COM1008: Web and Internet Technology

Dr. Steve Maddock,
s.maddock@sheffield.ac.uk

Exercise sheet 3: A solution

The following presentation discusses a possible solution for Exercise Sheet 3. It is not the only solution.

1. Create a Web page that has four areas: a header, a navigation area, a main content area and a footer. The four areas should be wrapped in a <div id="wrapper"> element. Random text can be added to each of the areas. For the navigation area, add three random pieces of text to represent three links, but do not add anchor tags to these. For the main content area, add a header and three paragraphs of random text. Do not write the CSS yet.

Figure 1 gives a solution.

2. Attach a CSS reset or normalize to your Web page. You choose which. This will change the display of your HTML file.

The following line is added to the head of the HTML file in figure 1:

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

3. Now create your own CSS file to style the page. Set the width of the wrapper to 320px, i.e. in a CSS file: #wrapper { width: 320px; margin: 0 auto; }

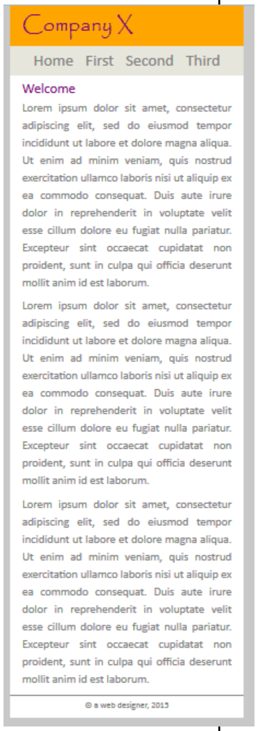
Figure 2 gives an example CSS file.

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="utf-8" />
  <title>Company X</title>
</head>

<body id="home">

<div id="wrapper">

  <header>
    <h1>Company X</h1>
  </header>

  <nav>
    <ul>
      <li>Home</li>
      <li>First</li>
      <li>Second</li>
      <li>Third</li>
    </ul>
  </nav>

  <main>
    <h1>Welcome</h1>
    <p>Lorem ipsum dolor …</p>
    <p>Lorem ipsum dolor …</p>
    <p>Lorem ipsum dolor …</p>
  </main>

  <footer>
    <p>&copy; a web designer, 2015</p>
  </footer>

</div> <!-- wrapper -->

</body>

</html>
```

Figure 1: A solution for exercise 1

```
/* ~~~general tags~~~~~~~~~~ */

body {
  background-color: rgb(200,200,200);
  color: gray;
  font-family: Calibri, sans-serif;
  line-height: 1.5;
  text-align: justify;
}

h1, h2 {
  color: purple;
}

h1 {
  font-size: 120%;
}

h2 {
  font-size: 100%;
}

p {
   margin-bottom: 0.5em;
}

/* ~~~wrapper~~~~~~~~~~~~~~~~ */

#wrapper {
  width: 320px;
  margin: 0 auto;
  background-color: white;
}

/* ~~~header~~~~~~~~~~~~~~~~~~~~ */

header {
  padding: 1% 5%;
  background-color: orange;
}
```

```
header h1 {
  padding: 1% 0;
  color: purple;
  font-family: Papyrus, serif;
  font-size: 1.8em;
}

/* ~~~navigation~~~~~~~~~~~~~~~~ */

nav {
  padding: 1% 5%;
  background-color: rgb(230,230,220);
  color: gray;
  font-size: 1.4em;
  text-align: center;
}

nav ul {
  padding: 0px;
  list-style: none;
}

nav ul li {
  display: inline;
  padding: 2%;
}

/* ~~~main content~~~~ */

main {
  padding: 1% 5%;
}

/* ~~~footer~~~~~~~~~~~~~~~~~~ */

footer {
  clear: both;
  font-size: 75%;
  text-align: center;
  border-top: 1px solid gray;
  padding: 3px 5%;
}

/* ~~~end~~~~~~~~~~~~~~~~~~~~*/
```



Welcome

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

© a web designer, 2015

Figure 2: A solution for exercise 3 with a screenshot of the resulting display

4. Change the width of the wrapper area so that it is expressed as a %, e.g. width: 100% instead of a fixed width of 320px. Your page will grow to the size of the browser window and the text display should now be ‘liquid’ as the browser window is resized, by which I mean it reflows to fit the available space. However, undoubtedly the display looks better for the 320px width it was originally designed for rather than the large desktop display. Some media queries will now be used to address this. But first, add a min-width to the wrapper:
   #wrapper { min-width: 320px; width: 100%; margin: 0 auto; }
   This will stop the wrapper becoming less than 320px in width.

This is a straightforward change and results in a resizeable window.

5. Add a media query so that aspects of the design change when the width of the browser window is greater than or equal to 750px. For example, add some extra CSS rules to this media query so that the size of the header and its text is bigger. Perhaps change the width of the wrapper to 80%.

Figure 3 lists the rules that are added to the css file. You should compare this with Figure 2 to try and predict what the visual changes would be before you display the file that is included with this solution.

```
@media screen and (min-width: 750px) {
  #wrapper {
    width: 80%;
  }
}

/* ~~~~~~~~*/

@media screen and (min-width: 750px) {
  header {
    padding: 1% 10%;
  }

  header h1 {
    font-size: 4em;
    line-height: 1.5;
  }
}

/* ~~~~~~~~*/

@media screen and (min-width: 750px) {
  nav {
    padding: 1% 10%;
  }
}

/* ~~~~~~~~*/

@media screen and (min-width: 750px) {
  main {
    padding: 1% 10%;
  }
}

/* ~~~~~~~~*/

@media screen and (min-width: 750px) {
  footer {
    padding: 1% 10%;
  }
}
```

Figure 3: A solution for exercise 5

6. Add a media query so that aspects of the design change when the width of the browser window is greater than or equal to 970px. Introduce an extra column to the right hand side of the main content area. You can choose to do this in one of two ways: (i) use float; (ii) use flexbox. Then make sure this extra content is at the bottom of the main content area for browser window widths of less than 970px. Figure 1 gives the general layout for the two designs. (There is a nice article on RWD patterns at https://developers.google.com/web/fundamentals/layouts/rwd-patterns/?hl=en. Also, Firefox describe use of the flexbox at https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_Flexible_Box_Layout/Using_CSS_flexible_boxes)

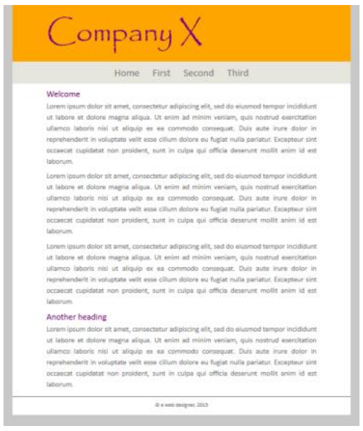
(i) float

The HTML file needs to be changed slightly to make two columns inside the main element, as shown in Figure 4. The accompanying CSS file is also changed, as shown in Figure 4. The first rule says that each column is 100% width. This will produce the display on the left hand side of Figure 5. The riles in the @media statement cause the columns to float to the left and right respectively when the screen width is greater than or equal to 970px wide. The full width of the two columns is width of left column content + width of padding + width or right column content = 55+5+40 = 100% of the parent element. I know there is no extra width as all margins and padding were set to 0 by the CSS reset.

```
<main>
  <div id="left_column">
    <h1>Welcome</h1>
    <p>Lorem ipsum ...      </p>
    <p>Lorem ipsum ...      </p>
    <p>Lorem ipsum ...      </p>
  </div>
  <div id="right_column">
    <h1>Another heading</h1>
    <p>Lorem ipsum ...      </p>
  </div>
</main>
```

```
#left_column, #right_column {
  width: 100%;
}

@media screen and (min-width: 970px) {
  #left_column {
    float: left;
    width: 55%;
    padding-right: 5%;
  }

  #right_column {
    float: right;
    width: 40%;
  }
}
```

Figure 4: A solution for exercise 6a



Welcome

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

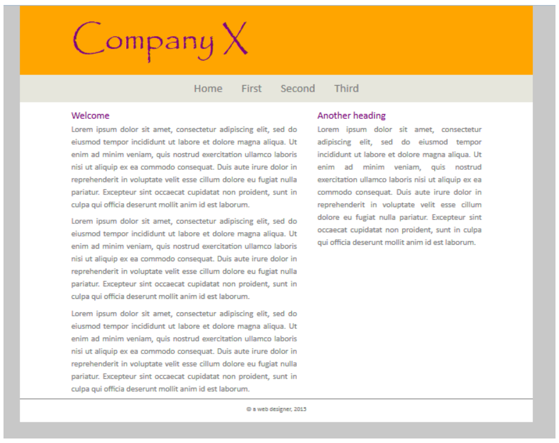
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Another heading

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

© a web designer, 2015



Welcome

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Another heading

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

© a web designer, 2015

Figure 5: Screen shots for exercise 6a

Some of you may have experienced a slightly different display when you floated the two columns. For example, the footer may have moved above the floated elements. The reason for this is the way that floated elements behave. If a container contains only floats, as my main element does in the example in Figure 4, then it effectively has no height – the height collapses to nothing. The following page explains the problem and possible solutions: https://css-tricks.com/all-about-floats/

To solve the problem, I chose to add 'clear: both' to the CSS rule for the footer – check the footer rule at the bottom of figure 2. This states that the footer cannot have any floated elements on either side of it, so it is therefore forced to appear after the floats.

(ii) flexbox

The same HTML as for the solution in '(i) float' is used. The CSS file changes are shown in Figure 6. Note that the main element (the parent container) is set to "display: flex". The initial direction, which is for our mobile screen is a column of boxes. The left and right columns are set to 100% width and will stack on top of each other.

When the screen size is greater than or equal to 970px the direction of the boxes contained in the main element are set to display in a row. There is now no need for float to be used for the left and right columns.

```
main {
  display: flex;
  flex-direction: column;
  padding: 1% 5%;
}

#left_column, #right_column {
  width: 100%;
}

@media screen and (min-width: 750px) {
  main {
    padding: 1% 10%;
  }
}

@media screen and (min-width: 970px) {
  main {
    flex-direction: row;
  }

  #left_column {
    width: 55%;
    padding-right: 5%;
  }

  #right_column {
    width: 40%;
  }
}
```

Figure 6: A solution for exercise 6b

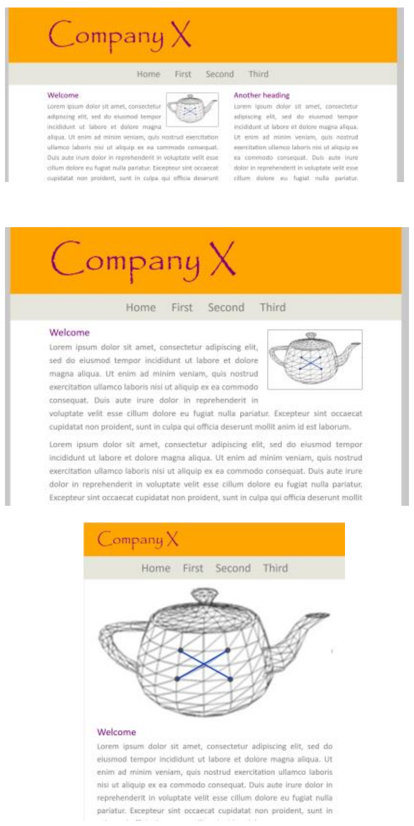
7. Set a max-width of 1170px for the wrapper. This will mean that for very high resolution monitors, all that will happen when the browser window gets bigger than 1170px is that the margin either side of the wrapper will grow larger. The arrangement of the contents of the wrapper will remain the same and be centred in the browser window. This is currently an accepted practice for Web site design, rather than produce extra layout designs for larger and larger displays.

Add “max-width: 1170px;” to the rule for the wrapper.

8. Add an image to the page. At a width of 320-749px, the image should be as wide as the wrapper. For larger page sizes it should occupy only a small proportion of the width of the main element (or, more likely, a proportion of the width of the left hand column for the two column display shown in Figure 1). You choose values that look right.

An image is first added to the HTML file. I’ve put it just before the h1 element in the left column, as shown in Figure 7. The relevant CSS is also shown in Figure 7, together with some screen shots for different display widths.

```
<main>
  <div id="left_column">
    <figure>
      <img src="./images/tpot1.jpg" alt="a teapot" />
    </figure>
    <h1>Welcome</h1>
```



```
Main   figure img {
  max-width: 100%;
  width: 100%;
  height: auto;
}

main figure {
  width: 100%;
}

@media screen and (min-width: 750px) {
  main {
    padding: 1% 10%;
  }

  main figure {
    float: right;
    width: 30%;
    margin: 10px 0 10px 3%;
    border: 1px solid gray;
  }
}
```

Figure 7: A solution for exercise 8

9. Test the web page using Google’s mobile-friendly test: https://www.google.com/webmasters/tools/mobile-friendly/

You need to copy the full website to your mypublic_html folder before you can use Google’s test, as you have to supply a URL on a web server.

10. Progressively enhance the Web site with some CSS3 features. For example, add a text-shadow effect and a transition or transformation effect to one or more of the elements.

This depends on which CSS3 you have used. The solution should be clear when the effect is added, and there are example in the lecture notes and previous exercise sheets that can be used.

11. Now duplicate the page three times and add links to each of the pages in the navigation area.

Since the template for one page has now been completed, multiple pages should be straightforward. The main extra is to add the anchor tags to the navigation bar list item elements. This has been illustrated in numerous previous examples.

12. Add extra items to the navigation area so that there are now seven links. You don’t have to create extra pages – just link to the same pages you have already created; that will suffice for the purposes of this exercise. The layout of the navigation area may now ‘break’ (look wrong) when the browser window width becomes small or the items in the navigation area will wrap, increasing the size of the navigation area so that it dominates the small screen size. This is a common problem when laying out navigation areas and different solutions have been proposed. The article at http://responsivenavigation.net/index.html presents a range of these approaches. Read this article and decide which approach you favour. Then try to implement that approach.

The solution to this exercise depends on which approach you have used from those at the above URL.