**HTML**

HEADLINE & SUB-HEADLINE & PARAGRAPH

<h1> This is the main headline </h1>

<h2> This is the second headline </h2>

<h3> It can go up to sub-headline </h3>

<p> I am a paragraph </p>

<!— You can in insert comment inside these tags -->

HTML5 introduces more descriptive HTML tags. These include header, footer, nav, video, article, section and others.

These tags make your HTML easier to read, and also help with Search Engine Optimization (SEO) and accessibility.

The main HTML5 tag helps search engines and other developers find the main content of your page.

Example;

<main>

This is the main paragraph.

</main>

**ADDING IMAGE ON THE WEBSITE**

You can add images to your website by using the img element and point to a specific image's URL using the src attribute.

An example of this would be:

<img src="https://www.your-image-source.com/your-image.jpg">

Note that img elements are self-closing.

All img elements **must** have an alt attribute. The text inside an alt attribute is used for screen readers to improve accessibility and is displayed if the image fails to load.

Note: If the image is purely decorative, using an empty alt attribute is a best practice.

Ideally the alt attribute should not contain special characters unless needed.

Let's add an alt attribute to our img example above:

<img src="https://www.your-image-source.com/your-image.jpg" alt="Author standing on a beach with two thumbs up.">

You can use anchor elements to link to content outside of your web page.

Anchor elements need a destination web address called an href attribute. They also need anchor text. Here's an example:

<a href="https://freecodecamp.org">this links to freecodecamp.org</a>

Then your browser will display the text **"this links to freecodecamp.org"** as a link you can click. And that link will take you to the web address **https://www.freecodecamp.org**.

**You can nest links within other text elements.**

<p>  
Here's a <a href="http://freecodecamp.org"> link to freecodecamp.org</a> for you to follow.  
</p>

Let's break down the example:

Normal text is wrapped in the p element:  
<p> Here's a ... for you to follow. </p>

Next is the anchor element <a>(which requires a closing tag </a>):  
<a> ... </a>

href is an anchor tag attribute that contains the URL address of the link:  
<a href="http://freecodecamp.org"> ... </a>

The text, **"link to freecodecamp.org"**, within the anchor element called anchor text, will display a link to click:  
<a href=" ... ">link to freecodecamp.org</a>

The final output of the example will look like this:

Here's a [link to freecodecamp.org](http://freecodecamp.org) for you to follow.

The current value of the href attribute is a link that points to "http://freecatphotoapp.com". Replace the href attribute value with a #, also known as a hash symbol, to create a dead link.

For example: href="#"

You can make elements into links by nesting them within an a element.

Nest your image within an a element. Here's an example:

<a href="#"><img src="https://bit.ly/fcc-running-cats" alt="Three kittens running towards the camera."></a>

Remember to use # as you’re a element's href property in order to turn it into a dead link.

NESTING ELEMENTS

<p> My cat is <strong>very</strong> grumpy. </p>

Output: My cat is **very** grumpy.

BLOCK VS INLINE ELEMENTS

There are two important categories of elements in HTML which you should know about. They are block-level elements and inline elements.

* Block-level elements form a visible block on a page — they will appear on a new line from whatever content went before it, and any content that goes after it will also appear on a new line. Block-level elements tend to be structural elements on the page that represent, for example, paragraphs, lists, navigation menus, footers, etc. A block-level element wouldn't be nested inside an inline element, but it might be nested inside another block-level element.
* Inline elements are those that are contained within block-level elements and surround only small parts of the document’s content, not entire paragraphs and groupings of content. An inline element will not cause a new line to appear in the document; they would normally appear inside a paragraph of text, for example an [<a>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/a) element (hyperlink) or emphasis elements such as [<em>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/em) or [<strong>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/strong).

Take the following example:

<em>first</em><em>second</em><em>third</em>

<p>fourth</p><p>fifth</p><p>sixth</p>

[<em>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/em) is an inline element, so as you can see below, the first three elements sit on the same line as one another with no space in between. On the other hand, [<p>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/p) is a block-level element, so each element appears on a new line, with space above and below each (the spacing is due to default [CSS styling](https://developer.mozilla.org/en-US/docs/Learn/CSS/Introduction_to_CSS) that the browser applies to paragraphs).

Attributes

Elements can also have attributes, which looks like this:



Attributes contain extra information about the element which you don't want to appear in the actual content. In this case, the class attribute allows you to give the element an identifying name that can be later used to target the element with style information and other things.

An attribute should have:

1. A space between it and the element name (or the previous attribute, if the element already has one or more attributes.)
2. The attribute name, followed by an equals sign.
3. An attribute value, with opening and closing quote marks wrapped around it.

ADDING ATTRIBUTES TO AN ELEMENT

Another example of an element is [<a>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/a) — this stands for "anchor" and will make the piece of text it wraps around into a hyperlink. This can take a number of attributes, but several are as follows:

* href: This attribute specifies as its value the web address that you want the link to point to; where the browser navigates to when the link is clicked. For example, href="https://www.mozilla.org/".
* title: The title attribute specifies extra information about the link, such as what the page is that you are linking to. For example, title="The Mozilla homepage". This will appear as a tooltip when hovered over.
* target: The target attribute specifies the browsing context which will be used to display the link. For example, target="\_blank" will display the link in a new tab. If you want to display the link in the current tab just omit this attribute.

Edit the line below in the Input area to turn it into a link to your favorite website. First, add the <a> element. Second, add the href attribute and the title attribute. Lastly, specify target attribute to open the link in the new tab. You'll be able to see your changes update live in the Output area. You should see a link that when hovered over displays the title attribute's content, and when clicked navigates to the web address in the href element. Remember that you need to include a space between the element name, and each attribute.

Example:

<p>A link to my <a href="https://www.mozilla.org/" title="The Mozilla homepage" target="\_blank">favorite website</a>.</p>

Boolean Attributes

You'll sometimes see attributes written without values — this is perfectly allowed. These are called Boolean attributes, and they can only have one value, which is generally the same as the attribute name. As an example, take the disabled attribute, which you can assign to form input elements if you want them to be disabled (greyed out) so the user can't enter any data in them.

<input type="text" disabled="disabled">

As shorthand, it is perfectly allowable to write this as follows (we've also included a non-disabled form input element for reference, to give you more of an idea what is going on):

<input type="text" disabled>

<input type="text">

Both will give you an output as follows: