

COMP3006 – Full-Stack Development

Workshop 1 – HTML, CSS and JavaScript

Autumn 2020

Depending on your experience, this first workshop has two purposes. If you haven't encountered web development before then it will enable you to have some practice at building software for the web before we move on to more advanced topics. If you have, then this is a good opportunity to revise your knowledge early in the module.

The workshop is structured into three parts. The first should be completed **before the workshop** session you attend. Any problems with these questions can be discussed with the workshop supervisor during the session, but they must have been attempted first. The second section is for completion **during the workshop**, and the third section is an **extension task** that you can complete in your own time. It is entirely possible that you will need to finish the workshop material in your own time.

A few days after the workshop sessions are complete model answers will be posted for the prior and workshop exercises. No model answers will be posted for the extension work, but the workshop supervisor will be happy to look at your solution in later workshops.

Before the Workshop

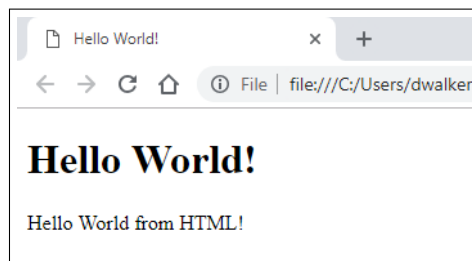
Exercise 1

In the first exercise you will implement a Hello World program using **HTML**. Create a file called *hello.html* and add the following code to it:

```
<!DOCTYPE html>
<html>
<head>
  <title>Hello World!</title>
</head>
<body>
  <h1>Hello World!</h1>
  <p>Hello World from HTML!</p>
</body>
</html>
```

You can use any text editor and browser combination to do this – even the basic Notepad editor will do. I recommend **VS Code** or **Atom**, and **Chrome** for this module.

When you have finished the code, save it and launch it in the browser. You should see the following:



Exercise 2

The simple HTML example created in the previous exercise is somewhat plain. Add some CSS to the page to make it look a bit more attractive. Begin by creating a new file called *hello.css*, and add the following code:

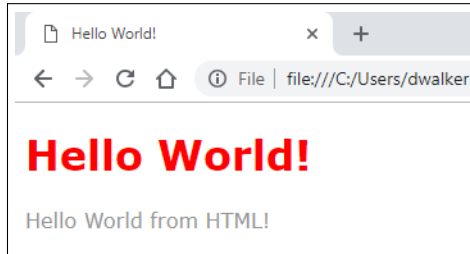
```
body { font-family: verdana, sans-serif; }
h1 { color: red; }
p { color: #969696; }
```

There are several ways CSS can be included – you could do this using the HTML `<style>` element, for example).

This will change the font of all elements within the `body` element to have the Verdana font (or Sans-Serif, if Verdana isn't available), and apply red and grey colouring to the heading and paragraph elements, respectively. You need to link the CSS file to the HTML page by adding the following line of code to the HTML page's `<head>` section:

```
<link rel="stylesheet" href="hello.css" />
```

The page now looks like this:



Exercise 3

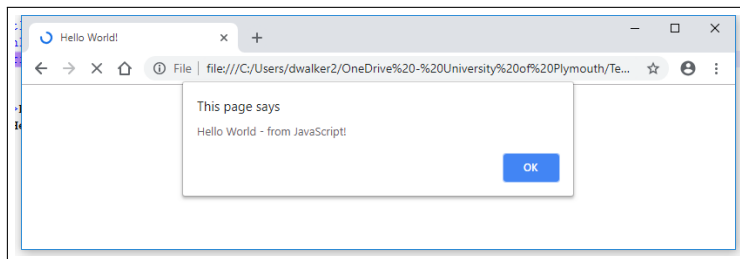
Complete the page by adding a JavaScript welcome message for the users. In another new file (*hello.js*) add the following code:

```
window.onload = function() {  
  alert("Hello World - from JavaScript!");  
};
```

As with the CSS exercise, the JavaScript file needs to be linked to the HTML page. Do this by inserting the following into the `<head>` section:

```
<script src="hello.js"></script>
```

When you load the page you will see the following message:



When you clear the message by clicking OK, the styled website should then return.

Again, you could do this in the HTML document without worrying about the extra file. You would use the `<script>` tag for this. The downside is this would then only be available to the one HTML file, whereas `.js` files can be shared between HTML files.

During the workshop

Exercise 4

This exercise requires you to use HTML, CSS and JavaScript to develop a webpage that fulfils a specific set of features as detailed below. You will get automated feedback on your efforts in terms of a set of unit tests.

Start by downloading the provided zip archive called **COMP3006_Wk1.zip**. This archive contains the files you need to edit, **COMP3006_Wk1.html**. The archive also contains a number of other files that enable automated testing of your code, in particular the **COMP3006_Wk1_Tests.js** file. You must not modify this file.

You need to add HTML, CSS and JavaScript code to provide the elements, styles and behaviours specified below.

The tests have been developed with **Mocha** and **Chai** – you will learn how to build your own tests using these tools later in the module.

You can either create CSS and JavaScript files or include the CSS and JavaScript code within the HTML file.

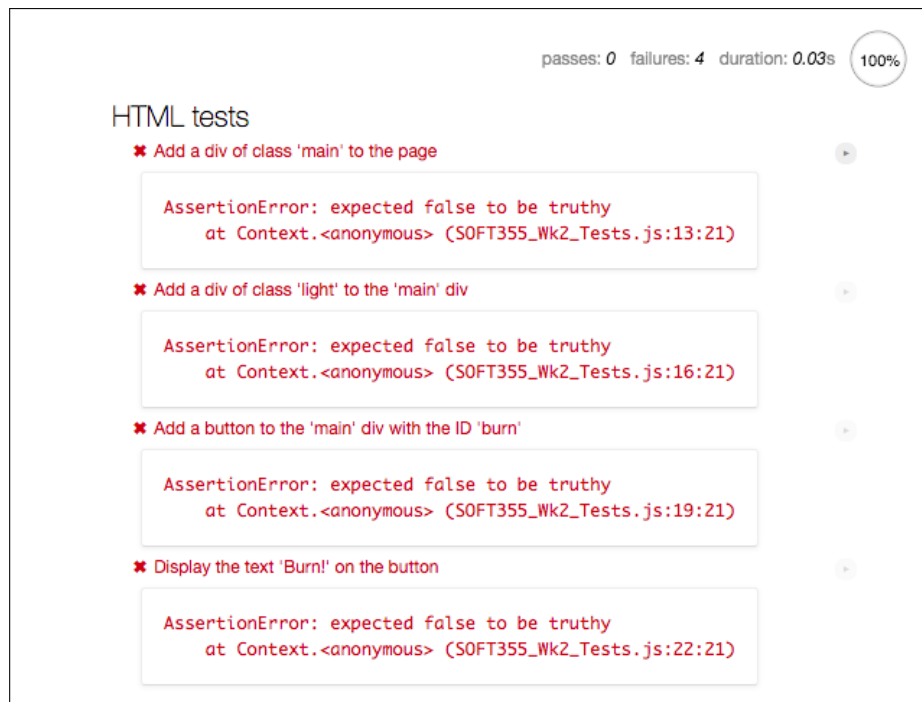


Figure 1: Failing unit tests.

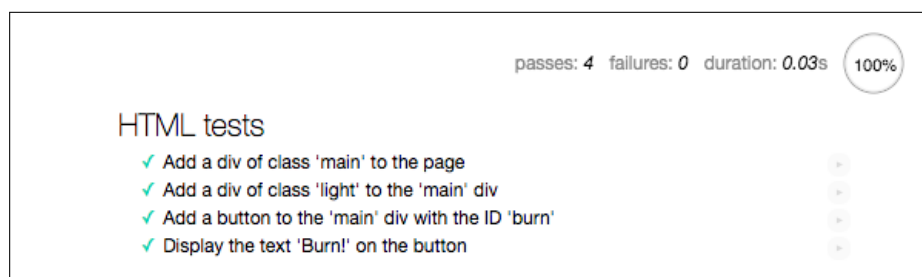


Figure 2: Passing unit tests.

You can run the tests at any point by opening the **COMP3006_Wk1.html** file in Chrome. The test results will appear as a separate element in the webpage. An example page with failing tests is shown in Figure 1. Example output for when all tests succeed is given in Figure 2.

Required HTML Features

1. Add a new `div` to the page using the `<div>` element. Define the new `div` to be of class "main".
2. In the main `div`, add a second `div`. Define this to be of class light.
3. Below the light `div` add a button. Set its ID to be "burn".
4. Display the text "Burn!" on the button.

Required CSS Features

5. Set the properties of the main `div` so that it is 250 pixels wide and has a background colour of #d0d0d0 (pale grey).

6. Set the light div to have a width and height of 50 pixels, and a background colour of #00ffff (cyan).
7. Set the burn button to a width of 150 pixels and the font style to bold.

Required JavaScript Features

8. When the burn button is clicked, the following should happen:
 - (a) If the background colour of the light div is #00ffff (cyan), it should change to #ffff00 (yellow).
 - (b) If the background colour of the light div is yellow, it should change to #ff00ff (magenta).
 - (c) If the background colour of the light div is magenta, it should change to cyan.

If using jQuery for this you must use the pattern `$(window).on("load", function() {})` to write your code. This has been included in the HTML template for you.

Mini Project

Throughout this module the labs will contain exercises that will enable you to complete a mini project combining all of the aspects of full stack development you are learning about.

For the mini project you will develop an online chat system. The system will allow two users to send instant messages to each other, and the messages will be logged in a database. Users will have to log in to the system to use it.

Exercise 5

We will consider the functionality of the system as we work through the module. For now, the task is to create a page that displays a chat history between two users, and has a form containing a text box for new messages and a submit button.

Create a HTML file and construct the page as described above. Then, add some CSS to it to make the page attractive and user-friendly.

Extension task

Exercise 6

You have started work on your mini project. Your project will also require users to log in, so you should create two additional pages: one will allow users to register, and the other will enable them to log in once they have created an account. You should make appropriate use of HTML form elements.

Your username should be an email address – you should include a message to tell the users that. Don't worry about validating the form for now, you will do that in a later session.