

JavaScript Orientation

EE1707 Programming for Digital Media 2

Workshop 1 (Peter Broomhead & Ali Mousavi)

This workshop introduces you to JavaScript development using NetBeans and reinforces material presented as part of the accompanying lecture course.

The work sheet itself is self-paced in the sense there is no set allocation of work to be carried out during the timetabled activity. But remember you learn to program by writing code, solving problems, observing what others have done, emulating others and most importantly by experimentation; that is: *'suck-it-and-see'*.

Please ensure you read the worksheet carefully and completely understand what is being asked of you in terms of the tasks before attempting them. Remember if you are lost at this stage, matters will only get worst later on.

In this workshop you are expected to:

- Use comments to make the program more readable
- Use correct program layout
- Test your program at each stage
- Create a NetBeans HTML5 project
- Create a HTML file in the project
- Use a JavaScript attribute string as an event handler
- Use an inline JavaScript function as an event handler
- Include an external JavaScript file

- 1) The first step is to create a NetBeans PHP project on media you 'own', in other words either use your University h: drive (the preferred option) or a USB stick. This is a *one-off* task and once completed we will just add additional files to the project as required in this and subsequent workshops.

A set of instruction describing the process of setting up a NetBeans HTML project is available on Blackboard as part of this workshop.

- 2) Add a new HTML file (File > New File... > HTML5 > HTML File) to your NetBeans project for this week's workshop. Set the File Name to *workshop1* or similar.

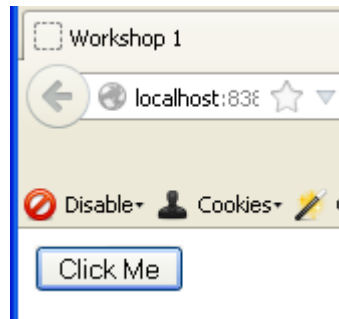
Set the title to Workshop 1 and replace the <div> in the body with an input button. The HTML content should look like this:

```

<html>
  <head>
    <title>Workshop 1</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width">
  </head>
  <body>
    <input type="button" value="Click Me" name="button" />
  </body>
</html>

```



Now right-click on the edit surface and select Run File to render the HTML:



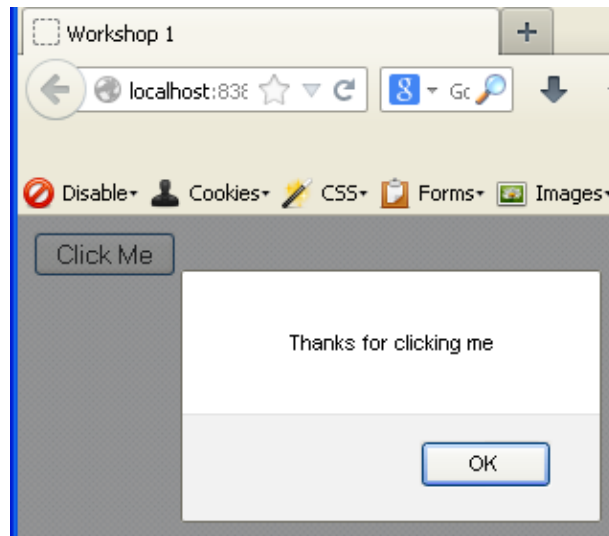
- 3) Debugging JavaScript is not easy and any help you can get in tracking down bugs and errors is a worthwhile activity. While Firefox is running take the opportunity to add some add-on's to the mix:
1. Firebug - use Google to locate this add-on and then add it to Firefox, install and finally restart Firefox itself
 2. Web Developer (by Chris Pederick) – again use Google to locate this add-on and then add it to Firefox, install and finally restart Firefox itself.

Such add-on's will make life easier when it comes to tracking down errors in your JavaScript code. If all is well then something like this should be visible in the top right hand corner of the browser. The first tick is the HTML, the second



is the CSS and the third is for the JavaScript. If errors occur in any of the three categories then a 'Red' cross:  will be displayed. Hover or click on the  cross to obtain a message relating to the error condition.

- 4) Now add an 'onclick' attribute to your <input> button whose value contains the JavaScript string: `alert('Thanks for clicking me');`. When the button is clicked a dialog box should appear displaying the specified text:



- 5) To the HTML header add a <script> tag:

```
<script>  
    // JavaScript goes here  
</script>
```

In the <script> tags body write a function called: 'clickMe' that simply call the alert function to display the message and then call the 'clickMe' function from the <input> buttons *onclick* attribute - replace the call to alert() with a call to clickMe(). Run the file and click on the rendered button to test the alert is still working – the response should be the same.

- 6) Add a JavaScript file to the project, in the *project* window right click on the 'Site Root' node and select a New > JavaScript File... set the File Name to *myAlert*.

Now transfer your 'clickMe' function from the <script> in the HTML header to your myAlert.js JavaScript file. Next add a 'src' attribute to the <script> tag in the header whose value is set to 'myAlert.js'. Run the file and click on the rendered button to test the alert is working.

- 7) Just for the craic, replace the call to alert() in your 'clickMe' function with `document.write('boom')`. Run the file and click on the rendered button to observe the effect on your HTML DOM tree. Clearly using `document.write()` at run-time i.e. as part of an event handler is not a sensible course of action.