# Learn HTML

**Contents**

[Learn HTML 1](#_Toc122025417)

[HTML 2](#_Toc122025418)

[HTML Elements 3](#_Toc122025419)

[HTML Attributes 5](#_Toc122025420)

[HTML Tables 6](#_Toc122025421)

[HTML Forms 8](#_Toc122025422)

[Submitting a Form 8](#_Toc122025423)

[Form Elements 9](#_Toc122025424)

[<Input> Types 9](#_Toc122025425)

[<input> Attributes 10](#_Toc122025426)

[HTML Example – Portfolio Project 11](#_Toc122025427)

## HTML

<head>

  <link href="style.css" type="text/css" rel="stylesheet">

  <link href="https://fonts.googleapis.com/css?family=Droid%20Serif" rel="stylesheet">

</head>

<a href="http://www.codecademy.com" target=”\_blank”>>Visit this site</a>

<a href="http://www.codecademy.com" <img src="logo.jpg">Click this image</a>

<a href="./about.html">The URL for this anchor element is a relative file path.</a>

<p> <a href="https://www.codecademy.com" title="Codecademy is an online learning platform">Codecademy</a> is the best place to learn to code!</p>

<p id="id-of-element-to-link-to">A different part of the page!</p>

<a href="#id-of-element-to-link-to">Take me to a different part of the page</a>

<video src="4kvideo.mp4">video not supported</video>

<audio src="koreanhiphop.mp3"></audio>

<embed src="babyyoda.gif"/>

<img src="path/to/image" alt="text describing image" />

<figure>

     <img src="qwerty.jpg">

     <figcaption>The image shows the layout of a qwerty keyboard.</figcaption>

</figure>

<section>

  <h2>Top Sports league in America</h2>

  <article>

    <p>One of the top sports league is the nba.</p>

  </article>

  <aside>

  <!--Additional information-->

  </aside>

</section>

<h2 style="text-align: center;">Centered text</h2>

<p style="color: blue; font-size: 18px;">Blue, 18-point text</p>

<div class="value1 value2 value3"></div>

## HTML Elements

**<!DOCTYPE html>** is required as the first line of an HTML document. The doctype declaration is an instruction to the browser about what type of document to expect and which version of HTML is being used.

**<html>** is the root of an HTML document and should be added after the !DOCTYPE declaration. All content/structure for an HTML document should be contained between the opening and closing <html> tags.

**<head>** contains general information about an HTML page that isn’t displayed on the page itself. This information is called metadata and includes things like the **title** of the HTML document and **links** to stylesheets.

**<title>** is displayed in the browser’s title bar or tab in which the HTML page is displayed. The <title> element can only be contained inside a document’s <head> element.

**<link>** is used to link HTML documents to external resources like CSS files. It commonly uses:

* **href** attribute to specify the URL to the external resource
* **rel** attribute to specify the relationship of the linked document to the current document
* **type** attribute to define the type of content being linked

**<body>** represents the content of an HTML document. Content inside <body> tags are rendered on the web browsers. There can be only one <body> element in a document.

**<header>** describes the content at the top of the page <body>. It may include a logo, navigational links or a search bar.

**<nav>** encapsulates the page’s navigational links. It is often placed inside the <header> or <footer>.

**<main>** encapsulates the main content of a page between the header/navigation and the footer areas.

**<footer>** includes the page’s footer content at the bottom of the <body>.

**<section>** defines elements in a document, such as chapters, headings, or any other area of the document with the same theme.

**<article>** holds content that makes sense on its own such as articles, blogs, and comments. Generally developers will use **<section>** to define a theme for the webpage and use **<article>** to write independent content for that theme.

**<aside>** is used to mark additional information that can enhance another element but isn’t required to understand the main content. Usually, this information would be in a sidebar or a location where it doesn’t obstruct the main piece of content.

**<a>**anchor element is used to create hyperlinks in an HTML document. The **href** determines the location the anchor element points to.

* Relative file paths begin with ./ followed by a path to the local file. ./ tells the browser to look for the file path from the current folder.
* The anchor can create hyperlinks to different parts of the same HTML document using the **href** attribute with **#** followed by the id of the element to link to.
* The **target** attribute specifies where a hyperlink should be opened. A target value of **"\_blank"** will tell the browser to open the hyperlink in a new tab.

**<video>** embeds a media player for video playback. The **src** attribute will contain the URL to the video. Adding the **controls** attribute will display video controls in the media player. The content inside the opening and closing tag is shown as a fallback in browsers that don’t support the element.

**<audio>** allows us to implement audio into our website.

**<embed>** can be used to implement any type of media. These elements are universal in that they all use the **src** attribute to link the source of the content. <video> and <audio> requires a closing tag while <embed> is a self-closing tag.

**<img>** embeds images in documents. The **src** attribute contains the image URL and is mandatory. <img> is an *empty element* meaning it should not have a closing tag. An <img> element can have alternative text via the **alt** attribute. The alternative text will be displayed if an image fails to render. The text will be read aloud if screen reading software is used and helps support visually impaired users by providing a text descriptor for the image content on a webpage.

**<figure>** is used to encapsulate media such as an image, diagram, or code snippet. The **<figcaption>** element is used to describe the media encapsulated within the <figure> element. Developers will normally use <figcaption> within the <figure> element to group the media and description. This way, if a developer decides to change the position of the media, the description will follow along with it.

**<li>** creates list items inside:

* Ordered lists **<ol>**
* Unordered lists **<ul>**

**<em>** emphasizes text and browsers will usually *italicize* the emphasized text by default.

**<br>** line break element will create a line break in text. The line break element requires only an opening tag and must not have a closing tag.

**<span>** is an inline container for text and can be used to group text for styling purposes. Its use should be avoided if a more semantic element is available.

**<strong>** highlights important, serious, or urgent text and browsers will normally render this highlighted text in **bold** by default.

**<table>** has content that is used to represent a two-dimensional table made of rows and columns.

## HTML Attributes

Some HTML attributes can have multiple attribute values. Multiple attribute values are separated by a space between each attribute.

**id**attributes can be assigned to elements to differentiate between them. Unique **id** attributes begin with a letter and only contain letters (a-Z), digits (0-9), hyphens (-), underscores (\_), and periods (.).

**title** attribute can be provided to any HTML element and is used for additional context or advisory text for clickable elements (tooltip).

**Comments** are added between an opening <!-- and closing -->. They can span single or multiple lines.

**Inline CSS styles** can be directly added to HTML elements by using the **style** attribute in the element’s opening tag. Each style declaration is ended with a semicolon.

## HTML Tables

<table>

  <thead>

  <tr>

    <th scope="col">Company Name</th>

    <th scope="col">Number of Items to Ship</th>

    <th scope="col">Next Action</th>

  </tr>

  </thead>

<tbody>

  <tr>

    <td>Adam's Greenworks</td>

    <td rowspan="2">14</td>

    <td>Package Items</td>

  </tr>

  <tr>

  <td colspan="2">Davie's Burgers</td>

  <td>2</td>

  <td>Send Invoice</td>

</tr>

(…)

<tr>

  <td>Strike Fitness</td>

  <td>1</td>

  <td>Enter Order</td>

</tr>

</tbody>

<tfoot>

  <tr>

    <td>Total</td>

    <td>28</td>

  </tr>

</tfoot>

</table>

**<table>** has content that is used to represent a two-dimensional table made of rows and columns.

**<thead>** defines the headings of table columns encapsulated in table rows.

**<tbody>** is a semantic element that will contain all table data other than table heading and table footer content. If used, <tbody> will contain all table row <tr> elements, and indicates that <tr> elements make up the body of the table. <table> cannot have both <tbody> and <tr> as direct children.

**<tfoot>** uses table rows to give footer content or to summarize content at the end of a table.

**<tr>** is used to add rows to a table before adding table data and table headings.

**<th>** is used to add titles to rows and columns of a table and must be enclosed in a table row element, **<tr>**.

**<td>** can be nested inside a table row element, **<tr>**, to add a cell of data to a table.

**colspan** attribute on a table header **<th>** or table data **<td>** element indicates how many columns that particular cell should span within the table. The **colspan** value is set to 1 by default and will take any positive integer between 1 and 1000.

**rowspan** attribute on a table header or table data element indicates how many rows that particular cell should span within the table. The **rowspan** value is set to 1 by default and will take any positive integer up to 65534.

## HTML Forms

<form action="/action\_page.php">

  <input type="text" name="country\_code" pattern="[A-Za-z]{3}" title="Three letter country code">

  <input type="submit">

</form>

<form method="POST" action="http://server1">

<select name="rental-option">

  <option value="small">Small</option>

  <option value="family">Family Sedan</option>

  <option value="lux">Luxury</option>

</select>

<input list="ide">

  <datalist id="ide">

    <option value="Visual Studio Code" />

    <option value="Atom" />

    <option value="Sublime Text" />

  </datalist>

<input type="checkbox" name="breakfast" value="bacon">Bacon 🥓<br>

<input type="checkbox" name="breakfast" value="eggs">Eggs 🍳<br>

<input type="checkbox" name="breakfast" value="pancakes">Pancakes 🥞<br>

<input name="delivery\_option" type="radio" value="pickup" />

<input name="delivery\_option" type="radio" value="delivery" />

<input type="range" name="movie-rating" min="0"  max="10"  step="0.1"  >

<textarea rows="10" cols="30" name="comment"></textarea>

<input type="text" name="username" minlength="8" maxlength="14" />

<label for="password ">Password:</label>

<input type="password" " id="password" name="password" required />

<input type="number" min="1" max="20"  />

<input type="submit" value="Submit">

### Submitting a Form

The **action** attribute tells the form to send the information. A URL is assigned that determines the recipient of the information. The **method** attribute tells the form what to do with that information once it’s sent. An HTTP verb is assigned to the **method** attribute that determines the **action** to be performed.

### Form Elements

**<form>** is used to collect and send information to an external source. <form> can contain various <input> elements. When a user submits the form, information in these input elements is passed to the source which is named in the action attribute of the form.

**<input>** is used to render a variety of input fields on a webpage including text fields, checkboxes, buttons, etc. <input> elements have a **type** attribute that determines how it gets rendered to a page. The value of the <input>‘s name and value attributes of the element are sent as a key-value pair when the form is submitted.

**<textarea>** is used when creating a text box for multi-line input. The element supports the **rows** and **cols** attributes which determine the height and width of the element. When rendered by the browser, textarea fields can be stretched/shrunk in size by the user, but the rows and cols attributes determine the initial size. Unlike the <input> element, the <textarea> element has both opening and closing tags. The value of the element is the content in between these tags (much like a <p> element).

**<select>** can be used to create a dropdown list. A list of choices for the dropdown list can be created using one or more **<option>** elements. By default, only one <option> can be selected at a time. The value of the selected <select>’s name and the <option>’ s value attribute are sent as a key-value pair when the form is submitted.

**<datalist>** is used to store a list of **<option>**s and is ideal when providing users a list of pre-defined options, but to also allow them to write alternative inputs as well. The <input>’s list value must match the value of the **id** of the <datalist>. As the user starts typing, the dropdown list will be updated to show elements that best match what has been typed into the input field. The actual list items are specified as multiple option elements nested inside the <datalist>.

**<label>** provides identification for a specific <input> based on matching values of the <input>‘s **id** attribute and the <label> **for** attribute.

### 

### <Input> Types

**type=”number”** field allows the user to enter only numbers and a few special characters inside the field.

**type="text"** renders a single row input field that users can type text inside.

**type="checkbox"** renders a single checkbox item. A group of checkboxes related to the same topic should all use the same name attribute. Multiple checkboxes can be selected for the same topic.

**type="range"** creates a range slider that will act as a selector between a minimum and a maximum value. The slider can be adjusted to move in different steps or increments using the **step** attribute.

**type="radio"** renders a single radio button. Multiple radio buttons of a related topic are given the same **name** attribute value. Only a single option can be chosen from a group of radio buttons.

**type="password"** allows the user to type censored text inside the field.

**type="submit"** renders a submit button and, by default, will submit the <form> and execute its action. The text of a submit button is set to Submit but can be changed by modifying the **value** attribute.

### 

### <input> Attributes

**name** will become the key and the value of the input will become the value the form submits corresponding to the key. The **ID** and the **name** of the input may be the same, but the **value** will only be submitted if the **name** attribute is specified.

**required** specifies that the field must include a value. The attribute can be written as **required="true"** or simply **required**.

**max** specifies the maximum value for the input field.

**min** attribute specifies the minimum value.

**maxlength** specifies the maximum number of characters that can be entered into the field.

**minlength** specifies the minimum number of characters.

**pattern** uses a regular expression to match against (or validate) the value of the <input>.

To specify patterns for the computer to recognize, we use a special language called regular expressions — also known as **regex** or **regexp**. A regular expression is a sequence of characters representing a pattern. We can use that pattern to match a string, match parts of a string, confirm that data is formatted acceptably, or even replace parts of strings with different characters.

### Form Example

    <h1>Welcome To This Form</h1>

    <form action="" method="POST">

        <label for="text">You can enter text here:</label>

        <input type="text" name="text">

        <hr>

        <label for="num">You can enter a number here:</label>

        <input type="number" name="num">

        <hr>

        <label for="slider">You can slide this:</label>

        <br>

        <span>Left</span>

        <input type="range" name="slider" value="3" min="1" max="5">

        <span>Right</span>

        <hr>

        <label for="boxes">You can check these:</label>

        <input type="checkbox" name="boxes" value="first">

        <label for="first">First</label>

        <input type="checkbox" name="boxes" value="second">

        <label for="second">Second</label>

        <input type="checkbox" name="boxes" value="third">

        <label for="third">Third</label>

        <hr>

        <label for="radio">You can select one of these:</label>

        <input type="radio" name="radio" value="true">

        <label for="true">TRUE</label>

        <input type="radio" name="radio" value="false">

        <label for="false">FALSE</label>

        <hr>

        <label for="dropdown">You can select one of these</label>

        <select name="dropdown">

            <option value="first">First</option>

            <option value="second">Second</option>

            <option value="third">Third</option>

        </select>

        <hr>

        <input type="submit" value="Submit to Reset">

    </form>

## JavaScript Code in HTML

<script>

    function blooming() {

        let image = document.getElementById('myImage');

        if (image.src.match("normal")) {

            image.src = "flower.png";

        } else {

            image.src = "normal.png";

        }

    }

</script>

<script>

console.log("Hello world!");

</script>

<head>

    <link rel="stylesheet" href="style.css">

    <script src="./script.js"></script>

</head>

<body onclick="newStyle();">

<script src="example.js" defer></script>

<script src="example.js" async></script>

<script type="module" src="./secret-messages.js"> </script>

**<script>** allows HTML files to load and execute JavaScript. The JavaScript can either go embedded inside of the **<script>** tag or the script tag can reference an external file.

Browsers come equipped with HTML parsers that help them render the elements accordingly. By default, elements are parsed in the order they appear. If one script depends on another script, they should be placed in that very order inside the HTML file.

**defer** executes the script after the HTML file is completely parsed. When a script contains functionality that requires interaction with the DOM, the **defer** attribute is the way to go.

**async** loads and executes the script asynchronously with the rest of the webpage. Similar to the **defer** attribute, the parser will continue parsing the rest of the HTML as the script is downloaded in the background. However, the script will execute immediately after it has been downloaded. If it does not matter exactly at which point the script file is executed, asynchronous loading is the most suitable option as it optimizes web page load time.

**type='module'** is used to use a JavaScript module.

## Accessibility

Some examples include (but are not limited to):

* Screen readers that parse a website for a user with visual impairments
* Videos on websites are closed-captioned for individuals with hearing impairments
* Images include “alt text” for individuals with visual impairments
* Websites must be navigable by keyboard for users who may not be able to operate a mouse (i.e., navigating using the “Tab” on a keyboard)

The keys to building a more visually inclusive Web are:

* Using contrast so our text and colors provide better visual cues
* Using correct font size so our content is legible
* Creating user interfaces that are enhanced by, but not dependent on, color

Using ARIA (Accessible Rich Internet Applications) roles and properties, alt attributes, and semantic elements in your HTML is a simple way to make your website accessible to visually-impaired Internet users.

* Using semantic HTML elements whenever possible (such as <header> instead of <div id="header">) will allow screen reader users to navigate your website more efficiently.
* The **role** attribute is used to communicate information about the role of specific elements.
* **role="presentation"** allows a screen reader to skip markup elements that don’t directly contain useful information.
* **aria-label** and other ARIA properties can be used to help users perceive information that is communicated visually but not through text.
* The **alt** attribute should be added to every image element (and all other elements that support it) instead of aria-label. When used, its value should be a useful description of the image.

<https://www.w3.org/TR/html-aria/#allowed-aria-roles-states-and-properties>

Screen reader options:

* <https://www.nvaccess.org/>
* <https://chrome.google.com/webstore/detail/screen-reader/kgejglhpjiefppelpmljglcjbhoiplfn?hl=en>

<https://www.a11yproject.com/checklist/>

<https://developer.mozilla.org/en-US/docs/Learn/Accessibility/CSS_and_JavaScript>

## HTML Example – Portfolio Project

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <link rel="stylesheet" href="./style.css" type="text/css" />

    <script src="./script.js" defer></script>

    <title>Portfolio</title>

</head>

<body>

    <nav>

        <ul>

            <li><a href="#about-me">About Me</a></li>

            <li><a href="#projects">Projects</a></li>

            <li><a href="#contact">Contact</a></li>

        </ul>

    </nav>

    <header>

        <h1>Portfolio</h1>

    </header>

    <content>

        <section id="about-me">

            <h2>About Me</h2>

            <p>My name is Annie McMahon. I am originally from Canada (Quebec), and currently live in NJ.</p>

            <p>I recently enrolled in Codecademy's Full-Stack Engineer program

                to learn software development and possibly make a career change.

                This is a place to display all my projects.</p>

            <p>Please contact me if you would like to give me feedback on any of those projects.

                I'm open to learning!</p>

        </section>

        <section>

            <h2>Projects</h2>

            <h4>Click on each image to see the project details</h4>

            <div id="projects">

                <div class="project-box">

                    <div class="front">

                        <h3>CSS Cheatsheet</h3>

                        <img src="./Images/cheatsheet.png" />

                    </div>

                    <div class="back">

                        <h3>CSS Cheatsheet</h3>

                        <p>Technology used: HTML, CSS</p>

                        <p>Summary: I created an HTML table to display all the different ways

                            to define colors in CSS.</p>

                        <p>GitHub link: (not yet)</p>

                        <p>Web page link: (not yet)</p>

                    </div>

                </div>

                <div class="project-box">

                    <div class="front">

                        <h3>Style Guide</h3>

                        <img src="./Images/styleguide.png" />

                    </div>

                    <div class="back">

                        <h3>Style Guide</h3>

                        <p>Technology used: HTML, CSS</p>

                        <p>Summary: I created a style guide (colors, font families, and text styles)

                            for one of my websites using flexboxes.</p>

                        <p>GitHub link: (not yet)</p>

                        <p>Web page link: (not yet)</p>

                    </div>

                </div>

                <div class="project-box">

                    <div class="front">

                        <h3>Company Website</h3>

                        <img src="./Images/company.png" />

                    </div>

                    <div class="back">

                        <h3>Company Website</h3>

                        <p>Technology used: HTML, CSS</p>

                        <p>Summary: I built a fictitious company's website using a flexbox layout.</p>

                        <p>GitHub link: (not yet)</p>

                        <p>Web page link: (not yet)</p>

                    </div>

                </div>

                <div class="project-box">

                    <div class="front">

                        <h3>Club Website</h3>

                        <img src="./Images/club.png" />

                    </div>

                    <div class="back">

                        <h3>Club Website</h3>

                        <p>Technology used: HTML, CSS</p>

                        <p>Summary: I created a fictitious club website using responsive design tools.</p>

                        <p>GitHub link: (not yet)</p>

                        <p>Web page link: (not yet)</p>

                    </div>

                </div>

                <div class="project-box">

                    <div class="front">

                        <h3>Mad Lib Generator</h3>

                        <img src="./Images/mad-lib.png" />

                    </div>

                    <div class="back">

                        <h3>Mad Lib Generator</h3>

                        <p>Technology used: HTML, CSS, JavaScript</p>

                        <p>Summary: The purpose of this project was to generate random

                            messages using JavaScript coding.</p>

                        <p>GitHub link: <a href="https://github.com/AnnieMcMahon/mixed\_messages"

                                target="\_blank">mixed\_messages</a></p>

                        <p>Web page link: <a href="https://anniemcmahon.github.io/mixed\_messages/"

                                target="\_blank">mixed\_messages</a></p>

                    </div>

                </div>

            </div>

        </section>

    </content>

    <footer>

        <h2 id="contact">Contact Me</h2>

        <p>E-mail: anniemcmahon20@gmail.com</p>

        <p>GitHub: <a href="https://github.com/AnnieMcMahon" target="\_blank">https://github.com/AnnieMcMahon</a></p>

    </footer>

</body> </html>