Multimedia Authoring using Bootstrap, HTML 5, CSS & JS

Aim: To consolidate the basics of creating media-rich and dynamic web pages with HTML 5, Cascading Style Sheets (CSS), JavaScript (JS) and Bootstrap, and to develop more advanced skills.

In these practicals you will start with a skeleton web-based multimedia application that is built with Bootstrap and you will go through a series of exercises to add content to this application. Instructions are not detailed, and you will need to create your own code to implement what is asked for. You will need to consult learning resources as you proceed, and each other. The text here is written for someone who is a bit shaky on web pages and HTML etc, so we don't leave anyone behind by starting too high. There is lots of scope for those of you who are more familiar with web tools to get more advanced more quickly. There are no checkpoints, or assessments associated with these sheets: they're to help guide your learning.

Basic Introduction: HTML 5, CSS, JS and Bootstrap

HTML 5, Cascading Style Sheets (CSS) and JavaScript (JS) are used together to create web-based applications. As well as traditional web pages, many mobile apps are built using these technologies. In these practical sessions you will gain some experience in how they can be used to create dynamic, media-rich web applications.

HTML 5 is the latest version of the HyperText Markup Language (HTML) that is used to specify the structure and content of web pages.

CSS is a separate computer language for specifying the style and layout of web pages.

JS is a fully-fledged programming language that can be executed by web browsers. It is used in many ways, but particularly to add functions for reacting to events, such as button clicks and data entry in forms, to perform any needed calculations, and to alter dynamically the HTML and CSS of a page to change its content, presentation and layout.

Bootstrap is a framework that provides pre-built components, using HTML, CSS and JS, for use in creating sophisticated web pages. Its CSS classes are designed to work seamlessly from large screens down to mobile phone screens, resizing and repositioning components as appropriate for the current screen size.

Resources

To learn more and to complete the more advanced parts of this practical you should consult the ***W3Schools*** tutorials and references on HTML, CSS, JS and Bootstrap at **http://www.w3schools.com/**

Introduction to Our Multimedia Web Application

Is this a web page or a multimedia application? They look pretty similar nowadays, and we will use the terms interchangeably. Our basic web page will contain different screens that are accessed by buttons. Each screen will give an example of different Bootstrap elements. There are some ideas below, but you are encouraged to be creative and explore other options. The skills learnt here will be employed in the module assignment.

Preparatory work

Starter files for this practical session are in the **WebMM** ZIP file on the week 1 Multimedia materials page on **Canvas**. Download this file to somewhere convenient in your file space and unzip it.

Development tools: choose what you like to use. A fully-fledged IDE such as **Netbeans** or **Eclipse**: these tools make it easier to keep track of files in single projects, and provide useful editing features, such as code completion. **TextPad** is a fairly intuitive text editor, and provides syntax colouring. To view your application and to **debug** it, you need to display it in a web browser, such as **Chrome** or **Firefox**. The **developer tools** in these browsers will help you to debug your code, particularly your Javascript, by enabling you to set breakpoints and watch variable values.

To start things off:

1. Open your copy of the **WebMM** folder. You will see a couple of HTML files and a subfolder containing some media to use.
2. Open **MMappBS.html** in the browser and your IDE. This is the basic starter file. JavaScript is used to change the content in the main panel when a new button is clicked (an alternative might be to display a whole new web page).
3. Open **MMappBSfin.html** in the browser. This is a slightly more enhanced version. Check out briefly what there is by clicking each of the buttons in the left-hand menu in turn. You might like to leave this open in your web browser so you can refer back to it as you progress with building your own application.
    a. Note of course that this file contains code for all of the exercises given below, but you are strongly advised NOT to look at this unless you are really stuck. What you will build should not be an exact replica of this version, and you are encouraged to develop your own code (there will be more than one way to do some of the things you are asked for).

4. Familiarise yourself with the basic code. How is the document constructed (scripts and divs)? What are the main components? How do these relate to what you see on the screen? How do styles affect the screen presentation? If you're not sure, change some elements in a small way to experiment.
5. Button operation is controlled using **collapse.** Make sure you understand what happens.
6. Experiment with styling, to change the look of the paragraph text. Add more text if you want to play around with different styles for different paragraphs.
7. (Remember that you need to **Refresh** the web page in your browser to see the change that you have made.)
8. Add a transition to a text element so that it responds to mouse hover.

Feel free to play with different CSS styles for this page before moving on to the next exercise.

Slide show

Add a new scene, with its associated button, to our app: the slide show.

9. Add a new button for scene1 directly below the code for the scene0/Home button.
10. Directly below the closing </div> tag that finishes “scene0” (but before the </div> that finishes the “stage”) create a new scene with ID “scene1”, following the template of the home screen (you could cut-and-paste the entire “scene0” <div>…</div> and edit as needed.
11. A slide show is nicely presented using the **carousel** component. See w3schools for the details. The Media folder contains three images: image1.jpg, image2.jpg, image3.jpg, but feel free to source your own images.

If things do not work as expected and you cannot work out why by looking at your code, you could try using the debugger in the Developer Tools of your browser. Again, please ask if you need help: use the Teams channel, or ask at the discussion sessions.

Video

HTML5 allows the easy inclusion of videos in web pages via the <video> element. The next screen you will build will display a short video clip and provide suitable user controls to play it.

12. Following the template for screens 0 and 1, create a new <div> with ID “**scene 2**”. Include a sensible <h2> heading and a single paragraph with a style that will centre its text.
13. A video clip is stored in the file “Media/LMFAO.mp4”. It is 423 pixels wide by 240 pixels high. Add a suitable <video> tag to the empty paragraph to include this video, using the “controls” attribute to give it user controls.
14. It is also possible to add your own buttons for user control. Give your <video> tag an ID by including the attribute id=”vid”. This allows us to refer to this tag by name. Create a new button for this scene by adding the following line after the paragraph containing the video:

```
<button onclick="document.getElementById('vid').play();">Play</button>
```

15. Now, in a similar fashion, add buttons to pause and rewind the video and give them a try.

Canvas Graphics

You can create drawings directly in a web page, using the HTML5 Canvas object and JavaScript (JS) programming. This exercise provides an introduction to this approach to web graphics. Sometimes it’s useful to be able to sketch an animation rather than make a video.

16. Create a new “scene 3” with a heading as content, and an associated menu button, as you have done for the other scenes.
17. Below the heading in this scene, create an HTML canvas with a suitable ID, a width of 400 pixels and a height of 280 pixels. You might like to give it a border too.
18. Now after the end of the “scene 3” div, create a <script> element to which you will add JS code to draw to your canvas. It will probably start something like the following (depending on your canvas ID and what variable names you choose to use):

```
<script>
var c = document.getElementById("myCanvas");
var ctx = c.getContext("2d");
ctx.fillStyle = "blue";
</script>
```

19. The exercise is to add code to draw a simple Scottish Saltire with some text: take a look at the finished application to see what we are aiming at. Add code to your script to achieve this (consulting appropriate <canvas> documentation (e.g. on W3Schools) as the need arises; and feel free to ask for help as well!)

Why not code directly in HTML5?

Web technologies are constantly evolving: tools that we used just five years ago are no longer relevant (this can make being a computer scientist / software engineer / web developer / data scientist / …. quite challenging).

To get a feel for what you have gained in using Bootstrap, see this alternative web application which is built with HTML, CSS and JS.

20. Open **MMfinalraw.html** in your favoured IDE and also in your browser. It has similar elements to those you've coded above. Notice that everything is done directly in HTML, CSS, or using JS and the jQuery library. (Now is a very good time to explore a bit about JQuery in the W3Schools site, if you are not familiar with it.)

How much time do you think it would have taken to code this yourself directly? That's why we use libraries! It's also much easier to keep up with developments such as mobile-first, new technologies, and evolving styles and fashions in web development.

Close this file and feel relieved.

The Beginning

That's the minimum. But there's a lot more to learn, so keep playing and modifying to learn about other Bootstrap features. Be adventurous!

But…

MAKE SURE YOU KEEP ALL OF THIS PRACTICAL WORK, AS YOU WILL RETURN TO IT LATER!!

In a later practical session you will add more buttons to this application that will demonstrate different techniques for adding animation to your web pages.