

Introduction to HTML

HTML, which stands for Hypertext Markup Language, is the standard markup language used to create web pages.

HTML consists of *HTML elements*, which define the different parts of the document. The web developer uses *HTML tags* to specify which elements go on a given page. For example, you can use HTML tags to specify headings, paragraphs, lists, tables, images and much more.

What do I need to create HTML?

You don't need any special equipment or software to create HTML. In fact, you probably already have everything you need. Here is what you need:

- Computer

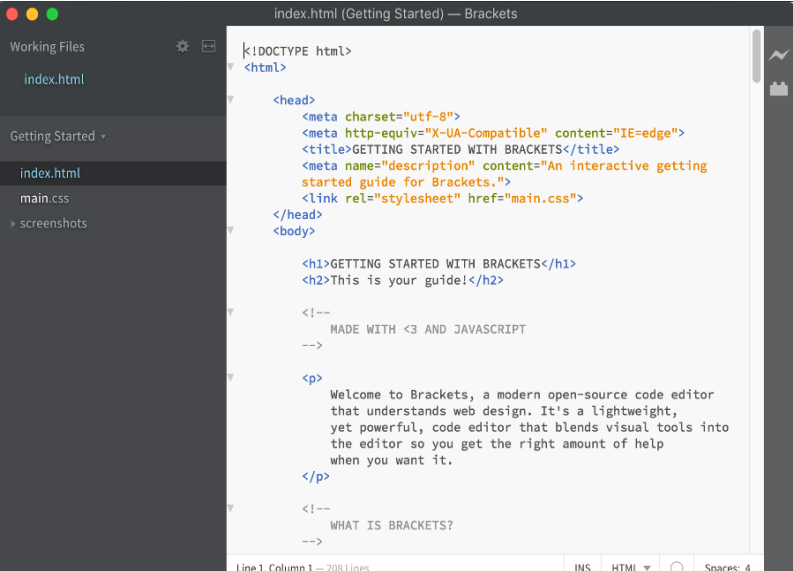
This is probably an obvious one. Tablets should be fine too, and maybe even your smartphone.

- HTML Editor

An HTML editor is an application that helps you create HTML pages.

While you can certainly create HTML documents using a simple text editor, an HTML editor will make your life easier — especially if you plan to write a lot of code.

HTML editors usually highlight your code so that it's easier to read, and they will typically have an autocomplete feature that picks up on what you're typing, and makes valid HTML suggestions.

A screenshot of the Brackets code editor interface. The window title is "index.html (Getting Started) — Brackets". On the left, a sidebar shows a "Working Files" pane with "index.html" selected, and a "Getting Started" pane with "index.html", "main.css", and "screenshots". The main editor area displays HTML code with syntax highlighting. The code includes a DOCTYPE declaration, a head section with meta tags for charset, UA compatibility, title, and description, and a link to a stylesheet. The body contains a heading, a sub-heading, a comment, and a paragraph of text. The status bar at the bottom indicates "Line 1, Column 1 — 208 Lines" and "INS HTML Spaces: 4".

```
<!DOCTYPE html>
<html>

  <head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <title>GETTING STARTED WITH BRACKETS</title>
    <meta name="description" content="An interactive getting
started guide for Brackets.">
    <link rel="stylesheet" href="main.css">
  </head>
  <body>

    <h1>GETTING STARTED WITH BRACKETS</h1>
    <h2>This is your guide!</h2>

    <!--
      MADE WITH <3 AND JAVASCRIPT
    -->

    <p>
      Welcome to Brackets, a modern open-source code editor
      that understands web design. It's a lightweight,
      yet powerful, code editor that blends visual tools into
      the editor so you get the right amount of help
      when you want it.
    </p>

    <!--
      WHAT IS BRACKETS?
    -->
```

Free HTML Editors

Many HTML editors cost money, but there are also a few free ones around.

Examples include Brackets (pictured), Notepad++, SeaMonkey, CoffeeCup free version and TextPad.

Or a Text Editor

If you don't have an HTML editor, and you don't want to download one just now, a text editor is fine. Most computers already have a text editor. Examples of text editors include Notepad, (for Windows), Pico (for Linux), or Simpletext/Text Edit/Text Wrangler (Mac).

- Web Browser

To view the results of your HTML code.

OK, technically, you can create HTML without viewing the results, but where's the fun in that!

Do I need to be online?

No, you do not need to be online to create web pages. You can create web pages on your local machine. You only need to go online when you want to publish your web page to the web .

The next lesson will show you how to create a web page in less than 5 minutes.

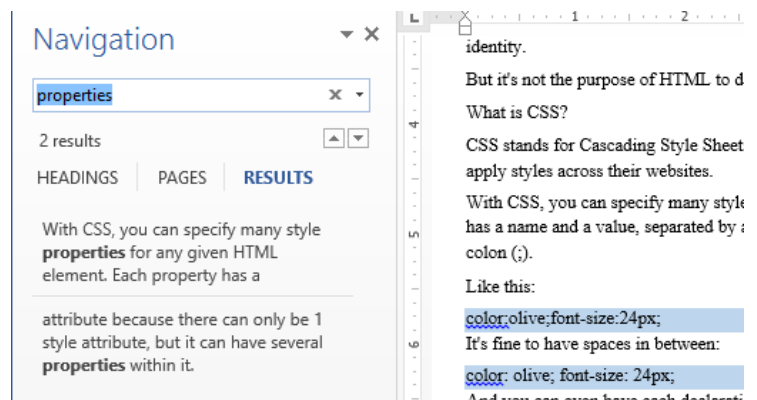
About HTML

HTML stands for Hypertext Markup Language, and it's the standard markup language used to create web pages.

HTML is a subset of Standard Generalized Markup Language (SGML) and is developed by the World Wide Web Consortium (W3C) and Web Hypertext Application Technology Working Group (WHATWG).

Tool Tip

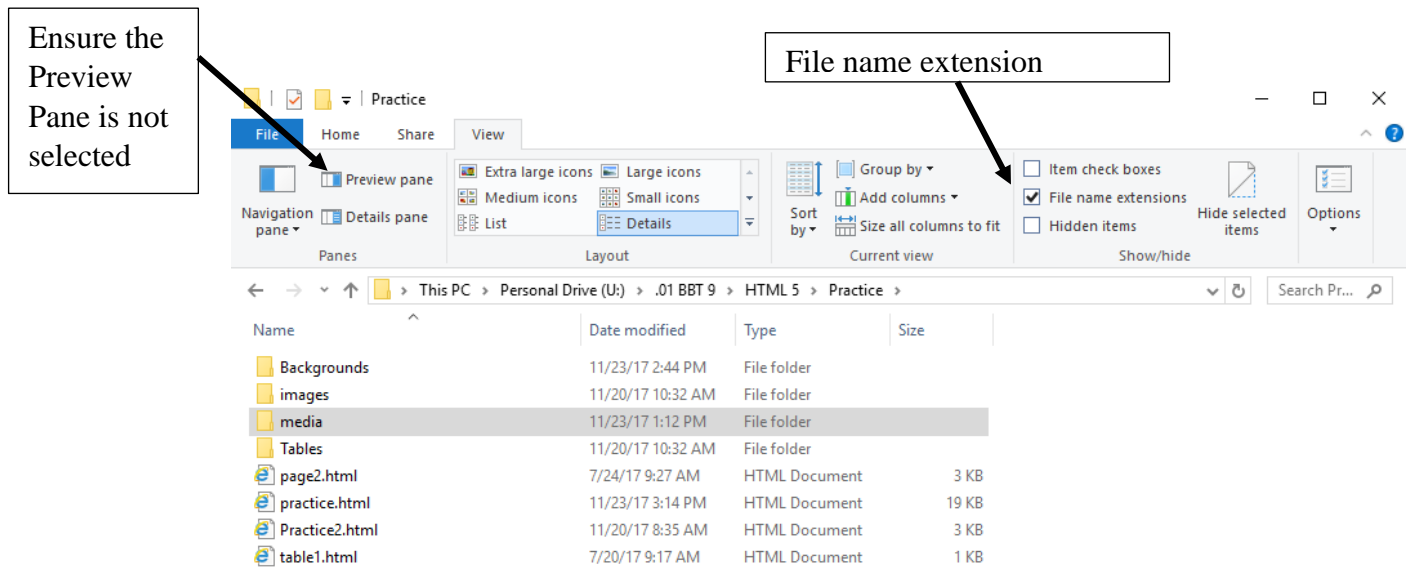
This module will present you with a number of new terms and techniques which you may not be able to remember when you need them. Luckily, there is assistance in the form of a constant search box. All you need to do is press the **ctrl** and **F** keys at the same time and a “find” search tool opens – simply type your search term in the box and the program will take you to the next time that term is used in the document. This tool works on most text based documents (Word, pdf, and internet applications).



HTML Tips

1. Before you begin HTML ensure that your folder options are set so that the preview pane is not selected. To check go to your folder button from your task bar, then select the view tab and turn it off. This is important because it can interfere with your ability to save files.
2. While in the view tab select the “file name extensions check box. This will show the extensions in your folders. This is helpful when coding so you can see the file whole file name.





Unit 1

Lesson 1 - Create your First Web Page!

Let's create a webpage using HTML, then view it in a web browser.

First let's create a folder named HTML. Then create a folder inside this folder named Practice.

All the files you need to create your webpage need to be in the same folder – so this webpage files will all be inside your Practice folder.

OK, let's get straight into it. Here, you will learn just how easy it is to create a web page. In fact, by the time you've finished with *this* Module, you will have created *your own* web page!

When you create a web page you will usually do something like this:

1. Create an HTML file
2. Type some HTML code
3. View the result in your browser
4. Repeat the last 2 steps (if necessary)

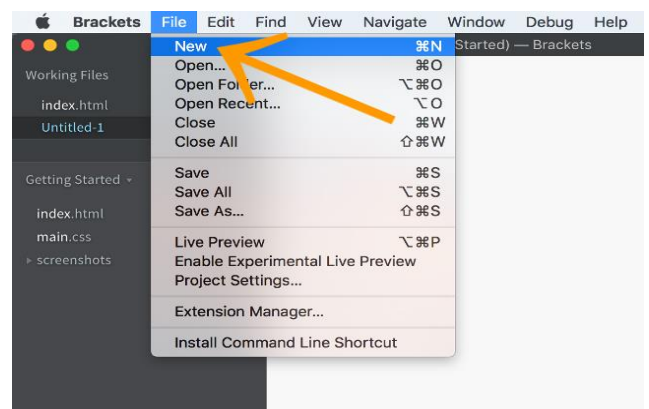
Create a Webpage

OK, let's walk through the above steps in more detail.

1. Create an HTML file

An HTML file is simply a text file saved with an .html extension (i.e. as opposed to a .txt extension).

1. Open up your HTML editor (such as Brackets or Notepad++). You could use your computer's normal plain text editor if you prefer.



2. Create a new file (if one wasn't already created)
3. In your BBT folder of your U drive, create a new folder named practice and Save the file as practice.HTML inside the folder named practice.

2. Type some HTML code

Type the following code into your editor (Brackets or Notepad++): (Pay attention to upper and lower case type)

```

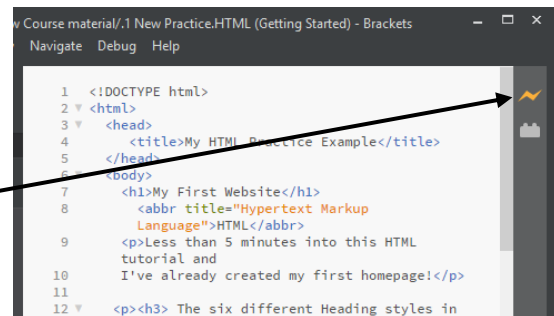
1 <!DOCTYPE html>
2 <html>
3 <head>
4,5 <title>My HTML Practice Example</title>
6 </head>
7 <body>
  <h1>My First Website</h1>
  <p>Less than 5 minutes into this HTML tutorial and
  I've already created my first homepage!</p>
8 </body>
9 </html>

```

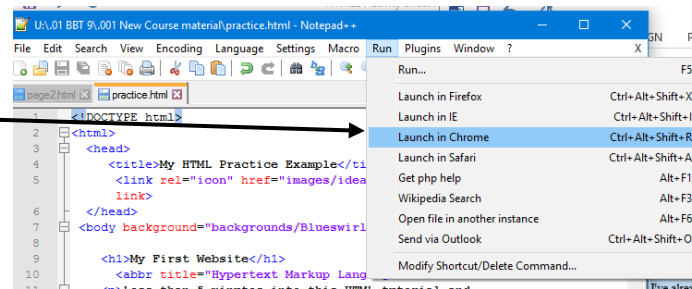
These are the 9 basic tags required to create a webpage in HTML 5. (Hint these 9 are required to create a second and third page – you will need to remember this later)

3. View the result in your browser

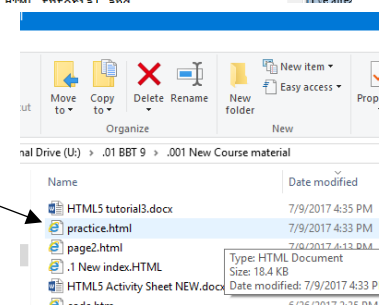
If you are using Brackets you can simply save your file as **Practice1.html** in your BBT folder and click the lightning bolt in the top right portion of your Brackets window – a browser will open displaying the result of your coding.



If you are using Notepad++ you can select the Run command from the main window and select the launch in Chrome option



Or you can navigate to your file in your U drive (practice1.html) then double click on the html file.



4. Repeat the last 2 steps until you're satisfied with the result

It's unrealistic to expect that you will always get it right the first time around. Don't worry — that's OK! Just try again and again — until you get it right.

Part of the fun of creating web pages is experimenting with different elements, styles, approaches, etc, and then checking out the result.

Explanation of code

OK, before we get too carried away, I'll explain what that code was all about.

We just coded a bunch of HTML tags. These tags tell the browser what to display and where. You may have noticed that for every "opening" tag there was also a "closing" tag, and that the content we wanted to display appeared in between. Most HTML tags have an opening and closing tag.

The next lesson goes into a bit more detail about HTML tags.

Lesson 2 - HTML Elements

HTML elements are the fundamentals of HTML. HTML documents are simply a text file made up of HTML elements. These elements are defined using HTML tags.

An HTML element usually consists of a **start** tag and **end** tag, with the content inserted in between:

```
<tagname>Content goes here...</tagname>
```

The HTML **element** is everything from the start tag to the end tag:

```
<p>My first paragraph.</p>
```

An *HTML element* is an individual component of an HTML document. Any given web page consists of many HTML elements.

HTML tags tell your browser which elements to present and how to present them. Where the element appears is determined by the order in which the tags appear.

HTML consists of over 100 tags. Don't let that put you off though - you will probably find that most of the time, you only use a handful of tags on your web pages.

(You can find a list of tags and how to use them here: <https://www.quackit.com/html/tags/>)

OK, lets look more closely at the example that we created in the previous lesson.

```
<!DOCTYPE html>
<html>
<head>
<title>My HTML Practice Example</title>
</head>
<body>
<h1>My First Website</h1>
<p>Less than 5 minutes into this HTML tutorial and
I've already created my first homepage!</p>
</body>
</html>
```

Show your teacher your first web page and have them initial your Activity Sheet!

Explanation of the above code:

- The [!DOCTYPE...](#) declaration tells the browser which version of HTML the document is using.
- The [html](#) element is the document's *root* element - it can be thought of as a container that all other tags sit inside (except for the !DOCTYPE declaration).
- The [head](#) tag contains information that is not normally viewable within your browser (such as meta tags, JavaScript and CSS), although the [title](#) tag is an exception to this. The content of the title tag is displayed in the browser's title bar (right at the very top of the browser).
- The [body](#) tag is the main area for your content. This is where most of your code (and viewable elements) will go.
- The [h1](#) tag defines a level 1 heading.
- The [p](#) tag defines a paragraph. This contains the body text.

Did you notice how the <title> element is inside the <Head> element and the <p> element is inside the <body> element? That is called nesting elements and it is a very essential part of the HTML structure. I will leave it at this for now, but I just wanted to introduce to you the idea of nesting elements.

(See: <http://www.html5-tutorials.org/introduction-to-html/your-first-webpage/>)

And: https://www.w3schools.com/html/html_elements.asp)

Closing your Tags

As mentioned in a previous lesson, you'll notice that all of these HTML elements have opening and closing tags, and that the content of the element is placed in between them. There are a few exceptions to this rule.

You'll also notice that the closing tag is slightly different to the opening tag — the closing tag contains a forward slash (/) after the <. This tells the browser that this tag closes the previous one.

Indents and Carriage Returns

You can indent the code and add extra carriage returns if you like. This can help with readability, and it doesn't affect the way the page is displayed in the browser.

In fact, code indenting is a universal practice in computer programming circles, and most HTML editors automatically indent the code for you as you type. (NB: using Notepad++ go to the settings menu, select preferences, and in New Document, select the default language as HTML).

For example, we could take the above example, and indent it like the following example, and it will still appear the same in the browser.

```
<!DOCTYPE html>
<html>
  <head>
    <title>My HTML Practice Example</title>
  </head>
  <body>
```

```
<h1>My First Website</h1>
<p>Less than 5 minutes into this HTML tutorial and
I've already created my first homepage!</p>
</body>
</html>
```

UPPERCASE or lowercase?

You can use uppercase or lowercase when coding HTML, however, HTML 5 requires all lowercase and most developers use lowercase. This helps the readability of your code, and it also saves you from constantly switching between upper and lower case.

Therefore...

- **Good:** <head>
- **OK:** <HEAD>

In the next lesson, we learn about some of the more common formatting tags.

What you have learned so far:

- Always declare the doctype - <!DOCTYPE html>
- Elements require opening and closing tags
- Use the element <html> to identify the start and end of your HTML programming
- Use the element <head> to separate the information about your page (and titles)
- Use your <title> element to display your webpage name on the title bar
- Use the element <body> to display all the content of your webpage
- Indenting your markup makes it easier to comprehend
- You can nest one element inside another element

*Unit 1 Activity Sheet response:

Show your teacher your HTML editor and first webpage displayed in your browser (Internet Explorer or Chrome).

Activity 1 – Review Questions (Answer in your Activity Sheet)

Now it's time to review what you've learned in this activity. Enter the letter of the correct answer on your Activity Sheet.

1. What does HTML stand for?

- A. Hot Mail
- B. Hypertext Mark-up Language
- C. Hypertext Made-up Language
- D. Hypertext Mail Language

2. What is another name for HTML tags?

- A. Hypertexts
- B. Browsers
- C. Elements
- D. Links

3. The opening and closing tags for the title of a web page, in order, are:

- A. <title> and </title>
- B. </title> and <title>
- C. title and /title
- D. <title> and </head>

4. In the correct order the nine tags that are the basics for an HTML Web page are:

- A. <doctype html> <head> </head> <title> </title> <body> </body> </html> </doctype>
- B. <html> <!doctype > <body> <head> <title> </title> </head> </body> </html>
- C. <!doctype html> <html> <head> <title> </title></head> </html> <body> </body> </doctype html>
- D. <!DOCTYPE html> <html> <head> <title> </title> </head> <body> </body> </html>

5. What is a file extension?

- A. A period and letters added to the end of a file name to identify the type of file it is.
- B. The name of a folder.
- C. The name of a text editor program.
- D. An addition to the name of a file so that it is different from another file of the same name.

Unit 2

HTML Formatting and Semantics

Here, we look at formatting and semantics within HTML documents.

Semantics refers to the use of HTML markup to reinforce the meaning, of the information in webpages and web applications rather than merely to define its presentation or look.
(https://en.wikipedia.org/wiki/Semantic_HTML)

HTML provides the structure of the document (consisting of all the individual HTML elements on the page).

HTML also provides the semantics of an HTML document. Elements are generally used for a particular meaning. For example, a heading level 1 is more important than a heading level 2.

Most formatting and styling on web pages is done using [CSS](#). Browsers format certain HTML elements in a standard way without the need for CSS. However, CSS is becoming the international standard for formatting instructions on webpages and deprecated elements (those used within HTML) are being discouraged. Therefore if you are to be doing webpage design learning CSS is encouraged. Learning CSS, however, is too large a scope for this module, but we will introduce its basics later in the module.

Below are examples of common HTML elements, with an explanation of their usage, as well as an example of their basic format when applied to an HTML document. Many of these elements are being discouraged in favor of using CSS to format HTML, however they are all widely supported and will still function.

Headings

Headings Are Important - Search engines use the headings to index the structure and content of your web pages. Users skim your pages by its headings. It is important to use headings to show the document structure.

<h1> headings should be used for main headings, followed by <h2> headings, then the less important <h3>, and so on.

****NB:** Use HTML headings for headings only. **Don't use headings to make text BIG or bold.**

There is a special tag for specifying headings in HTML. There are 6 levels of headings in HTML ranging from <h1> for the most important, to <h6> for the least important.

They are listed below. **Type this code in your web page above the closing body tag (</body>).**

```
<p><h3> The six different Heading styles in HTML are:</h3>
<h1>Heading 1</h1>
<h2>Heading 2</h2>
<h3>Heading 3</h3>
<h4>Heading 4</h4>
<h5>Heading 5</h5>
<h6>Heading 6</h6></p>
```

The <p> Element

The <p> element adds a paragraph to the body of your text. It adds white space as a margin at the top and bottom of the paragraph. It ignores spacing in the coding when it displays on the browser. Place the code below, under your last paragraph but before your closing </body> tag.

```
<p>
This paragraph
contains a lot of lines
in the source code,
but the browser
ignores it.
</p>
<p>
This paragraph
contains    a lot of spaces
in the source    code,
but the    browser
ignores it.
</p>
```

The Poem Problem

This poem will display on a single line: Enter the code below above the closing body tag (</body>).

```
<p>
My Bonnie lies over the ocean.
My Bonnie lies over the sea.
My Bonnie lies over the ocean.
```

```
Oh, bring back my Bonnie to me.
</p>
```

The HTML `<pre>` Element

The HTML `<pre>` element defines preformatted text.

The text inside a `<pre>` element is displayed in a fixed-width font (usually Courier), and it preserves both spaces and line breaks: Enter this code above the closing body tag (`</body>`).

```
<pre>
My Bonnie lies over the ocean.

My Bonnie lies over the sea.

My Bonnie lies over the ocean.

Oh, bring back my Bonnie to me.
</pre>
```

The `` Element

To place a strong importance on a piece of text, use the `` element.

Place the code below, under your last paragraph but before your `</body>` tag.

```
<p><strong>Attention:</strong> Please leave the door open.</p>
```

****NB:** This element replaces the older `` element. But the `` element is still widely supported, so it is acceptable to use `` instead of `` in this class as all our browsers will support the element. Try it: Place the following line below the previous one –

```
<p><b>Attention</b> Please leave the door open.</p>
```

The `` Element

You can place an emphasis on text by using the `` element.

Place the code below, under your last paragraph but before your `</body>` tag.

```
<p>Strawberries are absolutely <em>delicious</em>!</p>
```

****NB:** This element replaces the older `<i>` element. But the `<i>` element is still widely supported, so it is acceptable to use `<i>` instead of `` in this class as all our browsers will support the element. Try it: Place the following line below the previous one –

```
<p>Strawberries are absolutely <i>delicious</i>!</p>
```

The `<mark>` Element

You can mark or highlight text by using the `<mark>` element.

Place the `<mark>` element **within the previous paragraph like this:**

```
<p>Strawberries are absolutely <em><mark>delicious</mark></em>!</p>
```

The `<big>` Element

The `<big>` element can be used to make certain text one size larger than the surrounding text.

This can be used to make a word or phrase stand out from the text. As mentioned earlier use this tag instead of a heading tag. Add the `<big>` tag into your previous paragraph in your HTML editor:

```
<p>Strawberries are absolutely <big><em>delicious</em></big>!</p>
```

****NB:** this tag is obsolete in HTML 5, but still is widely supported. You can add them together to increase the size of the text another size. Try adding another `<big>` to your Strawberries `<p>`.

```
<p>Strawberries are absolutely <big><big><em>delicious</em></big></big>!</p>
```

The `<small>` Element

The `<small>` element can be used to make certain text one size smaller than the surrounding text. This can be used to make a word or phrase stand out from the text. As mentioned earlier use this tag instead of a heading tag. Type a new strawberries `<p>` below your previous one and add the `<small>` tag:

```
<p>Strawberries are absolutely <small><em>delicious</em></small>!</p>
```

****NB:** this tag is obsolete in HTML 5, but still is widely supported. You can add them together to decrease the size of the text another size. Try adding another `<small>` to your Strawberries `<p>`. (do you think this small change effects the meaning of the sentence?)

```
<p>Strawberries are absolutely <small><small><em>delicious</em></small></small>!</p>
```

The `<q>` and `<blockquote>` Elements

The `<q>` tag is used for short quotations and browsers usually insert quotations around the `<q>` element. Type the following code below the previous paragraph.

```
<p>WWF's goal is to: <q>Build a future where people live in harmony with nature.</q></p>
```

The `<blockquote>` element defines a section that is quoted from another source and browsers usually indent `<blockquote>` elements. Type the following code below the previous paragraph.

```
<p>Here is a quote from WWF's website:</p>
```

```
<blockquote cite=http://www.worldwildlife.org/who/index.html>
```

```
For 50 years, WWF has been protecting the future of nature. The world's leading conservation organization, WWF works in 100 countries and is supported by 1.2 million members in the United States and close to 5 million globally. </blockquote>
```

****NB:** Whenever using abbreviations it is a good idea to insert the `<abbr>` tag with a title attribute (more on this later) like this:

The **Title attribute** shows the viewer text if they hover over the abbreviation. **Change the code in your editor to look like the code below.** Save and refresh your browser and hover your mouse over the abbreviation WWF – you should see a tooltip that provides the meaning for the abbreviation.

```
<p>Here is a quote from <abbr title="World Wildlife Fund">WWF's</abbr> website:</p>
```

****NB:** There is no need to do this for every occurrence of the abbreviation, however, you should do it if the abbreviation appears on different pages within your website because viewers might not view all your pages.

The Comment Element

You can add comments to your HTML source by using the following syntax:

```
<!-- Write your comments here --> (Please note, the 2 dashes are required)
```

Notice that there is an exclamation point (!) in the opening tag, but not the closing tag. Comments are not displayed by the browser, but they can help document your HTML source code. They can be used by the original designer as aids to help with a future task, and they can also be useful to other designers who may want to view your source code – and comments.

Hint: You can use comments and extra spaces at the start of each new activity in your editor to assist you if you should have to come back and look for a mistake or to make changes.

Type the following into your HTML editor below the blockquote but before the closing body tag (don't forget to include the 2 dashes -- they are required):

```
<!-- This is a comment -->
<p>This is a paragraph.</p>
<!-- Remember to add more information here -->
```

When you save and refresh your browser all that is seen is the “this is a paragraph”

Line Breaks

You can force a line break by using the `
` element.

Place the code below, under your last paragraph but before your `</body>` tag.

```
<p>Here is a <br>line break.</p>
```

****NB** another way to create space on your webpage is by using the ` ` which is a non breaking space. You can use this between paragraphs, or to indent lines of text. **Try it now – go to the strawberries line** (`<p>Strawberries are absolutely delicious!</p>`)

and place five spaces before the `<p>` element like this:

```
&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;<p> Strawberries are absolutely <em>delicious</em>!</p>
```

Save your changes and refresh your web page -- What happens?

You should notice that you added five line spaces between the line that reads: **Attention** Please leave the door open. And the Strawberries line.

Now highlight the five spaces in your text editor (` `) and copy them then paste them to the right of the `<p>` tag. Save your changes and refresh your web page – what happens?

You should notice that you added some space between the left margin and the start of your Strawberries line in your webpage. **Your code should look like this:**

```
<p>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;Strawberries are absolutley <em>delicious</em>!</p>
```

Thematic Break

You can create a paragraph-level thematic break by using the `<hr>` element. Therefore, this element allows you to separate different topics into logical groups.

In early versions of HTML, this element represented a "horizontal rule". However, HTML5 gave it a specific semantic meaning (it represents a paragraph-level thematic break).

Place the code below, under your last paragraph but before your `</body>` tag.

```
<p>Here's one topic... </p>
<hr>
<p>Here's another topic.</p>
```

There are also ways to enhance the look of these horizontal rules using *attributes*. You can add thickness, and shadow effects. Later we will also look at using images as horizontal and thematic breaks. The process for adding this element is: type `<hr` where you want it to appear in your webpage. Then if desired choose a size type `size="n"` where n is the rule's height in pixels. And if desired add a `width="w"` where w is the width of the rule in pixels or as a percentage of the documents width. You can also align the rule to the left, right or center of the page by adding the align attribute thus: `align="direction"` where direction refers to left, right, or center, however, this attribute only applies if you have made the rule narrower than your page. Also if you want a solid line add `noshade="noshade"`

Let us try a few examples:

```
<hr size=12 width=50% align=right noshade="noshade"></hr>
```

This example will place a thick rule on the right side of your webpage

```
<hr size=8 width =50% align=left noshade="noshade"></hr>
```

This example will place a heavy line on the left side of the webpage

```
<hr size=4 width=75% align=center noshade="noshade"></hr>
```

This example will place a heavy line in the center of the webpage

Place at least 4 more examples of your own coding choice under the provided examples.

Show your teacher your 4 horizontal rules and have them initial your Activity sheet!

In this activity we have learned about elements (tags) that change the appearance of text on our webpage. **Go back to the poem and change it using the elements below** to discover more options for designing your webpage.

```
<hr><p>
  <tt>My<i> Bonnie </i>lies over the ocean.
  My <i>Bonnie</i> lies over the sea.
  My <i>Bonnie</i> lies over the ocean.
  Oh, bring back my <big><b><i>Bonnie</i></b></big> to me.</tt>
</p><hr>
```

Notice that the `<tt>` element changes the font to a typewriter font.

Activity 2 – Review Questions (Answer in your Activity Sheet)

Enter the letter of the correct answer on your Activity Sheet.

1. The heading level that is the biggest is:

- A. H1 B. H3 C. H5 D. H6

2. The heading level that is the smallest is:

- A. H1 B. H3 C. H5 D. H6

4. Headings are used to:

- A. make the Web page longer.
 B. confuse the user by making information harder to find.
 C. help organize text and images on the Web page.
 D. create Web pages.

5. Horizontal rules can be used for decoration or for thematic purpose in a website. Choose the correct method for inserting one into a webpage:

- A. <HR>
- B. <hr> </hr>
- C. <hr>
- D. hr/;>

Unit 3

Unordered (un-numbered) List

To create an unordered list, use the [](#) element to define the list, and the [](#) element for each list item. Enter this code above the closing body tag (</body>).

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

Now change the code to include a level 1 heading and a level 2 subheading and all list items as level 4 text like this:

```
<h1> Unordered List</h1>
<h2> Title of list</h2>
<h4><ul>
  <li>List item 1</li>
  <li>List item 2</li>
  <li>List item 3</li></h4>
</ul>
```

Ordered (numbered) List

To create an ordered list, use the [](#) element to define the list, and the [](#) element for each list item. Enter this code above the closing body tag (</body>).

```
<ol>
  <li>List item 1</li>
  <li>List item 2</li>
  <li>List item 3</li>
</ol>
```

Now change the code to include a level 1 heading and a level 2 subheading and all list items as level 4 text like this:

```
<h1> Ordered List</h1>
<h2> Title of list</h2>
<h4>
```

```
<ol>
<li>List item 1</li>
<li>List item 2</li>
<li>List item 3</li>
</h4>
</ol>
```

Ordered List or Unordered List?

You may occasionally find yourself wondering which list is the best one to use. Here's some info you can use as a guide.

Ordered Lists

Ordered lists should be used for when the order is important. They should be used only in cases where, if you were to change the order, it would change the meaning.

For example, a list of step by step instructions should be placed in an ordered list if the order of the steps are important (i.e. the steps need to be carried out exactly in the same order that is presented in the list).

Another example could be a list ordered by ranking, such as a "top 10" list.

***Ordered List Activity (Do this below your last paragraph </p>).**

Use a Horizontal Line to create a thematic break and then create a top 10 list for your Practice web page. Place it above the closing body tag, introduce it as **Ordered List Activity**. Use **Heading level 1 for the introduction (List Activity)**, **Heading 2 for the title of the list**, and **Heading 3 for the list items**.

Unordered Lists

Unordered lists should be used whenever the order is *not* important. A grocery list is a typical example of an unordered list (as long as it doesn't matter which order you get each item).

***Unordered List Activity (Do this below your Ordered List activity).**

Use a Horizontal Line to create a thematic break and then create an unordered list for your Practice web page. Place it above the closing body tag, introduce it as **Unordered List Activity**. Use **Heading level 1 for the introduction (List Activity)**, **Heading 2 for the title of the list**, and **Heading 3 for the list items**.

***Activity 3 check (the last 2 lists are displayed on your webpage):** website has a horizontal rule, then an <h1> title, then a <h2> title, then a <h3> numbered list, then another horizontal rule, another <h1> title, another <h2> title and a <h3> bulleted list.

Show your teacher your 2 horizontal rules, 2 h1 titles, 2 h3 titles numbered list and bulleted list and have them initial your Activity sheet!

We will be covering more HTML elements throughout this tutorial, but before we do that, you should know about [attributes](#).

Unit 4

HTML Attributes: What they are and How to use them

HTML attributes can be added to HTML elements to provide further information about that element. We used attributes in the Horizontal rule activity earlier.

HTML tags can contain one or more *attributes*. Attributes are added to a tag to provide the browser with more information about how the tag should appear or behave. Attributes consist of a *name* and a *value* separated by an equals (=) sign.

Example

Consider this example:

Abbr is the opening tag for the abbreviation element (remember the element is all of the information from opening to closing tag).

title is an attribute of the abbr element

“Hypertext Markup Language” is the Value for the title attribute.

HTML is the display text. – this is the text that will display on the webpage screen – notice it is in between the > < brackets but still enclosed in the whole tag (before the closing tag).

</abbr> is the closing tag for this element.

```
<abbr title="Hypertext Markup Language">HTML</abbr>
```

This example uses the **<abbr>** tag, which is used to indicate an abbreviation or acronym. But something extra has been added to the tag — an attribute. This particular attribute (the title attribute) provides a title for the element.

The name of the attribute is title. In this example, we've given it a value of Hypertext Markup Language.

The title attribute can be used on any HTML element to provide extra information about the element's contents. When used with the **<abbr>** tag, it allows us to provide an expansion of the abbreviation or acronym (i.e. we can say what the acronym stands for).

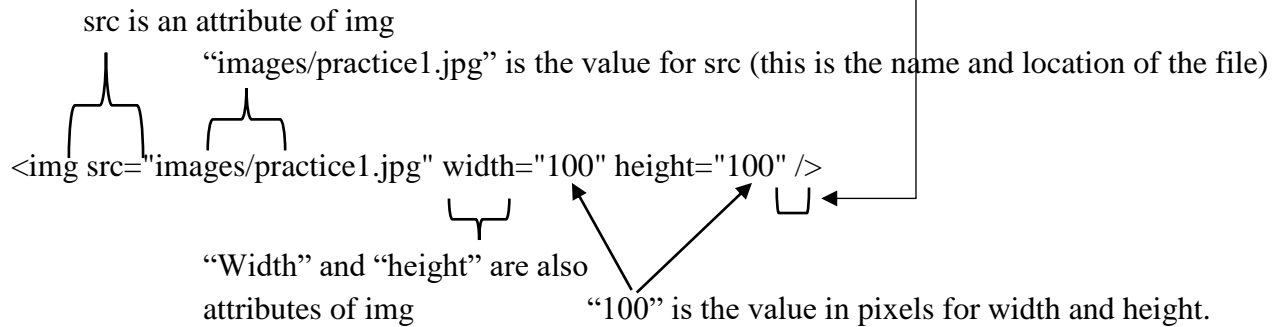
When using the title attribute, most browsers will display its value as a "tooltip" when the user hovers over the element.

Place the **<abbr>** code above **at line 8 in your editor** (just below your “My first website” inside **the body tag**). You should see an “HTML” displayed on your webpage, and when you hover over it with your mouse it should provide a “tooltip” that reads “Hypertext Markup Language”.

More Attributes - Size Attributes

HTML images are defined with the `` tag.

The filename of the source (**src**), and the size of the image (**width** and **height**) are all provided as **attributes**: `img` is an element, but it is an empty element as it is only for displaying an image. In older HTML versions a closing tag was not needed, in HTML5 a space and a forward slash are required. (although I would suggest using ``).



Go to the Student shared drive – BBT9 – HTML5 folder and copy the entire folder into your practice folder. This folder also needs to contain your practice1.html notepad file.

Then type the **highlighted** code below into your Practice.html above the closing body tag (`</body>`).

```
</img>
```

Copy and paste the `img` code again. Save and refresh your page. Do you notice that you have 2 pictures on the same line. How could you put them on separate lines?

Try adding a couple of `
` tags at the end of the image line:

```
</img><br><br>
```

Insert the `flamingo1.jpg` picture below this image in your HTML editor.

```
 </img><br><br>
```

You will notice the picture is too large so let's change the code to have width and height to make the picture smaller:

```
</img><br><br>
```

****NB:** it should be noted that the size attribute for images does not change the file size of the image, only the size it will appear on the webpage. The image size is specified in pixels: `width="100"` means 100 screen pixels wide. If the file size is large it may be a good idea to make it smaller so the page loads more quickly. This can be easily accomplished using MS Paint.

Resizing Image file size:

Open Paint, then open the flamingo1.jpg file in your HTML folder

Select the resize button and ensure the Percentage button is selected and the Maintain Aspect Ratio box is selected.

Then change the 100 in the Horizontal box to 10.

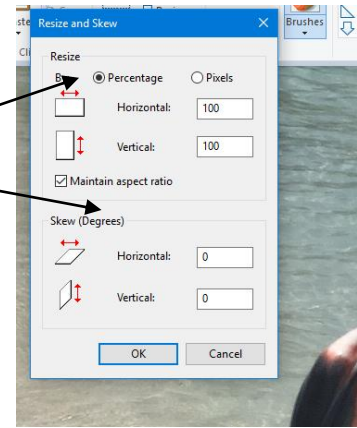
Now select the Pixels button and change the Horizontal number to 100.

This will change the Vertical to 133 because we have selected the Maintain aspect ratio box. This is the information we need to set the width and height of the picture in our img tag.

Now select the OK button, then choose File and Save as, select the JPEG option and save the file as **flamingo2.jpg** (save this in the images folder inside your Practichtml folder) and close the Paint program.

Next, navigate to your practice/images folder and compare the file sizes of both Flamingo1.jpg and flamingo2.jpg. You will notice that Flamingo2.jpg is much smaller than its original. In this case the original file size is not that large, however, the point is that the more and the larger the files that a browser needs to download the slower the browser will be.

You can use this process with any picture or drawing to decrease file size.



More about images (*the alt attribute*)

The **** tag is written as `` with the image file name **and** location (or URL) inserted between the double quotes of the src attribute.

The **alt attribute** provides alternative text for users who are unable to view the image. Some reasons why users can't view the image might include having a slow Internet connection, they are vision impaired and are using text-to-speech software, the image won't load for some reason, etc.

Note that the alt attribute is a required attribute for all image files.

Like this:

```

```

You can also provide the image dimensions using the width and height attributes.

```
</img>
```

Go back to all your image codes and place an alt attribute in all the codes like this:

```
</img><br><br>
```

****NB:** not all browsers will show the alt text on screen. IE for example will not show the text if the picture works, so if you want the viewer to see the text while the file is opening it is acceptable to also place a title attribute in the tag, like this:

```
</img><br><br>
```

Multiple Attributes

You can add more than one attribute to a given element.

Here's an example of adding two attributes to the <a> element (which is used for creating a hyperlink to another web page). **Type this code below your flamingo img code and before the closing body tag.**

```
<a href="https://grade9bbt.wikispaces.com" title="Best grade 9 BBT tutorials on the web!">BBT Wikisite</a>  
  
<br><hr><br> <!--this is just for a thematic break between the images and color-->
```

The **href** attribute specifies the location of the web page that we're linking to.

We also use the title attribute to provide some advisory text (a tool tip for mousing over).

There will be more on links later in the module.

More Attributes

There are many different attributes available to HTML elements. Some attributes can be used on every HTML element some are available on many (but not all) elements, while other attributes are only available on one specific element. But there are a lot of attributes and you don't need to memorize them just yet. The good thing about attributes is that, in most cases, they are optional.

Many HTML elements assign a default value to its attributes — meaning that, if you don't include that attribute, a value will be assigned anyway. Having said that, some HTML tags do require an attribute (such as the hyperlink example above).

(full list here: https://www.quackit.com/html_5/tags/html_5_global_attributes.cfm)

Or here: https://www.w3schools.com/tags/ref_attributes.asp)

Code Assist

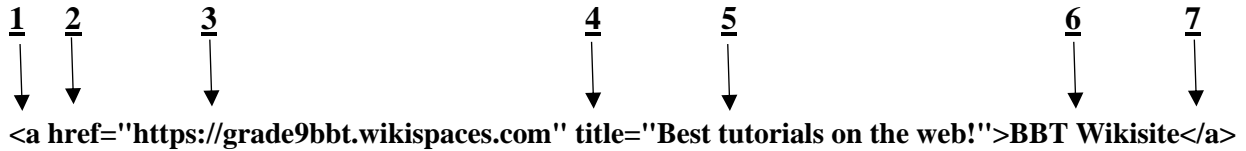
Many HTML editors provide a "code assist" feature, which presents all available attributes when you begin to enter an HTML tag.

This can help tremendously, and saves you from having to memorize every single attribute available for every single element.

What you have learned so far:

- Some HTML elements can have attributes
- The pattern for an attribute is attribute="value"
- Attributes provide additional information about an element
- Attributes are always specified in the start tag
- Attributes require quotation marks, this also makes it easier to check code at a future date.
- Multiple attributes can be used inside a tag or element.

***Activity 4 – Name the parts:**

$\underline{1}$ $\underline{2}$ $\underline{3}$ $\underline{4}$ $\underline{5}$ $\underline{6}$ $\underline{7}$

`BBT Wikisite`

Use the possible answers below and place the correct letter in the activity sheet blank.

A – Opening tag B – Closing Tag C – Attribute D- Value E – Display text F – Element

Unit 5**HTML Styles**

CSS allows you to specify how each HTML element is presented to the user.

HTML is quite limited when it comes to the appearance of its elements. This is not so good if you're trying to give a website a unique look and feel, or you're trying to adhere to a corporate identity.

But it's not the purpose of HTML to do such things. That's the job of CSS.

What is CSS?

CSS stands for Cascading Style Sheets, and its purpose is to enable web authors/designers to apply styles across their websites.

With CSS, you can specify many style properties for any given HTML element. Each property has a name and a value, separated by a colon (:). Each property declaration is separated by a semi-colon (;).

Like this:

```
color:olive;font-size:24px;
```

It's fine to have spaces in between:

```
color: olive; font-size: 24px;
```

And you can even have each declaration on its own line:

```
color: olive;
font-size: 24px;
```

CSS Inline Styles

The above code is an example of inline styles. It is called *inline* because we declared the styles within the HTML tag itself.

Type the code below above your closing body tag in your HTML editor.

```
<p style="color: olive; font-size: 24px;">HTML Styles with CSS</p>
```

The HTML Style Attribute

Setting the style of an HTML element, can be done with the **style attribute**.

The HTML style attribute has the following **syntax**:

```
<tagname style="property:value;">
```

The *property* is a CSS property. The *value* is a CSS value.

HTML Background Color

The **background-color** property defines the background color for an HTML element.

Go to your opening body tag in your HTML editor and type the code below for a powderblue background for your webpage.

```
<body style="background-color:powderblue;">
```

This example sets the background color for your page to powderblue:

HTML Text color

You can change the color of any text on your webpage by adding the CSS style property.

Type the following code below the HTML styles with CSS line:

```
<h1 style="color: blue;"> This is a heading:</h1>
<p style="color: red;"> This is a paragraph.</p><br>
```

Now type the following:

```
<p>I am normal</p>
<p style="color:red;">I am red</p>
<p style="color:blue;">I am blue</p>
<p style="color:red;font-size:36px;">I am big and red</p>
```

This is fine if you want to make a whole paragraph a certain color, but if you want to change just one word in the text you need to use the ** tag, like this:**

```
<p> <span style="color:red;">BBT</span> is the best course to learn about the use of technology in grade 9!</p>
```

Type this code below your last paragraph (I am big).

HTML Background Attribute: Color

The **background-color** property defines the background color for an HTML element.

This example sets the background color for a page to powderblue:

Example

Go to your **opening body tag** and insert the style and color as shown below:

```
<body style="background-color:powderblue;">
```

The background attribute can also be used on just lines of text like the following examples.

Type these into your HTML editor below your BBT line:

```
<h2 style="background-color:red">Background-color set by using red</h2>
```

```
<h2 style="background-color:orange">Background-color set by using orange</h2>
```

```
<h2 style="background-color:yellow">Background-color set by using yellow</h2>
```

```
<h2 style="background-color:blue;color:white">Background-color set by using blue</h2>
```

```
<h2 style="background-color:cyan">Background-color set by using cyan</h2>
```

There is much to learn about the use of color in HTML. In HTML4 there were only 16 named colors that browsers could recognize. Now in HTML5 there are 140 named colors, and many more that can be recognized using the color codes. Below is a list of the original 16 with their corresponding color codes. The format for using color is the same you can specify either the color name or the color code, like this:

```
<h2 style="background-color:#0000FF;color:#FFFFFF">
```

```
Background-color set by using blue, text color set using white</h2>
```

Enter this code below your last. It should look the same as the other background set in blue.

Background-color Activity

Add at least 3 more headings like the above examples exchanging the color name for the color codes, and where the default text color is changed add this in the screen text also (like the one above) – see the HTML color code chart below or the websites listed below the chart.

****NB:** the # symbol is required and 0s are zeros

Show your teacher your 3 headings with background color and text explaining the settings and have them initial your Activity sheet!

HTML Color Codes

Aqua #00FFFF	Navy #000080	Gray #808080	Silver #C0C0C0
Black #000000	Olive #808000	Green #008000	Teal #008080
Blue #0000FF	Purple #800080	Lime #00FF00	White #FFFFFF
Fuchsia #FF00FF	Red #FF0000	Maroon #800000	Yellow #FFFF00

There are plenty of web resources to assist you, here are 3:

<http://htmlcolorcodes.com/color-names/> , <http://htmlcolorcodes.com/>

https://www.w3schools.com/colors/colors_picker.asp

https://www.quackit.com/html/html_color_codes.cfm

HTML Borders

Similar to the background-color above, **borders** are like a picture frame as they set a box of color around an HTML element rather than filling it. **Borders** can be applied to most HTML elements within the body.

To make a border around an element, all you need is border-style.

The values can be solid, dotted, dashed, double, groove, ridge, inset and outset.

border-width sets the **width** of the border, most commonly using pixels as a value.

Finally, border-color sets the **color**.

Let us try a few combinations. Add the following code to your HTML editor:

```
<br><hr><br> <!--this is just for a thematic break between the background and border-->
<h2 style="color:navy;border-width:10px;border-style:solid;border-color:yellow;"><center>Border Activity</center></h2>
```

```
<p style="color:red;border-width:10px;border-style:solid;border-color:olivedrab;">
This text is red with a solid olive drab border 10 pixels wide.</p>
```

```
<p style="color:maroon;border-width:5px;border-style:dotted;border-color:aqua;">
This text is maroon with a dotted aqua border 5 pixels wide.</p>
```

```
<p style="color:Fuchsia;border-width:5px;border-style:dashed;border-color:purple;">
This text is fuchsia with a dashed purple border 5 pixels wide.</p>
```

Of course you can combine a background with a border, but you need to choose your colors carefully so that all the effects can be seen clearly, like this:

```
<p style="background-color:black;color:white;border-width:5px;border-style:double;border-color:yellow;">This text is white in a black background with a double yellow border 5 pixels wide.</p>
```

Border exercise: create the remaining border styles (groove, ridge, inset and outset) with a combined background color and text color of your choice.

Show your teacher your 4 borders and the display text explaining the colors and styles and have them initial your Activity sheet!

HTML Background image

The background image property sets an image for the page background. **This needs to be placed inside your opening body tag.**

```
<body background="backgrounds/practice2.jpg">
```

Ensure you have the image saved in your backgrounds folder and that the background folder is inside your practice folder then Go to your opening body tag and insert the background image as shown above. (While not HTML5, it is still widely supported).

Or visit this site for some free background images for your website:

https://www.quackit.com/web_graphics/free_website_graphics/free_website_background_patterns.cfm

If you do this ensure the files are saved as picture formats and saved to your backgrounds folder.

HTML Fonts

The **font-family** property defines the font to be used for an HTML element:

Type the following below your last paragraph.

```
<br><hr><br> <!--this is just for a thematic break between the borders and fonts-->
```

```
<h1 style="font-family:verdana;">This is a heading</h1>
```

```
<p style="font-family:courier;">This is a paragraph.</p>
```

Table of available Font Families:

Georgia	Palatino Linotype	Book antiqua
Times New Roman	Arial	Helvetica
Arial Black	Impact	Lucinda Sans Unicode
Tahoma	Verdana	Courier New
Lucida Console	initial	

```
<h1 style="text-align:center; font-family:times new roman; color:navy;border-width:10px;border-style:solid;border-color:yellow;"><strong>Font Activity</strong></h1>
```

Here we have combined the font-family with a border and font color styles (as earlier) as well as a strong and center tag to make the text bold, centered, heading 1 as well as in a specific font (times new roman).

```
<p style="font-family:helvetica;"> this paragraph is displayed using the <b>Helvetica</b> font-family.</p>
```

```
<p style="font-family:helvetica;"> There are many different reasons for creating personal Web pages. Many people create Web pages to share information about their families, pets, vacations, or favorite hobbies. Some people create Web pages to present a resume to potential employers or to display joke collections or humorous stories. You can also create Web pages to display information, pictures, sound clips, and videos about a favorite celebrity, sports team, or TV show. Web pages can be used to give information about an organization or club to which you belong – you can include a schedule of upcoming events and detailed information about the goals of the organization. </p><br>
```

****NB:** remember you can copy and paste text in your editor – so you can copy this paragraph and paste it and then change only the values that you need to change.

Font Exercise 1A: Choose three different font-families to write the paragraph above into your HTML editor: (3 paragraphs using 3 different fonts – ensure the title paragraph names the font you choose for each paragraph).

Go back to **one** of your previous 3 paragraph exercises and change the introductory paragraph to a font size of 200% like this:

```
<p style="font-family:Palatino Linotype;color:red;font-size:200%;"> This paragraph is displayed using the <b>Palatino Linotype </b> font-family.</p>
```

HTML Text Alignment

The **text-align** property defines the horizontal text alignment for an HTML element:

Type the following below your last paragraph.

```
<h1 style="text-align:center;">Centered Heading</h1>
<p style="text-align:center;">Centered paragraph.</p><br>
```

Text Exercise 2

Copy the **2nd introductory paragraph** from above and paste it below the last activity but align it to the right of the screen like this:

```
<p style="font-family:georgia;color:blue;text-align:right;"> This paragraph is displayed using the <b>Georgia</b> font-family.</p>
```

Copy and paste the paragraph again but align the text to the center, like this:

```
<p style="font-family:georgia;color:blue;text-align:center;"> This paragraph is displayed using the <b>Georgia</b> font-family.</p>
```

Copy and paste the paragraph again but align the text to the left, like this:

```
<p style="font-family:georgia;color:blue;text-align:left;"> This paragraph is displayed using the <b>Georgia</b> font-family.</p>
```

Copy and paste the paragraph again but take the align-text property and value out of the style attribute, like this:

```
<p style="font-family:georgia;color:blue;"> This paragraph is displayed using the <b>Georgia</b> font-family.</p>
```

Is there a difference between the last two paragraphs? Usually there is not because most browsers align text to the left by default so there is no need to use the left align command except in special cases like forms and tables – which is beyond the scope of this module.

What you have learned in this Unit:

- Use the **style** attribute for styling HTML elements
- Use **background-color** for background color
- Use **color** for text colors
- Use **font-family** for text fonts
- Use **font-size** for text sizes
- Use **text-align** for text alignment

CSS has two other ways to add styles to HTML documents, these are **Embedded styles** and **External styles** both of which are too large a scope for this module.

Activity 5 Review questions

Enter the letter of the correct answer on your Activity Sheet.

1. To make the background color red, the correct tag is:

- A. `<body background="red">`
- B. `<body style= red>`
- C. `< body style="background-color:red;">`
- D. `<body bgcolor="red">`

2. When using colored backgrounds, an important consideration is:

- A. to ensure that the image or color is pretty.
- B. to ensure that the image is colorful.
- C. to ensure that the image makes a statement.
- D. to ensure that the text information can be easily read.

3. In order to make one word in a line of text be red the proper code would be:

- A. `<p>the apple is <color="red">red.</p>`
- B. `<p>the apple is <text-color:red;>red.</p>`
- C. `<p>the apple is <"text-color:red;">red.</p>`
- D. `<p>the apple is red.</p>`

4. In order to make the background color yellow and the text color red in a line of text the proper code would be:

- A. `<p style="background-color:yellow">Display text</p>`
- B. `<p style="background-color:yellow;color:red;">Display text</p>`
- C. `<p style=background-color:yellow;color:red;>Display text</p>`
- D. `<p style="background-color;yellow;color;red;">Display text</p>`

5. Choose the correct response based upon the line of code: `<p style="background-color:black;color:white;border-width:5px;border-style:double;border-color:yellow;">`

- A. The following are all attributes: background-color, color, 5px, border-color
- B. The following are all values: black, white, 5px, double, yellow
- C. The following are all attributes: black, color, border-width,
- D. The following are all values:

6. Choose the correct code for centering an element:

- A. `<p style="text-align: center">`
- B. `<p><center></center></p>`
- C. `<h1 style="vertical-align: center">`
- D. `<p style="vertical-align: center">`

Unit 6**Aligning images with text**

Aligning images with text can be accomplished using the vertical-align property with the bottom, middle, or top values; and the float property with the right or left value.

Type the following below your last paragraph.

```
<br><hr><br> <!--this is just for a thematic break between sections-->
```

```
<h1 style="text-align:center; font-family:times new roman; color:navy;border-width:10px;border-style:solid;border-color:yellow;"><center><strong> Aligning images with text
</strong></center></h1>
```

```
<h4>Image with "vertical-align:bottom":</h4>
```

```
<p>This is some text.  This is some text.</p>
```

```
<h4>Image with "vertical-align:middle":</h4>
```

```
<p>This is some text.  This is some text.</p>
```

```
<h4>Image with "vertical-align:top":</h4>
```

```
<p>This is some text.  This is some text.</p>
```

```
<h4>Let the image float to the right:</h4>
```

```
<p>This is some text. This is some text. This is some text. This is some text.</p>
```

```
<h4>Let the image float to the left:</h4>
```

```
<p>This is some text. This is some text. This is some text. This is some text.</p>
```

Lets try a larger example – type the following paragraph into your HTML editor:

```
<br><b>Ludwig van Beethoven </b>was born in Bonn, Germany in 1770. He spent most of his life in Vienna where he earned a living giving concerts, teaching piano, and selling his compositions. One of the most fascinating aspects of Beethoven's life was his triumph over the deafness that struck him during adulthood. In fact, he composed some of his most powerful works after losing his hearing.<br><hr>
```

When you look at this in your browser you will notice that the image has no buffer between it and the text. **So a border should be added like this:**

```
<br><b>Ludwig van Beethoven </b>was born in Bonn, Germany in 1770. He spent most of his life in Vienna where he earned a living giving concerts, teaching piano, and selling his compositions. One of the most fascinating aspects of Beethoven's life was his triumph over the deafness that struck him during adulthood. In fact, he composed some of his most powerful works after losing his hearing.<br><br><br>
```

****NB:** note that the Float property and value (left) had to be moved inside the style attribute because there can only be 1 style attribute, but it can have several properties within it.

When you look at this on your browser the border does add some space between the image and the text, however **some blank space might be more attractive**. To accomplish this **you can add a margin around the image with the `hspace="10"` inserted inside the img tag like this:**

```
<BR><b>Ludwig van Beethoven </b>was born in Bonn, Germany in 1770. He spent most of his life in Vienna where he earned a living giving concerts, teaching piano, and selling his compositions. One of the most fascinating aspects of Beethoven's life was his triumph over the deafness that struck him during adulthood. In fact, he composed some of his most powerful works after losing his hearing.<br><br><br>
```

Reproduce the Beethoven paragraph another 3 times experimenting with image placement, border values, and `hspace` values. – Begin this section on your webpage with an h1 heading introducing the activity similar to the following:

```
<h1 style="text-align:center; font-family:times new roman; color:navy;border-width:10px;border-style:solid;border-color:yellow;">Beethoven image activity</h1>
```

Unit 7

Adding pages and Hyperlinks (taking it from a webpage to a website)

Hyperlinks

Links (hyperlinks) should be easy to find on your Web page and should always let users know where the link will take them.

Type the following below your last paragraph in your HTML editor.

```
<br><hr><br> <!--this is just for a thematic break between sections-->
```

```
<h1 style="text-align:center; font-family:times new roman; color:navy;border-width:5px;border-style:solid;border-color:yellow;"><strong>Adding Pages and Hyperlinks</strong></h1>
```

Now place this text below the previous code:

```
<a href="http://leohayeshigh.nbed.nb.ca"> Go to Leo Hayes Webpage</a><br>
```

Save and refresh and you can now visit the Leo Hayes website by clicking on the “go to Leo Hayes Website” link in your browser.

If you have difficulty seeing the link, change the color of your background. **If your background color or image is the same color as the link’s text, it will be almost impossible to see!** This is another important consideration for choosing a background color or image.

Here’s how to use an image for link.

The image link is inserted in the space between the Larger and Lesser signs where the ‘display text’ or link text space is. A required component of the link is the display text or an image in this space – this is the only part that shows on the screen so without this component there is no link to click on.

Delete your ‘display text’ link and insert the following (shown in bold):

```
<a href="http://leohayeshigh.nbed.nb.ca"></a><br>
```

Save and refresh and your link is now an image.

It is a good idea to label these images so that your viewers know they are links **like this**:

```
<a href="http://leohayeshigh.nbed.nb.ca">Click the picture to go to the Leo Hayes website.</a><br>
```

Centering images in HTML5 requires CSS programming and because this is beyond the scope of this module we will use some **coding trickery**. We will use a text element and nest the img element within the text element and this will allow us to center the image along with the text.

(This can be used with other elements like audio and video – which we will get to later)

First we choose a text element <p> and within the start tag we need to ensure the text will be centered between the margins and as you may recall from the text and images lesson above, we can do this with the text-align property and the center value (see below).

Second, we need to nest the <a> element within the text element and place an image in the link text space like above.

Third, we need to center the image with the text and as you may recall from the text and images lesson this is achieved by placing a vertical-align property and a middle value within the style attribute of the img element.

All these factors put together look like this (**enter this code below your last link**):

```
<p style="text-align:center;"><a href="http://leohayeshigh.nbed.nb.ca">Click the picture to go to the Leo Hayes website.</a></p>
```

If we wanted to get creative we could incorporate the picture into the text, like this:

```
<p style="text-align:center;font-size:300%;"><a href="http://leohayeshigh.nbed.nb.ca">Go to Le Hayes</a></p><br>
```

****NB:** This seems really complicated compared to the old HTML4 code which was achieved simply by adding a <center> tag like below, however, this tag may not be supported by all browsers so it is not advised to use it.

```
<center><a href="http://leohayeshigh.nbed.nb.ca">Click the picture to go to the Leo Hayes website.</a></center><br>
```

Adding Pages (go from a webpage to a website)

First, you need to create a second page for your website and this file needs to be in the same folder as the first page (practice.html). If you can't remember how to make a page go to the first instructions in the module and write the code in a new tab and save it as **page2.html** in the same folder as your practice.html file.

Second, In the bottom of your first page (practice.html), but before your closing body tag (</body>) **enter the following code (shown in bold)**.

```
<a href="page2.html"> Go to second page </a><br>
```

Save and refresh. You should now have a link that reads **Go to second page**. Try it.

You will also need to put a link on your second page to bring you back to your first.

Open the source in notepad for your second page. Enter the following (shown in bold):

```
<body>
```

```
<a href="practice.html"> Back to Home Page </a>
```

```
</body>
```

Save and Refresh. You should now have a link that reads **Back to Home Page**. Try your link. You should now have links to and from both of your Web pages. Hey, you have a Web site!

Special note about Hyperlinks – absolute and relative links

Links, otherwise known as *hyperlinks*, are defined using the <a> tag — otherwise known as the *anchor* element.

To create a hyperlink, you use the `<a>` tag in conjunction with the href attribute. The value of the href attribute is the URL, or, location of where the link is pointing to.

Hypertext references can use *absolute* URLs or *relative* URLs.

absolute

This refers to a URL where the full path is provided. For example:

```
<a href="https://www.quackit.com/html/tutorial/">HTML Tutorial</a>
```

relative

This refers to a URL where the path, relative to the current location, is provided.

For example, if we want to reference the `https://www.quackit.com/html/tutorial/` URL, and our current location is `https://www.quackit.com/html/`, we would use this:

```
<a href="tutorial/">HTML Tutorial</a>
```

****NB: all the references you will use to link to your own pages are relative links.** If you copy and paste links from your address bar to do this ensure that you are not providing an absolute reference – if you do your teacher will not be able to link to your pages because you are not posting your website to the internet and therefore an absolute reference will not work for anyone else (outside of your computer).

Link exercise:

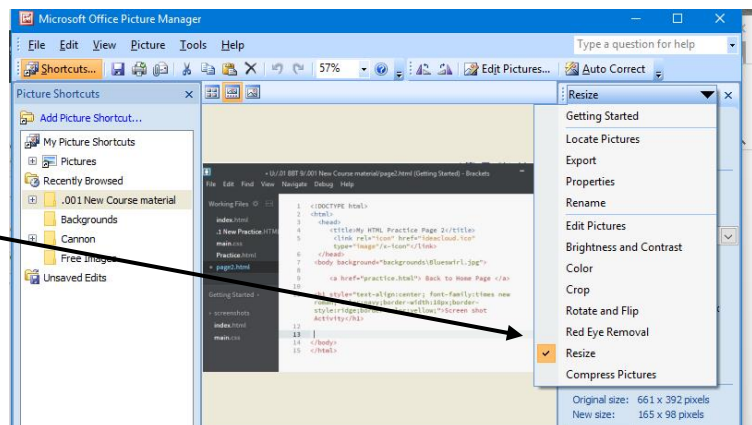
As you did previously, change the links above to include a picture centered on your page. (It would be cool if you could find a picture of a forward and backward arrow to use for these links).

Now let's add some content to the second page!

1. **First**, change the background to an image.
2. **Second**, place an h1 heading in a border to introduce the code for the second page. The text for this should read: **Screen shot Activity**

Then, take a screen shot of the code for your page2.html page. You can do this with word and save as a jpg image or you can use the snipping tool to snip the whole page and save it as **Screen.jpg** in your images folder inside your Practice folder. Then open the picture in office picture manager and crop it to include only the HTML editor and code. (You can do this in Paint as well).

Next, in the tools menu, resize the picture to 25% of the original and save this copy as **screen1.jpg** in your Practice HTML folder. (Remember you can do this in Paint as well).



Then, place some text above both the original and the resized picture. (That's 2 text elements and 2 pictures). The text should tell the viewer what they are looking at, something like this:

Here is a screenshot of the code in the program *brackets* used for the current webpage in its original size.

Image (code.jpg)

Here is a screenshot of the code in the program *brackets* used for the current webpage at 25% of its original size.

Image (code1.jpg)

3. Audio Content - Add some Audio content to your page

The **<audio>** tag is written as `<audio></audio>` with any attributes provided within the start tag. Attributes can specify the actual audio file to play, as well as other options such as whether to play automatically, whether the browser should display the controls or not, etc.

Like this:

```
<audio src="media/sundial.mp3" autoplay loop><p>if you are reading this, it is because your browser does not support the audio element</p></audio><br><br>
```

In this example I have selected an audio file from my media folder to auto play and loop – which means the viewer cannot turn it off.

If I wish to provide controls for the viewer I would either exchange the attributes `autoplay` and `loop` for controls, like this:

```
<audio src="media/sundial.mp3" controls><p>if you are reading this, it is because your browser does not support the audio element</p></audio><br><br>
```

Or I could simply add the `controls` attribute, which would make the audio auto play and loop, but provide the viewer the option to turn it off, like this:

```
<audio src="media/sundial.mp3" autoplay loop controls><p>if you are reading this, it is because your browser does not support the audio element</p></audio><br><br>
```

Notice that there is a `<p>` element within the audio element, which is the “fallback” content. This is used when the browser does not support the audio element.

****NB** there is much to learn about audio, but for this module keep it simple and use only **.mp3** files for audio content on your webpages as they are well supported.

Go to one of the websites noted below, or create your own music and add the file in mp3 format to your media folder and add it to your website.

You can download royalty free music for websites here:

<http://www.bensound.com/royalty-free-music/corporate-pop>

<http://www.purple-planet.com/>

You can create your own music on line here: <https://soundation.com/>

4. Video – Add your own video content to your webpage

The latest version of HTML (i.e. HTML5) makes it extremely easy to add video to a web page. HTML5 allows you to use a `<video>` tag, which, believe it or not, embeds video right into your web page!

The HTML 5 `<video>` tag is used for embedding video into an HTML document. The tag accepts attributes that specify how the video should be played.

Any content between the start and end `<video>` tags is *fallback content*. This content is displayed only by browsers that don't support the `<video>` element.

The `<video>` tag is typically written as `<video src=""></video>` with the URL of the video file inserted between the double quotes of the `src` attribute.

The `controls` attribute can be specified to provide user controls for the interface (so that the user can stop, start, change the volume, etc). The `width` and `height` attributes can also be used to provide the width and height of the video.

Like this:

```
<video src="media/sample.mp4" width="250" height="180" controls></video><br>
```

****NB** there is much to learn about video, but for this module keep it simple and use only mp4 files for video content on your webpages as they are well supported.

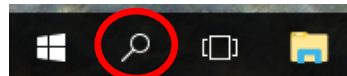
More information on video tags

https://www.quackit.com/html_5/tags/html_video_tag.cfm

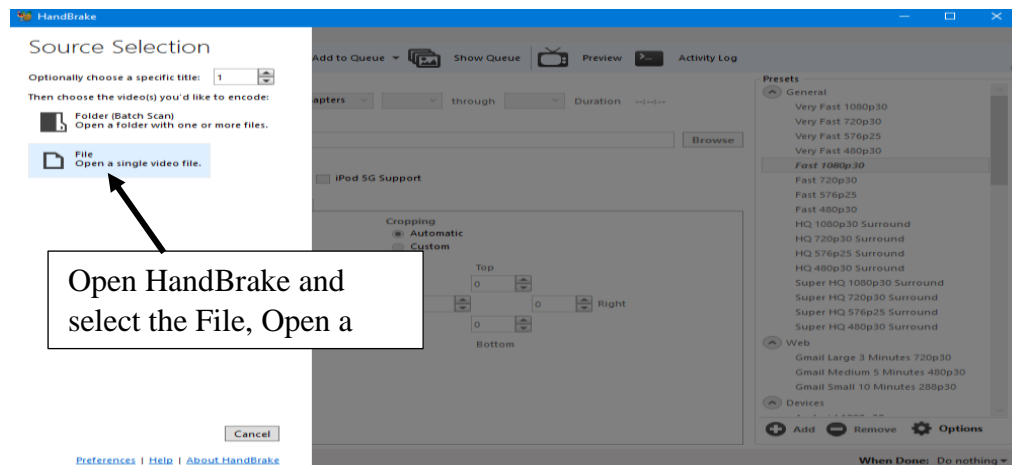
Converting Video format using Handbrake

Open the Handbrake program by double clicking on the desktop icon

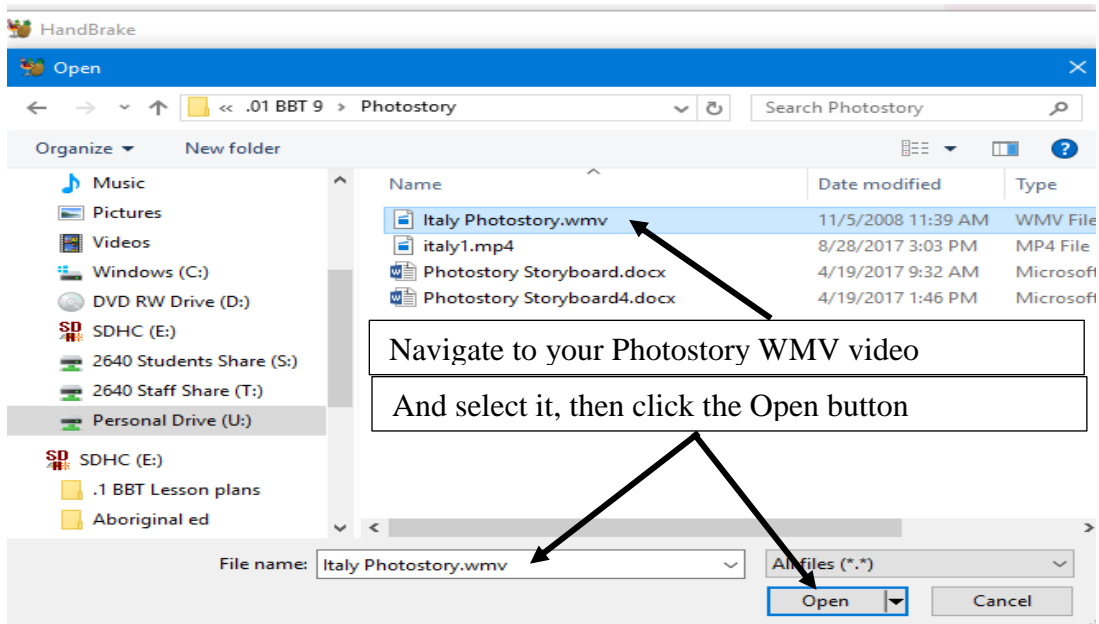
Or by typing “Handbrake” in the search windows



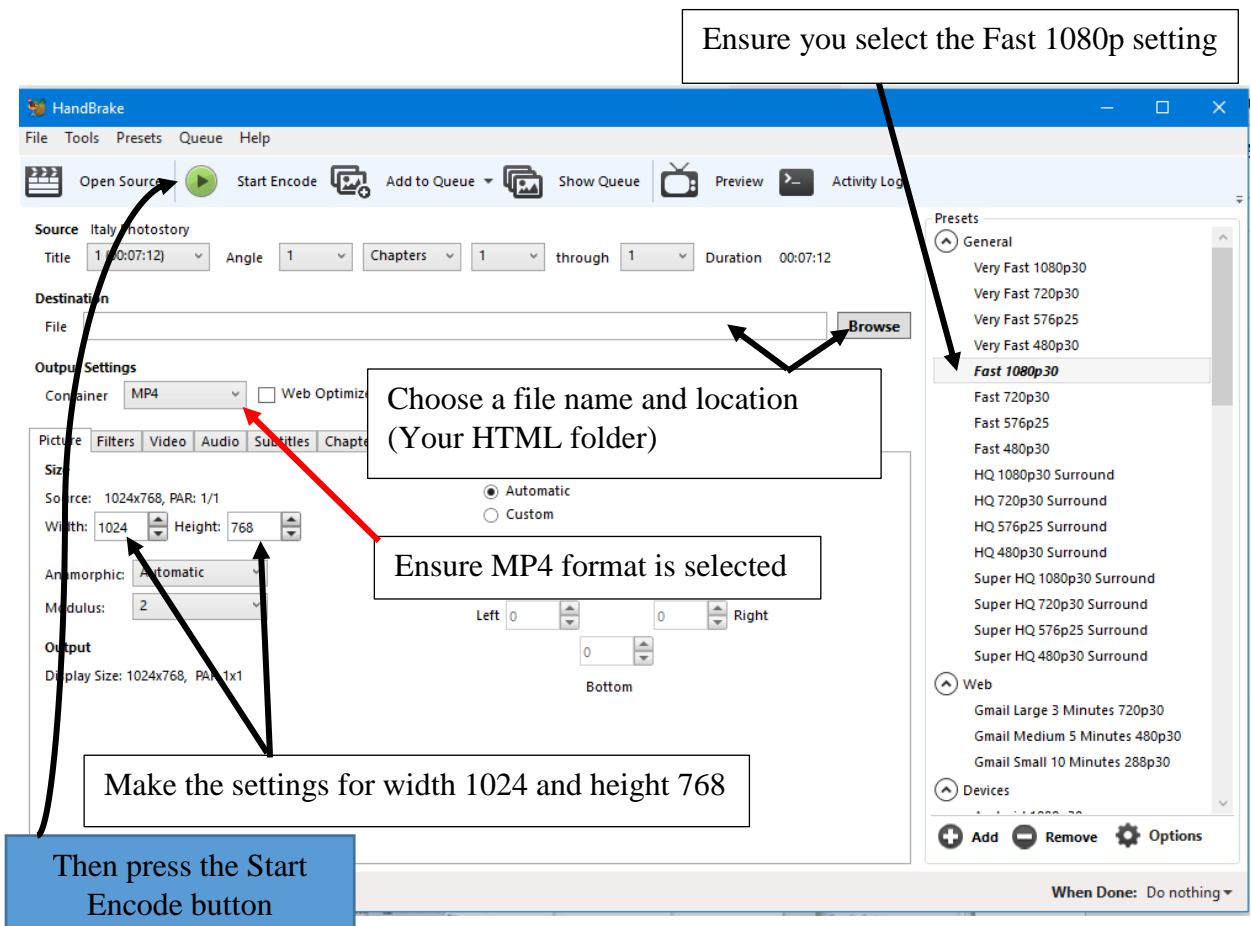
When the program opens, select the File, Open a single video file option as shown below



Then find the file you want to convert and click the Open button as shown below.



Then choose the location you want to save the new file to (your HTML folder), ensure the settings are correct, then click the start encode button, as shown below,



When the encoding process is finished your new mp4 video will be saved in your folder When the program is finished you should have both the original sample.wmv and a new sample.mp4 movie in your media folder.

****NB: If Handbrake is not working, you can convert the video to MP4 using MovieMaker (see instructions in the Photostory module)**

Now place an h1 heading with a background and border style and color to introduce a sample video in your page 2 source code.

Then insert the video code to play the sample.mp4 video, like this:

```
<video src="media/sample.mp4" width="250" height="180" controls></video><br>
```

This will place the video on the left margin of your page.

Challenge exercise: see if you can center the video on the screen.

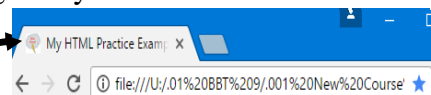
****Hint: you may have to use some *coding trickery* to achieve this****

Adding an Icon to your Web Site – (Bonus Activity)

1. To add an Icon for your web site first create a drawing in MS Word that captures the idea of your website. Use the drawing tools to accomplish this.
2. Next take a screenshot of your drawing and paste it into the same word document. Then select the pasted picture using the picture tools and right click and select the save as picture option. Save your picture in .BMP format in your HTML folder.
3. Next open your picture in Paint and resize it to 16 by 16 pixels and save it in “other formats” choose a simple name **with the extension .ico** in your HTML folder.
4. In your head section of your HTML document type:

```
<link rel="icon" href="iconname.ico" type="image/x-icon"></link>
```

where iconname.ico is the name of your icon. Save the changes to your HTML editor and refresh your webpage and you should see an icon in your address bar to the left of your address:



Creating Images

The above examples assumed that you already had an image to embed into your web page. To learn about creating images for the web, check out the [Web Graphics Tutorial](https://www.quackit.com/web_graphics/tutorial/) (https://www.quackit.com/web_graphics/tutorial/).

Activity 7 Review Questions:

1. A hyperlink is:

- A. a way to change text colors on a Web page.
- B. a device to take the user to another Web page or site.
- C. a way to insert images on a Web page.
- D. a device to create more Web pages.

2. To use the words **Click Here** as a link to take you to **www.snowboarders.com**, the correct HTML is:
- ` Click here `
 - ` Click here `
 - ` Click here `
 - ` Click here <a>`
3. To use the image **icecream.gif** as a link to take you to **icecreamshoppe.com**, the correct HTML is:
- ` `
 - ` `
 - ` `
 - ` `
4. To link to a second page within a Web site, **second.html**, that is in the same folder as the home page, using **Page 2** as the link text, the correct HTML is:
- ` Page 2 `
 - ` Page 2 `
 - ` Page 2 `
 - ` Page 2 `
5. Which type of list displays information using bullets:
- ``
 - ``
 - `<bl>`
 - `<lu>`
6. In order to make the music file “**sundial.mp3**” play continuously on a website the following code needs to be placed in the body tag:
- `<audio src="media/sundial.mp3" controls><p>your browser does not support the audio element</p></audio>

`
 - `<audio src="media/sundial.mp3" autoplay controls><p>your browser does not support the audio element</p></audio>

`
 - `<audio src="media/sundial.mp3" loop controls><p>your browser does not support the audio element</p></audio>

`
 - `<audio src="media/sundial.mp3" autoplay loop controls><p>your browser does not support the audio element</p></audio>

`

HTML5 Project

Congratulations! You have now finished all of the activities in this module. This means that you now have the basic skills needed to design your own Web site. Here’s your chance to let your creativity and knowledge shine!

Your assignment is to design a Three Page Website on a topic of your choice (with your teacher’s approval, of course). **Review the Basics of Good Design below before beginning your project.**

- Save page 1 as index.html

- Page 2 must include a picture of you as the web page designer
- Your website must contain the following:

Page 1 (Index)

1 page title

1 Heading Title with a background color and a colored border

At least 2 paragraphs of information on your topic

A background color

Music that autoplays and loops, with controls for the viewer to turn it off

At least 5 links to websites related to your topic

At least one image with a border

Another image centered on the page

At least 2 horizontal rules, one of which is less than the full width of the screen

Navigation links located at the top and bottom of the page linking to pages 2 and 3

Page 2

1 page title

1 Heading Title with a background color and a colored border

A picture and a paragraph about you as the web page designer

(align the picture with the text)

A background image

Music that autoplays and loops, with controls for the viewer to turn it off

A video with audio content and controls – does not autoplay – requires the viewer to start

At least 2 horizontal rules, one of which has some thickness

One list (one of the 2 types from the activities)

Navigation links located at the top and bottom of the page linking to pages 3 and 1

Page 3

1 page title

1 Heading Title with a background color and a colored border

References displayed as links (where you got your images and info)

One image centered on the page linking to page 1

Review the Basics of Good Design before you get started on your pages. Also, it would be a good idea to brain storm and get your ideas down on paper before creating your pages. Having your ideas drawn out on paper makes it easier to organize your thoughts once you begin creating your pages. You could even use a storyboard if you wish☺

The Basics of Good Design

1. Emphasize Important Information. Always display the most important information at the top of each Web page since some users will not take the time to scroll all the way through. These users will most likely miss your important information if it is not at the top of the page.

Including a Table of Contents that contains links to important areas of a Web page can help users access important information that appears further down the page.

Headings can also help to emphasize important information since users can quickly scroll through the page and find information of interest.

2. **Page Length.** Web pages should not be too short or too long. If a Web page is shorter than half a screen, try to combine it with another page. If a Web page is longer than five screens, try to break up the page into shorter pages. Users become frustrated if they have to scroll through a lot of information.

3. **Plan Your Layout.** There are two main types of layouts – linear and hierarchical. A linear layout reads like a book with each page providing a link forward and back, just like turning pages in a book. A hierarchical layout has all pages linking off of the home or main page. The home page usually contains a summary of the information contained in each link so that users can choose the order in which to view each page. Combining both types gives the most flexibility.

4. **Plan for Visual Appeal.** To make your web page easier to read and more visually appealing, remember these basics:

- Use a consistent color scheme throughout.
- Use a change in font color, style, and size to draw attention to what is most important.
- Make sure that your choice of background doesn't overpower the text making it difficult to read, or clash with any images on your page.
- Use headings and horizontal rules to organize your information by dividing it into easy to follow sections.
- Use links that are alike and are placed consistently throughout your pages.
- Ensure that each page has a link back to the home page/your first page.

Appendix - Useful Tags

The <a> tag is written as `` with the linked URL between the double quotes of the href attribute and the anchor text (i.e. the text that the user sees) between the start and end tags.

Like this: `Great Workout`.

The <abbr> tag is written as `<abbr></abbr>` with the abbreviation/acronym inserted between the start and end tags. You can add the title attribute to the first instance of the abbreviation to provide an expansion.

Like this: `<abbr title="HyperText Markup Language">HTML</abbr>`

The <audio> tag is written as `<audio></audio>` with any attributes provided within the start tag. Attributes can specify the actual audio file to play, as well as other options such as whether to play automatically, whether the browser should display the controls or not, etc.

Like this: `<audio src="" controls>(Fallback content goes here...)</audio>`

Example:

```
<audio src="/music/good_enough.mp3" controls>
<p>If you are reading this, it is because your browser does not support the audio element.</p>
</audio>
```

****NB** there is much to learn about audio, but for this module keep it simple and use only mp3 files for audio content on your webpages as they are well supported.

The ** tag** should be used to markup text as bold without conveying any extra importance, for example in article abstracts, where the beginning of an article is set in bold text. It should *not* be used to convey extra importance. To convey extra importance, use the tag. To emphasize text, use the tag. The **** tag is written as `` with the text to be bold inserted between the start and end tags.

Like this: `bold text`

The **
 element** is classified as a "void element" as it has no content. This also means that no end tag is required (or allowed for that matter). The `
` tag must be used only for line breaks that are actually part of the content, as in poems or addresses. It should not be used for separating thematic groups in a paragraph. To separate thematic groups, use multiple <p> tags instead.

The `
` tag is written as `
` at the point where the line break should occur.

Like this: Line 1
Line 2

The ** tag** is written as `` with the image file name and location (or URL) inserted between the double quotes of the `src` attribute.

The **alt attribute** provides alternative text for users who are unable to view the image. Some reasons why users can't view the image might include having a slow Internet connection, they are vision impaired and are using text-to-speech software, the image won't load for some reason, etc.

Note that the alt attribute is a required attribute.

Like this:

```

```

You can also provide the image dimensions using the `width` and `height` attributes.

```
</img>
```

The **<q> tag** represents a quotation from another source.

Browsers usually surround `<q>` text with quotation marks. If the quotation spans multiple lines, you should use the <blockquote> tag. Most browsers present blockquote text as indented text.

The `<q>` tag is written as `<q></q>` with the quotation inserted between the start and end tags.

Like this:

```
<q>Quotation text here...</q>
```

The tag

A `` element used to color a part of a text:

```
<p>My mother has <span style="color:blue">blue</span> eyes.</p>
```

The **HTML 5 <video> tag** is used for embedding video into an HTML document. The tag accepts attributes that specify how the video should be played.

Any content between the start and end `<video>` tags is *fallback content*. This content is displayed only by browsers that don't support the `<video>` element.

The `<video>` element can be used with the `<track>` tag to provide captions, subtitles, etc.

The `<video>` tag is typically written as `<video src=""></video>` with the URL of the video file inserted between the double quotes of the `src` attribute.

The `controls` attribute can be specified to provide user controls for the interface (so that the user can stop, start, change the volume, etc). The `width` and `height` attributes can also be used to provide the width and height of the video (see "Template" below for all attributes that can be used with the `<video>` element).

Like this:

```
<video src="video.mp4" width="170" height="85" controls>
```

****NB** there is much to learn about video, but for this module keep it simple and use only mp4 files for video content on your webpages as they are well supported.

For more information on HTML tags go here:

<https://www.w3schools.com/TAGS/>

https://www.quackit.com/html/tags/html_audio_tag.cfm

For more information and helpful tips check out the following resources:

<https://www.quackit.com/html/tutorial/introduction.cfm>

<http://www.htmldog.com/>

<https://www.w3schools.com/html/>

<http://www.html5-tutorials.org/introduction-to-html/the-structure-of-html/>