TUTORIAL: CREATING A WEBPAGE FROM SCRATCH

Part 1 – Background and HTML

What is HTML?

As its name suggests, HTML is a **Markup Language** which means you use HTML to simply "mark-up" a text document with tags that tell a Web browser how to structure it to display.

Hypertext refers to the way in which Web pages (HTML documents) are linked together. Thus, the link available on a webpage is called Hypertext.

HTML is written in the form of HTML elements consisting of **markup tags**. These markup tags are the fundamental characteristic of HTML. Every markup tag is composed of a keyword, surrounded by angle brackets, such as <html>, <head>, <body>, <title>, <p>, and so on.

How do we use HTML?

Most of the stuff on the web is no different than the stuff on your computer – it's just a whole load of files sorted into a whole load of directories.

HTML files are nothing more than simple text files, so to start writing in HTML, you need nothing more than a simple text editor.

Basic Construction of an HTML Page

Although the basics of HTML is plain text, we need a bit more to make it a nice and shiny HTML document.

HTML Tags

The basic structure of an HTML document includes tags, which surround content and apply meaning to it.

- These tags should be placed underneath each other **at the top of every HTML page** that you create. *This is how your average HTML page is structured visually.*

• <!DOCTYPE html> – is a **document type declaration** and it lets the browser know which flavor of HTML you're using (HTML5, in this case). It's very important to stick this in - If you don't, browsers will assume you don't really know what you're doing and act in a very peculiar way.

• <html> – This tag signals that from here on we are going to write in HTML code. It is the **opening tag** that kicks things off and tells the browser that everything between that and the </html> **closing tag** is an HTML document.

• <head> – This is where all the **metadata for the page** goes – stuff mostly meant for search engines and other computer programs.

• <body> – This is where the **content of the page** goes.

Tip: The <html>, <head>, and <body> tags make up the basic skeleton of every web page.

Content inside the <head> and </head> are invisible to users with one exception: the text between <title> and </title> tags which appears as the title on a browser tab.

HTML Tags, continued

HTML tags normally come in pairs like <html> and </html>. The first tag in a pair is often called the opening tag (or start tag), and the second tag is called the closing tag (or end tag).

An opening tag and a closing tag are identical, except a slash (/) after the opening angle bracket of the closing tag, to tell the browser that the command has been completed.

In between the start and end tags you can place appropriate contents. For example, a paragraph, which is represented by the p element, would be written as:

```
<p>This is a paragraph.</p>

<p>  This is <b>another</b> paragraph. </p>
```

Not all tags have closing tags like this (<html></html>) some tags, which do not wrap around content will close themselves.

The line-break tag for example, looks like this :
 - a line break doesn't hold any content, so the tag merrily sits by its lonely self.

We will come across these examples later. All you need to remember is that all tags with content between them should be closed, in the format of opening tag → content → closing tag.

Attributes

- Tags can also have **attributes**, which are extra bits of information. Attributes appear inside the opening tag and their values sit inside quotation marks.

- They look something like:
- <tag attribute="value">Margarine</tag>

- We will come across tags with attributes later.

Tags tend not to do much more than mark the beginning and end of an element. Elements are the bits that make up web pages.

You would say, for example, that everything that is in between (and includes) the <body> and </body> tags is the body element.

As another example, whereas “<title>” and “</title>” are **tags**, “<title>Rumple Stiltskin</title>” is a title **element.**

Elements

When you nest HTML elements, make sure you close them in the right order.

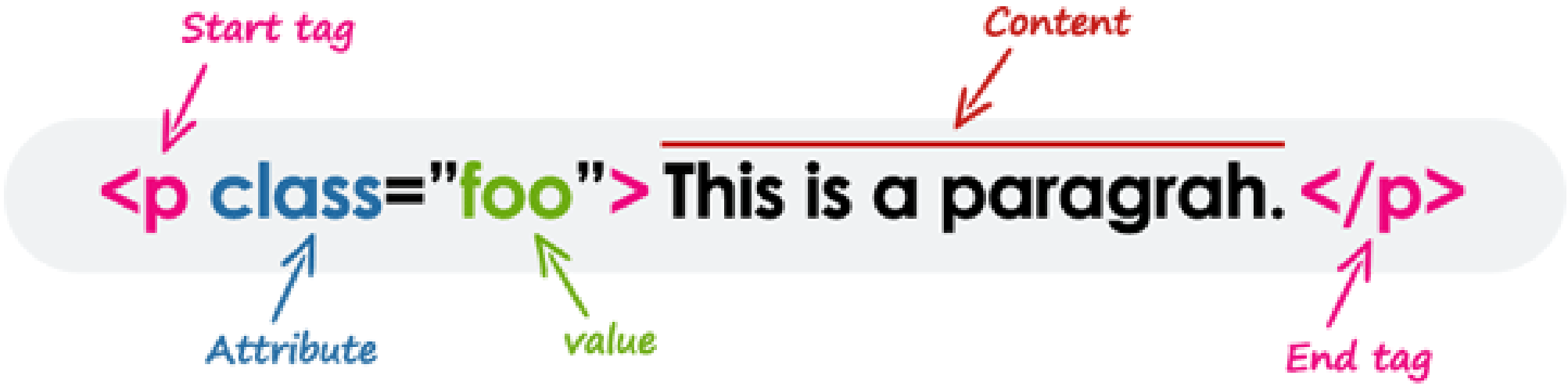
The element which was opened last must be the first to close:

Correct	<html><body> </body></html>
Incorrect	<html><body> </html></body>

HTML Element Syntax

- An HTML element is an individual component of an HTML document. It represents semantics, or meaning. For example, the title element represents the title of the document.

- Most HTML elements are written with a *start tag* (or opening tag) and an *end tag* (or closing tag), with content in between. Elements can also contain attributes that defines its additional properties. For example, a paragraph, which is represented by the p element, would be written as:



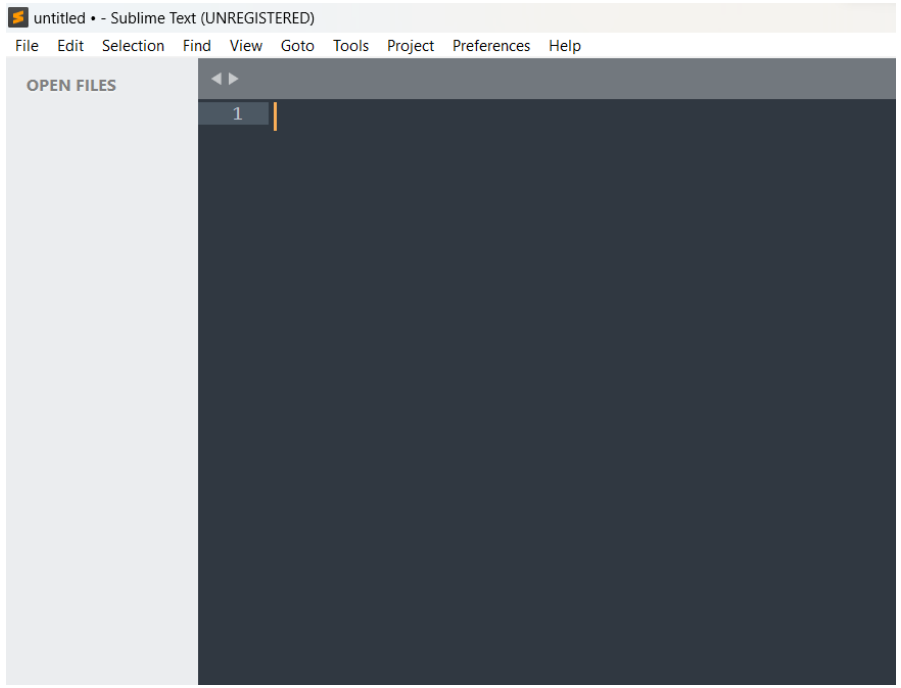
LET’S GET STARTED

Step 1: Setting Up

- In the **Applications** folder of your computer, double-click on the **Sublime** program to open it.

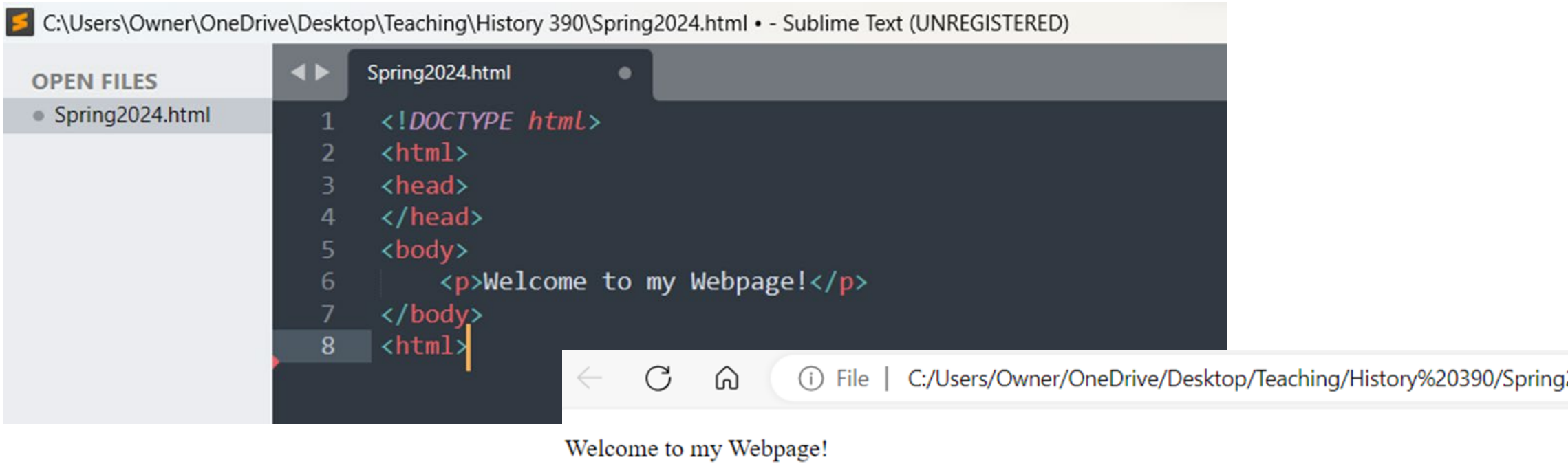
- **Sublime** is a (free) text editor, which is the name for the kind of program that you type code into. Just like Microsoft Word, it displays the text that you type. But unlike Microsoft Word, it doesn't (invisibly) surround your text with formatting information. So never use Microsoft Word for your code! Instead, use Sublime or one of the many alternatives, like Atom.

- Believe it or not, a plain text editor (like Sublime) and a web browser (like Chrome, Safari, or Firefox) are all you need to build even the fanciest webpage!



Step 2: Write Some HTML

- **Enter the document declaration and a little bit of text:**



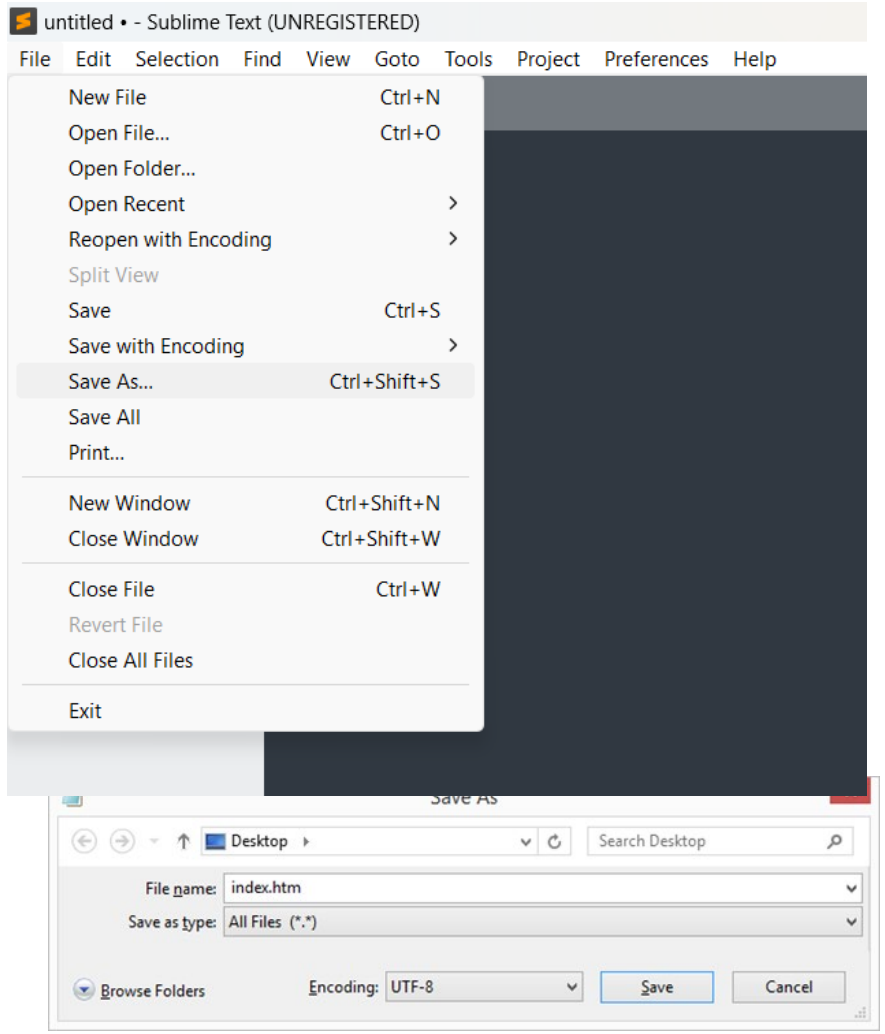
Step 3: Save the HTML Page

Select **File > Save as** in the Sublime menu.

Name the file **"index.html"**

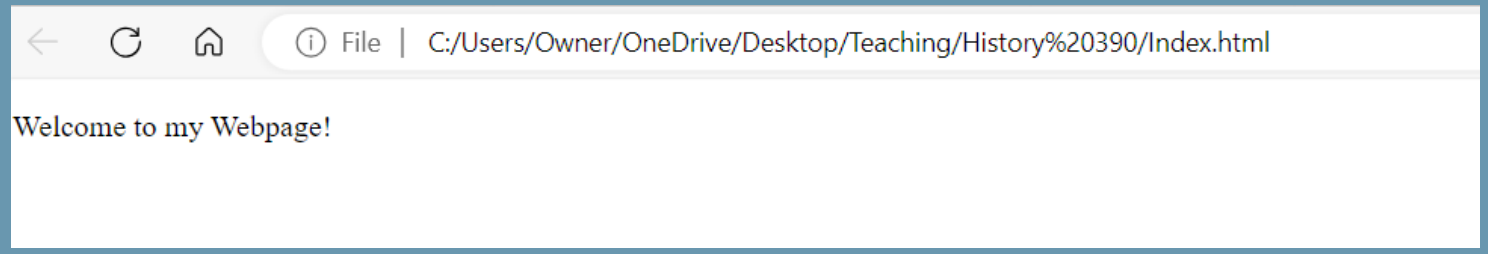
Make sure to put the file someplace that you will remember

If prompted set the encoding to **UTF-8** (which is the preferred encoding for HTML files).



Step 4: View the HTML Page in Your Browser

- Leave your document open and navigate to where you saved the file. Double-click the icon for the file you saved. By default, that file should open up in a browser. OR you can also right click in Sublime and hit “Open in Browser”.



Now you’ve opened the file in two different ways: as a webpage and as a text file. Arrange your windows so you can see the document open as a text document and as a webpage, simultaneously.

Step 5: Page Titles

- Try to add a title to your page.
  Change your code so that it looks like this:

```
<!DOCTYPE html>
<html>
<head>

    <title>My Webpage</title>

</head>
<body>
```

- If you look at this document in the browser (save and reload as before), you will see that “My Webpage” will appear on a tab or the title bar of the window (not the actual canvas area).

- The text that you put in between the title tags has become the title of the document (surprise!). If you were to add this page to your “favorites” (or “bookmarks”, depending on your browser), you would see that the title is also used there.

I have added two new elements here, that start with the head tag and the title tag.

The head element (that which starts with the <head> opening tag and ends with the </head> closing tag) appears before the body element (starting with <body> and ending with </body>) and contains information **about** the page.

The information in the head element does not appear in the browser window.

Step 6: Paragraphs

Now that you have the basic structure of an HTML document, you can mess around with the content a bit.

Go back to your text editor and add another line to your page:

```
<!DOCTYPE html>
<html>
<head>
    <title>My Webpage</title>
</head>
<body>
    This is my first web page
    How exciting
</body>
</html>
```

Look at the document in your browser.

You might have expected your document to appear as you typed it, on two lines, but instead you should see something like this:

```
This is my first web page How exciting.
```

This is because web browsers don't usually take any notice of what line your code is on.

It also doesn't take any notice of spaces (you would get the same result if you typed "This is my first web page How exciting").

Paragraphs, continued

If you want text to appear on different lines or, rather, if you intend there to be two distinct blocks of text (because, remember, HTML is about meaning, not presentation), you need to explicitly state that.

Change your two lines of content so that they look like this:

```
<p>This is my first web page</p>
<p>How exciting</p>
```

The p tag is used for **paragraphs**.

Look at the results of this. The two lines will now appear on two lines because the browser recognizes them as separate paragraphs.

Think of the HTML content as if it were a book - with paragraphs where appropriate.

Step 7: Line Breaks

The line-break tag can also be used to separate lines like the <p> tag, but it looks a bit different, like this:

This is my first web page.**
**

How exciting!

There's no content involved in breaking lines so there is no closing tag.

It could be tempting to over-use line breaks and <**br>** shouldn't be used if two blocks of text are intended to be separate from one another (because if that's what you want to do you probably want the <**p>** tag).

Step 8: Emphasis

-You can emphasize text in a paragraph using **em** (emphasis) and **strong** (strong importance).

```
<p>Yes, that really <em>is</em> exciting.
<strong>Warning:</strong> level of excitement may
cause head to explode.</p>
```

Step 9: Headings

- Headings do not have to appear in section elements, as shown in this example.

```
<h1>The Main Heading</h1> <!-- stuff -->

<section>
     <h2>A Subheading</h2>
     <!-- sub-stuff --> </section>

<section>
     <h2>Another Subheading</h2>
     <!-- sub-stuff -->

     <h3>A Subheading of a
Subheading</h3>
     <!-- sub-sub stuff --> </section>
```

Try Your Own Heading

If you have documents with genuine **headings**, then there are HTML tags specifically designed just for them.

- They are h1, h2, h3, h4, h5 and h6
- h1 being the almighty emperor of headings and h6 being the lowest plebe

Note that the h1 tag is only used once, as the main heading of the page. h2 to h6, however, can be used as often as desired, but they should always be used in order, as they were intended. For example, an h4 should be a sub-heading of an h3, which should be a sub-heading of an h2.

```
<!DOCTYPE html>

<html>
<head>
    <title>My first web page</title>
</head>

<body>
    <h1>Dr. Beasley's Webpage</h1>
    <h2>What this is?</h2>
    <p>A simple page put together using HTML</p>
    <h2>Why this is?</h2>
    <p>To learn HTML</p>
</body>

</html>
```

Step 10: Lists

There are three types of list; **unordered lists**, **ordered lists** and definition lists. We will look at the first two here (the last type is a little more advanced).

Unordered lists and ordered lists work the same way, except that the former is used for non-sequential lists with list items usually preceded by bullets and the latter is for sequential lists, which are normally represented by incremental numbers.

The **<ul>** tag is used to define unordered lists and the **<ol>** tag is used to define ordered lists. Inside the lists, the li tag is used to define each list item.

Try Your Own List (<ul>)

Change your code to the following:

```
<!DOCTYPE html>

<html>

<head>
    <title>My first web page</title>
</head>


<body>
    <h1>Dr. Beasley's Webpage</h1>

    <h2>What this is?</h2>
    <p>A simple page put together using HTML</p>

    <h2>Why this is?</h2>
    <ul>
      <li>To learn HTML</li>
      <li>To show off</li>
       <li>Because I need to pass this class</li>
    </ul>

</body>

</html>
```

If you look at this in your browser, you will see a bulleted list.

Lists can also be included in lists to form a structured hierarchy of items.

Try Your Own List (<ol>)

Simply change the **<ul>** tags to **<ol>** and you will see that the list will become numbered.

Replace the previous list code with the following:

```
<ul>
<li>To learn HTML</li>
     <li>
        To show off
        <ol>
           <li>To my boss</li>
           <li>To my friends</li>
           <li>To my cat</li>
<li>To the little talking duck in my brain</li>
        </ol>
      </li>
      <li>Because I need to pass this class</li>
</ul>
```

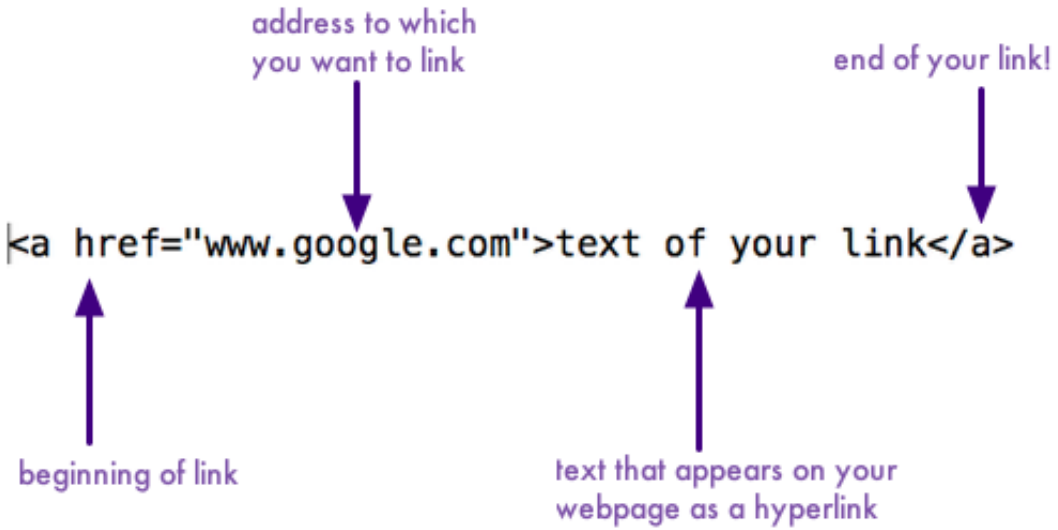
Et voilà. A list within a list. And you could put another list within that. And another within that. And so on and so forth.

Step 11: Links

- So far you've been making a stand-alone web page, which is all very well and nice, but what makes the Internet so special is that it all **links** together.

- The "H" and "T" in "HTML" stand for "hypertext", which basically means a system of linked text.

- An **anchor** tag (a) is used to define a link, but you also need to add something to the anchor tag – the **destination** of the link.



Add your Own Link

```
<!DOCTYPE html>
<html>
<head>
<title>My first web page</title>
</head>
<body>
<h1>My first web page</h1>
<h2>What this is?</h2>
<p>A simple page put together using HTML</p>
<h2>Why this is?</h2>
<p>To learn HTML</p>
<h2>Where can I find this tutorial?</h2>
 <p> On the class website
<a href="https://gebeasley.org/DH390"</a>under Week 7.</a>
</p>
</body>
</html>
```

Step 12: Images

- Things might seem a little bland and boring with all of this text formatting. Of course, the web is not just about text, it is a multi-media extravaganza and the most common form of sparkle is the image.
- **The img tag is used to put an image in an HTML document and it looks like this:**
  - **<img src="shutterstock_569455963.jpg (1024×512) (eirma.org)" width="120" height="90" alt="History">**
- The **src** attribute tells the browser where to find the image. Like the **<a>** tag, this can be absolute, as the above example demonstrates, but is usually relative.
- The **width and height** attributes are necessary because if they are excluded, the browser will tend to calculate the size as the image loads, instead of when the page loads, which means that the layout of the document may jump around while the page is loading.
- The **alt** attribute is the alternative description. This is an accessibility consideration, providing meaningful information for users who are unable to see the image (if they are visually impaired, for example).
- Note that, like the
 tag, because the <img> element does not enclose any content, no closing tag is required.

Linking Images

```
<img src="http://poolhouse.s3.amazonaws.com/
blog-assets-two/2015/03/sanedogs1.jpg">
```

OR

```
<img src="dog.jpg">
```

- You can link to an image in one of two ways. If there's an image out there on the Web that you'd like to embed on your page, you can grab the link to the image by right-clicking on the image and clicking Copy Image URL. Then you can embed the image by pasting the link, as above.

- If you have an image on your computer that you'd like to use, you can save it at the same place where your html file is saved and link directly to it by entering the file name, as above.

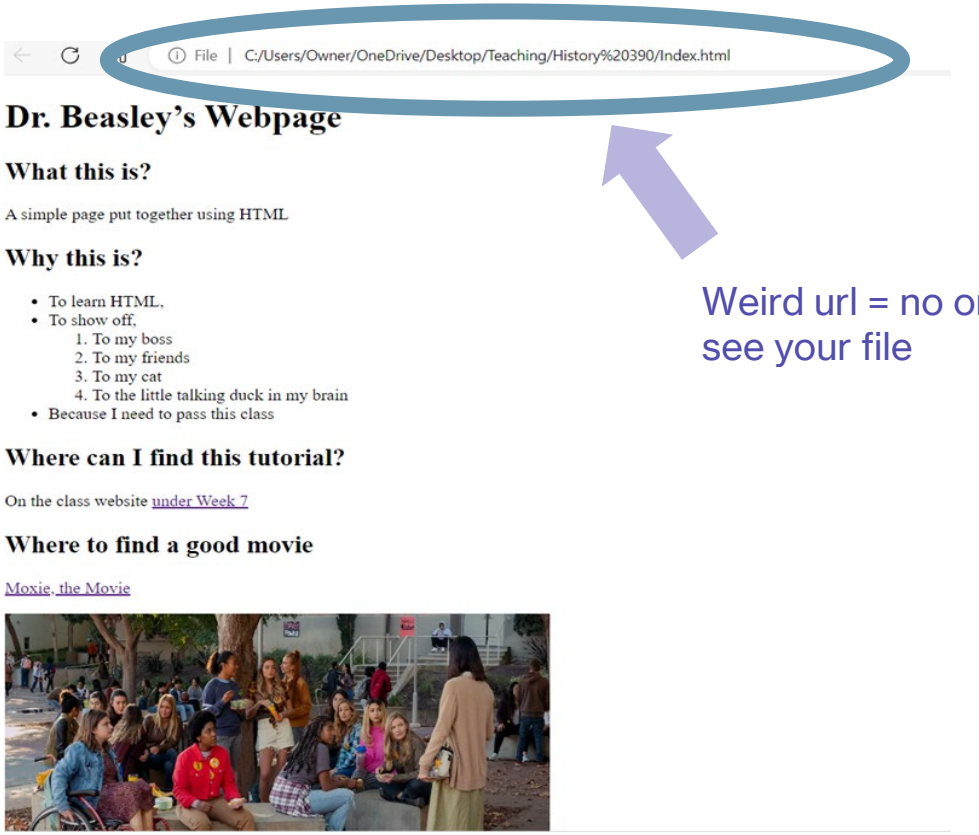
Adding an image into your webpage

```
<!DOCTYPE html>
 <html>
<head>
<title>My first web page</title>
</head>
<body>
 <h1>Dr. Beasley's Webpage</h1>
<h2>What this is?</h2>
<p>A simple page put together using HTML</p>
<h2>Why this is?</h2>
<ul>
<li>To learn HTML,</li>
 <li> To show off,
<ol>
<li>To my boss</li>
<li>To my friends</li>
<li>To my cat</li>
<li>To the little talking duck in my brain</li>
</ol> </li>
<li>Because I need to pass this class</li>
</ul>
<h2>Where can I find the tutorial?</h2>
<p> On the class website <a href="https://gebeasley.org/DH390"</a>under Week 7</a></p>
<h2>What is a movie you recommend? </h2>
<p> <a href=" https://www.imdb.com/title/tt6432466"> Moxie, the Movie </a> </p>
<p> <img src=" https://media.distractify.com/brand-img/HGZsg5R5F/0x0/moxiemovienetflix-1614801038540.jpg" width="500" height="300" alt="Riot Grrl"></p>
</body>
</html>
```

Step 13: Create your first HTML Page

If you have gone through all of the pages in this **HTML Tutorial,** then you should be a competent HTMLer. In fact, you should be better than most!!

But wait! No one can see your page!



But wait! No one can see your page!, continued

- If you look closely at the URL on your webpage, you'll see that it looks funny: it starts with **file:///** not **http://**
- That's because you're working on your file locally, meaning just on your own computer. No one else can see your webpage at the moment.
- In order for other people to see your file, it has to be hosted on a server, meaning moved to a special computer whose job is to broadcast files to the internet (like your WordPress site)
- See this tutorial on how to add HTMLto your WordPress site:
  - How to Edit HTML in WordPress Code Editor (Beginner's Guide) (wpbeginner.com)

Other Key Elements

This list includes the elements you most likely will use. For a full list of HTML Elements, go to this reference sheet: HTML Element Reference - By Category

Element	Meaning	Purpose
<b>	Bold	Highlight important information
****	Strong	Similarly to bold, to highlight key text
<i>	Italic	To denote text
****	Emphasised Text	Usually used as image captions
<mark>	Marked Text	Highlight the background of the text
<small>	Small Text	To shrink the text
<strike>	Striked Out Text	To place a horizontal line across the text
<u>	Underlined Text	Used for links or text highlights
<ins>	Inserted Text	Displayed with an underline to show an inserted text
<sub>	Subscript Text	Typographical stylistic choice
<sup>	Superscript Text	Another typographical presentation style