

Basic HTML and HTML 5

HTML is a markup language that uses a special syntax or notation to describe the structure of a webpage to the browser. HTML elements usually have opening and closing tags that surround and give meaning to content. For example, different elements can describe text as a heading, paragraph, or list item.

HTML Elements

First, you'll start by building a simple web page using HTML. You can edit code in your code editor, which is embedded into this web page.

In your code editor write the following

```
<h1>Hello</h1>
```

That's an HTML element.

Most HTML elements have an opening tag and a closing tag.

Opening tags look like this:

```
<h1>
```

Closing tags look like this:

```
</h1>
```

The only difference between opening and closing tags is the forward slash after the opening bracket of a closing tag.

Exercise

Change your h1 element's text to say Hello World.

```
<h1>Hello</h1>
```

Headline with the h2 Element

The h2 element you will be adding in this step will add a level two heading to the web page.

This element tells the browser about the structure of your website. h1 elements are often used for main headings, while h2 elements are generally used for subheadings. There are also h3, h4, h5 and h6 elements to indicate different levels of subheadings.

```
<h1>Hello World</h1>
```

Exercise

Add an h2 tag that says "CatPhotoApp" to create a second HTML element below your "Hello World" h1 element.

Inform with the Paragraph Element

p elements are the preferred element for paragraph text on websites. p is short for "paragraph".

You can create a paragraph element like this:

```
<p>I'm a p tag!</p>
```

Exercise

Create a p element below your h2 element, and give it the text Hello Paragraph.

Note: All HTML tags are written in lowercase, for example `<p></p>` and not `<P></P>`.

Fill in the Blank with Placeholder Text

Web developers traditionally use lorem ipsum text as placeholder text. The lorem ipsum text is randomly scraped from a famous passage by Cicero of Ancient Rome.

Lorem ipsum text has been used as placeholder text by typesetters since the 16th century, and this tradition continues on the web. Well, 5 centuries is long enough. Since we're building a CatPhotoApp, let's use something called "kitty ipsum" text.

Exercise

Replace the text inside your p element with the first few words of this kitty ipsum text:

Kitty ipsum dolor sit amet, shed everywhere shed everywhere stretching attack your ankles chase the red dot, hairball run catnip eat the grass sniff.

Uncomment HTML

Commenting is a way that you can leave comments for other developers within your code without affecting the resulting output that is displayed to the end user.

Commenting is also a convenient way to make code inactive without having to delete it entirely.

Comments in HTML start with `<!--` and end with `-->`

Exercise

Uncomment your h1, h2 and p elements.

Code

```
<!--  
<h1>Hello World</h1>  
  
<h2>CatPhotoApp</h2>  
  
<p>Kitty ipsum dolor sit amet, shed everywhere shed everywhere stretching attack your  
ankles chase the red dot, hairball run catnip eat the grass sniff.</p>  
-->
```

Comment out HTML

Remember that in order to start a comment, you need to use `<!--` and to end a comment, you need to use `-->`

Here you'll need to end the comment before your h2 element begins.

Exercise

Comment out your h1 element and your p element, but not your h2 element.

Code

```
<!--  
<h1>Hello World</h1>  
  
<h2>CatPhotoApp</h2>  
  
<p>Kitty ipsum dolor sit amet, shed everywhere shed everywhere stretching attack your ankles chase the red  
dot, hairball run catnip eat the grass sniff.</p>  
-->
```

Delete HTML Elements

Our phone doesn't have much vertical space.

Let's remove the unnecessary elements so we can start building our CatPhotoApp.

Exercise

Delete your h1 element so we can simplify our view.

Code

```
<h1>Hello World</h1>
```

```
<h2>CatPhotoApp</h2>
```

```
<p>Kitty ipsum dolor sit amet, shed everywhere shed everywhere stretching attack your ankles chase the red dot, hairball run catnip eat the grass sniff.</p>
```

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>
```

Exercise

Apply HTML5 elements to the webpage below

Code

```
<h2>CatPhotoApp</h2>
```

```
<p>Kitty ipsum dolor sit amet, shed everywhere shed everywhere stretching attack your ankles chase the red dot, hairball run catnip eat the grass sniff.</p>
```

Add Images to Your 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:

```

```

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:

```

```

Exercise

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 the url or file `"fcc-relaxing-cat.jpg"` Finally, don't forget to give your `img` element an `alt` attribute with applicable text.

Code

```
<h2>CatPhotoApp</h2>
```

```
<main>
```

```
  <p>Kitty ipsum dolor sit amet, shed everywhere shed everywhere stretching attack your ankles chase the red dot, hairball run catnip eat the grass sniff.</p>
```

```
  <p>Purr jump eat the grass rip the couch scratched sunbathe, shed everywhere rip the couch sleep in the sink fluffy fur catnip scratched.</p>
```

```
</main>
```

CatPhotoApp



Kitty ipsum dolor sit amet, shed everywhere shed everywhere stretching attack your ankles chase the red dot, hairball run catnip eat the grass sniff.

Purr jump eat the grass rip the couch scratched sunbathe, shed everywhere rip the couch sleep in the sink fluffy fur catnip scratched.

Freecatphotoapp.html

Cat Photos



Three kittens running towards the camera.



A cute orange cat lying on its back.

Link to External Pages with Anchor Elements

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://www.w3schools.com/">this links to w3Schools.com</a>
```

Then your browser will display the text this links to W3Schools as a link you can click. And that link will take you to the web address <https://www.w3schools.com>.

Exercise

Create an a element that links to freecatphotoapp.html and has "cat photos" as its anchor text.

Code

```
<h2>CatPhotoApp</h2>
<main>
  

  <p>Kitty ipsum dolor sit amet, shed everywhere shed everywhere stretching attack your ankles chase the red dot, hairball run catnip eat the grass sniff.</p>
  <p>Purr jump eat the grass rip the couch scratched sunbathe, shed everywhere rip the couch sleep in the sink fluffy fur catnip scratched.</p>
</main>
```

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.

Exercise

Change your external link to an internal link by changing the href attribute to "#footer" and the text from cat photos to Jump to Bottom.

Remove the target="_blank" attribute from the anchor tag since this causes the linked document to open in a new window tab.

Then add an id attribute with a value of footer to the <footer> element at the bottom of the page.

Code

```

<h2>CatPhotoApp</h2>

<main>

  <a href="freecatphotoapp.html" target="_blank">cat photos</a>

  

  <p>Kitty ipsum dolor sit amet, shed everywhere shed everywhere stretching attack your
  r ankles chase the red dot, hairball run catnip eat the grass sniff. Purr jump eat the
  grass rip the couch scratched sunbathe, shed everywhere rip the couch sleep in the si
  nk fluffy fur catnip scratched. Kitty ipsum dolor sit amet, shed everywhere shed every
  where stretching attack your ankles chase the red dot, hairball run catnip eat the gra
  ss sniff.</p>

  <p>Purr jump eat the grass rip the couch scratched sunbathe, shed everywhere rip the
  couch sleep in the sink fluffy fur catnip scratched. Kitty ipsum dolor sit amet, shed
  everywhere shed everywhere stretching attack your ankles chase the red dot, hairball
  run catnip eat the grass sniff. Purr jump eat the grass rip the couch scratched sunbat
  he, shed everywhere rip the couch sleep in the sink fluffy fur catnip scratched.</p>

  <p>Meowwww loved it, hated it, loved it, hated it yet spill litter box, scratch at o
  wner, destroy all furniture, especially couch or lay on arms while you're using the ke
  yboard. Missing until dinner time toy mouse squeak roll over. With tail in the air lou
  nge in doorway. Man running from cops stops to pet cats, goes to jail.</p>

  <p>Intently stare at the same spot poop in the plant pot but kitten is playing with
  dead mouse. Get video posted to internet for chasing red dot leave fur on owners cloth
  es meow to be let out and mesmerizing birds leave fur on owners clothes or favor packa
  ging over toy so purr for no reason. Meow to be let out play time intently sniff hand
  run outside as soon as door open yet destroy couch.</p>

</main>

<footer>Copyright Cat Photo App</footer>

```

Nest an Anchor Element within a Paragraph

You can nest links within other text elements.

<p>

Here's a link to www.w3schools.com 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> ...

target is an anchor tag attribute that specifies where to open the link. The value _blank specifies to open the link in a new tab. The href is an anchor tag attribute that contains the URL address of the link:

 link to www.w3schools.com

The text, link to www.w3schools.com, within the a element is called anchor text, and will display the link to click:

Exercise

Nest the existing <a> element within a new <p> element. Do not create a new anchor tag. The new paragraph should have text that says View more cat photos, where cat photos is a link, and the rest is plain text.

Code

```
<h2>CatPhotoApp</h2>
```

```
<main>
```

```
  <a href="freecatphotoapp.html" target="_blank">cat photos</a>
```

```
  
```

```
  <p>Kitty ipsum dolor sit amet, shed everywhere shed everywhere stretching attack your ankles chase the red dot, hairball run catnip eat the grass sniff.</p>
```

```
<p>Purr jump eat the grass rip the couch scratched sunbathe, shed everywhere rip the
couch sleep in the sink fluffy fur catnip scratched.</p>
</main>
```

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.

Exercise

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

For example: href="#"

Code

```
<h2>CatPhotoApp</h2>
<main>
  <p>Click here to view more <a href="freecatphotoapp.html" target="_blank">cat photos
</a>.</p>

  

  <p>Kitty ipsum dolor sit amet, shed everywhere shed everywhere stretching attack you
r ankles chase the red dot, hairball run catnip eat the grass sniff.</p>
  <p>Purr jump eat the grass rip the couch scratched sunbathe, shed everywhere rip the
couch sleep in the sink fluffy fur catnip scratched.</p>
</main>
```

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="#"></a>
```

Remember to use `#` as your `a` element's `href` property in order to turn it into a dead link.

Exercise

Place the existing image element within an `a` (anchor) element.

Once you've done this, hover over your image with your cursor. Your cursor's normal pointer should become the link clicking pointer. The photo is now a link.

Code

```
<h2>CatPhotoApp</h2>
<main>
  <p>Click here to view more <a href="#">cat photos</a>.</p>

  

  <p>Kitty ipsum dolor sit amet, shed everywhere shed everywhere stretching attack you
r ankles chase the red dot, hairball run catnip eat the grass sniff.</p>
  <p>Purr jump eat the grass rip the couch scratched sunbathe, shed everywhere rip the
couch sleep in the sink fluffy fur catnip scratched.</p>
</main>
```

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 `` element, followed by any number of `` elements. Finally, unordered lists close with a ``.

For example:

```
<ul>
  <li>milk</li>
  <li>cheese</li>
</ul>
```

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

Exercise

Remove the last two p elements and create an unordered list of three things that cats love at the bottom of the page.

Code

```
<h2>CatPhotoApp</h2>
<main>
  <p>Click here to view more <a href="#">cat photos</a>.</p>

  <a href="#"></a>

  <p>Kitty ipsum dolor sit amet, shed everywhere shed everywhere stretching attack you
r ankles chase the red dot, hairball run catnip eat the grass sniff.</p>
  <p>Purr jump eat the grass rip the couch scratched sunbathe, shed everywhere rip the
couch sleep in the sink fluffy fur catnip scratched.</p>
</main>
```

Create an Ordered List

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

Ordered lists start with an opening `` element, followed by any number of `` elements. Finally, ordered lists are closed with the `` tag.

For example:

```
<ol>
  <li>Garfield</li>
  <li>Sylvester</li>
```


would create a numbered list of Garfield and Sylvester.

Exercise

Create an ordered list of the top 3 things cats hate the most.

Code

```
<h2>CatPhotoApp</h2>
<main>
  <p>Click here to view more <a href="#">cat photos</a>.</p>

  <a href="#"></a>

  <p>Things cats love:</p>
  <ul>
    <li>cat nip</li>
    <li>laser pointers</li>
    <li>lasagna</li>
  </ul>
  <p>Top 3 things cats hate:</p>

</main>
```

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.

Exercise

Create an input element of type text below your lists.

Code

```
<h2>CatPhotoApp</h2>
<main>
  <p>Click here to view more <a href="#">cat photos</a>.</p>

  <a href="#"></a>

  <p>Things cats love:</p>
  <ul>
    <li>cat nip</li>
    <li>laser pointers</li>
    <li>lasagna</li>
  </ul>
  <p>Top 3 things cats hate:</p>
  <ol>
    <li>flea treatment</li>
    <li>thunder</li>
    <li>other cats</li>
  </ol>

</main>
```

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.

Exercise

Set the placeholder value of your text input to "cat photo URL".

Code

```
<h2>CatPhotoApp</h2>
<main>
  <p>Click here to view more <a href="#">cat photos</a>.</p>
```

```
<a href="#"></a>
```

```
<p>Things cats love:</p>  
<ul>  
  <li>cat nip</li>  
  <li>laser pointers</li>  
  <li>lasagna</li>  
</ul>  
<p>Top 3 things cats hate:</p>  
<ol>  
  <li>flea treatment</li>  
  <li>thunder</li>  
  <li>other cats</li>  
</ol>  
<input type="text">  
</main>
```

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 attribute on your form element.

For example:

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

Exercise

Nest the existing input element inside a form element and assign "https://www.freecatphotoapp.com/submit-cat-photo" to the action attribute of the form element.

Code

```
<h2>CatPhotoApp</h2>  
<main>
```

```

<p>Click here to view more <a href="#">cat photos</a>.</p>

<a href="#"></a>

<p>Things cats love:</p>
<ul>
  <li>cat nip</li>
  <li>laser pointers</li>
  <li>lasagna</li>
</ul>
<p>Top 3 things cats hate:</p>
<ol>
  <li>flea treatment</li>
  <li>thunder</li>
  <li>other cats</li>
</ol>
<input type="text" placeholder="cat photo URL">
</main>

```

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>
```

Exercise

Add a button as the last element of your form element with a type of submit, and Submit as its text.

Code

```

<h2>CatPhotoApp</h2>
<main>

```



```

<p>Click here to view more <a href="#">cat photos</a>.</p>

<a href="#"></a>

<p>Things cats love:</p>
<ul>
  <li>cat nip</li>
  <li>laser pointers</li>
  <li>lasagna</li>
</ul>
<p>Top 3 things cats hate:</p>
<ol>
  <li>flea treatment</li>
  <li>thunder</li>
  <li>other cats</li>
</ol>
<form action="freecatphotoapp.html/submit-cat-photo">
  <input type="text" placeholder="cat photo URL">
</form>
</main>

```

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>`

Exercise

Make your text input a required field, so that your user can't submit the form without completing this field.

Then try to submit the form without inputting any text. See how your HTML5 form notifies you that the field is required?

Code

```

<h2>CatPhotoApp</h2>

<main>

  <p>Click here to view more <a href="#">cat photos</a>.</p>

  <a href="#"></a>

  <p>Things cats love:</p>
  <ul>
    <li>cat nip</li>
    <li>laser pointers</li>
    <li>lasagna</li>
  </ul>
  <p>Top 3 things cats hate:</p>
  <ol>
    <li>flea treatment</li>
    <li>thunder</li>
    <li>other cats</li>
  </ol>
  <form action="freecatphotoapp.html/submit-cat-photo">
    <input type="text" placeholder="cat photo URL">
    <button type="submit">Submit</button>
  </form>
</main>

```

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 related input element.

For example:

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

```
<label for="indoor">Indoor</label>
```

We can also nest the input element within the label tags:

```
<label for="indoor">
```

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

```
</label>
```

Exercise

Add a pair of radio buttons to your form, each nested in its own label element. One should have the option of indoor and the other should have the option of outdoor. Both should share the name attribute of indoor-outdoor to create a radio group.

Code

```
<h2>CatPhotoApp</h2>
```

```
<main>
```

```
<p>Click here to view more <a href="#">cat photos</a>.</p>
```

```
<a href="#"></a>
```

```
<p>Things cats love:</p>
```

```
<ul>
```

```
<li>cat nip</li>
```

```
<li>laser pointers</li>
```

```
<li>lasagna</li>
```

```
</ul>
```

```
<p>Top 3 things cats hate:</p>
```

```

<ol>
  <li>flea treatment</li>
  <li>thunder</li>
  <li>other cats</li>
</ol>
<form action="freecatphotoapp.html/submit-cat-photo">
  <input type="text" placeholder="cat photo URL" required>
  <button type="submit">Submit</button>
</form>
</main>

```

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>

```

Exercise

Add to your form a set of three checkboxes. Each checkbox should be nested within its own label element. All three should share the name attribute of `personality`.

Code

```

<h2>CatPhotoApp</h2>

```

```

<main>
  <p>Click here to view more <a href="#">cat photos</a>.</p>

  <a href="#"></a>

  <p>Things cats love:</p>
  <ul>
    <li>cat nip</li>
    <li>laser pointers</li>
    <li>lasagna</li>
  </ul>
  <p>Top 3 things cats hate:</p>
  <ol>
    <li>flea treatment</li>
    <li>thunder</li>
    <li>other cats</li>
  </ol>
  <form action="freecatphotoapp.html/submit-cat-photo">
    <label for="indoor"><input id="indoor" type="radio" name="indoor-outdoor"> Indoor</label>
    <label for="outdoor"><input id="outdoor" type="radio" name="indoor-
outdoor"> Outdoor</label><br>
    <input type="text" placeholder="cat photo URL" required>
    <button type="submit">Submit</button>
  </form>
</main>

```

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.

Exercise

Give each of the existing radio and checkbox inputs the value attribute. Do not create any new radio or checkbox elements. Use the input label text, in lowercase, as the value for the attribute.

Code

```
<h2>CatPhotoApp</h2>

<main>

  <p>Click here to view more <a href="#">cat photos</a>.</p>

  <a href="#"></a>

  <p>Things cats love:</p>
  <ul>
    <li>cat nip</li>
    <li>laser pointers</li>
    <li>lasagna</li>
  </ul>
  <p>Top 3 things cats hate:</p>
  <ol>
    <li>flea treatment</li>
    <li>thunder</li>
    <li>other cats</li>
  </ol>
  <form action="freecatphotoapp.html/submit-cat-photo">
    <label for="indoor"><input id="indoor" type="radio" name="indoor-outdoor"> Indoor</label>
    <label for="outdoor"><input id="outdoor" type="radio" name="indoor-outdoor"> Outdoor</label><br>
    <label for="loving"><input id="loving" type="checkbox" name="personality"> Loving</label>
    <label for="lazy"><input id="lazy" type="checkbox" name="personality"> Lazy</label>
```

```

    <label for="energetic"><input id="energetic" type="checkbox" name="personality"> Energetic<
/label><br>

    <input type="text" placeholder="cat photo URL" required>

    <button type="submit">Submit</button>

</form>
</main>

```

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>
```

Exercise

Set the first of your radio buttons and the first of your checkboxes to both be checked by default.

Code

```

<h2>CatPhotoApp</h2>
<main>
  <p>Click here to view more <a href="#">cat photos</a>.</p>

  <a href="#"></a>

  <p>Things cats love:</p>
  <ul>
    <li>cat nip</li>
    <li>laser pointers</li>
    <li>lasagna</li>
  </ul>
  <p>Top 3 things cats hate:</p>
  <ol>
    <li>flea treatment</li>
    <li>thunder</li>
    <li>other cats</li>
  </ol>

```

```

</ol>
<form action="freecatphotoapp.html/submit-cat-photo">
  <label for="indoor"><input id="indoor" type="radio" name="indoor-
outdoor" value="indoor"> Indoor</label>
  <label for="outdoor"><input id="outdoor" type="radio" name="indoor-
outdoor" value="outdoor"> Outdoor</label><br>
  <label for="loving"><input id="loving" type="checkbox" name="personality" value="loving"> L
oving</label>
  <label for="lazy"><input id="lazy" type="checkbox" name="personality" value="lazy"> Lazy</l
abel>
  <label for="energetic"><input id="energetic" type="checkbox" name="personality" value="ener
getic"> Energetic</label><br>
  <input type="text" placeholder="cat photo URL" required>
  <button type="submit">Submit</button>
</form>
</main>

```

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>.

Exercise

Nest your "Things cats love" and "Top 3 things cats hate" lists all within a single div element.

Hint: Try putting your opening div tag above your "Things cats love" p element and your closing div tag after your closing ol tag so that both of your lists are within one div.

Code

```

<h2>CatPhotoApp</h2>
<main>
  <p>Click here to view more <a href="#">cat photos</a>.</p>

```



```

<a href="#"></a>

<p>Things cats love:</p>
<ul>
  <li>cat nip</li>
  <li>laser pointers</li>
  <li>lasagna</li>
</ul>
<p>Top 3 things cats hate:</p>
<ol>
  <li>flea treatment</li>
  <li>thunder</li>
  <li>other cats</li>
</ol>

<form action="freecatphotoapp.html/submit-cat-photo">
  <label for="indoor"><input id="indoor" type="radio" name="indoor-
outdoor" value="indoor" checked> Indoor</label>
  <label for="outdoor"><input id="outdoor" type="radio" name="indoor-
outdoor" value="outdoor"> Outdoor</label><br>
  <label for="loving"><input id="loving" type="checkbox" name="personality" value="loving" checked> Loving</label>
  <label for="lazy"><input id="lazy" type="checkbox" name="personality" value="lazy"> Lazy</label>
  <label for="energetic"><input id="energetic" type="checkbox" name="personality" value="energetic"> Energetic</label><br>
  <input type="text" placeholder="cat photo URL" required>
  <button type="submit">Submit</button>
</form>
</main>

```

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. Your HTML code would go in the space between the two html tags.

```
<!DOCTYPE html>
```

```
<html>
```

```
</html>
```

Exercise

Add a DOCTYPE tag for HTML5 to the top of the blank HTML document in the code editor. Under it, add opening and closing html tags, which wrap around an h1 element. The heading can include any text.

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>
```

```
    <meta />
```

```
  </head>
```

```
  <body>
```

```
    <div>
```

```
    </div>
```

```
  </body>
```

```
</html>
```

Exercise

Edit the markup so there's a head and a body. The head element should only include the title, and the body element should only include the h1 and p.

Code

```
<!DOCTYPE html>
<html>
  <title>The best page ever</title>

  <h1>The best page ever</h1>
  <p>Cat ipsum dolor sit amet, jump launch to pounce upon little yarn mouse, bare fangs at toy run hide in litter box until treats are fed. Go into a room to decide you didn't want to be in there anyway. I like big cats and i can not lie kitty ipsum dolor sit amet, shed everywhere shed everywhere stretching attack your ankles chase the red dot, hairball run catnip eat the grass sniff. Meow i could pee on this if i had the energy for slap owner's face at 5am until human fills food dish yet scamper. Knock dish off table head butt cant eat out of my own dish scratch the furniture. Make meme, make cute face. Sleep in the bathroom sink chase laser but pee in the shoe. Paw at your fat belly licks your face and eat grass, throw it back up kitty ipsum dolor sit amet, shed everywhere shed everywhere stretching attack your ankles chase the red dot, hairball run catnip eat the grass sniff.</p>

</html>
```