

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>
```

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="#"

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>
```

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>
```

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:

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>
```

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">
```

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.

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>
```

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.

Use `video` tag to play any video in our html, eg.

```
<video src="" controls></video>
```

Block and inline elements

By default, the width of the block elements is 100%. It doesn't matter how much content is present in it.

Inline element's width depends on the content it has inside of it.

egs of inline -> img, b, i, a, span

div -> division . This is a block level element.

more egs of block level elements -> p, h1 to h6, div

semantic html tags

semantic means meaning

Semantic html tags

```
<article>
  <aside>
    <details>
      <figcaption>
        <figure>
          <footer>
            <header>
              <main>
                <mark>
                  <nav>
                    <section>
                      <summary>
                        <time></time>
                      </summary>
                    </section></nav>
                  ></mark>
                </main>
              </header>
            </footer>
          </figure>
        </figcaption>
      </details>
    </aside>
  </article>
```

All these are block level semantic tags.

`` and `` are inline level semantic tags.

em -> emphasis (like italics)

In i tag, screen readers will read it normally, in em tag, they will read it with some emphasis on it.

strong -> (like bold)

In strong tag, same thing happens.

If possible, use semantic tags more.

`<h1></h1>` -> Tags

```
<fieldset>
  <label for="cars">Choose a car:</label>
  <select name="cars" id="dropdown">
    <option value="volvo">Volvo</option>
    <option value="saab">Saab</option>
    <option value="mercedes">Mercedes</option>
    <option value="audi">Audi</option>
  </select>
</fieldset>
```

This is a fieldset element (it has a lot of content inside of its opening and closing tags).

We have height and width of an element and not a tag.

Tags help us create an element in the browser.

`<hr>` tag creates an hr(horizontal rule/line) element in the browser.

Global and Custom Attributes

`<tag attribute='attribute-value'>`

Global Attribute -> Applicable to all html tags. eg. style, class, id, hidden

Custom attribute -> Applicable to specific tags. eg. src, href, alt

We can also give anything as an attribute eg. `<h1 ksd='hi'></h1>`

Title attribute is a global attribute, the value given in this attribute will be shown to you when you hover over the element(tooltip).

`<form>` tag creates a URL using the following things: action attribute value + name attribute value + input value given in form(query or q).

`<input>` is a self closing tag.

Different sites use different name attributes- eg. YT uses name='search_query' and flipkart uses name='q'.

We can also go to other websites from our form, just update the action attribute for this, eg:

```
<form action="https://www.youtube.com/results">
  <input type="text" name="search_query" />
</form>
```

```
<form action="https://www.flipkart.com/search">
  <input type="text" name="q" />
</form>
```

If we look closely, there is a submit field too in the input field, we can simply do:

```
<form action="/">
  <input type="submit" value="search" />
  <!-- this will make the text of our submit button from 'submit' to 'search' -->
</form>
```

Button also has many types, it has a default value of type "submit"

type='button' -> We can specify what action we have to perform when we click on this button. eg. We can make a back button out of this.

type='reset' -> This resets our form to the default value when we click on the button.

Also some other attributes are:

placeholder required

For input text values: minlength maxlength

For number input values: min max (min and max numbers, value (number given), it should be within the numbers entered).

🕒 To display label above the form input, wrap the label tag inside the div tag. Then use the input tag.

To give spaces between 2 form fields, wrap the input and the label inside the p tag.

`<!DOCTYPE html>` this tells the browser that we are using the latest version of HTML i.e. HTML5.

`<html>` This represents an html document.

title -> This value gets shown in the browser tab.

link:favicon -> shortcut for favicon.

lang='en' -> This helps the search engines, browsers and screen readers about what language our html document is in.

UTF-8 -> This includes all text and emojis (latest version)

`Go to Top` This takes us to the top of the page (without loading the page, use #). If we use no #, then our page will reload and then we go to top.

`<big>` this is used to emphasize the text.

`<small>` this is used to keep the text simple

`<pre></pre>` This tag is used to give spaces and line breaks.

target=_blank and target=_main are same