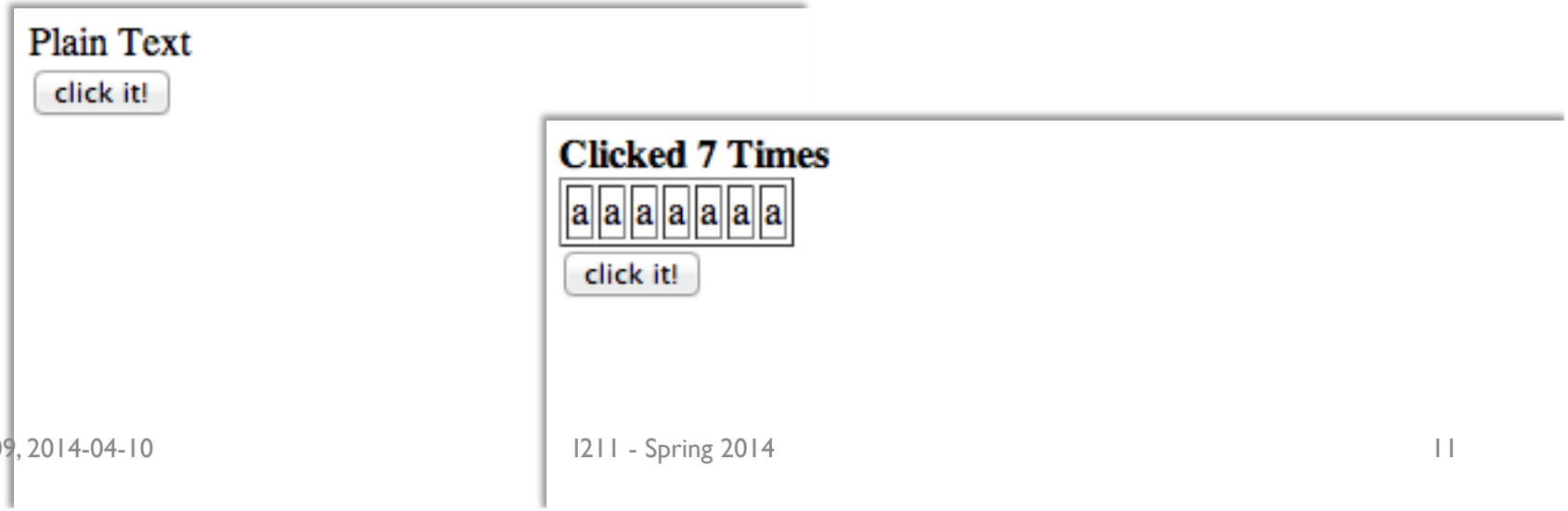
Solution:

```
<html>
  <head><title>Javascript Hello World 06</title></head>
  <body>

    <script type="text/javascript">

      var num = 0
      function startTime()  {
        num = num + 1
        var myDiv = document.getElementById('myDivText')
        myDiv.innerHTML="<b>Clicked " +num+ " Times</b>";
      }

    </script>
    <div id="myDivText">Plain Text</div>
    <input type=button value="click it!" onClick="startTime()">

  </body>
</html>
```

Group work:

Modify the script in the previous slide, so that it outputs a ***1-row*** table when you click.

The table row has to increase by 1 element every time you click.



Group work solution:

```
<html>
  <head><title>Javascript Hello World 07</title></head>
  <body>

    <script type="text/javascript">

      var num = 0
      var string = "empty"

      function startTime()  {
          num = num + 1
          var myDiv = document.getElementById('myDivText')
          myDiv.innerHTML="<b>Clicked " +num+ " Times</b>";

          string = "<table border=1>"
          string = string + "<tr>"

          for (var j = 0; j < num; j = j+1) {
              string = string + "<td>" + "a" + "</td>"

          }
          string = string + "</tr>"
          string = string + "</table>"

          myDiv.innerHTML = myDiv.innerHTML + string
      }

    </script>
    <div id="myDivText">Plain Text</div>
    <input type=button value="click it!" onClick="startTime()">
  </body>
</html>
```