Tutorial 2

Editor – Brackets
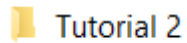
Goals

Create a website showcasing the following techniques
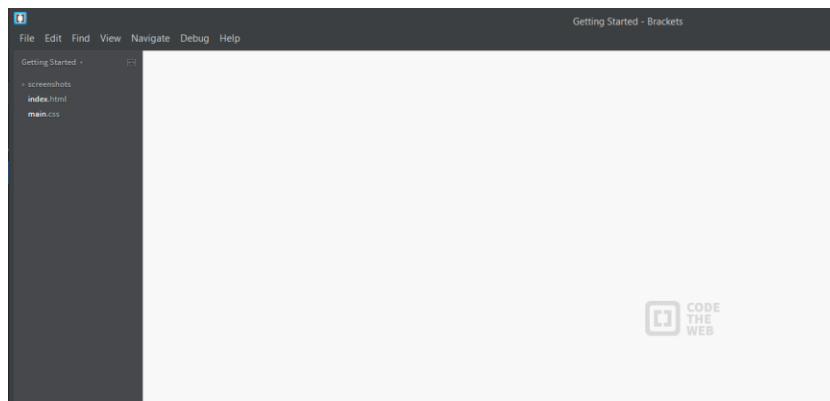
- Content switch with Javascript

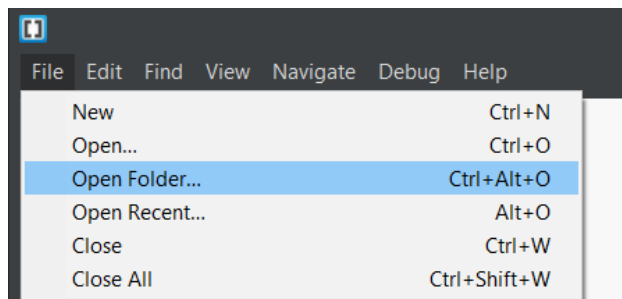## Website

- Create a folder on the desktop called tutorial 2
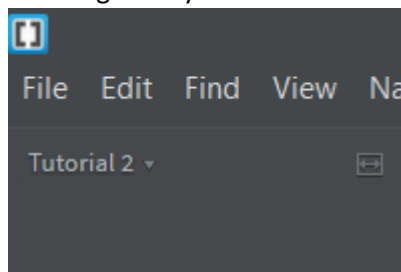
    - 📁 Tutorial 2
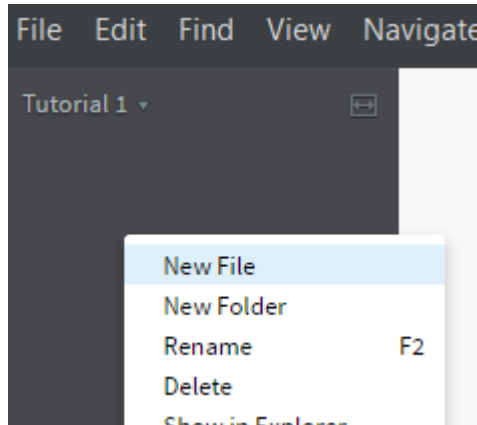
- Open Brackets

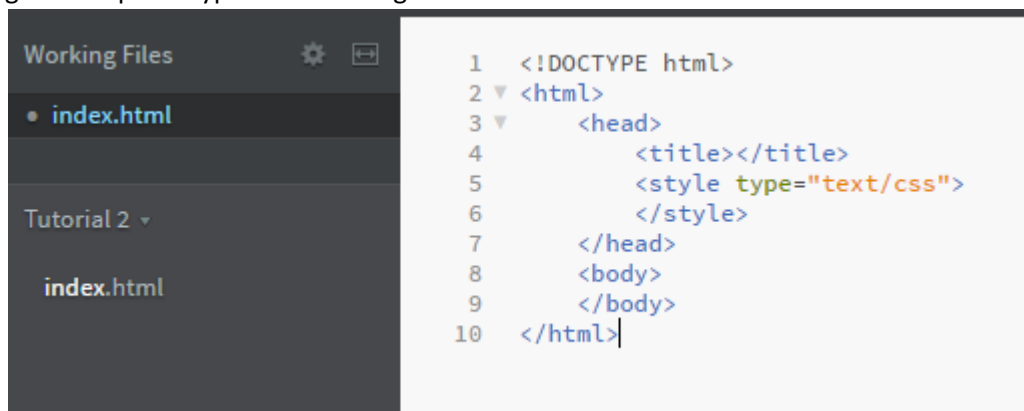- Open the tutorial 2 folder in brackets

- Click Select image and you should see the following folder structure:

- Next we need to create a html file. Right click below the tutorial folder and select new File

- Name the file index.html
- In the right hand panel type the following

- The above is the structure of any webpage, and we can duplicate this page to build any other page.
- Right click under index.html and create new folder, call it images

-
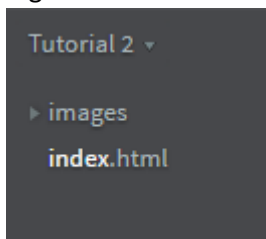
We will store a couple of images here in a bit.

Index Page

Open up index.html, we will be working on this to create the page.

- Add the text 'tutorial 2' in between the <title></title> syntax.

- Save the file (Ctrl + S or use the menu File-> Save).

Now we will build the following looking webpage:
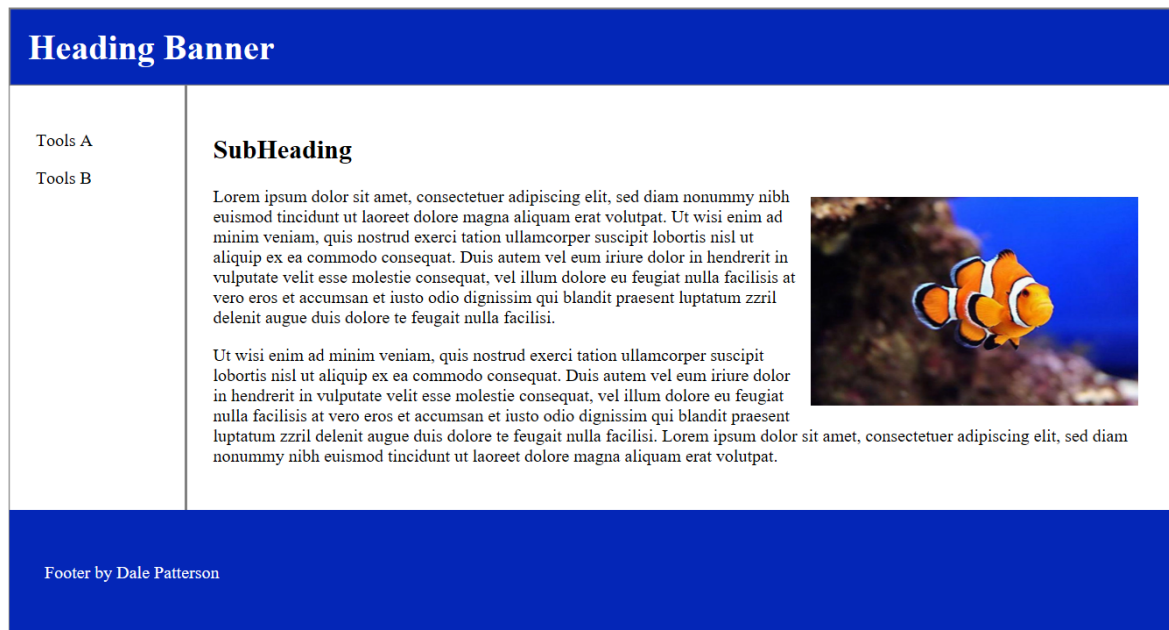
Take this design and break it into boxes like last week

-
- This box will hold the main content, so we shall call it content.
- As such, we now have 6 boxes to place on the page.
    - o Container
    - o Header
    - o Main
    - o Footer
    - o Nav
    - o Content

Now that we have the design, we need to start creating the page. This is done by creating a div tag (box) and then assigning it a colour. So you can see it in the browser.

Use the following div code:
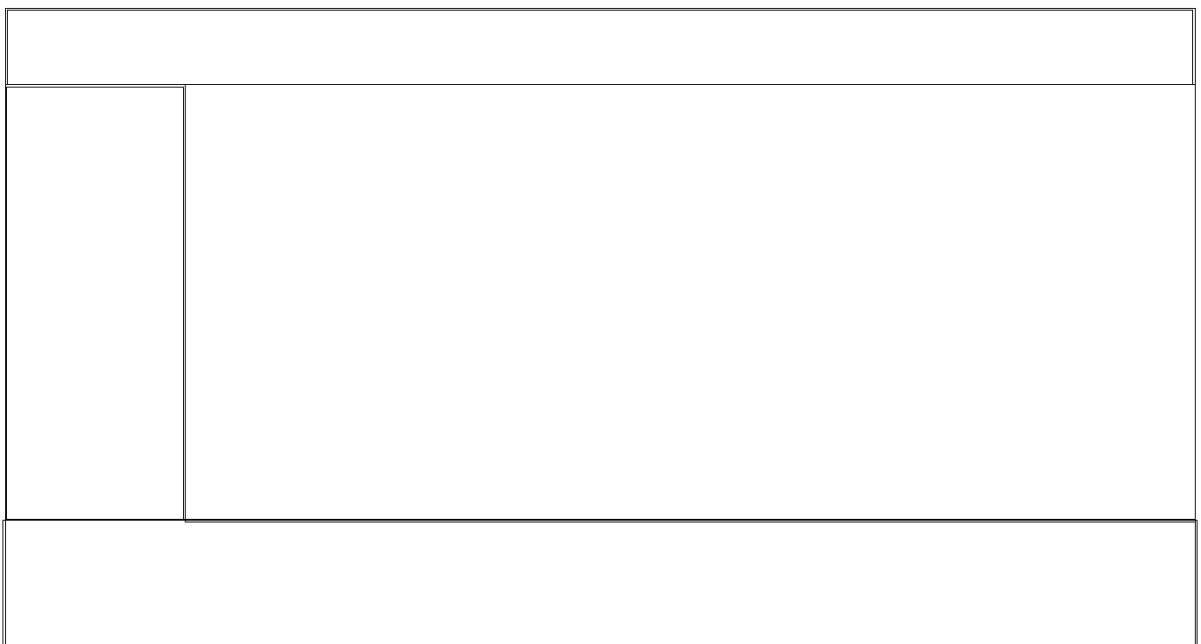
```html
1   <!DOCTYPE html>
2 ▼ <html>
3 ▼    <head>
4          <title>Tutorial 2</title>
5          <style type="text/css">
6          </style>
7      </head>
8 ▼    <body>
9 ▼        <div id="container">
10             <div id="header"></div><!-- eo header -->
11 ▼          <div id="main">
12                 <div id="nav"></div><!-- eo nav -->
13                 <div id="content"></div><!-- eo content -->
14             </div><!-- eo main -->
15             <div id="footer"></div><!-- eo footer -->
16         </div><!-- eo container -->
17     </body>
18 </html>
```

This will give the structure we need. Next, we need to add the Css code to be able to see what is going with the div boxes.

We will do the css in stages and test after each change. We do it in the following order:

- Container
- Header
- Main
- Footer
- Nav
- Content

All Css code goes inbetween the <style></style> tags. There are 3 ways to use css, inline, embedded and external. Inline means it occurs on the line of html code, embedded means it is located between the <style> tags and external, is where we put all of the styles into a separate page that we link to. At the moment, using embedded is the easiest way to go.

CSS code for container:

```
<style type="text/css">
    #container{width: 90%; min-height: 600px; margin: 0 auto; background-color: lime; border: 1px solid gray;}
</style>
```

Produces

Now, for the next 3 rows, the header, footer and Main. From here on out, you can select your own colour choice, remember you can select your colours via hex code #000000 (black) through to #FFFFFF (white); or you can use the name of the colour such as black, white, lime, red…. There are a couple of other methods such as RGB(0,0,0) or HSLA (0,0,0,0) but we will discuss those later.

Css code for header, main and footer:

```
<style type="text/css">
    #container{width: 90%; min-height: 600px; margin: 0 auto; background-color: lime; border: 1px solid gray;}
    #header{width: 100%; height: 100px; background-color: aliceblue; }
    #main {width: 100%; min-height: 400px; background-color: white;}
    #footer{width: 100%; height: 100px; background-color: aliceblue;}
</style>
```

Produces
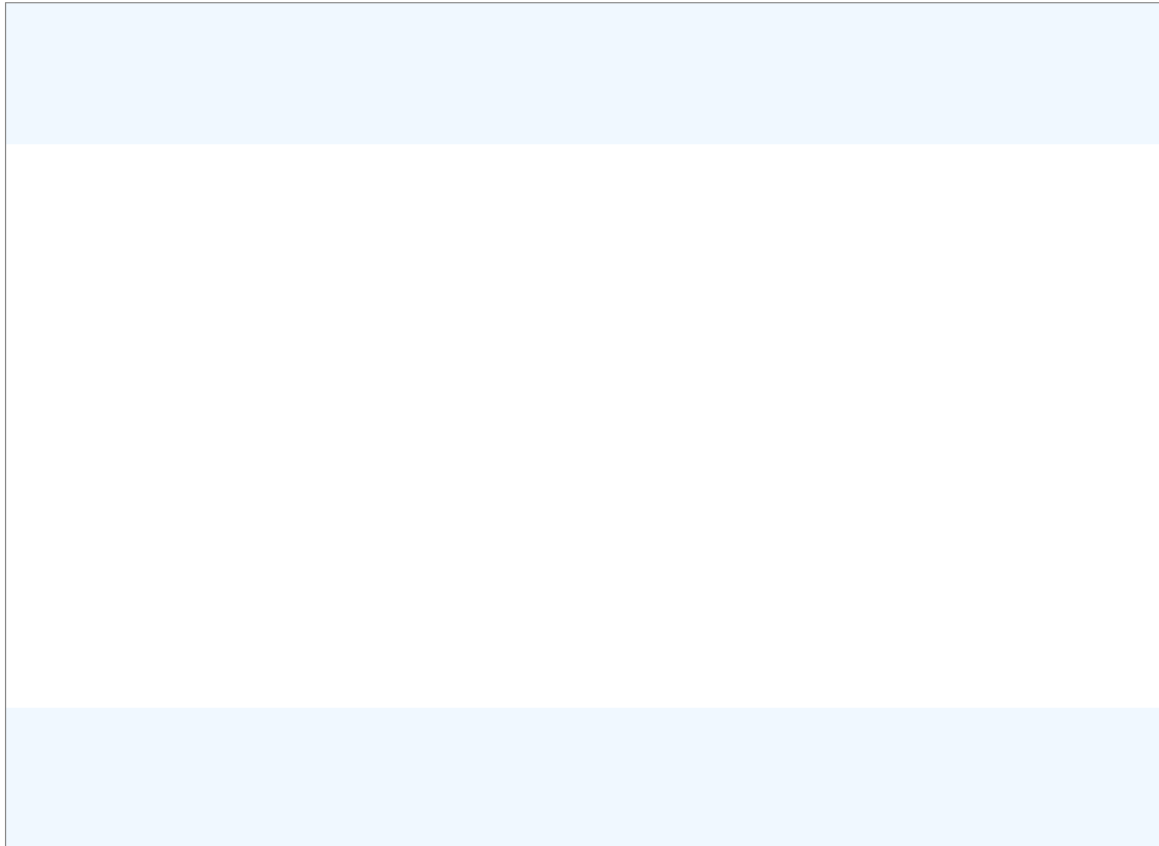
Next we have the two internal boxes, nav and content. These divs require the use of the float command, float allows for columns to exist in web pages. The way they work is that all columns have the float left command except for the very last, which has float right. So, if you have 5 columns, you would have 4 with float left assigned to them and the last with float right. At the moment, we only have the two columns.

Css for Nav and Content

```
<style type="text/css">
    #container{width: 90%; min-height: 600px; margin: 0 auto; background-color: lime; border: 1px solid gray;
    #header{width: 100%; height: 100px; background-color: aliceblue; }
    #main {width: 100%; min-height: 400px; background-color: white;}
    #footer{width: 100%; height: 100px; background-color: aliceblue;}
    #nav{width: 19%;min-height: 400px; float: left; background-color: cornsilk;}
    #content{width: 80%; min-height: 400px; float: right; text-align: left;}
</style>
```

Produces
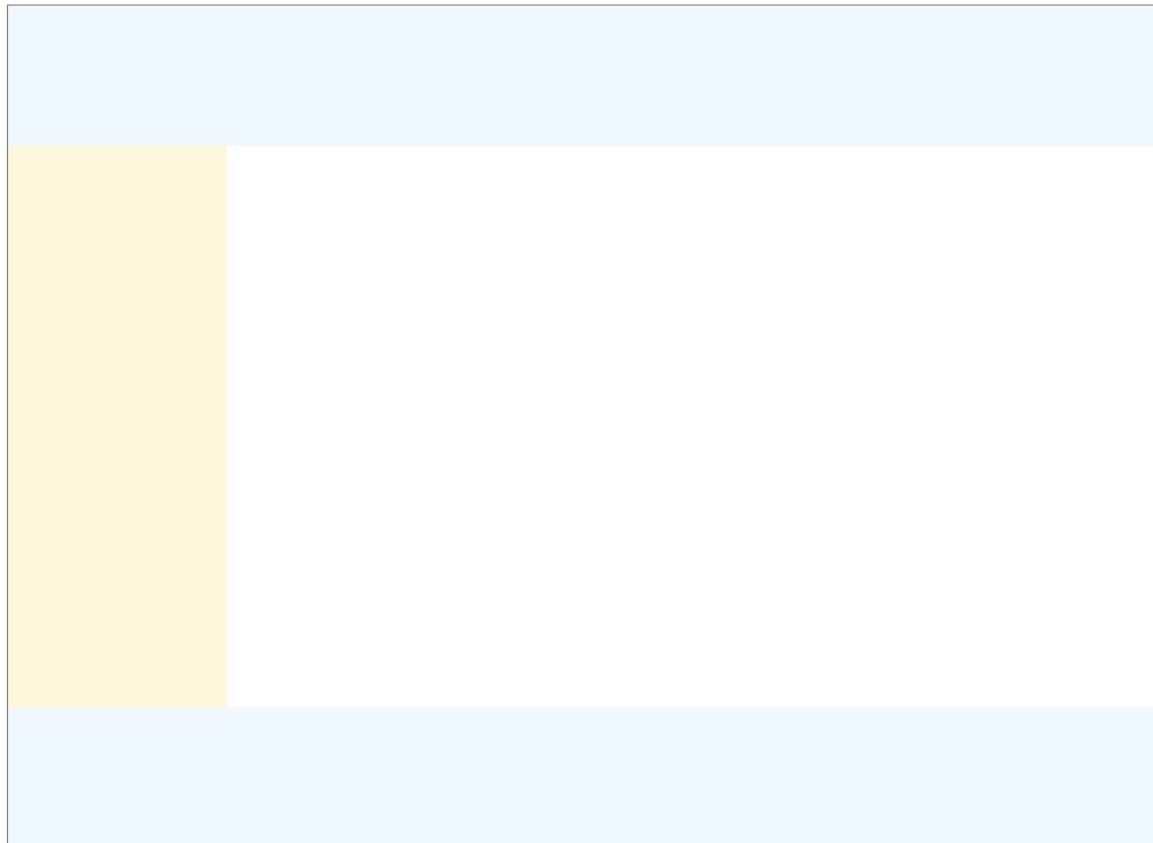
Remove the background colour of the container.

```
tle>Tutorial 2</title>
yle type="text/css">
 #container{width: 90%; min-height: 600px; margin: 0 auto; border: 1px solid gray;}
 #header{width: 100%; height: 100px; background-color: aliceblue; }
 #main {width: 100%, min height: 400px, background color: white;]
```

Now, we will remove the nav background colour and add some content to the site. Go grab 2 images off the net and some lorem ipsum text. Save the images into the images folder. If you are following along with the video tutorial, this is where you can grab the clownfish and swordfish images.

For this document, I will grab batman and superman images. As for the filler text, remember you can grab lorem ipsum text, in this case, I will use batman ipsum.

First, the paragraph text and image.

Div code that goes inside the div content

```html
<div id="content">
    <img src="images/batman.jpg" alt="batman" class="imgSize" id="mainImage">
    <p>Every year, I took a holiday. I went to Florence, this cafe on the banks of the Arno. Every fine e
    anything to me, nor me to you. But we would both know that you've made it, that you were happy. I nev
    <p>Batman may have made the front page but Bruce Wayne got pushed to page eight.</p>
    <p>You care about justice? Look beyond your own pain, Bruce. This city is rotting.</p>
    <p>Master Wayne, you've been gone a long time. You look very fashionable. Apart from the mud.</p>
</div><!-- eo content -->
```

Now the css. Notice, the css on the image line and removal of background colour on the nav line.

```css
#footer{width: 100%; height: 100px; background-color: aliceblue;}
#nav{width: 19%;min-height: 400px; float: left;}
#content{width: 80%; min-height: 400px; float: right; text-align: left;}
.imgSize{height: 200px; padding: 10px; float: right;}
style>
```

This will produce
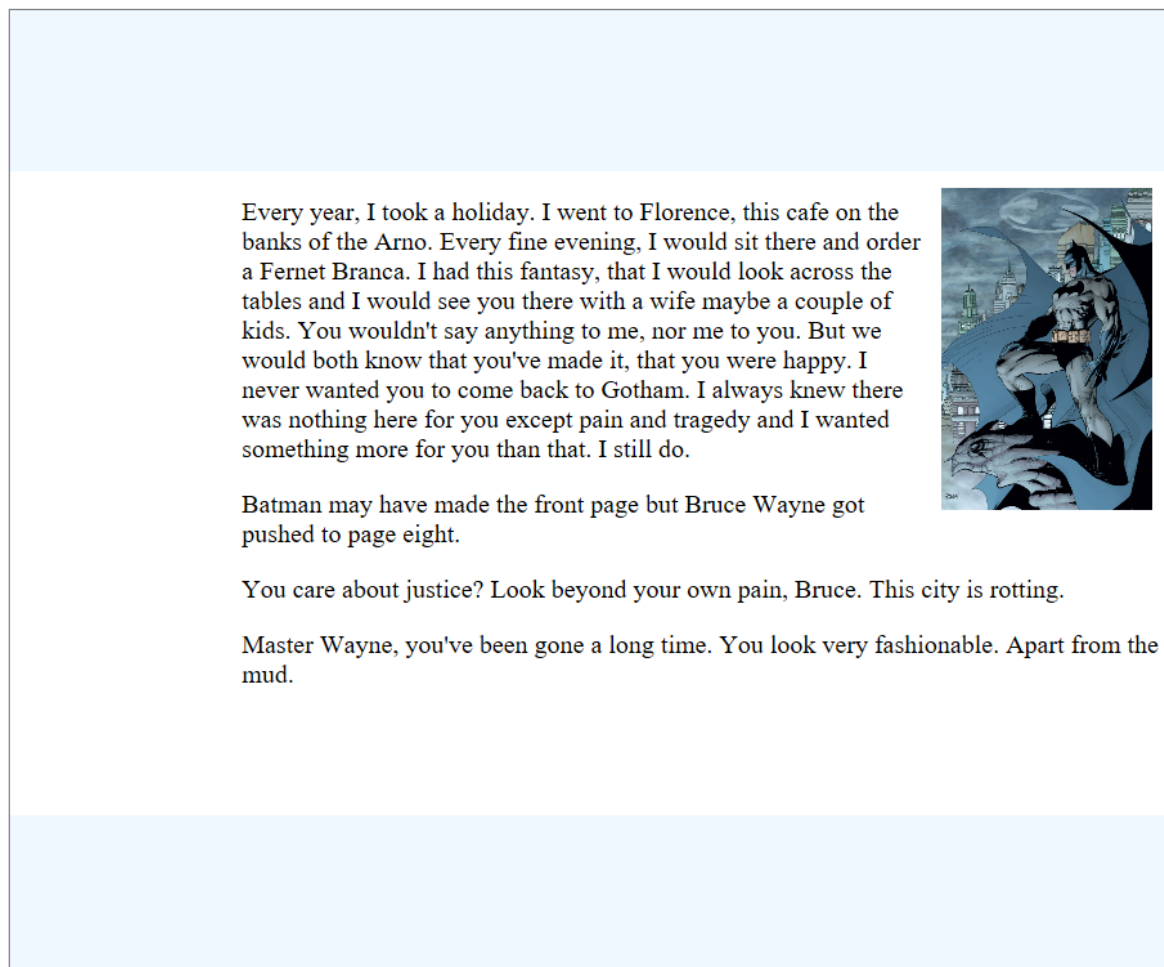
Next we will quickly add information into the header, footer and Nav section of the page

Div code for header

```
iv id="container" >
  <div id="header"><span class="headerText">Batman / Superman</span></div><!-- eo header -->
  <div id="main">
```

Div code for footer

```
</div><!-- eo main -->
  <div id="footer"><span class="footerText">&copy;2017  Your Name</span> </div><!-- eo footer -->
iv><!-- eo container -->
```

Div code for Nav

```
iv id= main >
  <div id="nav">
      <p class="navText">Click One</p>
      <p class="navText">Click Two</p>
  </div><!-- eo nav -->
```

Css code for Header, Footer and Nav

```
  .headerText{font-size:3em; padding: 10px;}
  .footerText{font-size: .7em; float: right; padding-top: 80px; padding-right: 20px;}
  .navText{padding: 10px;}
```

Produces

Now we play with JavaScript.

JavaScript Basics

Here we will test the basics of JavaScript, you will learn to look at the console to help determine and track the events that you program with Javascript.

To start with, we will create the javascript section on the page. This goes in the head section of the code.

Code:

```
            .navText[pt
        </style>
        <script>
        </script>
    </head>
    <body>
```

Now, to make javascript work, we can use the onClick command. The browser will recognise that an element on the page has been clicked and will search for the matching javascript function. So, we will write a very simple javascript function that we will then apply to the Click one content in the nav section of the page.

Javascript function

```
<script>
    function clickOne()
    {
        console.log("*** Element has been clicked *** \n");
    }
</script>
```

Applying the function to the page element

```
<div id="nav">
    <p class="navText" onClick="clickOne()">Click One</p>
    <p class="navText">Click Two</p>
</div><!-- eo nav -->
```

Save the page and open/refresh the page in your browser, open up the console

- Edge/Internet Explorer – Push F12
- Chrome – Ctrl + Shift + J
- Firefox – Ctrl + Shift + K

I am using edge, when I load the page and view the console, if I click on the Click One text in the Nav section, I see the following appear in the console

```
index.html
i Current window: /C:/Users/Sean/Desktop/Griffith%202017/Adv%20Web/Tutorial%202/index.html
  *** Element has been click ***


  index.html (20,17)
```

This shows that click event has been noticed by the browser and it has responded to it.

Removing content using JavaScript

Now that you have seen JavaScript in action ,its time to make it do something, in this case, we will remove the content layer from the page. To do this we will set the style of the content to make the div box disappear.

We will rewrite the ClickOne function. As we are changing the function, we do not have to modify the html that calls Javascript as it is the same one as before. Note, whenever using Javascript, ensure you have the console open to view any errors that might occur.

Javascript code

```
function clickOne()
{
    console.log("*** Element has been clicked *** \n");
    var contentDiv = document.getElementById("content");
    contentDiv.style.display = "none";
}
```

Produces
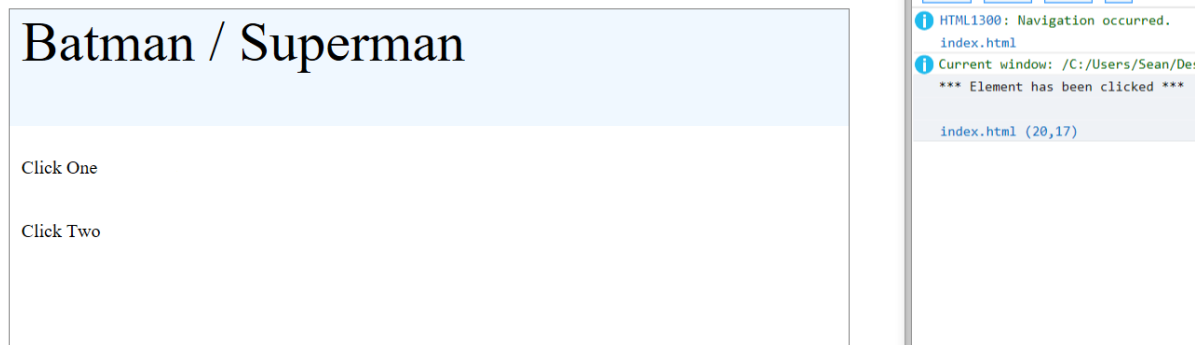
As you can see, by leaving in the console.log command, not only do I see that the content div has been removed, by it has been recorded in the console window as a successful element click.

Now that we know how to remove elements from a page, we need to learn how to show elements of code. To do so, we will re-write the javascript by adding another function that we will then hide one set of content and then show a new set of content.

To do this, we will first have to create a new set of content, in this case I will use the standard lorem ipsum and the superman pic. So, we are designing code to swap between the two differing sets of content.

Swapping Content with JavaScript

Start by making additional content. In this case we will generate a new div under the old div of content and call it content2. Once you have created the code and put it in view the page.

Div code

```
<div id="content">
    <img src="images/batman.jpg" alt="batman" class="imgSize" id="mainImage">
    <p>Every year, I took a holiday. I went to Florence, this cafe on the banks of the Arn
    anything to me, nor me to you. But we would both know that you've made it, that you we
    <p>Batman may have made the front page but Bruce Wayne got pushed to page eight.</p>
    <p>You care about justice? Look beyond your own pain, Bruce. This city is rotting.</p>
    <p>Master Wayne, you've been gone a long time. You look very fashionable. Apart from t
</div><!-- eo content -->
<div id="content2">
    <img src="images/superman.png" alt="Superman" class="imgSize" id="mainImage2">
    <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Quisque at leo lacus. Done
    a, interdum volutpat dui. Ut imperdiet lorem est. Etiam feugiat, elit eget tincidunt c
    <p>Sed ullamcorper leo quis leo mattis, at egestas eros luctus. Vivamus efficitur erat
    velit. Praesent mollis sem sapien, a molestie augue porttitor in. Sed non tempor velit
    condimentum dolor, ac commodo quam risus vitae libero. </p>
</div><!-- eo content2 -->
```

Produces

As you can see, content2 doesn't have the same styling that content has, as such, we need to make the content2 style.

Css for content2

```
    #content2{width: 80%; min-height: 400px; float: right; text-align: left;}
/style>
```

Produces
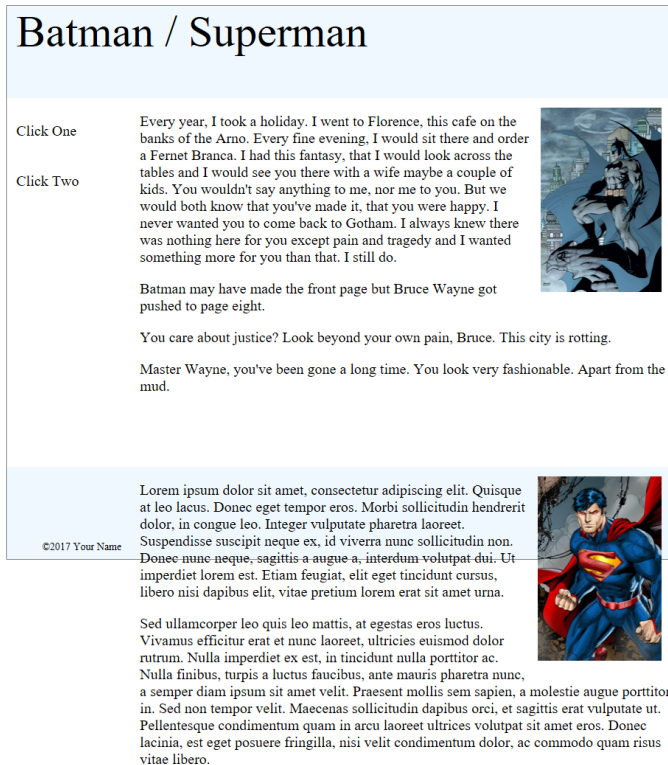
As you can see, content2 now has the styling, but it just falls into the page, from here, we need to modify more css and add some extra javascript to control what happens.

To start with, let's hide the content2 div.

Css to hide content2

```
    .navText{padding: 10px;}
    #content2{width: 80%; min-height: 400px; float: right; text-align: left; display: none;}
</style>
```

We need to modify the Click buttons Click One and Click two to be more user friendly and access javascript functions.

Div Code

```
<div id="nav">
    <p class="navText" onClick="clickOne()">Superman</p>
    <p class="navText" onClick="clickTwo()">Batman</p>
</div><!-- eo nav -->
<div id="content">
```

Starting Javascript Functions

```
<script>
    function clickOne()
    {
        console.log("*** Element One has been clicked *** \n");
        var contentDiv = document.getElementById("content");
        contentDiv.style.display = "none";
    }
    function clickTwo()
    {
        console.log("*** Element Two has been clicked *** \n");
    }
</script>
```

Now test and watch the console. You should see when Superman is clicked that the content disappears, but the console states that Element one has been clicked, and when you click on Batman, you should see the console record that Element two has been clicked.

Batman / Superman

Superman

Batman

```
HTML1300: Navigation occurred.
    index.html
Current window: /C:/Users/Sean/Desktop/Griffit
*** Element One has been clicked ***

    index.html (21,17)
*** Element Two has been clicked ***

    index.html (27,17)
```

Next we look into the functionality of what we are trying to do, each onClick function will hide one content box and show a new one. We can do this as two separate style modifications that occur one after the other in a function.

Javascript Show and Hide content

```
<script>
    function clickOne()
    {
        console.log("*** Element One has been clicked *** \n");
        var contentDiv = document.getElementById("content");
        var content2Div = document.getElementById("content2");
        contentDiv.style.display = "none";
        content2Div.style.display = "block";
    }
    function clickTwo()
    {
        console.log("*** Element Two has been clicked *** \n");
        var contentDiv = document.getElementById("content");
        var content2Div = document.getElementById("content2");
        contentDiv.style.display = "block";
        content2Div.style.display = "none";
    }
</script>
```

Examining the functions, the function clickOne will remove all of the content that is stored in the div layer called content and show all of the content that is stored in the div content2. Function clickTwo does the opposite, it hides the div content2 and shows the div content.

If you run it up and view the console, you should see the content swapping on the page and the functions activating in the console.

JavaScript Menu

In this part of the tutorial we will start a new page with a new layout.

Examine the following layout
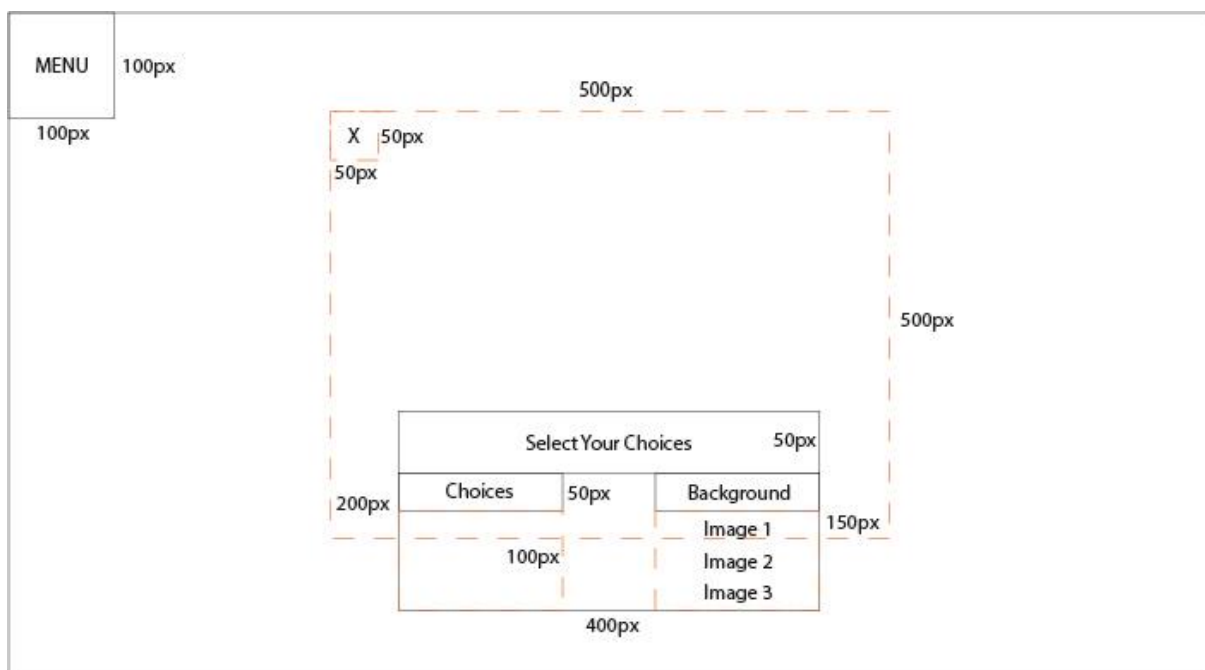
This design consists of using absolute positioning. But it is where we will work in the Javascript to control the menu and various other elements of the page.

To start with, create a new page called page2.html and a folder called css. We will be storing our stylesheet in the external method, that was discussed earlier. Also create a new file called stylesheet.css inside the css folder.

The new files and folder should look like this:

Tutorial 2 ▾

▾ CSS

   stylesheet.css

▸ images

index.html
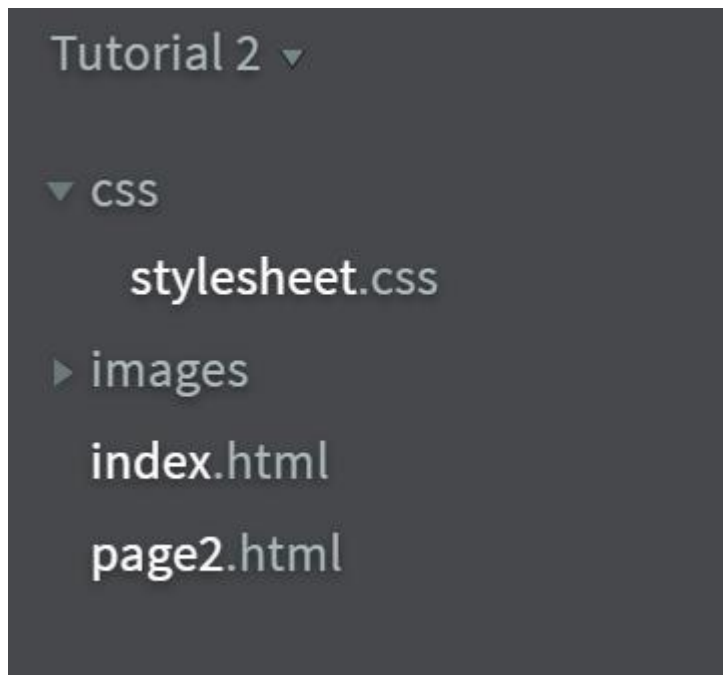
page2.html

To start with create the basic layout of the page in page2.html

```
1    <!DOCTYPE html>
2 ▾ <html>
3 ▾     <head>
4             <title>Page 2</title>
5             <style type="text/css">
6             </style>
7             <script>
8             </script>
9         </head>
10        <body>
11        </body>
12    </html>
```

Now we create the div boxes

```html
<body>
<div id="menuBtn"></div><!-- eo menuBtn -->
<div id="navMenu">
    <div id="closeBtn" ></div><!-- eo closeBtn -->
</div><!-- eo navMenu -->
<div id="box">
    <div id="bTop"></div><!-- eo bTop -->
    <div id="bBottom">
        <div id="choicesBtn">
            <div id="dDownChoices"></div><!-- eo dDownChoices -->
        </div><!-- eo choicesBtn -->
        <div id="backgroundBtn">
            <div id="dDownBackground">
            </div><!-- eo dDownBackground -->
        </div><!-- eo backgroundBtn -->
    </div><!-- eo bBottom -->
</div><!-- eo box -->
</body>
```

Add the images from the L@G zip file into the images folder. It should look like

Tutorial 2 ▾

▾ CSS

   stylesheet.css

▾ images

   island.jpg

   pinnacles.jpg

   rainforest1.jpg

   index.html

   page2.html
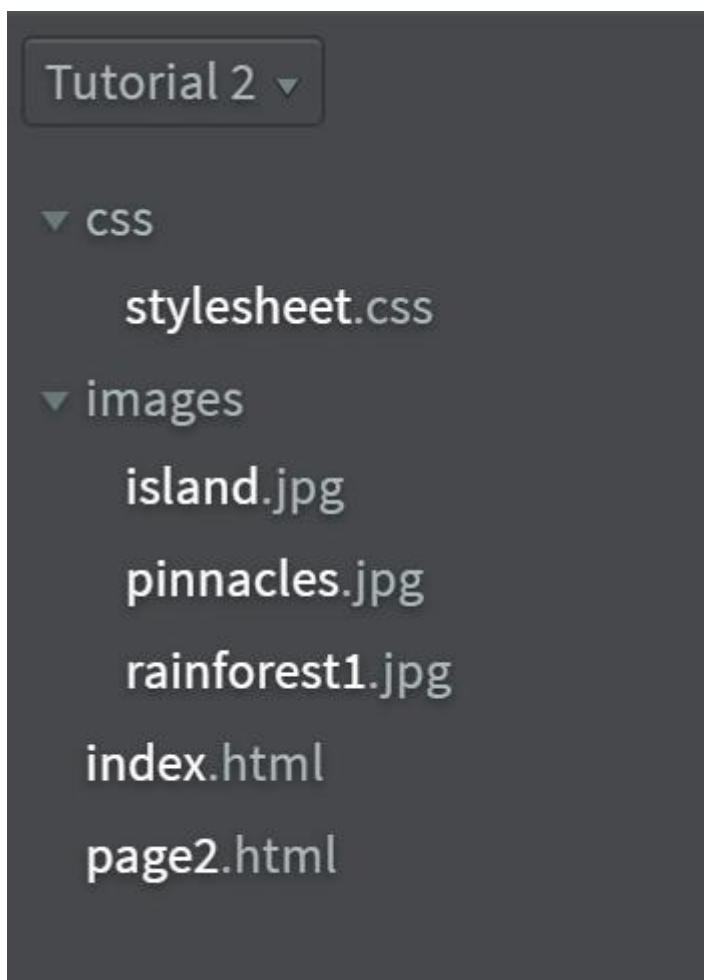
Next we add the styles to the elements

```
<style type="text/css">
    body {background-image:url('images/island.jpg');background-size:cover;background-repeat: no-repeat;
    background-attachment: fixed;background-position: center; /* this is CSS3 */}
    /* the menu and navigation menu styles */
    #menuBtn {width:100px;height:100px;position:absolute;top:0;left:0; background-color:orange;
        text-align:center;line-height:100px; font-family:arial;}
    #navMenu {width:500px;height:500px;position:absolute;top:0;right:0;bottom:0;left:0;margin:auto;
        background-color:white;opacity:0.8; /* this is CSS3 */z-index:5;visibility:hidden;}
    #closeBtn {width:50px;height:50px;background-color:orange;text-align:center;
        line-height:50px;font-family:arial;}
    /* the bottom box navigation styles */
    #box {width:400px;height:200px;position:absolute;left:0;right:0;margin:auto;
        bottom:10%;background-color:midnightblue;}
    #bTop {width:400px;height:50px;background-color:orange;text-align:center;
        line-height:50px;font-family:arial;}
    #bBottom {width:400px;height:150px;background-color:white;}
    #choicesBtn {width:150px;height:50px;position:absolute;left:0;background-color:slategrey;
        text-align:center;line-height:50px;font-family:arial;color:white;}
    #backgroundBtn {width:150px;height:50px;position:absolute;right:0;background-color:slategrey;
        text-align:center;line-height:50px;font-family:arial;color:white;}
    #dDownChoices {width:150px;height:100px;position:absolute;left:0;
        background:orange;opacity:0.5;visibility:hidden;}
    #dDownBackground {width:150px;height:100px;position:absolute;right:0;background:orange;
        opacity:0.5;line-height:14px;visibility:hidden;}
    /* the following nested styling will make the links on the background drop down list a different
    colour */
    #dDownBackground a {color:black;}
</style>
```

Save and test the site, you should see the following

From here, we need to add the content to the Div area.

```html
<body>
    <div id="menuBtn" onClick="openNav()">MENU</div><!-- eo menuBtn -->
    <div id="navMenu">
        <div id="closeBtn" onClick="closeNav()">X</div><!-- eo closeBtn -->
    </div><!-- eo navMenu -->
    <div id="box">
        <div id="bTop">Select Your Choices</div><!-- eo bTop -->
        <div id="bBottom">
            <div id="choicesBtn" onMouseOver="opendDownC()" onMouseOut="closedDownC()">
                Choices
                <div id="dDownChoices"></div><!-- eo dDownChoices -->
            </div><!-- eo choicesBtn -->
            <div id="backgroundBtn" onMouseOver="opendDownB()" onMouseOut="closedDownB()">
                Background
                <div id="dDownBackground">
                    <br><!-- the break <br> tag moves content down one line -->
                    <a href="" onClick="island(); return false;">Image 1</a>
                    <hr><!-- the horizontal rule <hr> tag creates a line -->
                    <a href="" onClick="rainforest(); return false;">Image 2</a>
                    <hr><!-- the horizontal rule <hr> tag creates a line -->
                    <a href="" onClick="pinnacles(); return false;">Image 3</a>
                </div><!-- eo dDownBackground -->
            </div><!-- eo backgroundBtn -->
        </div><!-- eo bBottom -->
    </div><!-- eo box -->
</body>
```

Note we have added in the javascript that will be used. As we are now happy with the layout, it's time to move all of the styles to the external style sheet, copy and paste the styles into the stylesheet.css.

We need to modify the background attribute of the body, as it referenced images in a forward folder such as images/imagename. But, as the code for the stylesheet is in another folder, we need to add ../ to that line, to allow for the style to see out of the css folder, back to root and then forwards into the image folder.

```css
body {background-image:url('../images/island.jpg');background-size:cover;background-repeat: no-repeat;
background-attachment: fixed;background-position: center; /* this is CSS3 */}
/* the menu and navigation menu styles */
```

Now, if we save and test the page, it will look like this

MENU
X
Select Your Choices
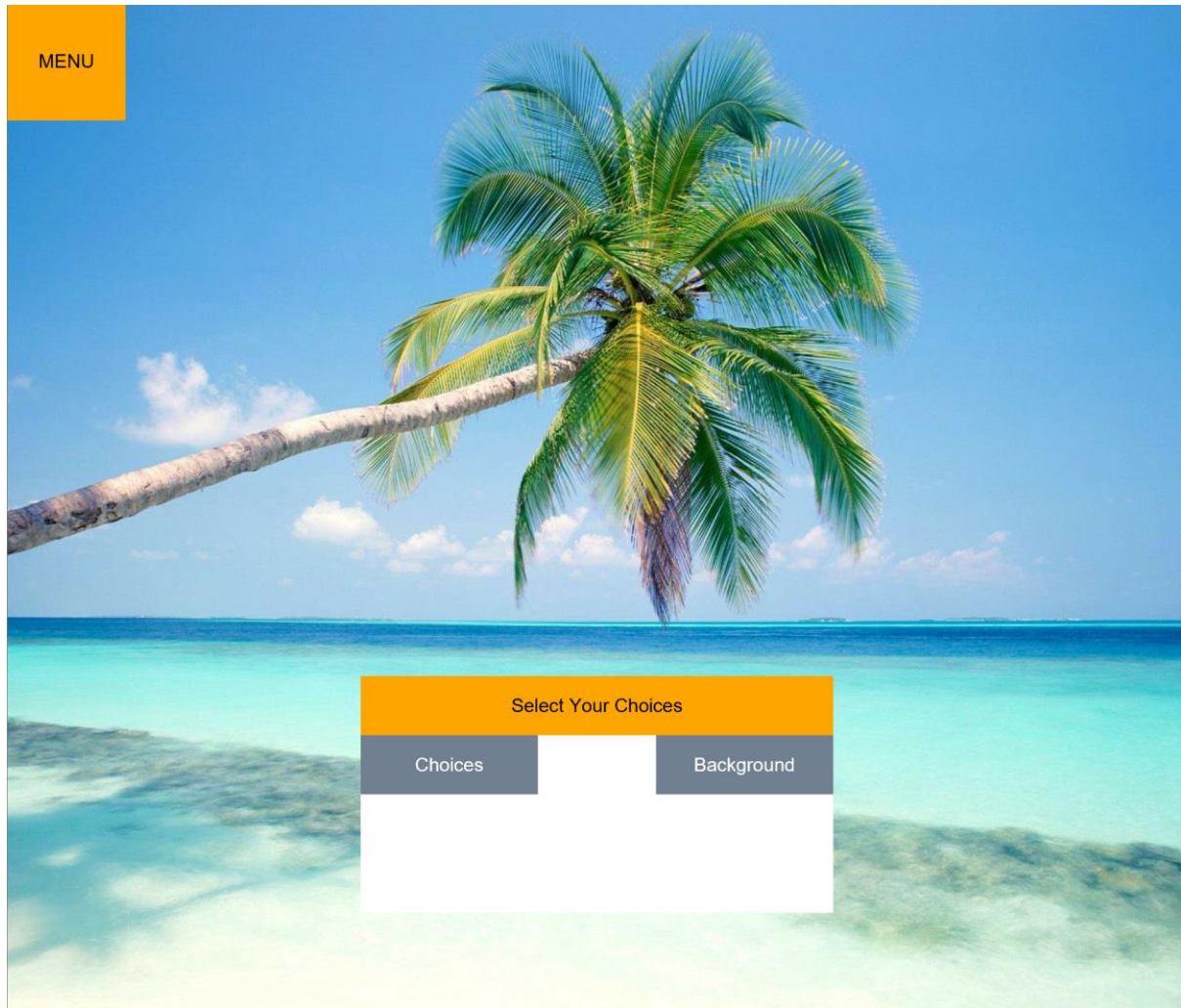Choices
Background

[Image 1](#)

[Image 2](#)

[Image 3](#)

This is because we have moved the styles, as such, now we need to link the stylesheet back to the page, this is done in the <head> part of the code. Use the following command:

```
<title>Page 2</title>
<link href="css/stylesheet.css" type="text/css" rel="stylesheet">
<style type="text/css">
</style>
<script>
```

Ensure all files a re saved and test, you should see the page looking like this

Now, we add the JavaScript to the page.

```
<script>
    function openNav()
    {
        // this function will open the navMenu div
        document.getElementById("navMenu").style.visibility = "visible";
    }
    function closeNav()
    {
        // this will close the navMenu div
        document.getElementById("navMenu").style.visibility = "hidden";
    }
</script>
```

Once this is in the page, save and test, you should be able to open the menu system and close it, when it is open, it will look like this:
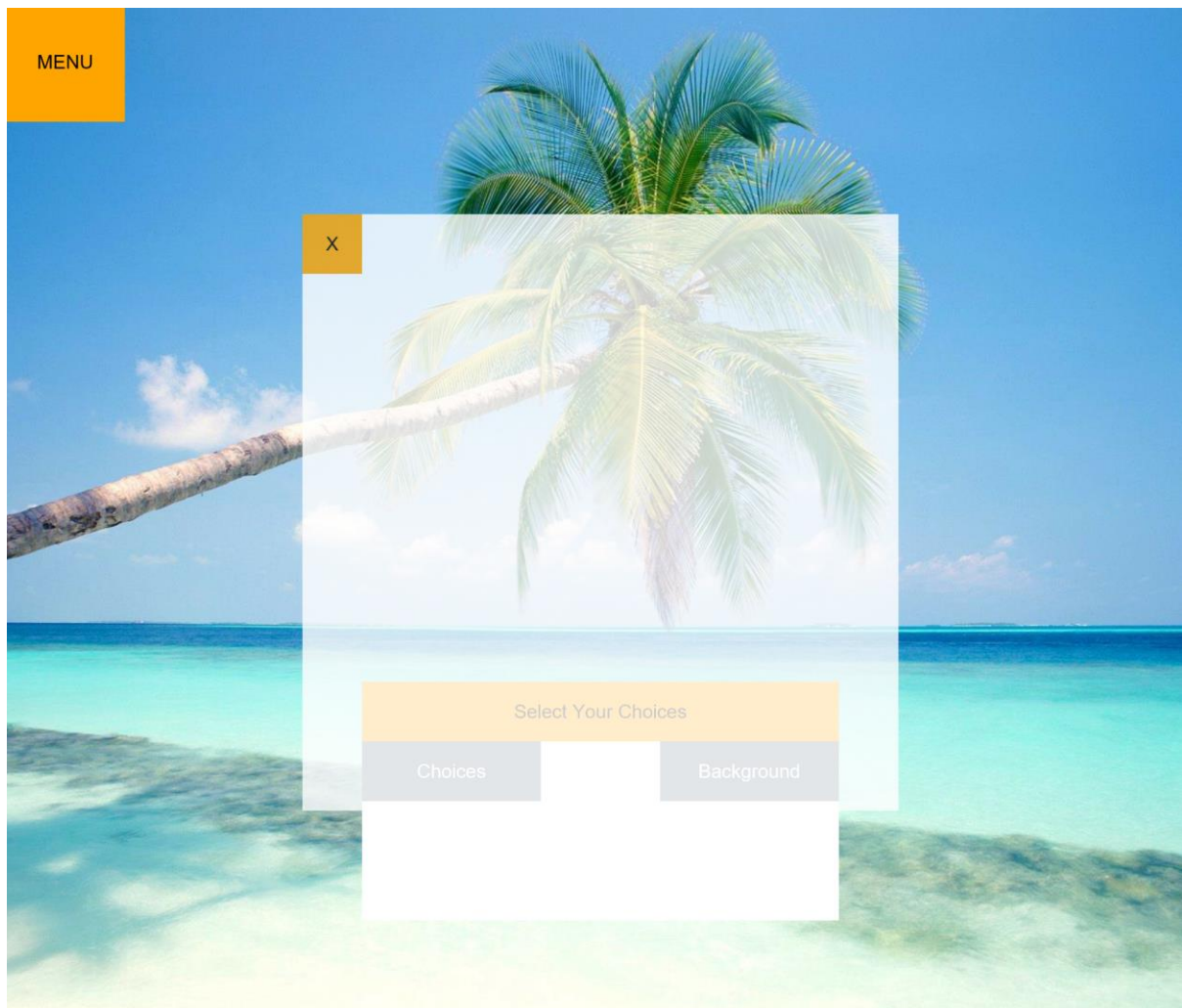
Now we add the javascript to control the drop down menus, this is choices and background.

Add the following javascript

```
}
function opendDownC()
{
    // open the choices drop down menu
    document.getElementById("dDownChoices").style.visibility = "visible";
}
function closedDownC()
{
    // close the choices drop down menu
    document.getElementById("dDownChoices").style.visibility = "hidden";
}
function opendDownB()
{
    // open the background drop down menu
    document.getElementById("dDownBackground").style.visibility = "visible";
}
function closedDownB()
{
    // close the background drop down menu
    document.getElementById("dDownBackground").style.visibility = "hidden";
}
/script>
```

It will produce:

Lastly, we need to be able to change the background images, this is done by adding the following javascript

```
    J
    function island()
    {
        // change the background to the island image
        document.body.style.backgroundImage = "url('images/island.jpg')";
    }
    function rainforest()
    {
        // change the background to the rainforest image
        document.body.style.backgroundImage = "url('images/rainforest1.jpg')";
    }
    function pinnacles()
    {
        // change the background to the pinnacles image
        document.body.style.backgroundImage = "url('images/pinnacles.jpg')";
    }
'script>
```

Once this is done, save and test, you should be able to rotate through the background images. You will see the following:
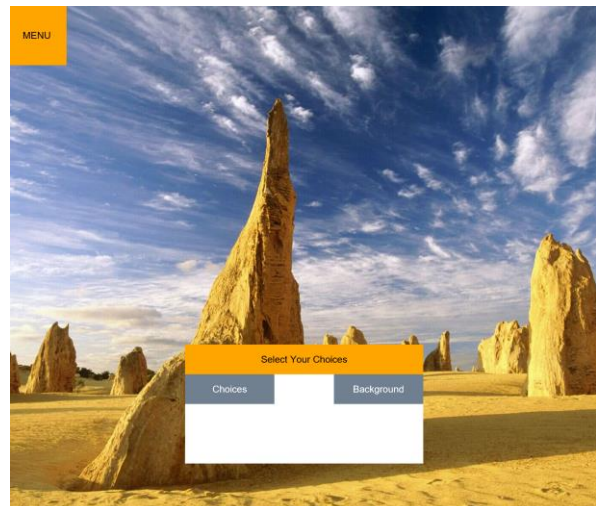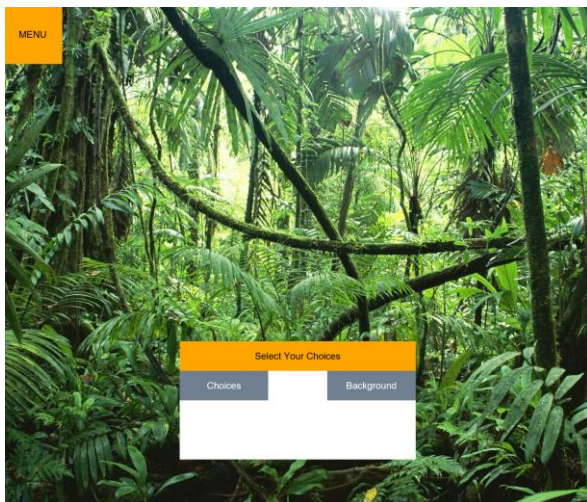
In addition to the island image.

Backup

This concludes tutorial. Save your work to USB, student drive, Onedrive, Dropbox or zip and email the folder to yourself.