Adapted from: https://www.w3schools.com/howto/howto_make_a_website.asp

# Lesson 3 - Create a Website

In this lesson you will create a multi page responsive (changes for different size displays) website with an external CSS file from scratch.

## Step 1 - Basic HTML Page

1. Open your HTML editor and create a new file.
2. Copy and paste the code below into your editor
3. Save your html code into a new file called **index.html** in your **myweb** folder

index.html

```
<html>
    <head>
        <title>Page Title</title>
        <meta name="viewport" content="width=device-width, initial-scale=1">
        <link rel="stylesheet" type="text/css" href="site.css">
    </head>
    <body>
    </body>
</html>
```

## index.html explained

- index.html is the default name for a home page in HTML
- The `<html>` element is the root element of an HTML page
- The `<head>` element contains meta information about the document
- The `<title>` element specifies a title for the document
- The `<meta>` element with name="viewport" makes the website look good on all devices and screen resolutions
- The `<link>` element points to an external file containing the CSS styles
- The `<body>` element contains the visible page content

---

## Step 2 - Creating Page Content

Inside the `<body>` element of our website, we will create:

- A header
- A navigation bar
- Main content
- Side content
- A footer

Make sure you save index.html after each insert of new code.

---

## Header

A header is usually located at the top of the body. See where the new (coloured) code is inserted inside the original code (grey):

index.html

```
<html>
     <head>
          <title>Page Title</title>
          <meta name="viewport" content="width=device-width,
initial-scale=1">
          <link rel="stylesheet" type="text/css"
href="site.css">
     </head>
     <body>
        <div class="header">
           <h1>My Website</h1>
           <p>A website created by me.</p>
        </div>
     </body>
</html>
```

## Navigation Bar

A navigation bar contains a list of links to help visitors navigating through your website:

```
<div class="navbar">
  <a href="#">Link</a>
  <a href="#">Link</a>
  <a href="#">Link</a>
  <a href="about.html" class="right">About</a>
</div>
```

## Main content

Create a 2-column layout, divided into a "side content" and a "main content".

```
<div class="row">
  <div class="side">...</div>
  <div class="main">...</div>
</div>
```

## Footer

At last, we will add a footer.

```
<div class="footer">
   <h2>Footer</h2>
</div>
```

CONGRATULATIONS YOU HAVE COMPLETED THE BASIC HTML STRUCTURE

## Step 3 - External CSS style file

Now that the HTML structure of our homepage is completed, we need to add the CSS styles to make it beautiful.

1. Use your HTML editor and create a new file.
2. Copy and paste the code below into your editor
3. Save your css into a new file called **site.css** in your **myweb** folder

site.css

```
.header {
   padding: 80px; /* some padding */
   text-align: center; /* center the text */
   background: #1abc9c; /* green background */
   color: white; /* white text color */
}

/* Increase the font size of the <h1> element */
.header h1 {
   font-size: 40px;
}
```

## site.css explained

- site.css is an external CSS file that can be linked to from any page
- `.header` is the name of a CSS class which contains the styles for that class
- `.header h1` defined the style updates for the h1 element inside .header
- /* */ symbols are CSS comments that help you document your code

## Navigation bar

Copy and paste the CSS styles for the navigation bar to the external CSS file **site.css**.

```
/* Style the top navigation bar */
.navbar {
  overflow: hidden; /* Hide overflow */
  background-color: #333; /* Dark background color */
}

/* Style the navigation bar links */
.navbar a {
   float: left; /* Make sure that the links stay side-by-side */
   display: block; /* Change the display for responsive */
   color: white; /* White text color */
   text-align: center; /* Center the text */
   padding: 14px 20px; /* Add some padding */
   text-decoration: none; /* Remove underline */
}

/* Right-aligned link */
.navbar a.right {
   float: right; /* Float a link to the right */
}

/* Change color on hover/mouse-over */
.navbar a:hover {
   background-color: #ddd; /* Grey background color */
   color: black; /* Black text color */
}
```

## Main content

This example uses CSS Flexbox to handle the two-column layout:

```
/* Ensure proper sizing, asterisk is a universal selector */
* {
   box-sizing: border-box;
}
```

```css
/* Column container */
.row {
   display: flex;
   flex-wrap: wrap;
}

/* Create two unequal columns that sits next to each other */

/* Sidebar/left column */
.side {
   flex: 30%; /* Set the width of the sidebar */
   background-color: #f1f1f1; /* Grey background color */
   padding: 20px; /* Some padding */
}

/* Main column */
.main {
   flex: 70%; /* Set the width of the main content */
   background-color: white; /* White background color */
   padding: 20px; /* Some padding */
}
```

Then add media queries to make the layout responsive. This will make sure that your website looks good on all devices (desktops, laptops, tablets and phones). Resize the browser window to see the result.

```css
/* Responsive layout - when the screen is less than 700px wide,
make the two columns stack on top of each other instead of next
to each other */
@media screen and (max-width: 700px) {
   .row {
      flex-direction: column;
   }
}
```

```
/* Responsive layout - when the screen is less than 400px wide,
make the navigation links stack on top of each other instead of
next to each other */

@media screen and (max-width: 400px) {
   .navbar a {
      float: none;
      width: 100%;
   }
}
```

Tip: To create a different kind of layout, just change the flex width (but make sure that it adds up to 100%).

---

## Footer

At last, we will code our footer CSS class

```
.footer {
   padding: 20px; /* Some padding */
   text-align: center; /* Center text*/
   background: #ddd; /* Grey background */
}
```

## Step 4 - About.html

In step 2 you created a navigation bar with a link to a file called **about.html**. In this step you will code the about.html so your web-page navbar link will work.

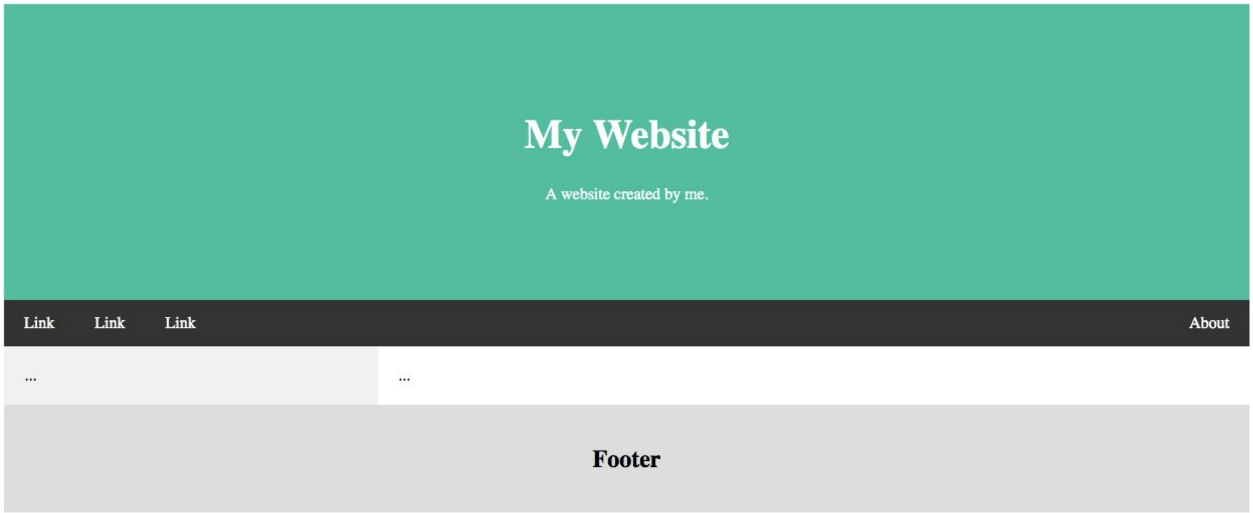
```
<a href="about.html" class="right">About</a>
```

1. Use your HTML editor to create a new file
2. Copy and paste the following code into the new file
3. Save your new file as **about.html** to your **myweb** folder

about.html

```
<html>
     <head>
          <title>About</title>
          <meta name="viewport" content="width=device-width,
initial-scale=1">
          <link rel="stylesheet" type="text/css"
href="site.css">
     </head>
     <body>
        <div class="header">
           <h1>My Website</h1>
           <p>A website created by me.</p>
        </div>
        <div class="navbar">
           <a href="index.html">Home</a>
           <a href="#">Link</a>
           <a href="#">Link</a>
           <a href="#" class="right">About</a>
        </div>
        <h1>About us</h1>
        <p>...</p>
     </body>
</html>
```

CONGRATULATIONS! YOU HAVE BUILT A RESPONSIVE WEBSITE FROM SCRATCH.



## Step 4 - Enrichment: Make the site your own

Well done, you have created a very basic structure of a website, with a linked subpage and an external CSS file.

Now it is your turn to learn some more web development techniques on your own.

Use the following links to integrate more functionality and content into your basic website.

HTML elements (tags)
https://www.w3schools.com/html/default.asp

CSS styles
https://www.w3schools.com/css/default.asp

## Step 5 - Share with peers and teachers

It is sometimes difficult to know if your page will be appealing to others. Be sure to share what you are doing with friends and your teachers.

At the end of the lesson, show your progress to your teacher to get points!