HTML: the language of structure

Below is the most basic, or skeleton, HTML document. All HTML documents need to include these tags. HTML files can be created in any text editor; they need the filename extension of .html. Each of the tags below may only appear once in an HTML document.

doctype
```
<!DOCTYPE html>
```

head
```
<html lang="en">
<head>
<meta charset= "utf-8" >
<title>Title of document goes here</title>
</head>
```

body
```
<body>
```
The content of your document goes here.
```
</body>
</html>
```

Creating a Web Page
You can use any program capable of creating a plain text file, such as Atom, which is free.

HTML files must end with the extension .html. Your home page (main page) must be called index.html or so the server knows that is the site's home page.

To simplify the organization of your files, keep the HTML files and any graphics in the same folder. Graphics can only be GIF (.gif) PNG (.png) or JPEG-compressed (.jpeg or .jpg) files. Since the files are only to be displayed on screen, they don't really have any physical size. Most computers display at least 1024 pixels horizontally and 800 pixels vertically; the majority have much higher resolutions. Make sure your images are sized appropriately for the web. Measure your graphics only by their width and height in pixels. 960 pixels wide is a good starting point. More complex techniques are needed for web sites that can adapt to a range of device sizes, like phones and tablets.

More HTML
Paragraphs should begin with the `<p>` tag and end with the `</p>`.

You can create headings by wrapping text in header tags:
`<h1>`This is a heading.`</h1>`
There are 6 levels of headings; `<h1>` is the most prominent, `<h6>` the least prominent.

Use the `<hr>` tag to insert a horizontal line. The `<hr>` does not need a closing tag. Use `<strong>` and `</strong>` for bold and `<em>` and `</em>` for italic.

To include an image, use the `img` tag. Make sure you put the name of your graphic file inside the quotation marks. "filename.jpeg" is just an example. Remember, GIF, PNG and JPEG files only.
```
<img src="filename.jpeg" alt="Description of image">
```

To create links to other files, use the `href` tag. Put the file you wish to link to in that same folder as the file you wish to link from. Remember, "filename.htm" is just the example.

`<a href="filename.html">`Here is the text which will serve as a link.`</a>`

To create links to other web sites, use the `href` tag but include the address (URL) of the web site in quotation marks.

`<a href="http://www.nscad.ca/">` Here is the text which will serve as a link.`</a>`

To use an image as a link (to make a button) simply put the `img` tag between the `<a href>` and `</a>` tags.

To create a list, use the `<li>` (list item) tags along with the `<ul>` or `<ol>` (unordered and ordered list) tags:

```
<ul>
<li> item 1 </li>
<li> item 2 </li>
<li> item 3 </li>
</ul>
```

To create comments, notes that are not displayed in your web browser, use the comment tags:

`<!--   comment text goes here   -->`

To find out about more tags, go to:
`http://www.w3schools.com/`

Make sure you are using resources for HTML5 and not other versions. Once you have mastered the basics, you only need to learn new tags to accomplish new effects. Use the "View Source" command in your web browser's View menu to see how other pages are created.

You can check the syntax of your HTML online at http://validator.w3.org/ but you will find some of the error messages hard to understand at first.

Once complete, the last stage would be to upload your files to a Web server. Server space can be rented; some organizations provide it for free, but often with strings attached. Use an FTP (File Transfer Protocol) program to copy the files up the server; I suggest Fetch.

Formatting text and positioning elements

While tags exist for formatting text and positioning elements such as text and graphics, avoid using them. Modern web pages should use CSS, Cascading Style Sheets, instead.

Troubleshooting

• Are your HTML files and images in the same folder?
• Are your filenames free of punctuation and spaces?
• Are your images saved as PNG or JPEG files?
• Have you spelled filenames exactly the same way as you named the files?
• Have you saved and reloaded changes?
• **Test all links and images before submitting any work.**

CSS: The language of style

The World Wide Web was originally envisaged as a way to present information simply, without elaborate formatting. As the web became popular, web browser manufacturers began adding HTML codes to control colours, typefaces and layout. However, with browser manufacturers inventing their own codes, the HTML language became polluted with incompatible formatting tags.

To order to this chaos, a new approach was needed. HTML would be kept as a structural language, and new language developed to handle presentation. This was CSS, or Cascading Style Sheets. Style sheets would control formatting, colour, positioning and other presentational aspects, while HTML would define the document's underlying structure.

One of the most powerful features of CSS is the ability to control the appearance of any number of HTML files from a single CSS style sheet. This allows the formatting for an entire web site to be changed by changing the linked style sheet.

Style sheets are collections of rules. Each rule controls the appearance of a specific part of a web page. Every rule has the same basic structure. Here is an example:

```
h1 {font-size: 100px; color: #FF0000;}
```
Note the colour codes: use the web colour value from Photoshop's colour picker. The example above uses hexadecimal (base16) notation. You can also use decimal RGB notation, shown below:
```
h1 {font-size: 100px; color: rgb(255,0,0);}
```

The selector
The selector specifies what elements the rule will apply to. In the example above, the rule will apply to all the H1 (heading level 1) tags in the HTML document.

Following the selector are one or more declarations. Declarations describe the style that will be applied to the elements specified by the selector. Declarations take the form of two parts: a property and a value. Properties describe what we can style, such as colour, typeface and size. Values describe how they will be styled. In the example above, the property color makes it clear that we are talking about the size of the h1 element. The value 100px says that the size of the h1 will be 100 pixels.

A rule may have multiple declarations, as in the example. Far more complex selectors are possible, too.

Using style sheets
When using style sheets, one of the decisions you will have to make is where to put the rules. There are a number of options, but only two of these are commonly used.

Option 1: in a separate linked file (best)
By placing your style sheet in a separate file, you need to only have a single style sheet for the entire web site. You do this by linking every page in your site to the css file, as follows:
```
<link href="global.css" rel="stylesheet">
```
In our examples, the file "global.css" contains the style rules that govern all pages linked to it. Make sure only style rules, and not HTML tags,are in your .css file. The contents of the file will look like this:

```
body
{
     font-family: "lucida grande", verdana, sans-serif;
     color: #555555;
}
h1
{
     font-size: 100px;
     color: #EE11EE;
     border-style: solid;
     border-width: 1px;
     border-color:#FF0000;
}
```

...and so forth, for all the rules you choose to define.

Option 2: within the <head> element (avoid in most cases)
You can choose to put your style sheet in the <head> element of your HTML document, as follows:

```
<style>
h1
{
    color: #FF0000;
    font-size: 30px;
}
</style>
```

This is the simplest method, but means that each HTML document will need to have its own style sheet. Since having one style sheet to control an entire web site's appearance is one of the main advantages of CSS, this method is usually used only by beginners looking for a quick way to test CSS, or for one-of-a-kind pages that use different formatting than other pages in the site.

Selectors
CSS allows a number of different ways of specifying which rules are to be applied to which elements. The simplest is the element selector. Element selectors specify the style for all instances of an HTML tag. Any HTML tag can serve as an element selector. For example, if you wish all the paragraphs of text to be blue, you could use:

```
p {color: #000099;}
```
as the style rule.

However, by using only element selectors, you would have no way to make one paragraph on a blue and the remainder grey, since the element selector applies to all matching elements.

Getting specific: class and id selectors
One way of specifying which elements you wish to style is the class selector. First, define a class in your style sheet:

```
.highlight {background-color: #FFFF00;}
```

This creates a class called highlight. Note the leading dot. Class names are up to you to create - you can call the class anything, as they have no predefined meaning.

Then, use the class property in your HTML code to specify how to style that particular element
```
<p class="highlight">
```A paragraph controlled by the "highlight" style```</p><p>``` and now some text that looks like all the other paragraphs.```</p>``` *Note the lack of the dot in the class name when used in your HTML code.*

Any element tagged as a member of the "highlight" class will be assigned the colour blue. ```<hr class="highlight" />``` for example, will create a blue horizontal rule.

To be even more specific, use id selectors. The syntax is similar. Define an id in your style sheet:
```
#menu {color: #FF0000;}
```
*note the # sign*

Then, in your HTML, tag an element with that id:
```
<p id="menu">
```A paragraph controlled by the "menu" style```</p><p>``` and now some text that looks like all the other paragraphs.```</p>```

Use class when more than one element will have that style applied to it. Use id when you want to identify a unique element on a page.

CSS basics: type
Some sample property/value pairs:

Controlling type
```
font-family: Arial, Helvetica, lucida, sans-serif;
font-size: 10px;
font-weight: bold;
```

Controlling colour
`color: #FF0000;`   *Changes colour of text or other elemements - you can look up the colour codes in Photoshop's colour picker.*

`background-color: #FFFFAA;`   *Changes the background colour of an element*

Controlling borders
```
border-width: 4px;
border-style: dotted; (some options are none/dotted/dashed/solid/double
```

Applying styles
To apply a style to a range of text, such as a phrase, use the span tag with an associated style.
This is `<span class="highlight">` some highlighted text`</span>` using the span tag with a style, class or id.

To apply a style to a range of other elements, such as multiple paragraphs of text, use the div tag. You can position elements by wrapping them in `<div>` tags, and using the positioning options shown above. A div creates a container in which text or images can be treated as a single unit. In the example below, the div with the id of "content" would be all have a background colour of light brown.

HTML:

```
<div class="content">
<h2>Choose a section</h2>
    <ul>
    <li><a href="1000-1.html">DSGN 1000-1: Thursdays</a></li>
    <li><a href="1000-2.html">DSGN 1000-2: Fridays</a></li>
    </ul>
</div>
```

CSS:
```
.content {

background-color: #F8F5E1;
}
```

Resources

http://wendypeck.com/css101.html
https://developer.mozilla.org/en-US/docs/Learn/CSS
https://www.w3schools.com/css/default.asp