Lab 2 – HTML forms and Introduction to CSS

1 Introduction

Welcome to Week 2!

This week we be expanding our knowledge of HTML by looking at how we can accept input from users. For this, we can use HTML forms. While our forms won't yet do anything exciting (that will come in later weeks), it's important that building HTML forms becomes second nature to us (because every website includes some kind of form).

In addition, we introduce Cascading StyleSheets (CSS) for adding style to our websites. Once you start playing around with CSS, it isn't long before you are able to create rich and complex user interfaces (UI's).

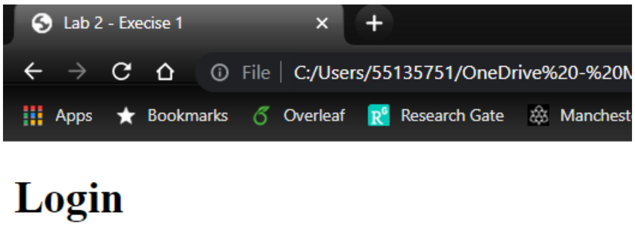
2 Learning Objectives

- To understand, and practice using, HTML forms
- To understand, and practice using, CSS for styling websites and building UIs

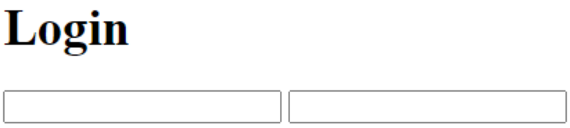
3 HTML Forms

3.1 Exercise 1: Basic form walkthrough

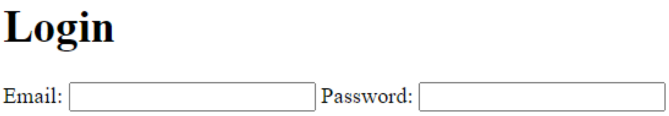
1. Create a new directory for this week's lab exercises. Inside, create a new HTML file
2. Add the basic templating code from last week and ensure that you're page loads correctly in Chrome
3. Give your page a <h1> header containing the word "Login"



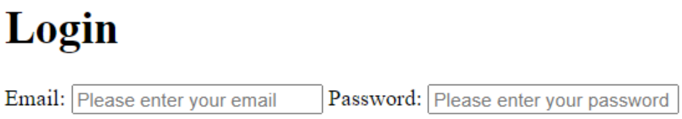
4. Underneath your header, create a new HTML form with the <form> tags. Inside, create two inputs, the first should be of type "email" and the second of type "password"



5. Provide each input with an associated <label> tag. Use W3Schools to help you.



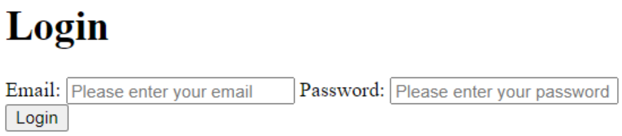
6. Using W3Schools to help you, add a placeholder into each element
7. Make sure that both elements are "required"



Login

Email: Please enter your email Password: Please enter your password

8. Underneath the inputs, add a new input of type "submit". Give this input a "value" of "Login"

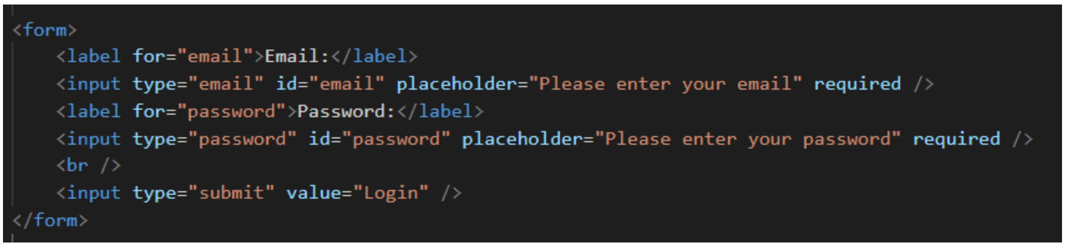


Login

Email: Please enter your email Password: Please enter your password

Login

9. Your final HTML should look something like the below

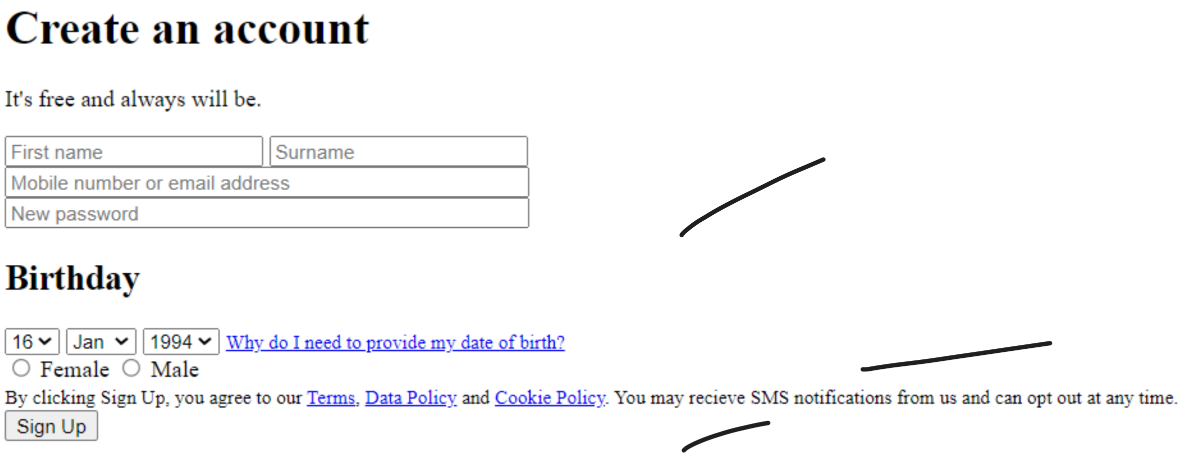


```
<form>
    <label for="email">Email:</label>
    <input type="email" id="email" placeholder="Please enter your email" required />
    <label for="password">Password:</label>
    <input type="password" id="password" placeholder="Please enter your password" required />
    <br />
    <input type="submit" value="Login" />
</form>
```

3.2 Exercise 2: Recreate this form

In a new HTML document try recreating this form. Do you recognise the form?

- Use the lecture slides and W3Schools to help you
- Make sure that you add in all relevant ID's
- The form doesn't use any CSS so you should be able to achieve it through HTML alone.
- You can set the "href" of any links to "#". This is often used as a placeholder while we develop front end interfaces
- We will go through the answers together later in the lab



Create an account

It's free and always will be.

First name Surname

Mobile number or email address

New password

Birthday

16 Jan 1994 Why do I need to provide my date of birth?

○ Female ○ Male

By clicking Sign Up, you agree to our Terms, Data Policy and Cookie Policy. You may recieve SMS notifications from us and can opt out at any time.

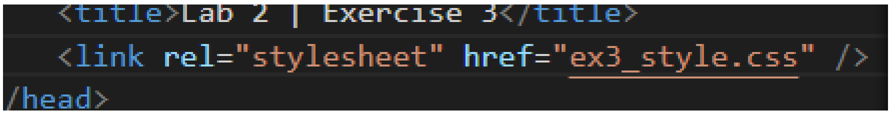
Sign Up

4 CSS for adding style

4.1 Exercise 3: Basic CSS walkthrough

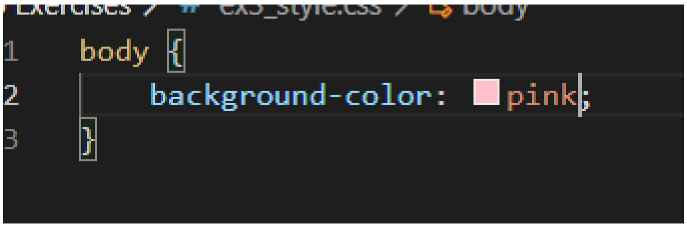
1. Create a new HTML document and add the basic template that we have been using.

2. In the same directory, create a new file called “ex3_style.css”
3. In the <head> tag of your HTML document, add the following line. This will link your stylesheet to your HTML document.



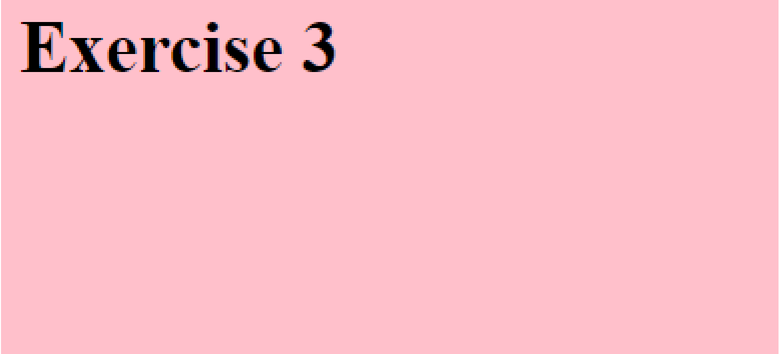
```
    <title>Lab 2 | Exercise 3</title>
    <link rel="stylesheet" href="ex3_style.css" />
/head>
```

4. In order to check that it has worked, lets add a style to the body element. We’ll change the background to pink so that it stands out. In your CSS file, add the following style declaration.

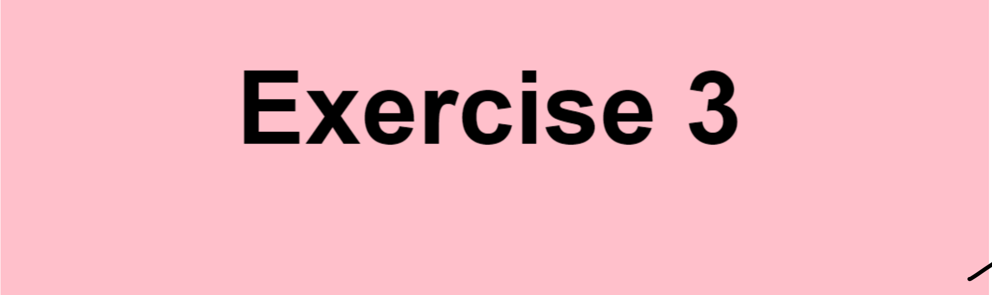


```
Exercises > # ex3_style.css > body
body {
    background-color: pink;
}
```

5. Save both files and open in Chrome. If you have a pink background then congratulations! You’ve successfully linked your CSS file to your web page.



6. We can also style the header. Let’s change the following:
   a. “font-family” to “Arial”
   b. “text-align” to “center”
   c. “font-size” to “72pt”
7. Does your page now look like the below? If not then try to debug using W3Schools and the lecture slides to help you

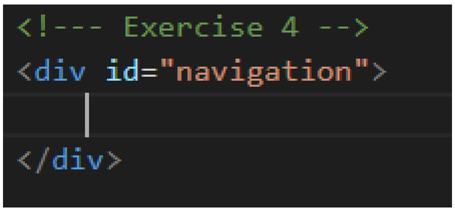


4.2 Exercise 4: Trying out different styles

The possibilities with CSS are endless and there are too many styles to learn them all. CSS can also be used to create animations and complex effects. As you practice web development, you will get used to using the most common style declarations.

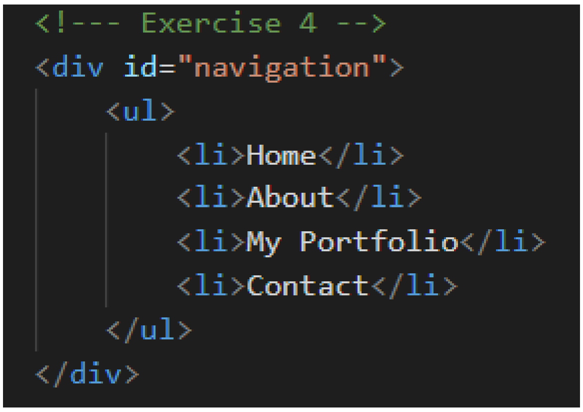
Let’s expand on Exercise 3 by trying out a few more styles.

1. Underneath your heading, add a <div> tag. A div tag (div for division) denotes a section of out web page and allows us to group content together. You can think of it as an invisible box. Give the div an id of “navigation”



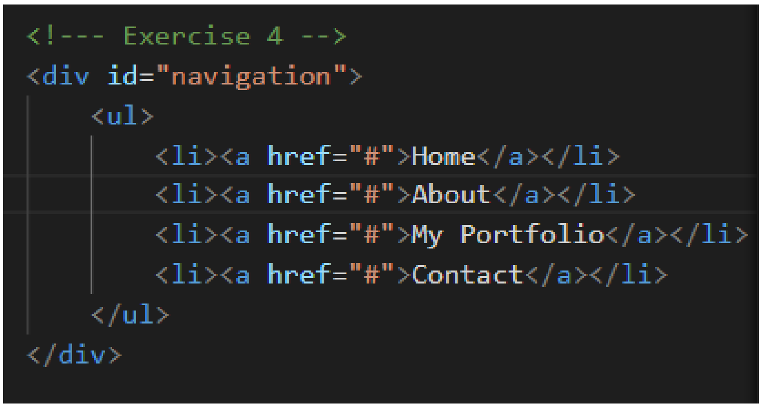
```
<!--- Exercise 4 -->
<div id="navigation">
    
</div>
```

2. Inside this div, we want to add some navigation to other pages on our site. Add an unordered list (<ul>). The list should include four items, one for each imaginary page (Home, About, My Portfolio, Contact)



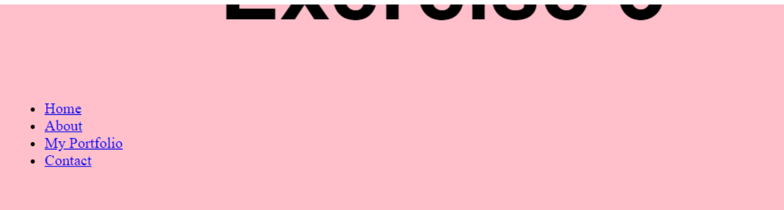
```
<!--- Exercise 4 -->
<div id="navigation">
    <ul>
        <li>Home</li>
        <li>About</li>
        <li>My Portfolio</li>
        <li>Contact</li>
    </ul>
</div>
```

3. Turn each item into a hyperlink with the <a> tag. For now, the href attributes can just be set to “#” as we did in exercise 2



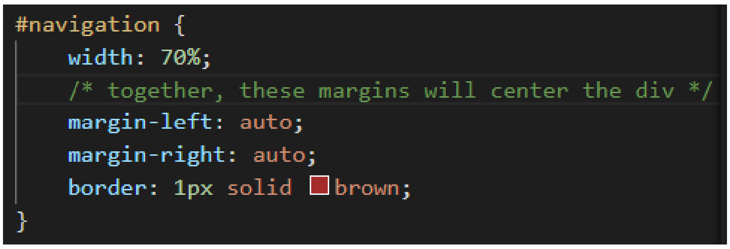
```
<!--- Exercise 4 -->
<div id="navigation">
    <ul>
        <li><a href="#">Home</a></li>
        <li><a href="#">About</a></li>
        <li><a href="#">My Portfolio</a></li>
        <li><a href="#">Contact</a></li>
    </ul>
</div>
```

4. It doesn’t look fantastic on the webpage, but we can change that with CSS



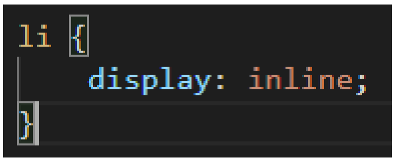
- Home
- About
- My Portfolio
- Contact

5. In the CSS file, add a style for the div. We can specify that we only want the styles to apply to the navigation div by using the ID. In CSS, we use a # before the ID to denote that we are referencing an ID. Below we are going to set the width of the div, the position on the page, and add a border so that we can see it



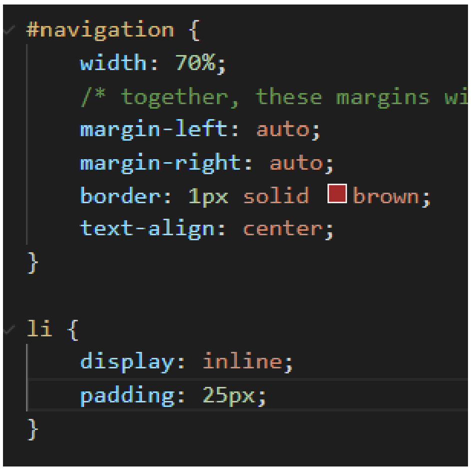
```
#navigation {
    width: 70%;
    /* together, these margins will center the div */
    margin-left: auto;
    margin-right: auto;
    border: 1px solid 🟥brown;
}
```

6. Save and check that it has worked in Chrome
7. The list items (<li>) are current displayed on top of each other. Let’s change that so they appear next to each other (inline).



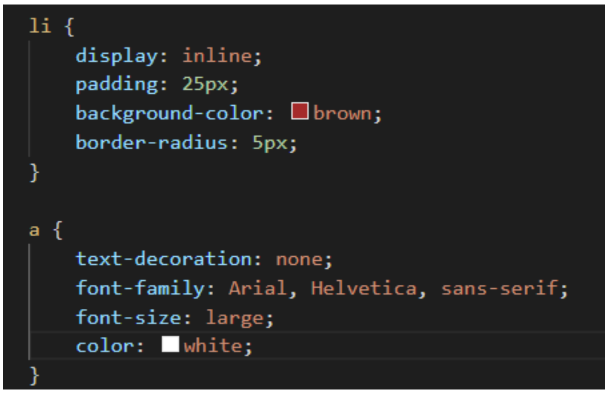
```
li {
    display: inline;
}
```

8. That’s worked and gotten rid of the bullet icons. However, they are all budged up on the left hand side. We can space them out by setting the padding, and center them by adding a text-align declaration in the #navigation style



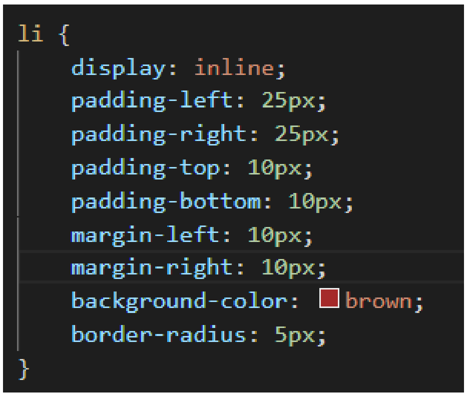
```
#navigation {
    width: 70%;
    /* together, these margins wi
    margin-left: auto;
    margin-right: auto;
    border: 1px solid 🟥brown;
    text-align: center;
}

li {
    display: inline;
    padding: 25px;
}
```

9. It is starting to look good, but the links still look a bit boring. We can remove the existing border and make each item look like a button by adding a background colour. The border-radius style curves the corners. In addition, we can add a style for the <a> tags so that the text appears nicer.



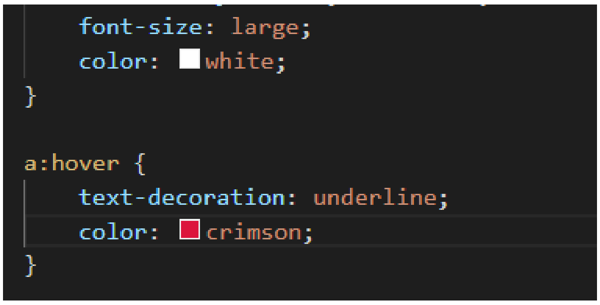
```
li {
    display: inline;
    padding: 25px;
    background-color: 🟥brown;
    border-radius: 5px;
}

a {
    text-decoration: none;
    font-family: Arial, Helvetica, sans-serif;
    font-size: large;
    color: ⬜white;
}
```

10. The buttons are a bit tall. Rather than set the padding all the way around to 25px, we can set the padding for each side individually. I’ve also added some extra margins to space the buttons out some more



```css
li {
    display: inline;
    padding-left: 25px;
    padding-right: 25px;
    padding-top: 10px;
    padding-bottom: 10px;
    margin-left: 10px;
    margin-right: 10px;
    background-color: 🟫brown;
    border-radius: 5px;
}
```

11. Finally, we can add a simple animation by using a selector. Below we use the :hover selector to underline each button when we hover over it and change the colour



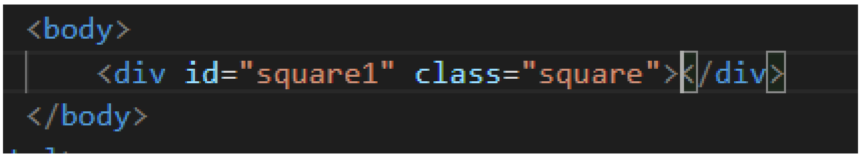
```css
    font-size: large;
    color: ⬜white;
}

a:hover {
    text-decoration: underline;
    color: 🟥crimson;
}
```

12. Save what you have done and try it out in Chrome

4.3 Exercise 5: classes and IDs

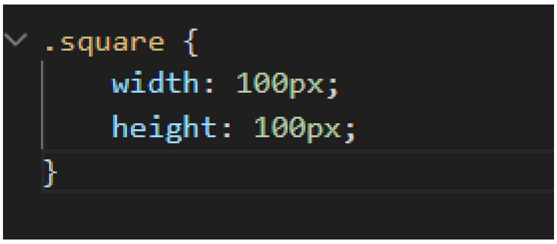
As we have seen, you can add an ID to an element so that it can be uniquely identified and used elsewhere. Classes on the other hand allow us to provide styles to multiple elements. This reduces repetition in your code and makes managing your web sites a lot easier.

1. Create a new HTML file and add the standard templating code.
2. Create a CSS file and link it to your HTML. Test by changing the background colour
3. In the HTML create an empty <div>. The div should have an ID of “square1” and a class of “square”



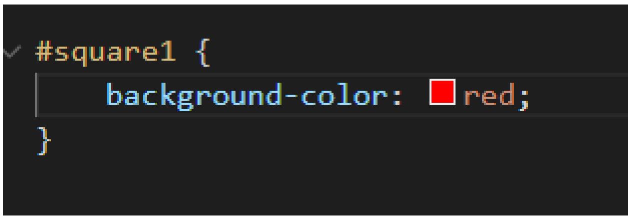
```html
<body>
    <div id="square1" class="square"></div>
</body>
```

4. In your CSS, add a class style for the square class. The style should set the width and heigh to 100px. Where ID’s are denoted with ‘#’, you can identify a class using a ‘.’



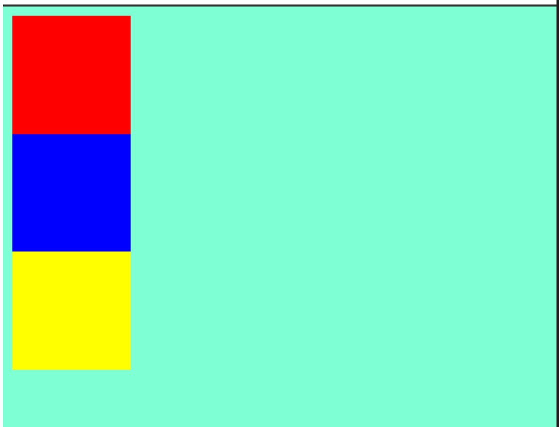
```css
.square {
    width: 100px;
    height: 100px;
}
```

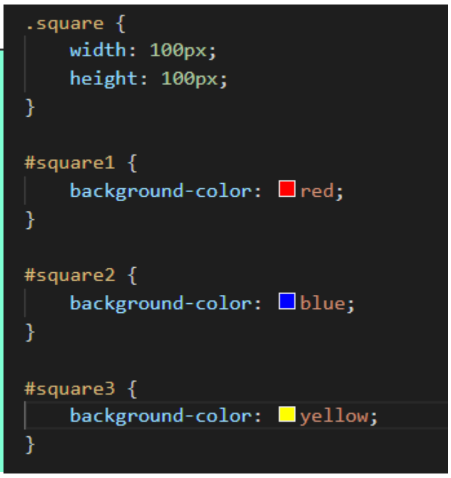
5. Set the background colour of square1 to be red, so that we can see it on screen.



```css
#square1 {
    background-color: red;
}
```

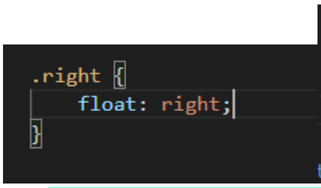
6. Add two more squares with unique IDs (square2 and square3). They can inherit the square class to set the sizing, but we will need to set their colours using their ID.



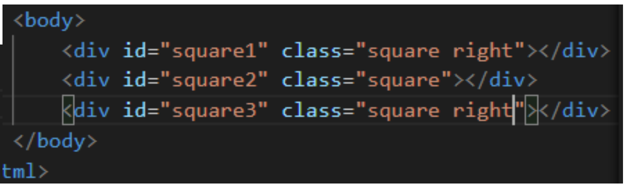


```css
.square {
    width: 100px;
    height: 100px;
}

#square1 {
    background-color: red;
}

#square2 {
    background-color: blue;
}

#square3 {
    background-color: yellow;
}
```

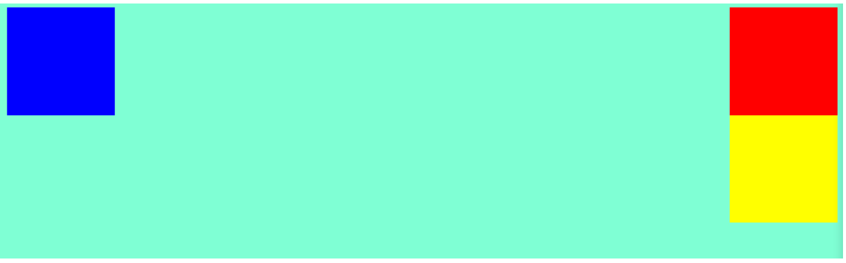
7. Change the width in the square class. All the squares update together because they all use the same class
8. Let's say that we want to move the red and yellow squares over to the right-hand side. We could add a rule in the individual squares, but that would mean repeating ourselves. Instead, we can create a new class to float objects to the right and apply that class to the relevant squares (elements can have more than one class)



```css
.right {
    float: right;
}
```

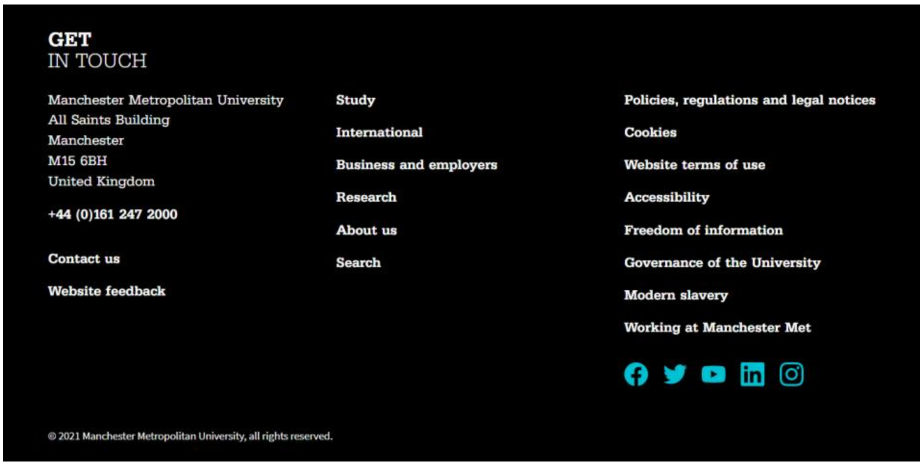


```html
<body>
    <div id="square1" class="square right"></div>
    <div id="square2" class="square"></div>
    <div id="square3" class="square right"></div>
</body>
tml>
```



4.4 Exercise 6: Nesting divs for building UIs

Quite often we need to nest multiple <div> tags inside of each other in order to build up desired UI elements. For example, consider this three-column footer on the MMU website:



GET
IN TOUCH

Manchester Metropolitan University
All Saints Building
Manchester
M15 6BH
United Kingdom

+44 (0)161 247 2000

Contact us

Website feedback

Study

International

Business and employers

Research

About us

Search

Policies, regulations and legal notices

Cookies

Website terms of use

Accessibility

Freedom of information

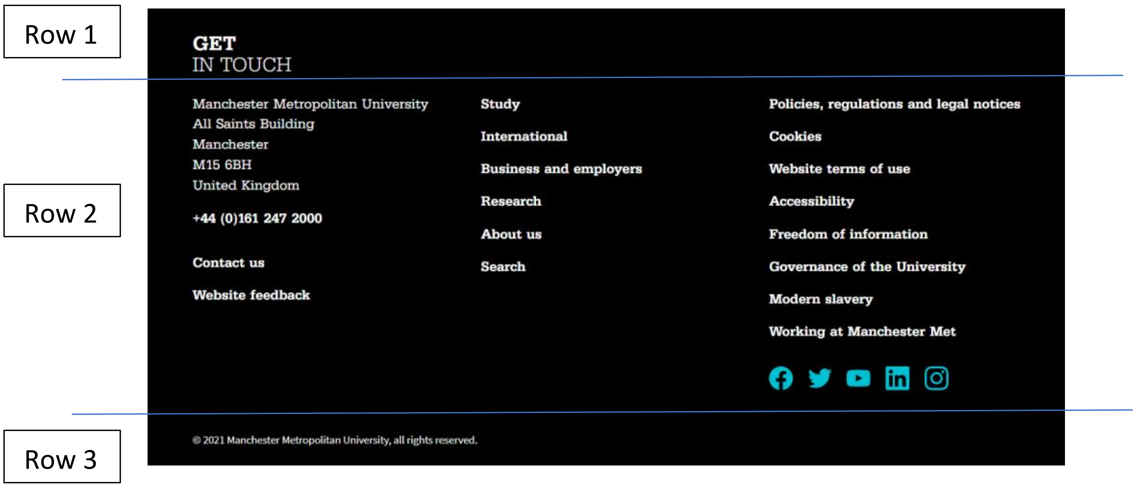
Governance of the University

Modern slavery

Working at Manchester Met

© 2021 Manchester Metropolitan University, all rights reserved.

First, it looks as though there are three rows in this footer. Each row would be a <div>:



GET
IN TOUCH

Manchester Metropolitan University
All Saints Building
Manchester
M15 6BH
United Kingdom

+44 (0)161 247 2000

Contact us

Website feedback

Study

International

Business and employers

Research

About us

Search

Policies, regulations and legal notices

Cookies

Website terms of use

Accessibility

Freedom of information

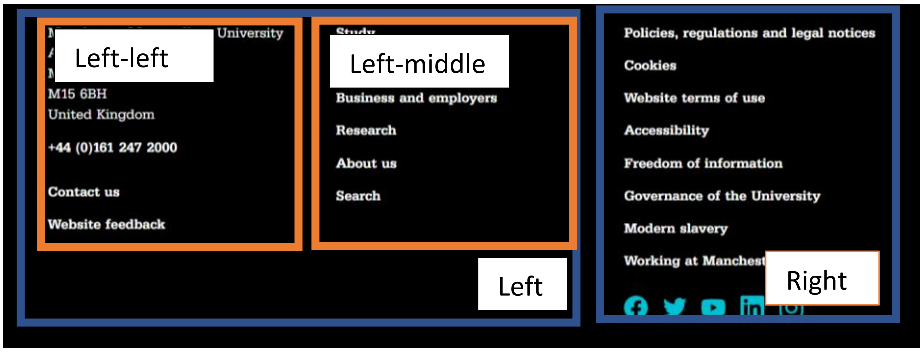
Governance of the University

Modern slavery

Working at Manchester Met

© 2021 Manchester Metropolitan University, all rights reserved.

Then row 2 has three columns. We can only float elements left or right, so one solution would be to break this row up into two divs (left and right) and then break the left div into two more divs (left-left and left-middle).

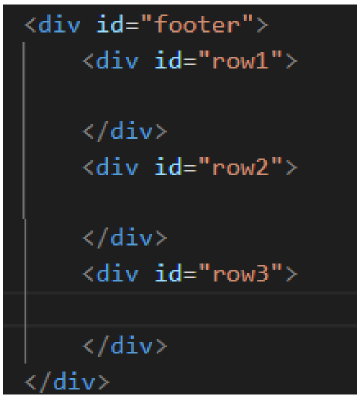


University
M15 6BH
United Kingdom
+44 (0)161 247 2000
Contact us
Website feedback
Business and employers
Research
About us
Search
Policies, regulations and legal notices
Cookies
Website terms of use
Accessibility
Freedom of information
Governance of the University
Modern slavery
Working at Manchest

Let’s create this structure in HTML and CSS:

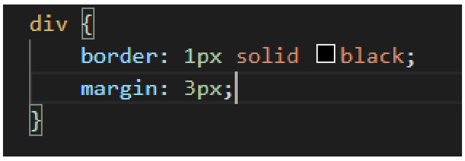
1. Create a new HTML file, add the template code
2. Create a new CSS file, link it to your HTML
3. In your HTML, create a <div> with an ID of “footer”. This will be the container for our entire footer

4. In the footer div, add three more div's. They should be given ID's to denote which row they are

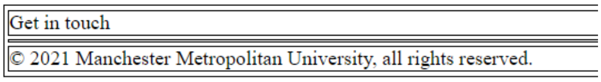


```
<div id="footer">
    <div id="row1">

    </div>
    <div id="row2">

    </div>
    <div id="row3">

    </div>
</div>
```

5. In CSS, I have just added a border and margin style on all divs so that we can see them in the browser. I've also added in some content in rows 1 and 3.



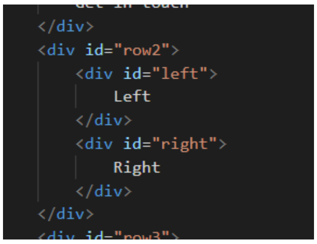
```
div {
    border: 1px solid black;
    margin: 3px;
}
```



Get in touch

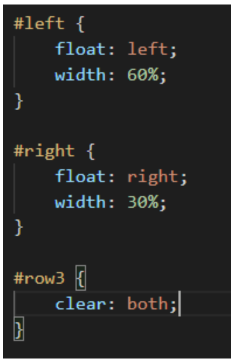
© 2021 Manchester Metropolitan University, all rights reserved.

6. In row 2, we need to add our left and right divs

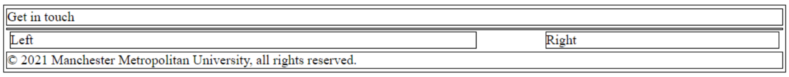


```
        Get in touch
    </div>
    <div id="row2">
        <div id="left">
            Left
        </div>
        <div id="right">
            Right
        </div>
    </div>
    <div id="row3">
```

7. For the styling, we float the left div to the left and the right to the right. We need row3 to clear both so that it doesn't appear on top of row 2. We also provide some widths.



```
#left {
    float: left;
    width: 60%;
}

#right {
    float: right;
    width: 30%;
}

#row3 {
    clear: both;
}
```

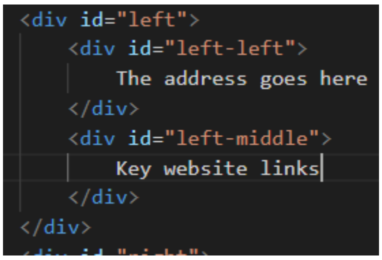


Get in touch

Left Right

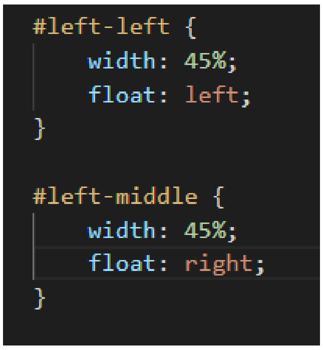
© 2021 Manchester Metropolitan University, all rights reserved.

8. That's it for the right column. It is now ready to add all of the links to policies and social media. The left column however needs breaking down further into left-left and left-middle. We start by adding the divs.



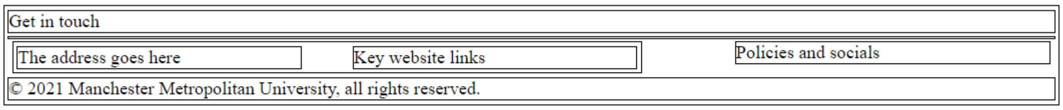
```
<div id="left">
    <div id="left-left">
        The address goes here
    </div>
    <div id="left-middle">
        Key website links
    </div>
</div>
<div id="right">
```

9. The styling works the same as before. We provide the divs with widths (note that the percentage is calculated based on the outer div (i.e. the left div is considered 100% now). We also float the divs to their relevant sides



```
#left-left {
    width: 45%;
    float: left;
}

#left-middle {
    width: 45%;
    float: right;
}
```

10. You should now see that we are set up in Chrome with the correct structure. We would next add the content and alter the styling so that the background was black, and the correct fonts and sizing is used.



Get in touch

The address goes here

Key website links

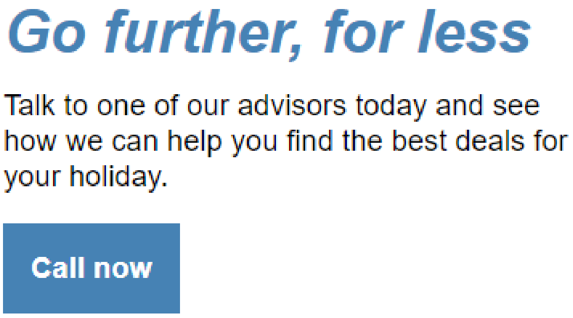
Policies and socials

© 2021 Manchester Metropolitan University, all rights reserved.

5 Putting it into practice by building complex UIs

The following exercises will bring together everything you've learnt this week. For each you will be provided with a screenshot of the finished artifact and you need to recreate them. Use the lecture slides, previous exercises, and W3Schools to help you. We will go through the answers together if there is time, or they will be provided to you at the end of the week.

5.1 Exercise 7: Recreate this UI element



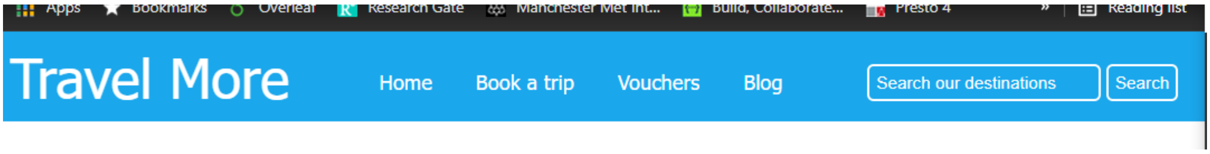
Go further, for less

Talk to one of our advisors today and see how we can help you find the best deals for your holiday.

Useful information:

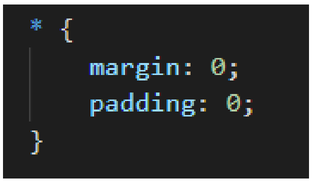
- Colour: 'steelblue'
- Total width: 350px
- Font: 'Helvetica'

5.2 Exercise 8: Recreate this UI element



Useful information:

- Colour: #1AA7EC (this is a HEX colour code; you can use it just like we have been using colours already)
- Font: Tahoma
- To push the header up to the top of the page, add the following style



```
* {
    margin: 0;
    padding: 0;
}
```

6 Expanding your CSS skills – Extension game

There are many resources on the Web for learning and improving your CSS skills. A good resource is CSSBattle (https://cssbattle.dev/). Sign up for an account and use your remaining time to work through the different levels.