

EE4216 Lab 1 – Responsive Web Design

You are required to submit a lab report to explain the results of each checkpoint with supplementary information such as screenshots and source code as appropriate. A public URL of your website must be provided for each checkpoint for the grader to verify the results.

Objectives:

- Learn to use flexible box layout and media query to create responsive web site.

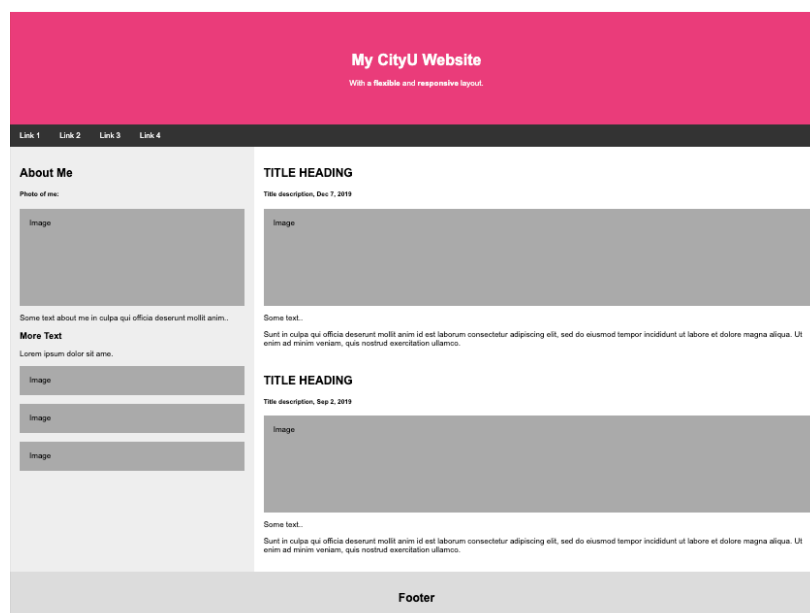
Responsive web design (RWD) is an approach to web design which makes web pages render well on a variety of devices and window or screen sizes. Content, design and performance are necessarily consistent across all devices to ensure usability and satisfaction. It has become much more important as the amount of mobile traffic now accounts for more than half of total internet traffic.

A site designed with RWD adapts the layout to the viewing environment by using fluid, proportion-based grids, flexible images, and CSS3 media queries.

- The fluid grid concept calls for page element sizing to be in relative units like percentages, rather than absolute units like pixels or points.
- Flexible images are also sized in relative units, so as to prevent them from displaying outside their containing element.
- Media queries (@media) allow the page to use different CSS style rules based on characteristics of the device the site is being displayed on, most commonly the width of the browser.

Task 1 – Building a Responsive Website Using Flexible Box Layout

Our aim is to create a website with a typical layout as follow: a header, a navigation bar, a side column, a main column, and a footer. The website will collapse to single column when viewing from a narrower screen.



1. Let's create the HTML structure first. We will decorate it with CSS later. Here is the header block where the website title or banner is usually placed.

```
12 <!-- Header -->
13 <div class="header">
14   <h1>My CityU Website</h1>
15   <p>With a <b>flexible</b> and <b>responsive</b> layout.</p>
16 </div>
```

2. Next, the navigation bar contains four hyperlinks.

```
18 <!-- Navigation Bar -->
19 <div class="navbar">
20   <a href="#">Link 1</a>
21   <a href="#">Link 2</a>
22   <a href="#">Link 3</a>
23   <a href="#">Link 4</a>
24 </div>
```

3. The content area is divided into side column and main column. We use a <div> block as a container of the two columns.

```
26 <!-- The flexible grid (content) -->
27 <div class="row">
28
29 <!-- side content and main content here -->
30
31 </div>
```

4. Then insert the side content block and main content block to the container above.

```
29 <div class="side">
30   <h2>About Me</h2>
31   <h5>Photo of me:</h5>
32   <div class="fakeimg" style="height:200px;">Image</div>
33   <p>Some text about me in culpa qui officia deserunt mollit anim..</p>
34   <h3>More Text</h3>
35   <p>Lorem ipsum dolor sit ame.</p>
36   <div class="fakeimg" style="height:60px;">Image</div><br>
37   <div class="fakeimg" style="height:60px;">Image</div><br>
38   <div class="fakeimg" style="height:60px;">Image</div>
39 </div>
41 <div class="main">
42   <div class="section">
43     <h2>TITLE HEADING</h2>
44     <h5>Title description, Dec 7, 2019</h5>
45     <div class="fakeimg" style="height:200px;">Image</div>
46     <p>Some text..</p>
47     <p>Sunt in culpa qui officia deserunt mollit anim id est laborum consectetur adipiscing elit,
sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis
nostrud exercitation ullamco.</p>
48   </div>
49
50   <br>
51
52   <div class="section">
53     <h2>TITLE HEADING</h2>
54     <h5>Title description, Sep 2, 2019</h5>
55     <div class="fakeimg" style="height:200px;">Image</div>
56     <p>Some text..</p>
57     <p>Sunt in culpa qui officia deserunt mollit anim id est laborum consectetur adipiscing elit,
sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis
nostrud exercitation ullamco.</p>
58   </div>
59 </div>
```

5. Last, the footer block usually contains some links to relevant resources and important information such as copyright notice and disclaimer.

```
63 <!-- Footer -->
64 <div class="footer">
65   <h2>Footer</h2>
66 </div>
```

6. Now let's beautify the website. Because different browsers have different default style settings. It is a good idea to reset the body margin to zero before applying any other rules. The box-sizing rule tells the browser to account for any border and padding in the values you specify for an element's width and height. If you set an element's width to 100 pixels, that 100 pixels will include any border or padding you added, and the content box will shrink to absorb that extra width. This typically makes it much easier to size elements.

```
1 * {
2   box-sizing: border-box;
3 }
4
5 /* Style the body */
6 body {
7   font-family: Arial;
8   margin: 0;
9 }
```

7. Here the style of the header is set.

```
11 /* Header/logo Title */
12 .header {
13   padding: 60px;
14   text-align: center;
15   background: #EC407A;
16   color: white;
17 }
```

8. Note that flexible box layout (display: flex) is used here to enable fluid positioning and proportion-based element sizing. The text decoration of the hyperlinks is removed to make them look more like buttons.

```
19 /* Style the top navigation bar */
20 .navbar {
21   display: flex;
22   background-color: #333;
23 }
24
25 /* Style the navigation bar links */
26 .navbar a {
27   color: white;
28   padding: 14px 20px;
29   text-decoration: none;
30   text-align: center;
31 }
32
33 /* Change color on hover */
34 .navbar a:hover {
35   background-color: #ddd;
36   color: black;
37 }
```

9. Similarly, flexible box layout is applied to the content container.

```
39 ▾ /* Column container */
40 ▾ .row {
41     display: flex;
42 }
```

10. The two content columns sitting next to each other share the space on a 3:7 ratio.

```
45 ▾ /* Sidebar/left column */
46 ▾ .side {
47     flex: 30%;
48     background-color: #f0f0f0;
49     padding: 20px;
50 }
51
52 ▾ /* Main column */
53 ▾ .main {
54     flex: 70%;
55     background-color: white;
56     padding: 20px;
57 }
```

11. Without using real image, a grey box is used to represent the images.

```
59 ▾ /* Fake image, just for this example */
60 ▾ .fakeimg {
61     background-color: #aaa;
62     width: 100%;
63     padding: 20px;
64 }
```

12. Here is the footer style which is quite similar to that of the header.

```
66 ▾ /* Footer */
67 ▾ .footer {
68     padding: 20px;
69     text-align: center;
70     background: #ddd;
71 }
```

13. For devices with smaller screen width, it is better to apply a column-oriented layout so that users can browse the content in a single direction (up and down only). The following flexbox rule makes the two columns stack on top of each other instead of next to each other when the screen is less than 700px wide.

```
74 ▾ @media screen and (max-width: 700px) {
75     .row, .navbar {
76         flex-direction: column;
77     }
78 }
```

14. Now you have all the codes to create a responsive website. Let's write it up and test it.

Checkpoint 1:

Demonstrate the responsiveness of your website's layout. Show the two different layouts for screen width larger than 700px and smaller than 700px respectively.

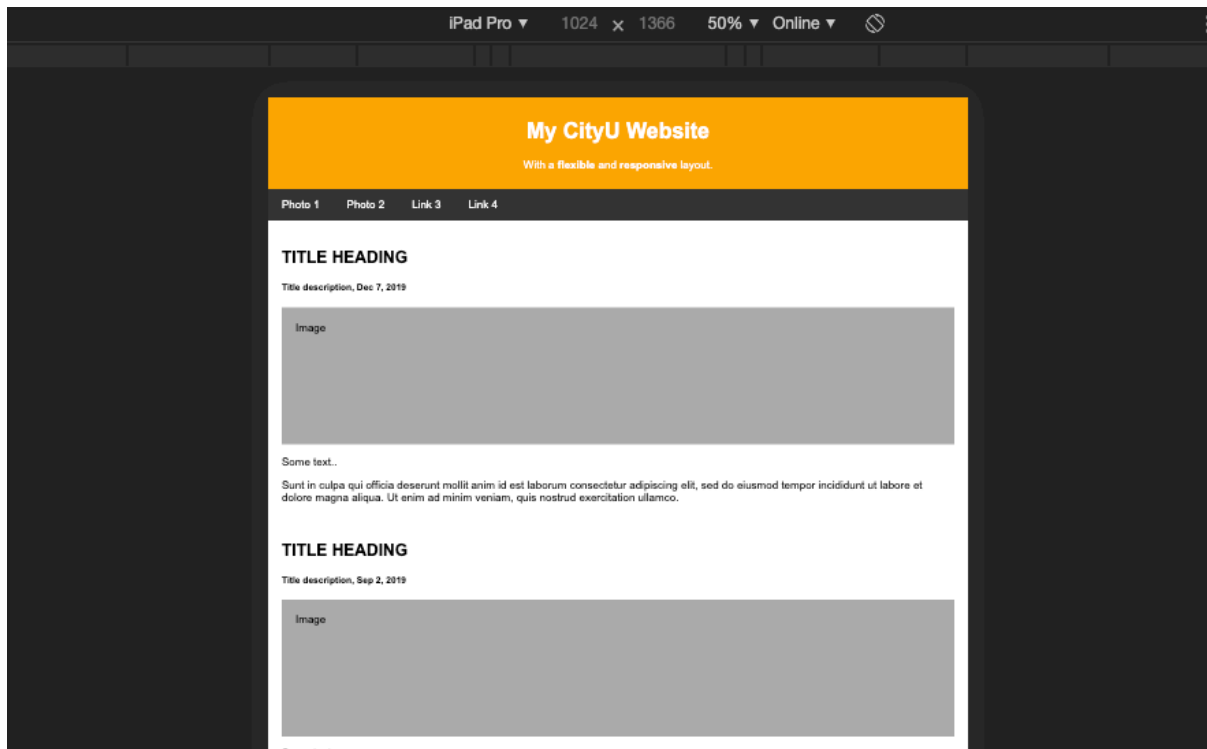
Checkpoint 2:

Replace the <div> block with corresponding HTML5 semantic elements such as header, nav, section, aside and footer. Explain the revised HTML code of your website.

Checkpoint 3:

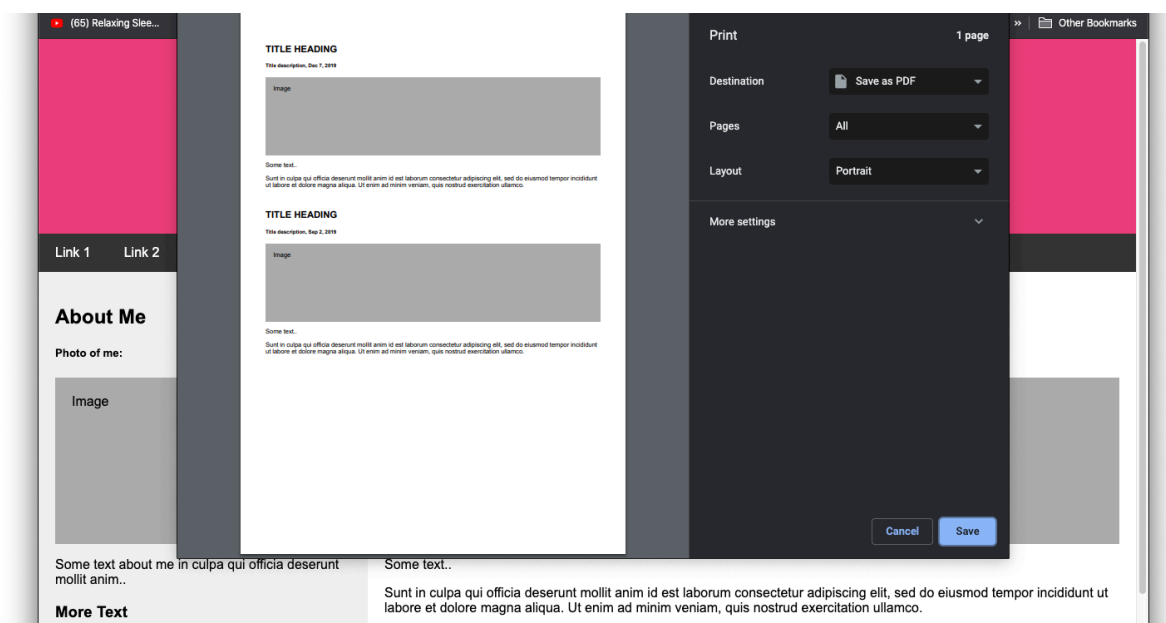
Rewrite the previous media query to apply the following changes when the website's orientation is portrait (i.e. height > width). Explain the revised media query.

- Placing the side column under the main column
- Changing the header's background colour to orange
- Reducing the header's padding to 10 px



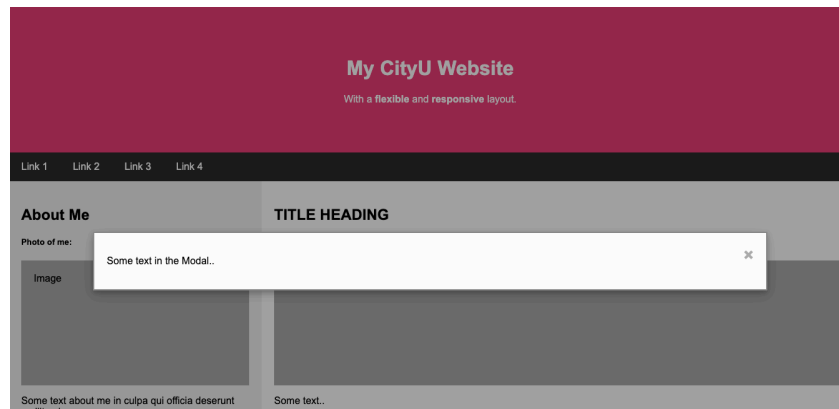
Checkpoint 4:

Design another media query such that only the main content is visible for printing. Explain the new media query.



Task 2 – Creating and Animating Modal

A modal is a dialogue box that pops up to display notifications, alerts, and handle register and login form. It is a very useful web UI component and can be made easily with custom CSS and simple scripts.



1. Again, we need to create the HTML structure for the modal box first. A modal box typically has two layers that the modal content is shown on top of an overlay. To make it available to the DOM, it can be inserted after the footer.

```
68 <!-- The Modal -->
69 <div id="myModal" class="modal">
70 <!-- Modal content -->
71 <div class="modal-content">
72 <span class="close">&times;</span>
73 <p>Some text in the Modal..</p>
74 </div>
75 </div>
```

2. The modal box should be hidden at the beginning (display: none). We will write script to control how to display it. Here is the style of the overlay - a full-screen semi-transparent layer shading the details of the page.

```
99 /* The Modal (background) */
100 .modal {
101   display: none; /* Hidden by default */
102   justify-content: center;
103   align-items: center;
104   position: fixed; /* Stay in place */
105   z-index: 1; /* Sit on top */
106   left: 0;
107   top: 0;
108   width: 100%; /* Full width */
109   height: 100%; /* Full height */
110   background-color: rgba(0,0,0,0.4); /* Black w/ opacity */
111 }
```

3. And here is the style of the modal content.

```
113 /* Modal Content/Box */
114 .modal-content {
115   background-color: #fefefe;
116   padding: 20px;
117   width: 80%;
118   border: 2px solid #888;
119   box-shadow: 0 4px 8px 0 rgba(0,0,0,0.2), 0 6px 20px 0 rgba(0,0,0,0.19);
120 }
```

- Then, move the close button to the right (float: right) and make it visually clickable.

```

130 /* The Close Button */
131 .close {
132     color: #aaa;
133     float: right;
134     font-size: 28px;
135     font-weight: bold;
136 }
137
138 .close:hover {
139     color: black;
140     text-decoration: none;
141     cursor: pointer;
142 }

```

Checkpoint 5:

Write a script to display the modal box upon clicking the navigation links. You should also implement the close button function for dismissing the modal box. (Hint: toggle the *display* property of the modal box between *flex* and *none*.) Explain your code.

Checkpoint 6:

Modify the CSS to create a fade-in animation effect for the modal box. Explain your code.

- Add two hyperlinks to the navigation bar. You may use any other royalty-free photos from Flickr instead.

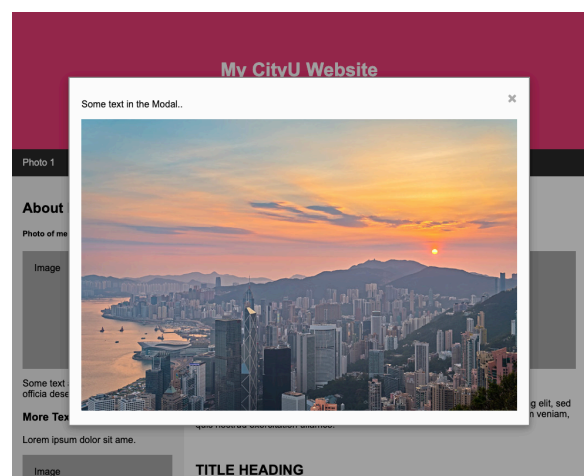
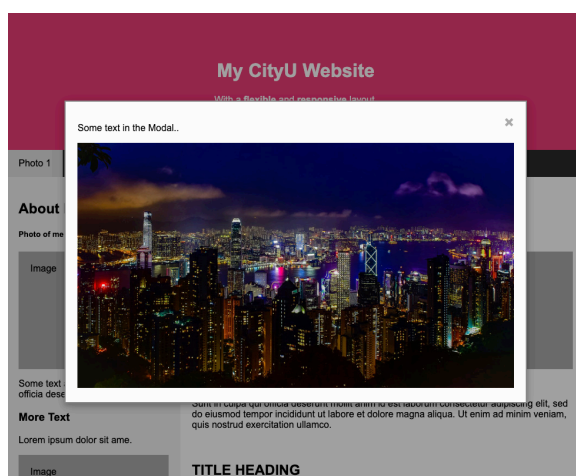
```

18 <!-- Navigation Bar -->
19 <div class="navbar">
20   <a href="https://live.staticflickr.com/1728/42736240511_91cd425b59_c_d.jpg">Photo 1</a>
21   <a href="https://live.staticflickr.com/1945/45582150001_7bda8a2ee0_c_d.jpg">Photo 2</a>
22   <a href="#">Link 3</a>
23   <a href="#">Link 4</a>
24 </div>

```

Checkpoint 7:

Modify your script such that the photos are displayed on the modal box by clicking the photo links. Explain your code.



END OF THE LAB