Lab 6 – CSS layout

Aims:

- Understand and apply the CSS Box model.
- Apply page layout techniques to a Web page
- Implement a Web page that maintains legibility under scaling
- Implement conditional CSS stylesheets for responsive design (using CSS media queries)

Task 1: Create a more advanced layout with CSS file

This task will not be assessed by your tutor, but you may find it useful for your assignment.

In this task you will apply some simple CSS styling to a Web page. This HTML file is in **Lab06.zip** and is called **lab06task1**.**html**. Put it in your **lab061** folder on your local machine.

With CSS, we will layout the HTML (**lab06task1**.**html)** to look something like below:

Using Notepad++ or similar editor, create a new text file called **main.css** and save it in **lab061/styles**. Add a comment header to the top of your CSS file similar to the following, replacing the text in italics:

```
/*
filename: your name author: your name
created: enter date last
modified: enter date
description: html files it refers to (if known) */
```

Create a link to this file from your web page **lab06task1**.**html** by adding a reference to the external stylesheet ***main.css*** in the <head> element.

Notice the files also reference an external font which will be imported.

Below is a CSS files using some of the techniques we coved in the Task 1. Review the CSS rules below to make sure you understand what they are doing, and then type or cut-paste them into your **main.css** file.

```css
/* main.css */
body {font-family: arial;}
article{
    max-width: 1220px;
    margin: 0 auto;
}
header {
    background-image: url(images/background.png);
    background-repeat: repeat-x;
    background-position: bottom;
    height: 158px;
    width: 100%;
}
header h1 {
    font-family: 'Love Ya Like A Sister', arial;
    letter-spacing: 4px;
    font-size: 72px;
    padding-top: .5em;
    padding-left: .5em;
}
nav ul {
    margin: 0;
    padding: 0;
    }
nav ul li {
    list-style: none;
    float: left;
    font-size: 93%;
}
nav ul li a {
    display: block;
    font-weight: bold;
    padding: .625em 3.5em; /* 10px 15px */
    text-decoration: none;
    color: #000;
    border-right: 1px solid #bababa;
}
img { max-width: 100%}
#section1, #section2, #section3 {
    float: left;
    width: 30%;
    text-align: justify;
    margin: 1em 1em 1em 1em;
}
footer { clear: both; }
footer h3 {
    font-family: 'Love Ya Like A Sister', arial;
    background-color: #94d4ff;
    padding: .5em 0 .5em .5em;
    background-repeat: repeat-x;
    background-position: bottom;
}
```

Font imported in HTML file

What is this doing?

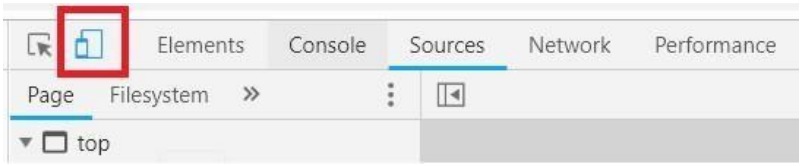
Task 2: Create a responsive layout(Optional)

In this task you will look at some of the problems that can occur when pages design for a desktop are displayed on smaller screens. You will then create some responsive CSS.

How can I test my webpage designs to see how responsive they are, without switching back and forth across devices?

Use the ***"Developer tools"*** available in Firefox and Chrome. Then each of these tools has a mode that shows how a web page will look on different size screens.

- In Firefox menu, select "Web Developer" then select the *"Responsive Design Mode" item*.
- In Google Chrome menu, select "More tools" then select "Developer tools". To activate device mode, click the device icon in the top left corner of the Developer Tools window.



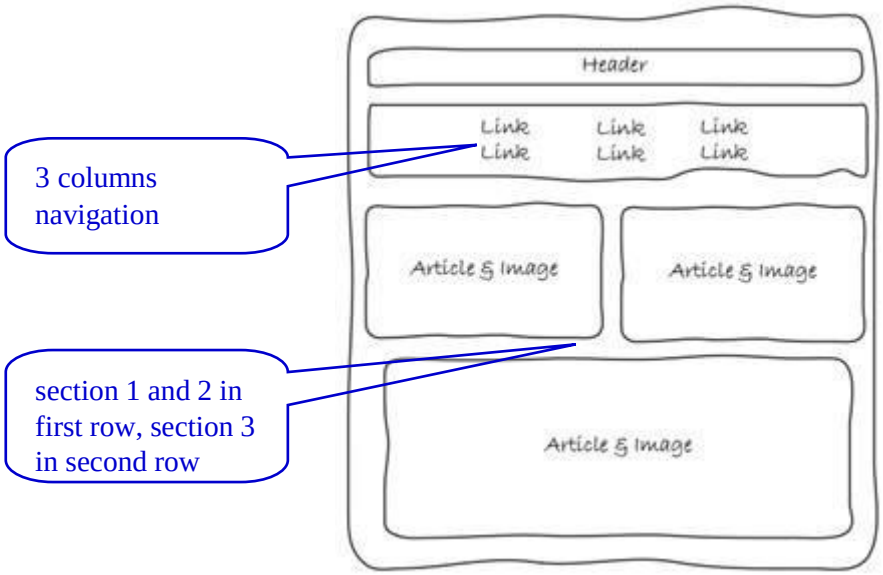
Configure the device resolution and test the result.

It is recommended that you use developer tools in Firefox or Chrome. The device emulation tool in Internet Explorer 11's developer tools window is more difficult to use for this purpose than either Firefox or Chrome.

Let's first check the Web page using a Responsive Web Developer Tool. We can identify the problems regarding responsiveness when we select various screen sizes as follows:
Select ***Medium Display (maximum display width is 1024 px, like iPad, Tablet, etc.)***:

The Web page will be something like below:

We can identify some problems here like navigation wraps, section columns are too narrow and misplaced, too much passive (unintentional) white space, and unexpected image size. We could design a mock-up for the above identified problems. The design could be something like below:



To implement a responsive design that can change to the above layout when the screen sizes changes we should follow the following steps:

Step 1: Setting the viewport scale to 1.0

The viewport is equal to the size of the browser window. The viewport on handheld devices is much narrower than a desktop/laptop browser window. The narrower viewport causes problems with responsive Web pages. Thus, you should include the following meta tag in the <head> element to the HTML file **lab06task1.html**, in order to set the viewport scale 1.0.

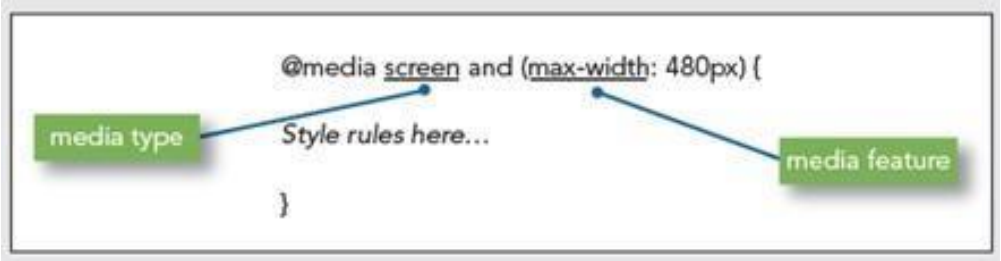
```
<head>
      <!-- Viewport set to scale 1.0 -->
      <meta name="viewport" content="width=device-width, initial-
       scale=1.0"/>

      <!-- References to external basic CSS file -->
      <link href= "styles/main.css" rel="stylesheet"/>
</head>
```

Step 2: Apply CSS Media Query

You should apply styles based on display device characteristics using CSS media query. A media query is an expression that lets you create precise rules for destination media/device. It contains both a ***media type*** and optional expressions that check conditions called ***media features***. Media features include ***variables*** such as the width or height of the destination device.

An example media query is shown below where the media type is ***screen*** and the media feature is ***max-width*** (the max-width value is set to 480 px). The max-width value is called a ***breakpoint*** and it is best measured in ems, because they are flexible (1 ***em*** equals 16 ***px,*** in most browsers when the default medium font is selected.).



Create a link to the HTML file **lab04task2.html** by adding a reference to the external responsive stylesheet ***responsive.css*** in the <head> element as follows:

```
<head>
      <!-- Viewport set to scale 1.0 -->
      <meta name="viewport" content="width=device-width, initial-
       scale=1.0"/>

      <!-- References to external basic CSS file -->
      <link href= "styles/main.css" rel="stylesheet"/>

      <!-- References to external responsive CSS file -->
      <link href="responsive.css" rel="stylesheet" media="screen and (max-
      width: 1024px)"/>
</head>
```

Create ***responsive.css*** file with the appropriate header in your **lab04/styles** folder then add the following CSS rules.

```
/* responsive.css  */
/* media query for display: over 500 */

@media screen and (min-width: 31.25em) {

            nav ul li {
                  width: 33%;
            }

            #section1 {
                  width: 43%;
            }

            #section2 {
                  width: 43%;
            }

            #section3 {
                  clear: both;
                  float: none;
                  width: 90%;
                  margin: 0 1em 1em 1.5em;
            }

            #section3 img {
                        float: left;
                        width: 30%;
                        margin: .5em 1em 1em 0;
            }

            article {
                        min-width: 600px;
            }
}
```

media query

3 columns navigation

section 1 and 2 set to 43%, i.e., in first row

section 3 sets to 90%, i.e., in second row

section 3 image wraps with content

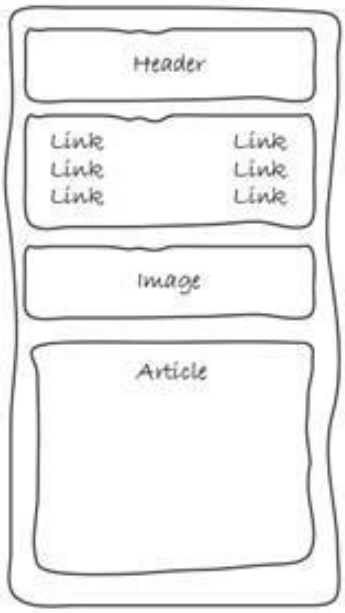
entire article sets to display window

Medium display: Your responsive HTML should now look something like:

Now let's create a design for a ***Small Display (maximum display width is 568 px)***:

The Web page will be something like below if we resize it to mobile size (:

The mock-up design and some hints have been given. The mock-up design will be something like below (***2 columns navigation and 1 column section***):



Create a separate CSS file for a small display like Apple iPhone 5.

Add the following conditional media query in your ***responsive.css*** file (complete the following CSS file and check your responsive Web page using Firefox Responsive Developer Tool).

```
/*  media query for display: under 500 px  */
@media screen and (max-width: 31.25em ) {
                header h1 {
                        font-size: 48px;
                        padding: 0;
                        text-align: center;
                }
                nav ul li {
                        width: ?
                }
                #section1, #section2, #section3 {
                        margin: 0;
                        float: none;
                        width: ?
                }
                img {
                  display: block;
                  margin: 0 auto;
                }
                article{
                                min-width: ?
                }
}
```

responsive content

What should this value be?

Small display (Apply iPhone 5):

Validate your HTML and CSS using the appropriate validators.

Want some more practice with CSS layout? If you are interested in exploring further here are some other tutorials:

http://learnlayout.com/

https://www.codecademy.com/courses/web-beginner-en-6merh/4/1?curriculum_id=50579fb998b470000202dc8