School of Computer Science
University of Lincoln
2018/19

CMP1130M Web Authoring

Lab (3)

Brief

The aim of this workshop is to familiarise yourself with the basic styling and some of the new HTML5 structuring elements. By the end of this workshop you are expected to be able to:

- Implement some of the HTML5 structuring elements to structure your page.
- Understand and utilise ID, Class, Tags and compound styles.
- Implement the basic styling to control the text contents and add simple mouse-hover effects and image icons using background positioning. (Using the cheat sheet provided and online sources)
- Able to add audio and video contents to your page, using the HTML5 elements.

I recommend working in pairs*. Find some one with a similar ability and take it turns to develop. Be proactive and resourceful. I want you to look at https://www.w3schools.com/ for reference and be creative in your development.

***Pair programming** is an agile software development technique in which two programmers work together at one workstation. One, the driver, writes code while the other, the observer or navigator, reviews each line of code as it is typed in. The two programmers switch roles frequently.*

While reviewing, the observer also considers the "strategic" direction of the work, coming up with ideas for improvements and likely future problems to address. This frees the driver to focus all of their attention on the "tactical" aspects of completing the current task, using the observer as a safety net and guide.

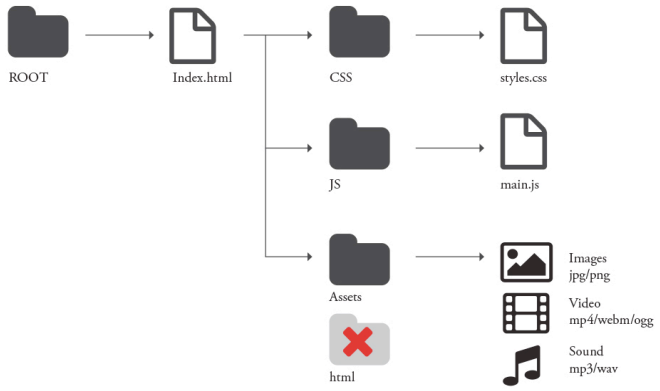
NOTE TASK 1 AND 2 ARE WHAT WE COVERED LAST WEEK.
**** if you copy and paste from this doc, watch the quote marks " sometimes it adds speech marks which are syntactically wrong.**

Images and files are in the ZIP provided.

Continued >

Task 1

Create a new folder (your_name_site_cache) with a folder structure below on your computer.



Best practices for naming files and folders

Follow these general guidelines for naming files and folders:

- File and folder names in websites should never contain spaces or any of the following characters: /\?%*:|"<>.
- Although other special characters are permitted, it is better to confine yourself to alphanumeric characters, hyphens, and underscores.
- Most file and folder names end up as part of a web page's URL, so keep them short, but meaningful. Long URLs are difficult for users to remember and type into browsers on a mobile device.
- Stick to lowercase letters for file and folder names to avoid file not found issues. Most web servers are Linux –based, which are case-sensitive.

2 Setting up a local version of your site

1. In Dreamweaver, choose Site > New Site.
2. In the Site Setup dialog box, make sure the Site category is selected.
3. In the Site Name text box, enter a name for your site. This name appears in the Files panel and in the Manage Sites dialog box; it does not appear in the browser.
4. In the Local Site Folder text box, specify the folder you identified earlier—the folder on your computer where you want to store the local version of your site files. Click the folder icon to the right of the text box to browse to the folder.

Task 2

Create your first html page and style sheet

Now you have your file structure and site cache set up, You can now your first page. In DW you can create a page that contains a predesigned CSS layout, or **create a completely blank page and then create a layout of your own** – you must do the later.

Go to window > workspace layout > developer

Select File > New.

In the New Document category, select the kind of page you want to create from the Document Type column. For example, select HTML to create a plain HTML page.

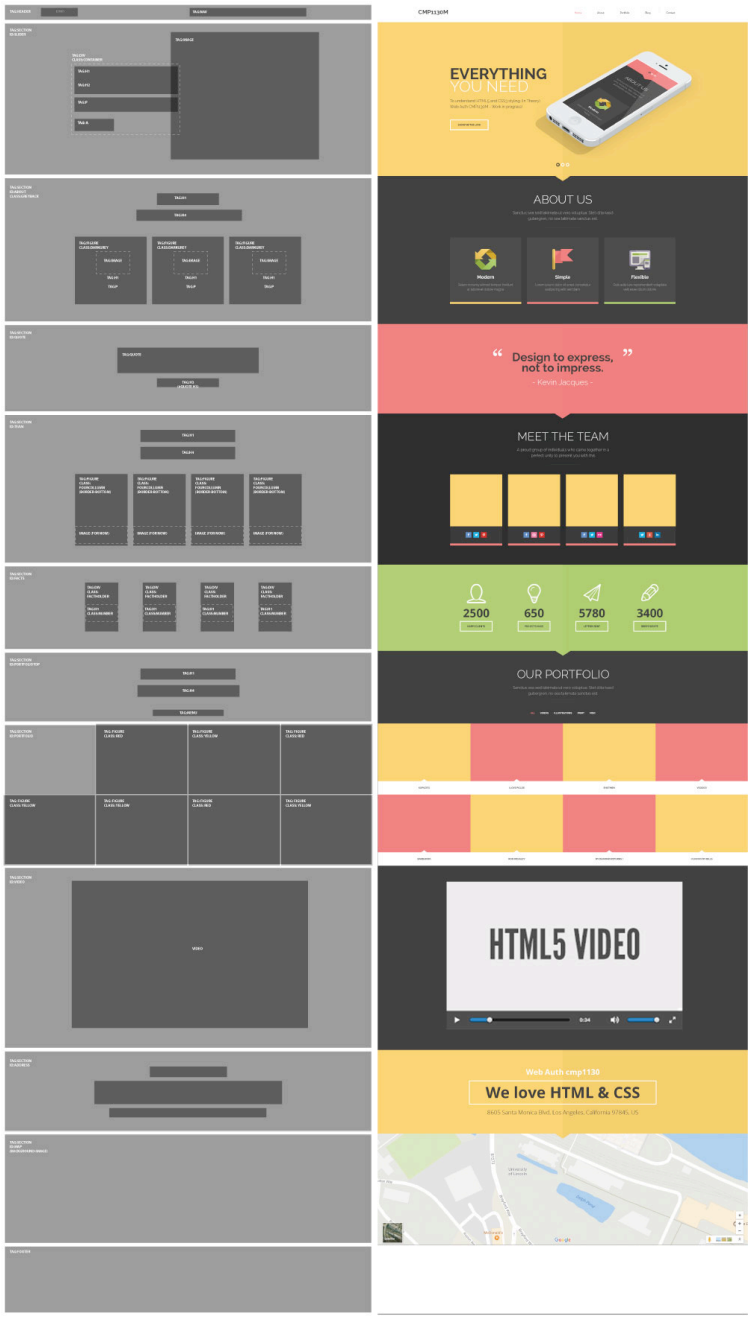
1. Select a document type from the DocType pop-up menu. In most cases, you can use the default selection, HTML5.
2. Select additional options depending on the type of page you want to create.
3. Select None: Select this option if you want to create a simple web page without using any framework. **Do not use Bootstrap:** Bootstrap templates are predefined layouts using the Bootstrap framework. It creates responsive web pages using the Bootstrap framework
4. Save as index.html in the site cache folder you have set up. (refer to folder diagram)
   Notice the header and body elements have been created for you. Look at the lecture slides to understand the different roles they play.
5. Create a new css file by going to new > css

6. Save it as styles.css in your css folder
7. Link the new css file to your webpage. All pages should link to one style sheet – consider what the benefits are? <link rel="stylesheet" href="where/your/style/sheet/was/saved.css">
8. Link a nice typeface https://fonts.google.com/specimen/Raleway?selection.family=Raleway follow the instructions

Try to avoid internal style sheets with the <style></style> tag or inline styles line

Task 3

Today we will turn a html5 semantic structure into a beautified webpage…DON'T PANIC, it does not have to be finished. It simply gives you some structure to focus your html and CSS. The wireframe and main design are supplied in jpg format. **The wireframe has hints on tags and classes. Remember to move the images into your assets folder.**



1. Create a div with an id of wrapper
   <div id-“wrapper”> website sections will go in here

2. Add a “header” tag. Inside this you will need a div for the log and a nav, ul, li and a tags to build the menu.
   ```
   <nav>
     <ul>
       <li>
         <a href=“”></a>
       </li>
     </ul>
   </nav>
   ```

3. Build each section in html first, giving each section an ID. They can be called what you want but have given you hint on the wireframe.

4. In the section with the id slider (section#slider) consider how you may split the content. You need to hold the text in one container and position an image
   section#slider img{ } try floating the image right, or absolutely positioning the image.

5. **A:** In section#about you will need to add h1 and h4 tags with css text-align:center; and font sizes – play with the size, test what looks similar.
   B: You will need to add 3 figure tags – with a width, height and float left. Try putting text align centre in the css for section#about or adding a div to hold the 3 items, setting a width and using margin:auto; // it needs a fixed size for margin auto to centre.
   The same process can be applied to the next few sections

6. **A**: Inside section#portfolio you will need 8 x figure tags, again float left will be useful but you will need to consider percentages or viewport width as measurements (width:25vw will give you 1 quarter of the screen size) **OR** look up the “flex column” CSS style and the “box-sizing” CSS style in the cheat sheet or online;
   B:You will need to apply a class for red and yellow (you can use CSS to check for odd or even :nth-child(odd) but we cover that later)

7. Inside section#video you will need to add a video tag. Note if the browser does not support video or the video formats the text will be displayed.

   ```
   <video width="320" height="240" controls>
     <source src="movie.mp4" type="video/mp4">
     <source src="movie.ogg" type="video/ogg">
   Your browser does not support the video tag.
   </video>
   ```

8. section#map add a background-image to the section. You will need to specify the height of the section otherwise it will collapse as there is no content to push it down.
   Use the map image provided. Try the background-size:contain or background-size:conver, what happens?

9. Look back through your html and css, can it be in another way?

10. Can you add animation to your images or links?

11. Can you add more specificity to your styles?

12. Feel free to add more html in order to position objects

BONUS
Who can add the triangle overlays on the sections in pure css – no images. (transform:rotate())

CSS TO CONSIDER:

1. padding:
2. margin:
3. transform: translate()
4. color:
5. background-color:
6. background-position:
7. background-size:
8. font-family:
9. font-size:
10. text-decoration:
11. position: absolute
12. text-align
13. width:
14. height:
15. box-shadow:
16. border-radius:
17. z-index:
18. float:
19. box-sizing:
20. flexbox:

COLOURS
GREY: #343434
DARK GREY # 242424
LIGHT GREY # 3e3e3e
WHITE #FFFFFF
GREEN # a8d164
RED: #ff6f6f
YELLOW # fbcf61

Remember

Always SAVE your work regularly on your own University storage. Backed up by ICT).

Always BACKUP your work regularly, into more than one extra storage (e.g. USB drive, DropBox, Google Drive,…….).

The lab machines can be wiped and/or re-ghosted at any time, without warning. Nothing should be stored in these machines.