

# Lab 7a: Web Page Basics

## Introduction

A website is a set of related web pages served from a single web domain. A website is hosted on at least one web server. It is accessible through a network such as the Internet or a private local area network through an Internet address known as a Uniform Resource Locator (URL). All publicly accessible websites make up the World Wide Web. Without websites, there is not WWW.

Today, you will create a single webpage.

The steps to building a website are gathering information, planning, designing (text, graphics, multimedia), and building your website. Following that, you can make it public by choosing a domain name and uploading it to a web hoster.

The process of creating a website is divided into two steps.

### **1) The first step is installing basic software.**

- A text editor is used to write code in.
  - This could be a free text editor (e.g. Notepad or Notepad++).
- Web browsers test your HTML and CSS.
  - Currently, the most-used browsers are Chrome, Firefox, Opera, Edge

We won't use these today, but they are useful when building a website.

- A graphics editor (Paint.NET, Photoshop) is used to make images for your web pages.
- An FTP program is used to upload web pages to your web hoster.

### **2) The next step is dealing with files.**

A website consists of many files: text content, code, stylesheet, media content, and so on. When you're building a website, you need to assemble these files into a sensible structure. When you are working on a website locally on your own computer, you should keep all the related files in a single folder that mirrors the published website's file structure on the server.

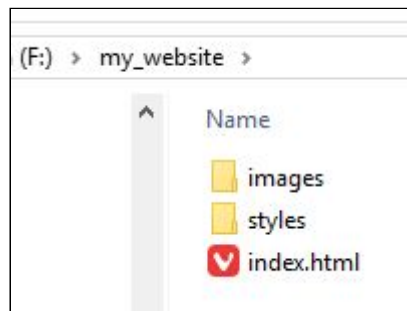
The most common items a website project has are index HTML file and folders that contain images, style files.

1. First, you will choose a place to store your website projects.
  - a. Since you are working in the university computer lab, we recommend using your flash drive.
2. Next, you will create an index HTML file and folders to contain images, stylesheet, and script files.

# Building your Website

## Step 1: Creating Your Files

1. Create a folder on flash drive or computer called: **lab7** or **lab7-firstname** (i.e., lab7-sara)
  - a. It is better to create it on your flash drive so that you can save the files
2. Create the following files or folders inside of your lab7 folder
  - a. **index.html**: this file will generally contain your homepage content, that is, the text and images that people see when they first go to your site.
    - i. Using your text editor, create a new file called **index.html** and save it just inside your site folder.
  - b. **images** folder: this folder will contain all the images that you use on your site.
    - i. Create a folder called **images** inside your site folder.
  - c. **styles** folder: this folder will contain the CSS code used to style your content (for example, setting text and background colors).
    - i. Create a folder called **styles** inside your site folder.



3. **Note:** You should create all web files, images, and other files in all lowercase without spaces.
  - a. Acceptable file names:  
my-file.html, my\_file.html, myfile.html
  - b. Incorrect file names (may cause problems):  
my file.html, Myfile.html, MyImage.jpg

## Step 2: Looking at HTML Syntax

Before you add some HTML, it is helpful to understand the syntax.

- You will use an HTML tool from <https://www.w3schools.com/> to help you.

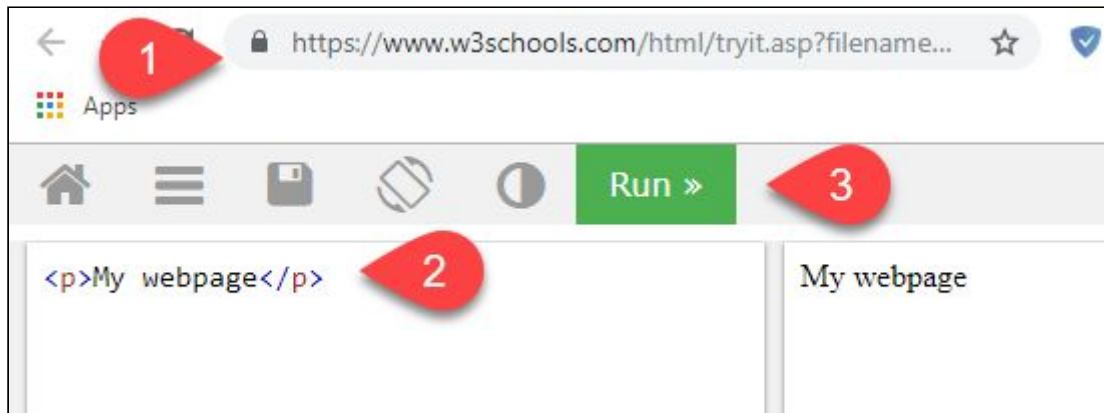
HTML is not a programming language; it is a markup language used to tell your browser how to display the web pages.

To create a new paragraph on the page, we will specify that it is a paragraph by enclosing it in a paragraph tag (`<p>`). The `p` tag is the same as pressing the Enter key on a keyboard.

<p>My webpage</p>

# Try now

Open up the [w3schools.com website](https://www.w3schools.com/html/tryit.asp?filename...) and try it!



The main parts of the HTML element are the opening tag `<p>`, closing tag `</p>`, and content `My webpage`.


- **The opening tag `<p>`:** This consists of the name of the element (in this case, `p`), wrapped in opening and closing angle brackets.
  - The opening tag determines where the element begins or starts to take effect – in this case where the start of the paragraph is.
- **The closing tag `</p>`:** This is the same as the opening tag, except that it includes a forward slash before the element name.
  - The closing tag determines where the element ends – in this case where the end of the paragraph is.
  - Failing to include a closing tag is a common error for beginners. It can lead to strange results.
- **The content `My webpage`:** This is the information that the web browser displays.
- **The element:** The complete structure, which contains the opening tag, content, and closing tag.

**A note about terminology:** We use tags and elements to describe HTML. Many people use these terms interchangeably, but there is a slight difference. If you want to know the difference, you can read all about it. <https://www.lifewire.com/html-tag-vs-element-3466507>

You can put elements inside other elements too – this is called nesting. We could wrap the word “webpage” in a `<strong>` tag to make the word **bold**:

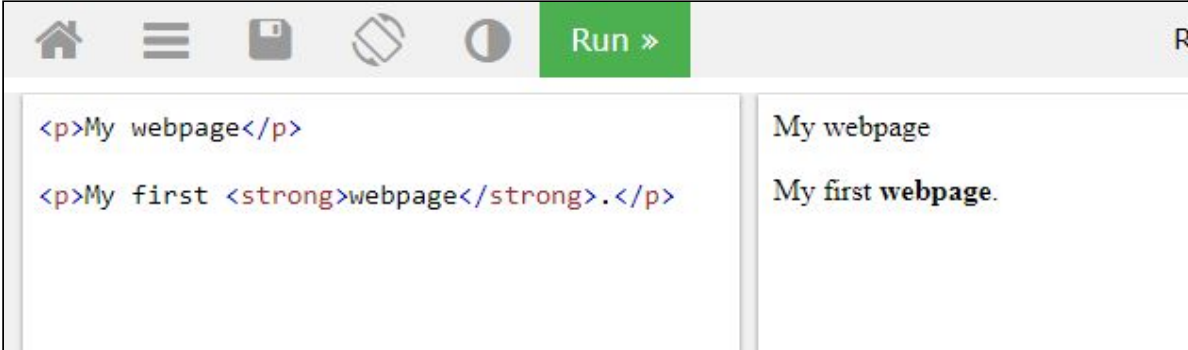
```
<p>My first <strong>webpage</strong>.</p>
```

My first **webpage**.



Open up the [w3schools.com website](https://www.w3schools.com)

- Try creating some bold text




The image shows a code editor interface with a toolbar at the top containing icons for home, menu, save, undo, redo, and a green 'Run' button. The editor has two panes: a left pane for code and a right pane for rendered output. The code pane contains two lines of HTML: `<p>My webpage</p>` and `<p>My first <strong>webpage</strong>.</p>`. The rendered output pane shows the corresponding visual result: 'My webpage' and 'My first **webpage**'.

The elements have to open and close in the same order. One needs to be inside of the other one (nesting elements) `<p><strong></strong></p>`. The `<p>` tag started on the outside so it must stay on the outside.

#### HTML Examples

- Correct open and close: `<p>My web page</p>`
- Incorrect open and close: `<p>My web page`
  - The web browser does not know where to end the paragraph
- Correct nesting: `<p>My <i>web page</i></p>`
- Incorrect nesting: `<p>My <i>web page</p></i>`
  - The order of the `<p>` and `<i>` tags are not consistent. The `<i>` tag should go inside.



Open up the [w3schools.com website](https://www.w3schools.com)

- Try creating some italic or emphasized text

Some elements have no content, and are called empty elements. Take the `<img>` element for adding an image:

```

```

The `img` element contains two attributes, but there is no closing `</img>` and no inner content. This is because an image element doesn't wrap content. It is called a self-closing tag. Instead, the elements that are self-closing store data in attributes.

## Try now

Open up the [w3schools.com website](https://www.w3schools.com)

1. Add the `img` tag: ``
  - a. Notice that the image is missing.
  - b. Instead, it displays the text in the `alt` attribute.



2. Replace the `src` value with a valid URL to an image:  
<https://resources.bilimedtech.com/ict/html/html.png>



### Step 3: Building your HTML file

The next step is adding the content using HTML.

In Figure 1, look at how individual elements are combined to form an entire HTML page.

### 1) Open your `index.html` file and type this HTML

```
<html>
<head>
<title>My first web page</title>
</head>
<body>
<p>Your web page content goes here</p>
</body>
</html>
```

Figure 1: Basic HTML structure

The above code is a very basic example of the code that helps make up every webpage.

1. As you can see, the code starts with `<html>`, which is defining that everything within `<html>` is HTML code.
2. Next, you have `<head>`, which is defining the heading of your HTML document.
3. Third, we have the `<title>` section within `<head>`, which defines the web page title that is displayed at the top of the Internet browser window.
4. Finally, the `<body>` section contains what is shown on the web page.

### 2) Verify your HTML

Now that you have created a basic website, you may want to verify how the website looks (Figure 2). Since you have files locally stored on your computer, you will not need to connect to the Internet to view your web page. You can double-click your `index.html` file to open it in a browser automatically.

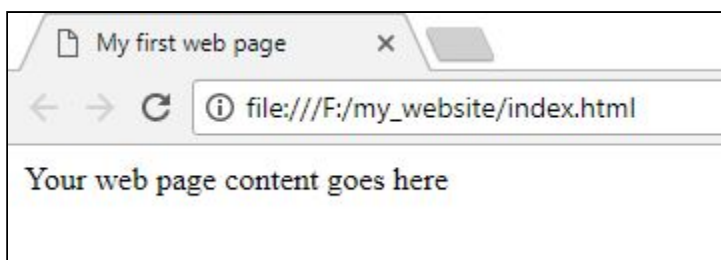


Figure 2: Viewing your `index.html` page

### 3) Add some headings to your HTML

Heading elements allow you to specify that certain parts of your content are headings – or subheadings – of your content. In the same way that a book has the main title (h1), chapter titles (h2), and subtitles (h3-h6), an HTML document can too. HTML contains six heading levels (`<h1–<h6>`).

#### 3.1) Add these heading elements to your HTML

```
<h1>My main title</h1>
<h2>My top level heading</h2>
```

```
<h3>My subheading</h3>
<h4>My sub-subheading</h4>
```

```
<html>
<head>
<title>My first web page</title>
</head>
<body>
<h1>My main title</h1>
<h2>My top level heading</h2>
<h3>My subheading</h3>
<h4>My sub-subheading</h4>
<p>Your web page content goes here</p>
</body>
</html>
```

Figure 3: Heading elements

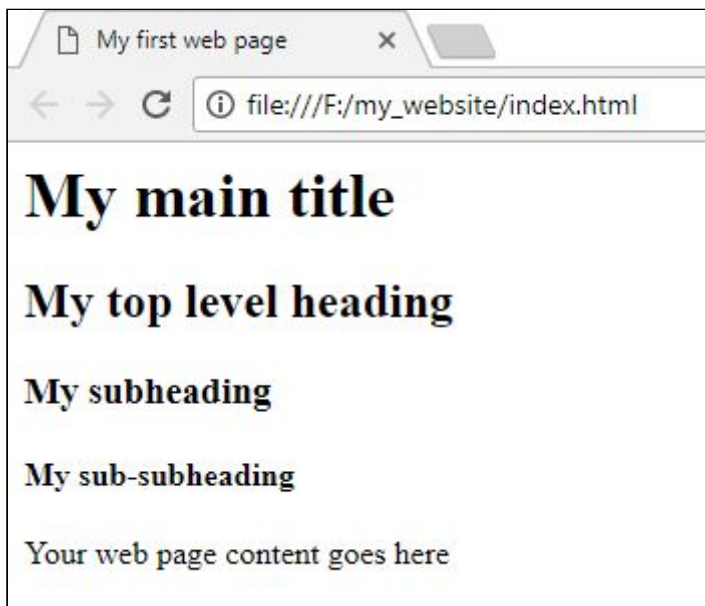


Figure 4: Headings in the browser

#### 4) Personalize your web page

To make your site more interesting, change the content to be about you or your interests. You don't have to use this topic, but you need to focus on something general and then focus to specifics.

#### Topic examples

Your city	Sports Team
<ul style="list-style-type: none"><li>• <code>&lt;title&gt;</code>: Your name</li><li>• <code>&lt;h1&gt;</code>: Your name</li></ul>	<ul style="list-style-type: none"><li>• <code>&lt;title&gt;</code>: Team Name</li><li>• <code>&lt;h1&gt;</code>: Team Name</li></ul>

<ul style="list-style-type: none"> <li>• <code>&lt;h2&gt;</code>: Your city</li> <li>• <code>&lt;h3&gt;</code>: Your district</li> <li>• <code>&lt;h4&gt;</code>: Your street</li> </ul>	<ul style="list-style-type: none"> <li>• <code>&lt;h2&gt;</code>: About the sport</li> <li>• <code>&lt;h3&gt;</code>: About the team,</li> <li>• <code>&lt;h4&gt;</code>: About</li> </ul>
<p>For example:</p> <ul style="list-style-type: none"> <li>• Jimmy's Site</li> <li>• Edinburgh, Scotland</li> <li>• Newington</li> <li>• E. Preston Street</li> </ul>	<p>For example:</p> <ul style="list-style-type: none"> <li>• Barça</li> <li>• About Football</li> <li>• FC Barcelona</li> <li>• Ronaldinho</li> </ul>

```

<html>
<head>
<title>Jimmy's site</title>
</head>
<body>
<h1>Jimmy's site</h1>
<h2>Edinburgh, Scotland</h2>
<h3>Newington</h3>
<h4>E. Preston Street</h4>
<p>Your web page content goes here</p>
</body>
</html>

```

Figure 5: Personalized site



Figure 6: Personalized site in the browser

### 5) Add more HTML elements

Lists are commonly used in HTML to display content. Creating lists uses two elements, which are the list type and then the list items. The most common list types are ordered



(numbers) and unordered (bullets) lists. Unordered lists are for lists where the order of the items doesn't matter. These are wrapped in a `<ul>` tag. Ordered lists are for lists where the order of the items does matter. These are wrapped in an `<ol>` tag.

Each item inside the lists is put inside an `<li>` (list item) tag. See Figure 7 for the rendered HTML.

Ordered list structure

```
<ol>
  <li>The first list item</li>
  <li>The second list item</li>
  <li> ... list item n</li>
</ol>
```

Unordered list structure

```
<ul>
  <li>The first list item</li>
  <li>The second list item</li>
  <li> ... list item n</li>
</ul>
```

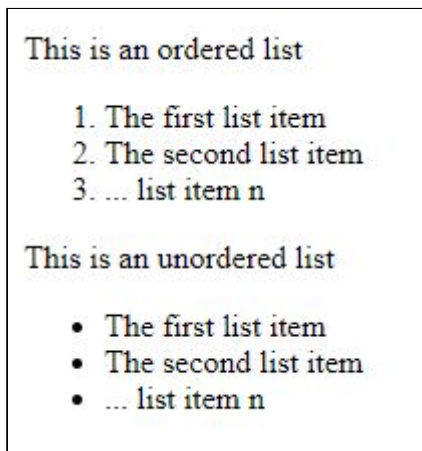


Figure 7: Ordered and unordered lists

5.1) Create an ordered list (`<ol>`) under your city of the top 3 things you like the most about it

1. Create several new lines after the closing `<h2>` tag.
2. Add your list (`<ol>`) and list items (`<li>`)

Figure 8 shows how your HTML should look. Figure 9 shows the view in the web browser.

```
<html>
<head>
<title>Jimmy's site</title>
</head>
<body>
<h1>Jimmy's site</h1>
<h2>Edinburgh, Scotland</h2>
<ol>
  <li>Visiting old castles</li>
  <li>Hanging out on Princes Street</li>
  <li>Hiking Arthur's Seat</li>
</ol>

<h3>Newington</h3>
<h4>E. Preston Street</h4>
<p>Your web page content goes here</p>
</body>
</html>
```

Figure 8: HTML ordered list <ol>

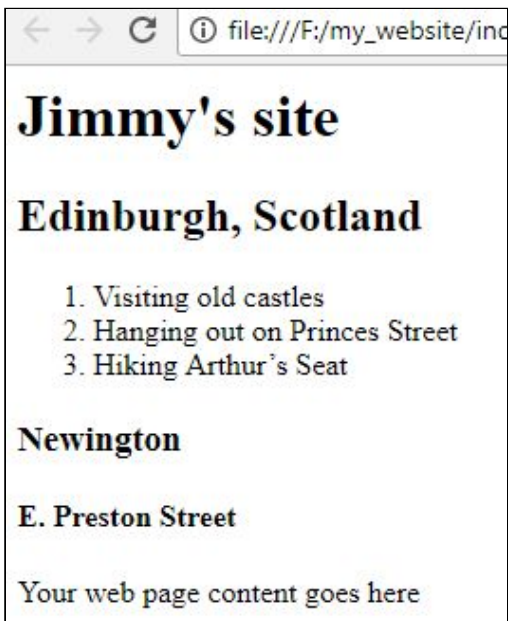


Figure 9: Ordered list in the web browser

5.2) Create an unordered list (<ul>) with three items (<li>) under your district or microregion that describe your area.

Figure 10 shows how your HTML should look. Figure 11 shows the view in the web browser.

```
<html>
<head>
<title>Jimmy's site</title>
</head>
<body>
<h1>Jimmy's site</h1>
<h2>Edinburgh, Scotland</h2>
<ol>
  <li>Visiting old castles</li>
  <li>Hanging out on Princes Street</li>
  <li>Hiking Arthur's Seat</li>
</ol>
<h3>Newington</h3>
<ul>
  <li>The best place to live in Edinburgh!</li>
  <li>The easternmost district of the area formerly covered by the Burgh Muir</li>
  <li>Many students live here</li>
</ul>
<h4>E. Preston Street</h4>
<p>Your web page content goes here</p>
</body>
</html>
```

Figure 10: HTML unordered list <ul>

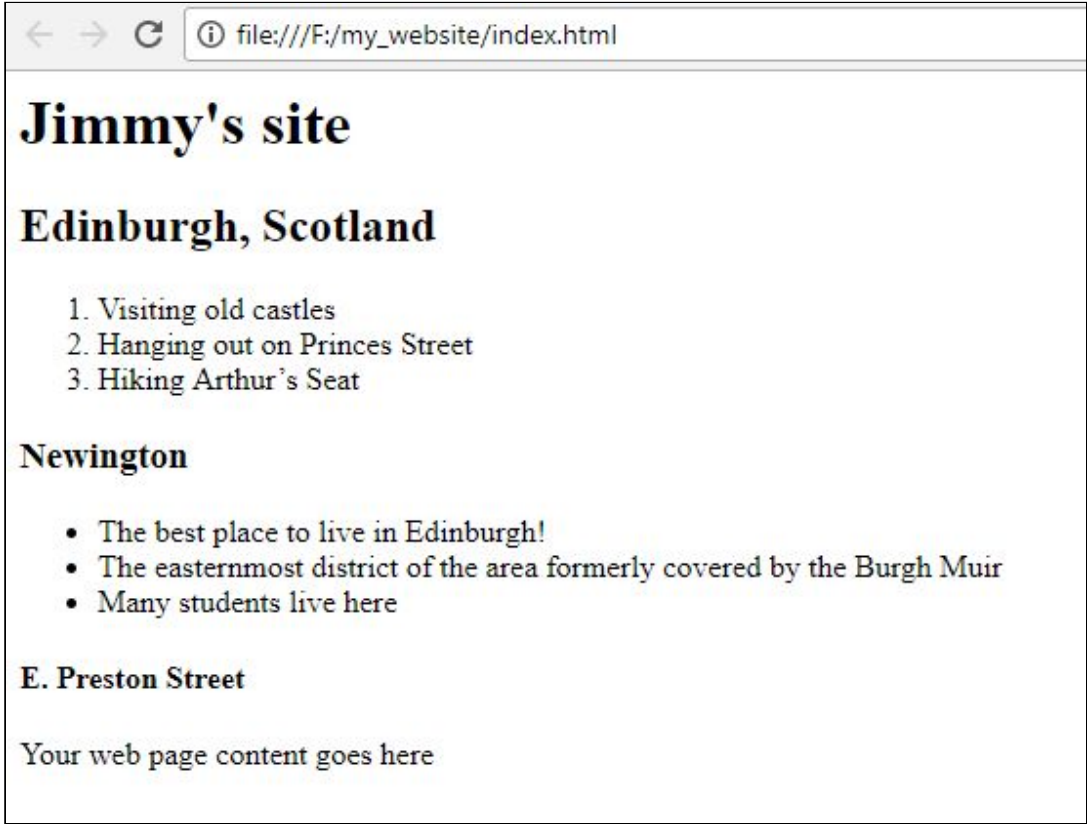


Figure 11: Unordered list in the browser

6) Adding Images

Images use a self-closing tag because they do not contain text. Instead, they use attributes. The attributes tell the web browser which image to display.

```

```

Attributes contain information about the element.

1. src (source): The path to the image file
  - a. According to the file, icon.png is inside of the images folder.
2. alt (alternate name): Descriptive text for users who cannot see the image
  - a. If the image does not load for some reason, the web browser will display the text in the alt attribute.

6.1) Add an image to your HTML file about your city.

1. Search Google images to find an interesting picture. Try not to find a huge picture (smaller is better).
2. Save it to your [images](#) folder.
3. Rename the file as [mycity.jpg](#)
  - a. You might need to change the extension in the HTML if you use a different type of image files, such as a .png or .gif.



4. Under the under the closing `</ol>` list tag, create an empty `<p>` tag to hold the image element.

```
<ol>
  <li>Visiting old
  <li>Hanging out
  <li>Hiking Arthu
</ol>
<p></p>
```

5. Inside of the `<p></p>` element, add the HTML for the image.

```

```

Figure 12 shows how your HTML should look. Figure 13 shows the view in the web browser.

```

<html>
<head>
<title>Jimmy's site</title>
</head>
<body>
<h1>Jimmy's site</h1>
<h2>Edinburgh, Scotland</h2>
<ol>
  <li>Visiting old castles</li>
  <li>Hanging out on Princes Street</li>
  <li>Hiking Arthur's Seat</li>
</ol>
<p></p>
<h3>Newington</h3>

```

Figure 12: HTML image element `<img />`



Figure 13: Image displayed in the browser

## 7) Adding Hyperlinks

Links are very important – they are what makes the Web A WEB. To add a link, we need to use a simple element – `<a>` – the `a` is short for “anchor”.

### 7.1) Create a link to a website about your city

1. Under the `<h2>` element, create a new paragraph element that is about your city. You will turn this text into a link.

```
<body>
<h1>Jimmy's site</h1>
<h2>Edinburgh, Scotland</h2>
<p>About Edinburgh</p>
<ol>
  <li>Visiting old castles</li>
```

2. Wrap the text inside of the `<p>` tag in an `<a>` element, like so:

```
<a>About Edinburgh</a>
```

```
<h1>Jimmy's site</h1>
<h2>Edinburgh, Scotland</h2>
<p><a>About Edinburgh</a></p>
<ol>
  <li>Visiting old castles</li>
```

3. Give the `<a>` element an `href` attribute, like so:

```
<a href="">About Edinburgh</a>
```

4. Use a search engine to find a website or page about your city. Copy the URL.

- a. For example: <https://www.scotlandinfo.eu/edinburgh/>

5. Paste the URL in the href attribute (it must go between the double quotes " ").

```
<a href="https://www.scotlandinfo.eu/edinburgh/">About Edinburgh</a>
```

6. Refresh the page in the browser and then test the link. It should take you the other website.

### 7.2) Add another link on your page

1. Search the internet for a website about any of interesting items in the ordered or unordered lists (about your city or district).

- a. For example, Hiking Arthur's Seat

2. Add an `<a>` element inside of the `<li>` that links to another site.

Figure 14 shows how your HTML should look. Figure 15 shows the view in the web browser.

```
<html>
<head>
<title>Jimmy's site</title>
</head>
<body>
<h1>Jimmy's site</h1>
<h2>Edinburgh, Scotland</h2>
<p><a href="https://www.scotlandinfo.eu/edinburgh/">About Edinburgh</a></p>
<ol>
  <li>Visiting old castles</li>
  <li>Hanging out on Princes Street</li>
  <li><a href="https://www.earthtrekkers.com/arthurs-seat-volcano-edinburgh/">Hiking Arthur's Seat</a></li>
</ol>
<p></p>
```

Figure 14: HTML anchor element `<a>`



Figure 15: A hyperlink in the browser

Congratulations! You have a good start creating your web page.

## Step 4: Adding Style to Your Page

CSS (Cascading Style Sheets) is the code you use to style your webpage. Like HTML, CSS is not a programming language. It is a style sheet language, that lets you apply styles selectively to elements in HTML documents. For example, to select all the heading `<h1>` elements on an HTML page and turn the text within them red and center it, you'd write this CSS:

```
h1 {  
  color: orange;  
  text-align: center;  
}
```

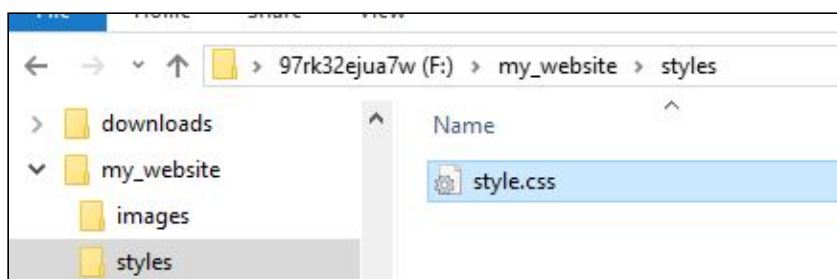
### 1) Style the `h1` element

4.1) Let's try it out. We'll style our heading 1 as red:

1. Create a new document in your text editor
2. Paste these three lines of CSS into a new file in your text editor

```
h1 {  
  color: orange;  
  text-align: center;  
}
```

3. Save the file as `style.css` in your `styles` directory.



Next, we need to link `style.css` to the web page. Otherwise, the CSS styling won't affect how your browser displays the HTML document.

1. Open your `index.html` file and paste the following line inside the `head` section of the HTML document, which is between the `<head>` and `</head>` tags. A good place is to place it under the `<title>` tag.  
`<link href="styles/style.css" rel="stylesheet" type="text/css">`
2. Save `index.html` and load it in your browser (figures 16 and 17).

```
<html>
<head>
  <title>Jimmy's site</title>
  <link href="styles/style.css" rel="stylesheet" type="text/css">
</head>
<body>
  <h1>Jimmy's site</h1>
  <h2>Edinburgh, Scotland</h2>
</body>
</html>
```

Figure 15: Linking the CSS file in HTML



Figure 16: Styled heading 1 tag

### Anatomy of a CSS Rule.

Let's examine CSS in a bit more detail now that you've seen what it does. Take a moment to study Figure 17.



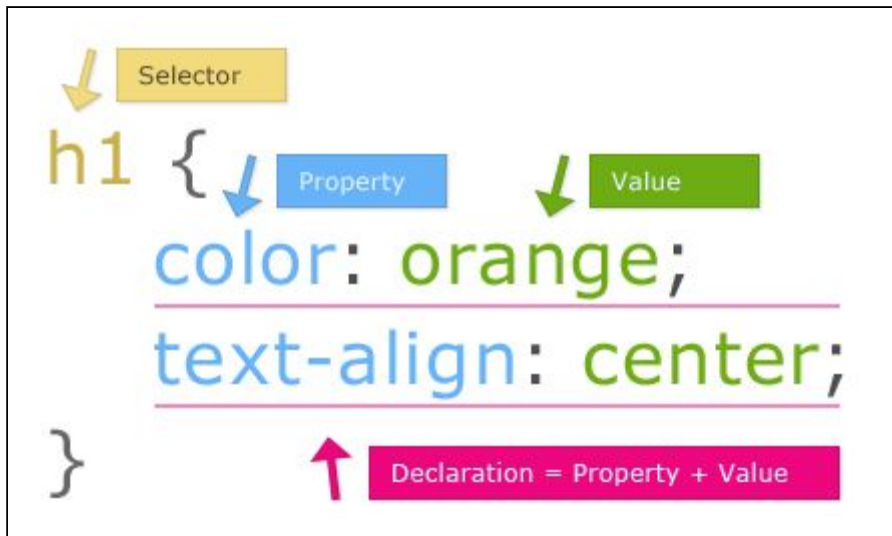


Figure 17: Anatomy of a CSS Rule

A CSS rule is a complete unit that contains a selector and one or more declaration

- **Selector:** The HTML tag name at the start of the rule set
  - It selects the element(s) to be styled (in this case, `h1` elements).
  - To style a different element, just change the selector.
- **Declaration:** A complete entity that contains the property and value
  - In this case, the entire declaration is: `color: orange;`
- **Properties:** The specific property of the HTML element to style or change
  - In this case, `color` and `text-align` are the properties of the style.
  - There are many CSS properties:
    - <https://www.w3schools.com/cssref/default.asp>
- **Property value:** The value to assign to the property
  - In this case, `orange` is the value of the `color` property
- **Note the other important parts of the syntax:**
  - Each rule set after the selector must be wrapped in curly braces (`{ }`)
  - A colon (`:`) separates the property from its value
  - A declaration always terminates with a semicolon (`;`)

CSS rules are flexible. For example, you can also select multiple types of elements and apply a single rule set to all of them. To do this, you would include multiple selectors separated by commas. For example:

```
P, li, h1 {
  color: red;
}
```

There are many different types of selectors. Above, we only looked at element selectors, which select all elements of a given type in the given HTML documents.

Now that we've explored some CSS basics let's start adding some more rules and information to our `style.css` file to make our example look nice. First of all, let's get our fonts and text looking a little better.

## 2) Add font styles

1. Delete the existing rule you have in your `style.css` file.
2. Add the following CSS rules in Figure 18.
3. Save your file and then refresh your page. It should look similar to Figure 19.

```
html {
  font-size: 12px;
  font-family: "Arial", Helvetica, sans-serif;
}

h1 {
  font-size: 24px;
  text-align: center;
}

p, li {
  font-size: 14px;
  line-height: 1.5;
  letter-spacing: 1.25px;
}
```

Figure 18: CSS styles

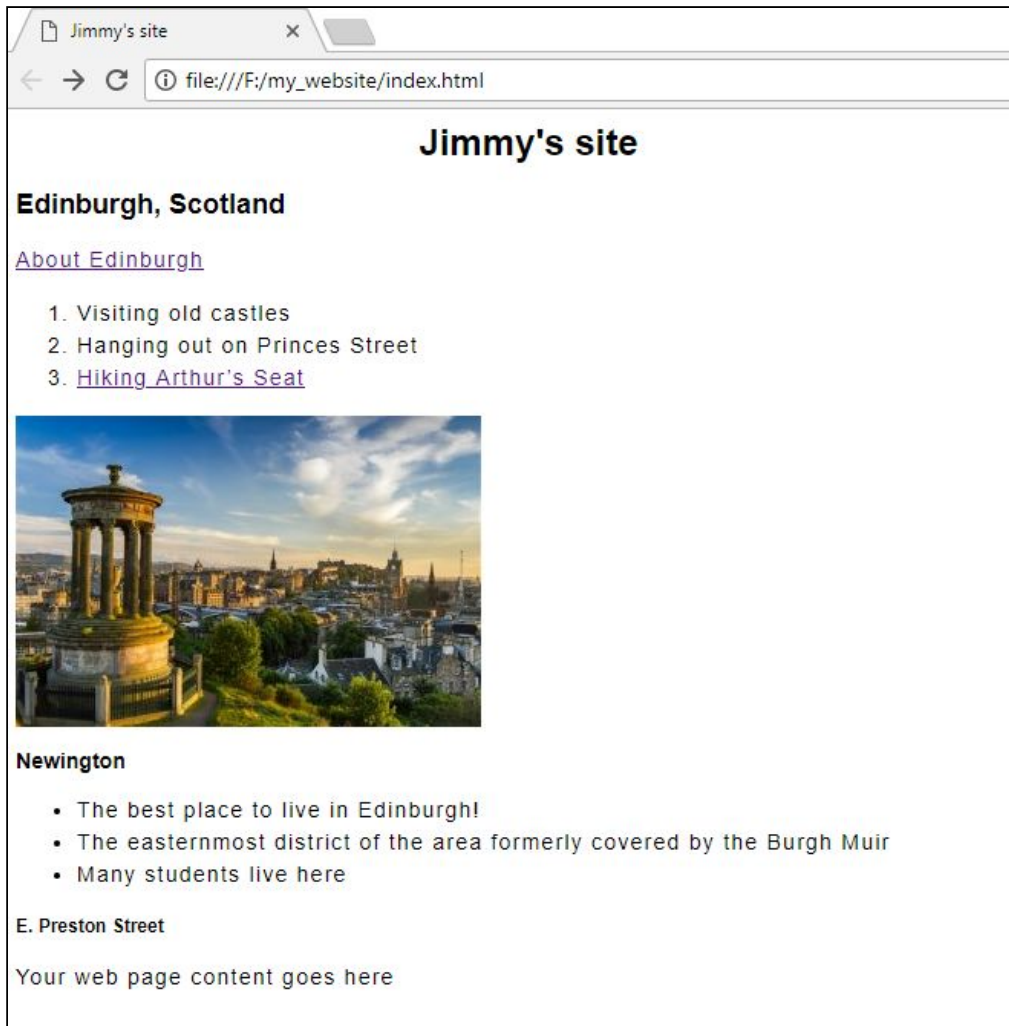


Figure 19: Web page with CSS

## Step 5: Show or submit your work

Congratulation! You have just completed a webpage that you styled using CSS.

Please show your instructor or upload your files to the submission box.