**<html> is the root element and defines the whole HTML document**

**<body> defines the document’s body. It contains the h1, h2, p, etc**

**<br> tag defines a line break**

**List of HMTL tags in order:** [**https://www.w3schools.com/tags/default.asp**](https://www.w3schools.com/tags/default.asp)

## **Introduction to HTML5 Elements**

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

## **These tags give a descriptive structure to your HTML, make your HTML easier to read, and 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 usage, a main element with two child elements nested inside it:**

## **<main>**

## **<h1>Hello World</h1>**

## **<p>Hello Paragraph</p>**

## **</main>**

## **Note: Many of the new HTML5 tags and their benefits are covered in the Applied Accessibility section.**

## 

## **Create a second p element after the existing p element with the following kitty ipsum text: Purr jump eat the grass rip the couch scratched sunbathe, shed everywhere rip the couch sleep in the sink fluffy fur catnip scratched.**

## **Then, create a main element and nest the two p elements inside the main element.**

## **Add Images to Your Website**

## **Passed**

## **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.freecatphotoapp.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.freecatphotoapp.com/your-image.jpg" alt="A business cat wearing a necktie.">**

## 

## **Let's try to add an image to our website:**

## **Within the existing main element, insert an img element before the existing p elements.**

## **Now set the src attribute so that it points to this url:**

## **https://bit.ly/fcc-relaxing-cat**

## **Finally, don't forget to give your img element an alt attribute with applicable text.**

## **Link to External Pages with Anchor Elements**

## **Passed**

## **You can use a (*anchor*) elements to link to content outside of your web page.**

## **a 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.**

## 

## **Create an a element that links to https://freecatphotoapp.com and has "cat photos" as its anchor text.**

## 

## 

## 

## 

## **Link to Internal Sections of a Page with Anchor Elements**

a (*anchor*) elements can also be used to create internal links to jump to different sections within a webpage.

To create an internal link, you assign a link's href attribute to a hash symbol ***# plus the value of the id attribute*** for the element that you want to internally link to, usually further down the page. You then need to add the same id attribute to the element you are linking to. ***An id is an attribute that uniquely describes an element.***

Below is an example of an internal anchor link and its target element:

<a href="#contacts-header">Contacts</a>

...

<h2 id="contacts-header">Contacts</h2>

When users click the Contacts link, they'll be taken to the section of the webpage with the **Contacts** header element.

## **Nest an Anchor Element within a Paragraph**

You can nest links within other text elements.

<p>

Here's a <a target="\_blank" 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> target is an anchor tag attribute that specifies where to open the link and the value "\_blank" specifies to open the link in a new tab 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 a 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.

## **Make Dead Links Using the Hash Symbol**

Sometimes you want to add a elements to your website before you know where they will link.

This is also handy when you're changing the behavior of a link using JavaScript, which we'll learn about later.

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

For example: href="#"

## **Turn an Image into a Link**

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 your a element's href property in order to turn it into a dead link.

## **Create a Bulleted Unordered List**

HTML has a special element for creating unordered lists, or bullet point style lists.

Unordered lists start with an opening <ul> element, followed by any number of <li> elements. Finally, unordered lists close with a </ul>

For example:

<ul>

<li>milk</li>

<li>cheese</li>

</ul>

would create a bullet point style list of "milk" and "cheese".

## **Create an Ordered List**

HTML has another special element for creating ordered lists, or numbered lists.

Ordered lists start with an opening <ol> element, followed by any number of <li> elements. Finally, ordered lists are closed with the </ol> tag.

For example:

<ol>

<li>Garfield</li>

<li>Sylvester</li>

</ol>

would create a numbered list of "Garfield" and "Sylvester".

## **Create a Text Field**

## Now let's create a web form.

input elements are a convenient way to get input from your user.

You can create a text input like this:

<input type="text">

Note that input elements are self-closing.

## **Add Placeholder Text to a Text Field**

Placeholder text is what is displayed in your input element before your user has inputted anything.

You can create placeholder text like so:

<input type="text" placeholder="this is placeholder text">

**Note:** Remember that input elements are self-closing.

## **Create a Form Element**

You can build web forms that actually submit data to a server using nothing more than pure HTML. You can do this by specifying an action on your form element.

For example:

<form action="/url-where-you-want-to-submit-form-data"></form>

## **Add a Submit Button to a Form**

## 

Let's add a submit button to your form. Clicking this button will send the data from your form to the URL you specified with your form's action attribute.

Here's an example submit button:

<button type="submit">this button submits the form</button>

## **Use HTML5 to Require a Field**

You can require specific form fields so that your user will not be able to submit your form until he or she has filled them out.

For example, if you wanted to make a text input field required, you can just add the attribute required within your input element, like this: <input type="text" required>

## **Create a Set of Radio Buttons**

You can use radio buttons for questions where you want the user to only give you one answer out of multiple options.

Radio buttons are a type of input.

Each of your radio buttons can be nested within its own label element. By wrapping an input element inside of a label element it will automatically associate the radio button input with the label element surrounding it.

All related radio buttons should have the same name attribute to create a radio button group. By creating a radio group, selecting any single radio button will automatically deselect the other buttons within the same group ensuring only one answer is provided by the user.

Here's an example of a radio button:

<label>

<input type="radio" name="indoor-outdoor">Indoor

</label>

It is considered best practice to set a for attribute on the label element, with a value that matches the value of the id attribute of the input element. This allows assistive technologies to create a linked relationship between the label and the child input element. For example:

<label for="indoor">

<input id="indoor" type="radio" name="indoor-outdoor">Indoor

</label>

## **Create a Set of Checkboxes**

Forms commonly use checkboxes for questions that may have more than one answer.

Checkboxes are a type of input.

Each of your checkboxes can be nested within its own label element. By wrapping an input element inside of a label element it will automatically associate the checkbox input with the label element surrounding it.

All related checkbox inputs should have the same name attribute.

It is considered best practice to explicitly define the relationship between a checkbox input and its corresponding label by setting the for attribute on the label element to match the id attribute of the associated input element.

Here's an example of a checkbox:

<label for="loving"><input id="loving" type="checkbox" name="personality"> Loving</label>

## **Use the value attribute with Radio Buttons and Checkboxes**

When a form gets submitted, the data is sent to the server and includes entries for the options selected. Inputs of type radio and checkbox report their values from the value attribute.

For example:

<label for="indoor">

<input id="indoor" value="indoor" type="radio" name="indoor-outdoor">Indoor

</label>

<label for="outdoor">

<input id="outdoor" value="outdoor" type="radio" name="indoor-outdoor">Outdoor

</label>

Here, you have two radio inputs. When the user submits the form with the indoor option selected, the form data will include the line: indoor-outdoor=indoor. This is from the name and value attributes of the "indoor" input.

If you omit the value attribute, the submitted form data uses the default value, which is on. In this scenario, if the user clicked the "indoor" option and submitted the form, the resulting form data would be indoor-outdoor=on, which is not useful. So the value attribute needs to be set to something to identify the option.

## **Check Radio Buttons and Checkboxes by Default**

You can set a checkbox or radio button to be checked by default using the checked attribute.

To do this, just add the word "checked" to the inside of an input element. For example:

<input type="radio" name="test-name" checked>

## **Nest Many Elements within a Single div Element**

The div element, also known as a division element, is a general purpose container for other elements.

The div element is probably the most commonly used HTML element of all.

Just like any other non-self-closing element, you can open a div element with <div> and close it on another line with </div>.

## **Declare the Doctype of an HTML Document**

The challenges so far have covered specific HTML elements and their uses. However, there are a few elements that give overall structure to your page, and should be included in every HTML document.

At the top of your document, you need to tell the browser which version of HTML your page is using. HTML is an evolving language, and is updated regularly. Most major browsers support the latest specification, which is HTML5. However, older web pages may use previous versions of the language.

You tell the browser this information by adding the <!DOCTYPE ...> tag on the first line, where the ... part is the version of HTML. For HTML5, you use <!DOCTYPE html>.

The ! and uppercase DOCTYPE is important, especially for older browsers. The html is not case sensitive.

Next, the rest of your HTML code needs to be wrapped in html tags. The opening <html> goes directly below the <!DOCTYPE html> line, and the closing </html> goes at the end of the page.

Here's an example of the page structure:

<!DOCTYPE html>

<html>

<!-- Your HTML code goes here -->

</html>

## **Define the Head and Body of an HTML Document**

You can add another level of organization in your HTML document within the html tags with the head and body elements. Any markup with information about your page would go into the head tag. Then any markup with the content of the page (what displays for a user) would go into the body tag.

Metadata elements, such as link, meta, title, and style, typically go inside the head element.

Here's an example of a page's layout:

<!DOCTYPE html>

<html>

<head>

<!-- metadata elements -->

</head>

<body>

<!-- page contents -->

</body>

</html>