LABORATORY WORK «CREATING A WEBSITE»

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. All publicly accessible websites collectively constitute the World Wide Web.

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

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). Web browsers is used to test code in. Currently the most-used browsers are Firefox, Chrome, Opera, Internet Explorer. A graphics editor (Paint.NET, Photoshop) is used to make images for your webpages. An FTP program is used to upload webpages, is used to a server for public viewing.

The steps are gathering information for website, planning, design (text, graphics, multimedia), choosing a domain: a domain is also known as URL or address of the website.

The next step is dealing with files.

A website consists of many files: text content, code, style sheets, 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.

Choose a place to store your website projects. Create an index HTML file and folders to contain images, style files, and script files. The most common things we'll have on any website project we create are an index HTML file and folders to contain images, style files.

Let's create these now:

- 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. Using your text editor, create a new file called index.html and save it just inside your site folder.
- images folder: this folder will contain all the images that you use on your site. Create a folder called images, inside your site folder.
- styles folder: this folder will contain the CSS code used to style your content (for example, setting text and background colors). Create a folder called styles, inside your site folder.

You should create all web pages, images, and files in all lowercase.

The next step is coding (HTML).

HTML is not a programming language; it is a markup language, and is used to tell your browser how to display the webpages.

If we wanted the line to stand by itself, we could specify that it is a paragraph by enclosing it in a paragraph tag (<p>) element:

```
<p> My webpage</p>
```

The main parts of element are opening tag, closing tag, content, and element.

The opening tag: this consists of the name of the element (in this case, p), wrapped in opening and closing angle brackets. This states where the element begins, or starts to take effect – in this case where the start of the paragraph is.

The closing tag: this is the same as the opening tag, except that it includes a forward slash before the element name. This states where the element ends – in this case where the end of the paragraph is. Failing to include a closing tag is one of the common beginner errors and can lead to strange results.

The content: this is the content of the element, which in this case is just text.

The element: the opening tag, plus the closing tag, plus the content, equals the element.

The elements have to open and close correctly so they are clearly inside or outside one another (nesting elements).

You can put elements inside other elements too – this is called nesting. We could wrap the word “webpage” in a <strong> element, which means that the word is to be strongly emphasized:

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

The elements have to open and close correctly so they are clearly inside or outside one another.

Some elements have no content, and are called empty elements. Take the <img> element we already have in our HTML:

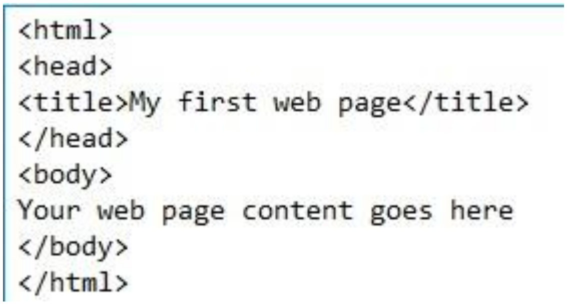
```
<img src="images/icon.png" alt="My image">
```

This contains two attributes, but there is no closing </img> tag, and no inner content. This is because an image element doesn't wrap content to have an effect on it. Its purpose is to embed an image in the HTML page, in the place it appears.

Anatomy of an HTML document.

Now look at how individual elements are combined to form an entire HTML page.

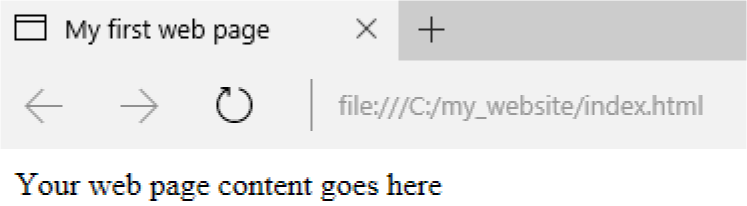
Insert the below source code into your page (index.html).



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

The above code is a very basic example of the code that helps make up every webpage. As you can see, the code starts with <html>, which is defining that everything within <html> is HTML code. Next, you have <head>, which is defining the heading of your HTML document. 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. Finally, the <body> section contains what is shown on the web page. Below is additional code that can be placed in the <body> section of the code to help familiarize you with some of the most commonly used HTML commands.

Now that you have created a basic website, you may want to verify how the website looks. Being that 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 the web page file so it opens in a browser automatically (figure 1).



Your web page content goes here

Figure 1 – Viewing the webpage

Let's turn our attention to the image element again:

```
<img src="images/icon.png" alt="My image">
```

It embeds an image into the page in the position it appears. It does this via the src (source) attribute, which contains the path to our image file. In alt (alternative) attribute, you specify descriptive text for users who cannot see the image, possibly because something has gone wrong causing the image to not display.

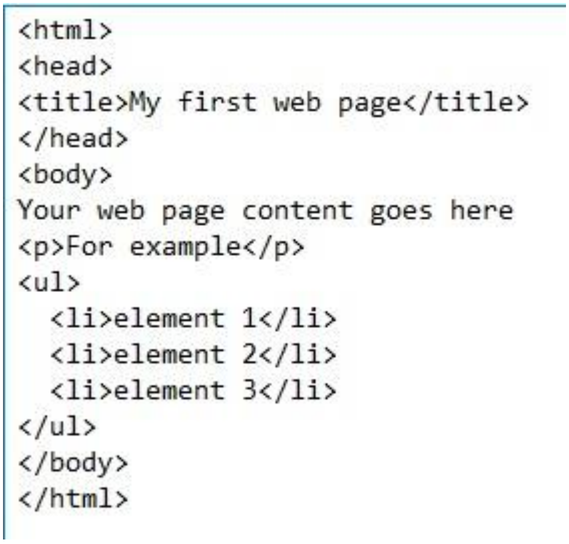
Try adding an image to your HTML page.

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 a main title, chapter titles and subtitles, an HTML document can too. HTML contains six heading levels (<h1>–<h6>).

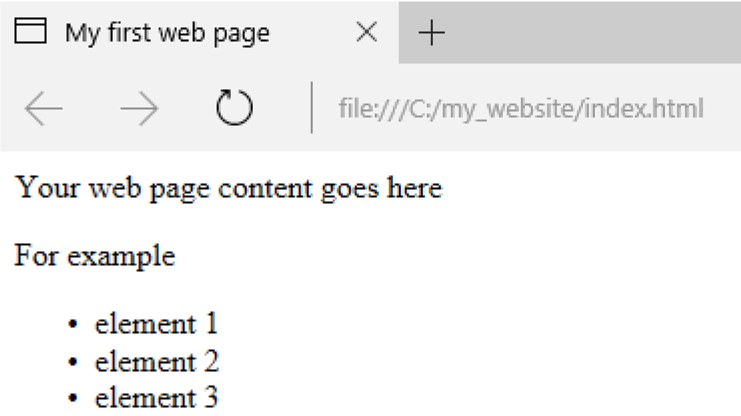
```
<h1>My main title</h1>
<h2>My top level heading</h2>
<h3>My subheading</h3>
<h4>My sub-subheading</h4>
```

A lot of the Web's content is lists and HTML has special elements for these. Marking up lists always consist of at least two elements. The most common list types are ordered and unordered lists. Unordered lists are for lists where the order of the items doesn't matter. These are wrapped in a <ul> element. Ordered lists are for lists where the order of the items does matter. These are wrapped in an <ol> element.
Each item inside the lists is put inside an <li> (list item) element.

For example, figure 2.



```
<html>
<head>
<title>My first web page</title>
</head>
<body>
Your web page content goes here
<p>For example</p>
<ul>
  <li>element 1</li>
  <li>element 2</li>
  <li>element 3</li>
</ul>
</body>
</html>
```



Your web page content goes here

For example

- element 1
- element 2
- element 3

Figure 2 – Viewing the webpage

Try adding an ordered or unordered list to your example page.

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* being short for "anchor". To make text within your paragraph into a link, follow these steps (figure 3):

1. Choose some text. We chose the text "Next page".
2. Wrap the text in an <a> element, like so:

<a>Next page</a>

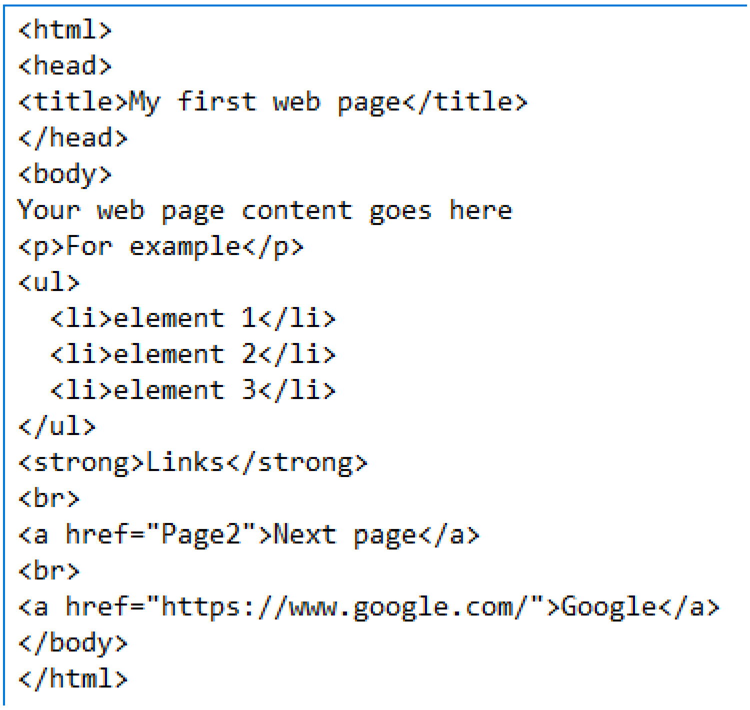
3. Give the <a> element an href attribute, like so:

<a href=""> Next page</a>

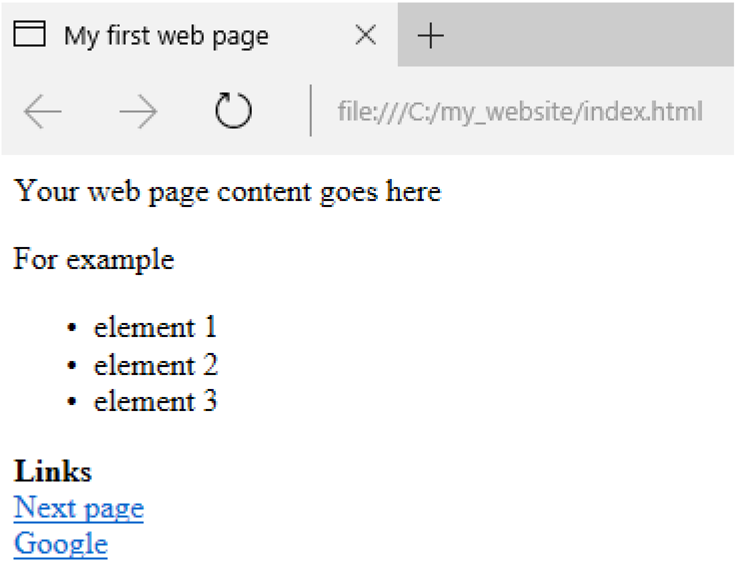
4. Fill in the value of this attribute.

<a href="Page2">Next page</a> or
<a href="https://www.google.com /">Google</a>

The HTML <br> tag is used for specifying a line break. The <br> tag is an empty tag.



```
<html>
<head>
<title>My first web page</title>
</head>
<body>
Your web page content goes here
<p>For example</p>
<ul>
  <li>element 1</li>
  <li>element 2</li>
  <li>element 3</li>
</ul>
<strong>Links</strong>
<br>
<a href="Page2">Next page</a>
<br>
<a href="https://www.google.com/">Google</a>
</body>
</html>
```



Your web page content goes here

For example

- element 1
- element 2
- element 3

Links
Next page
Google

Figure 3 – Viewing the webpage

Add a link to your page.

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

```
p {
  color: red;
}
```

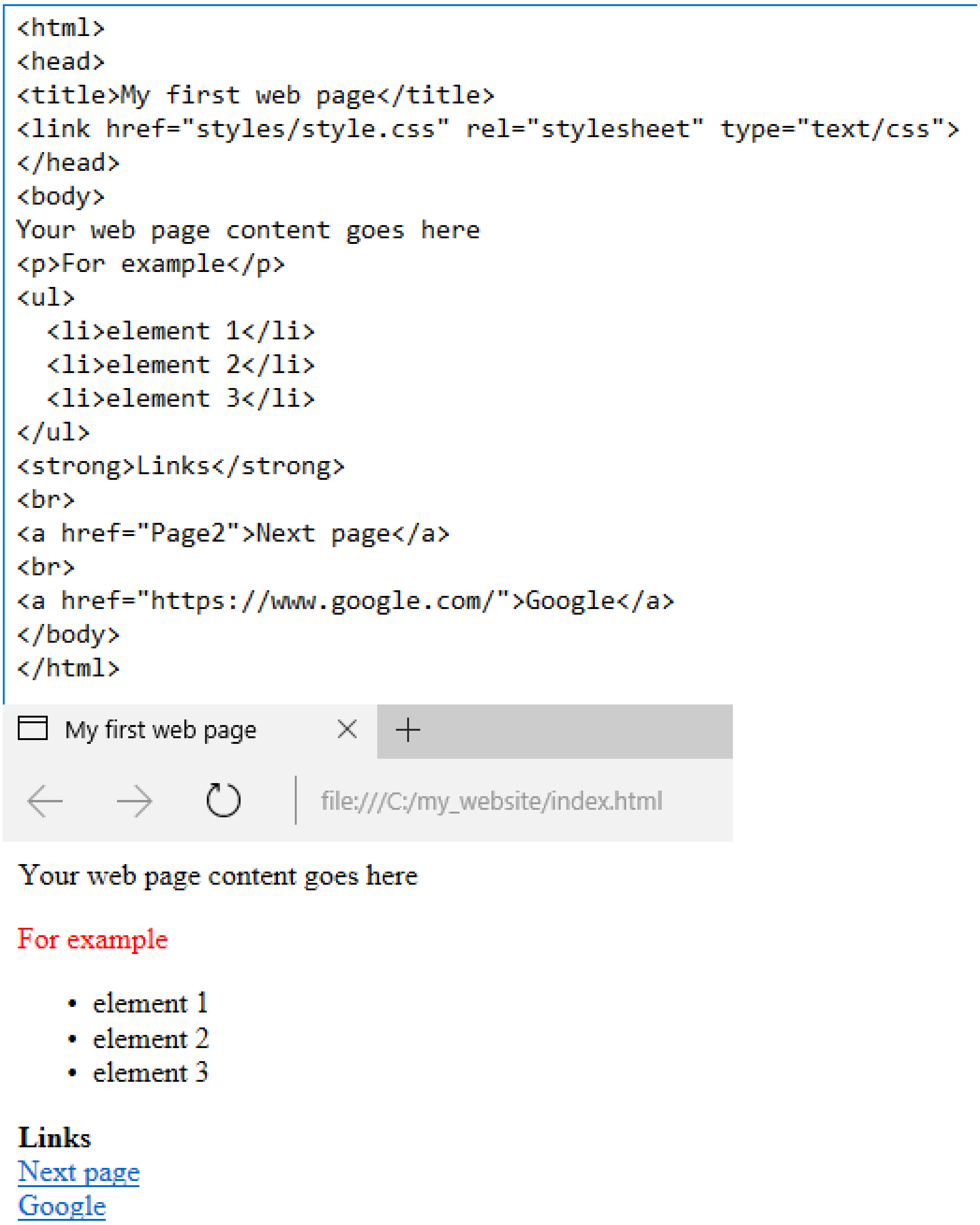
Let's try it out: paste those three lines of CSS into a new file in your text editor, and then save the file as style.css in your styles directory.

But we still need to apply the CSS to your HTML document; 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 somewhere in the head, that is, between the <head> and </head> tags:

<link href="styles/style.css" rel="stylesheet" type="text/css">

2. Save index.html and load it in your browser (figure 4).



```
<html>
<head>
<title>My first web page</title>
<link href="styles/style.css" rel="stylesheet" type="text/css">
</head>
<body>
Your web page content goes here
<p>For example</p>
<ul>
  <li>element 1</li>
  <li>element 2</li>
  <li>element 3</li>
</ul>
<strong>Links</strong>
<br>
<a href="Page2">Next page</a>
<br>
<a href="https://www.google.com/">Google</a>
</body>
</html>
```

Your web page content goes here

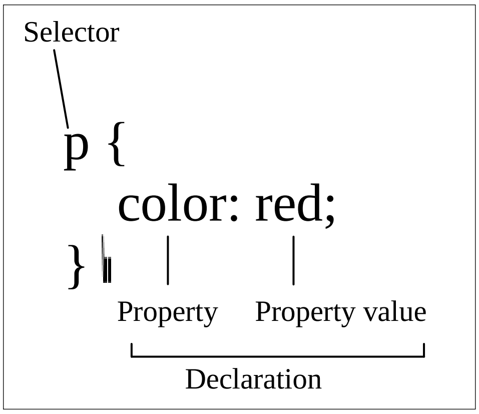
For example

- element 1
- element 2
- element 3

Links
Next page
Google

Figure 4 – Viewing the webpage

Anatomy of a CSS rule set. Let's look at the above CSS in a bit more detail:



The whole structure is called a rule set (but often “rule” for short). Note also the names of the individual parts.

Selector

The HTML element name at the start of the rule set. It selects the element(s) to be styled (in this case, p elements). To style a different element, just change the selector.

Declaration

A single rule like color: red; specifying which of the element's properties you want to style.

Properties

Ways in which you can style a given HTML element. (In this case, color is a property of the p elements.) In CSS, you choose which properties you want to affect in your rule.

Property value

To the right of the property, after the colon, we have the property value, to choose one out of many possible appearances for a given property.

Note the other important parts of the syntax:

- each rule set (apart from the selector) must be wrapped in curly braces ({});
- within each declaration, you must use a colon (:) to separate the property from its values;
- within each rule set, you must use a semicolon (;) to separate each declaration from the next one.

So to modify multiple property values at once, you just need to write them separated by semicolons, like this:

```
p {
  color: red;
  width: 500px;
  border: 1px solid black;
}
```

You can also select multiple types of elements and apply a single rule set to all of them. Include multiple selectors separated by commas. For example:

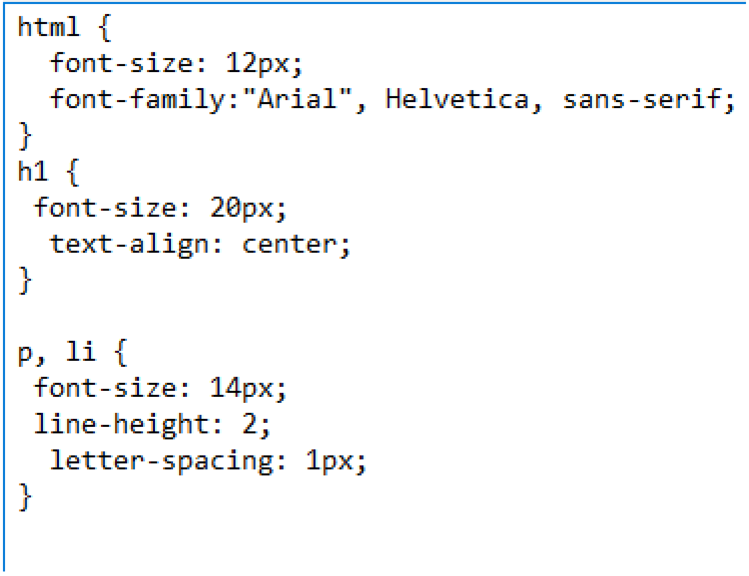
```
p,li,h1 {
  color: red;
}
```

There are many different types of selector. 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 up, let's get our fonts and text looking a little better.

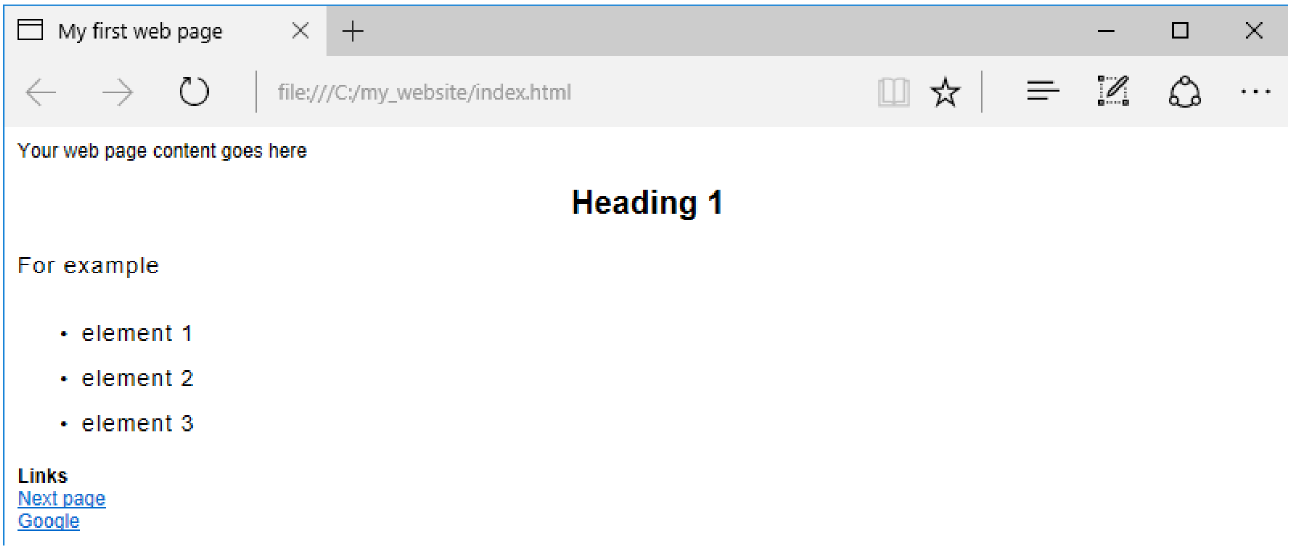
Delete the existing rule you have in your style.css file.

Add the following lines in its place and viewing the webpage (figure 5).



```
html {
  font-size: 12px;
  font-family:"Arial", Helvetica, sans-serif;
}
h1 {
 font-size: 20px;
  text-align: center;
}

p, li {
 font-size: 14px;
 line-height: 2;
  letter-spacing: 1px;
}
```



Your web page content goes here

Heading 1

For example

- element 1
- element 2
- element 3

Links
Next page
Google

Figure 5 – Viewing the webpage

Anything in a CSS document between /* and */ is a CSS comment, which the browser ignores when it renders the code. This is a place for you to write helpful notes on what you are doing.

You can adjust these px values to whatever you like to get your design looking how you want.