

Learn HTML Basics for Beginners in Just 15 Minutes

 freecodecamp.org/news/html-basics-for-beginners

January 12, 2021



January 12, 2021 / [#Web Development](#)



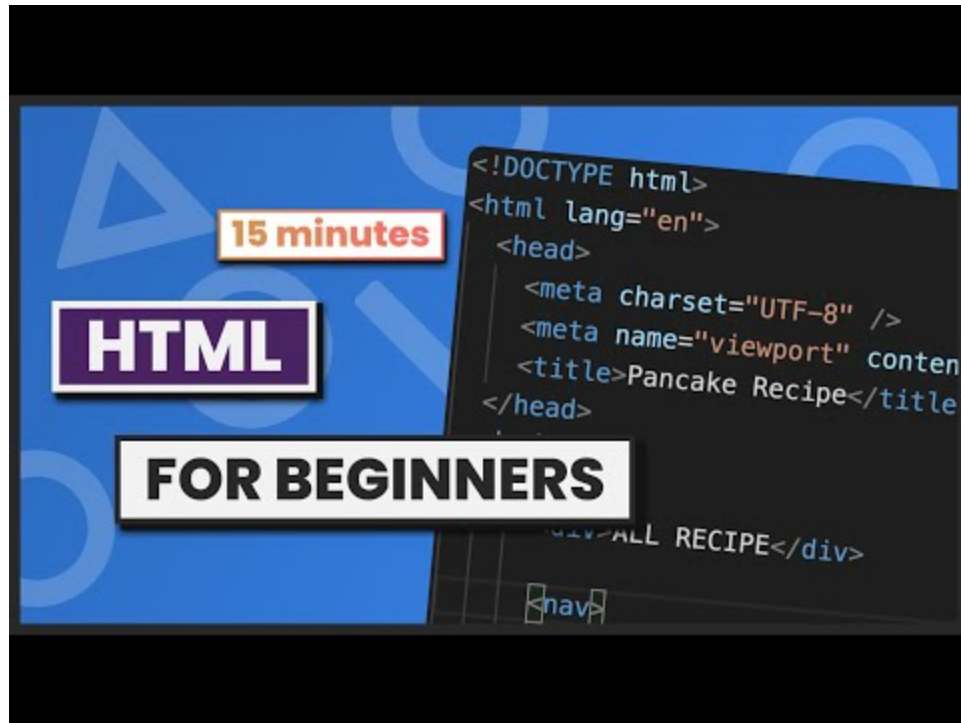
Thu Nghiem

 [Learn HTML Basics for Beginners in Just 15 Minutes](#)

If you want to build a website, the first language that you need to learn is HTML.

In this article, we are going to go through the basics of HTML. At the end, we are going to build a basic website using only HTML.

Here's a video you can watch if you want to supplement this article:



Watch Video At: <https://youtu.be/pMJ0NI3OkYA>

If you prefer video, you can watch it here

What Is HTML?

HTML, which stands for Hypertext Markup Language, is a pretty simple language. It consists of different elements which we use to structure a web page.

HTML – HyperText Markup Language

```
<body>
<h1>Heading</h1>

<p>This is a paragraph.</p>

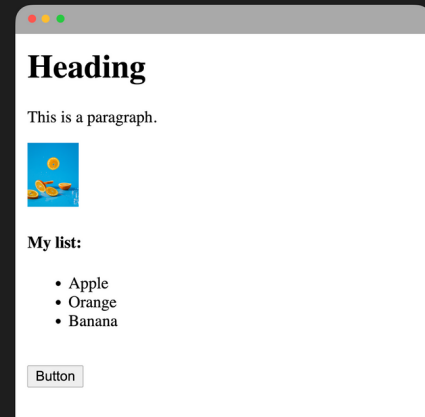


<div class="my-list">
  <h4>My list:</h4>

  <ul>
    <li>Apple</li>
    <li>Orange</li>
    <li>Banana</li>
  </ul>
</div>

<br>

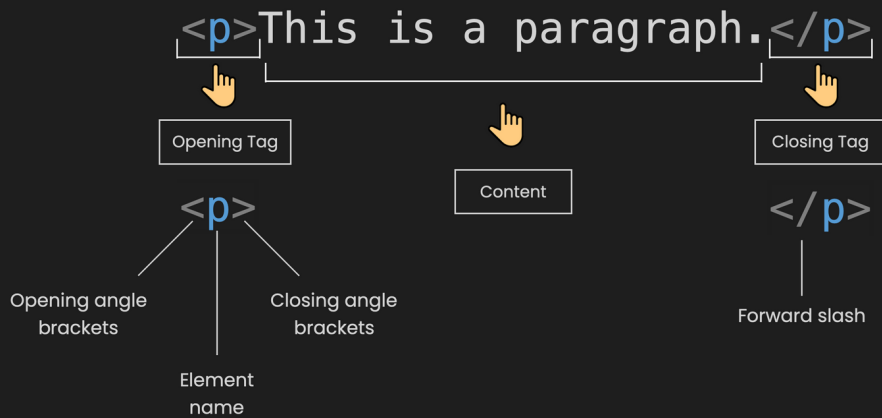
<button onclick="alert('Submit')">Button</button>
</body>
```



What is HTML?

What Are HTML Elements?

HTML Element



HTML elements

The element usually starts with an opening tag, which consists of the name of the element. It's wrapped in opening and closing angle brackets. The opening tag indicates where the element begins.

Similar to the opening tag, the closing tag is also wrapped in opening and closing angle brackets. But it also includes a forward slash before the element's name.

Everything inside the opening and closing tags is the content.

But not all elements follow this pattern. We call those that don't empty elements. They only consist of a single tag or an opening tag that cannot have any content. These elements are typically used to insert or embed something in the document.

For example, the `` element is used to embed an image file, or the `<input>` element is used to insert an input onto the page.

```

```

In the example above, the `` element only consists of one tag that does not have any content. This element is used to insert an image file from [Unsplash](https://images.unsplash.com/photo-1610447847416-40bac442fbe6) in the document.

How to Nest HTML Elements

```

<div class="my-list">
  <h4>My list:</h4>

  <ul>
    <li>Apple</li>
    <li>Orange</li>
    <li>Banana</li>
  </ul>
</div>

```

Elements can be placed inside other elements. This is called Nesting. In the example above, inside the `<div>` element we have an `<h4>` element and an `` or unordered list element. And Similarly inside the `` element, there are 3 `` or list item elements.

Basic nesting is quite straight-forward to understand. But when the page gets larger, nesting can become complicated.

Therefore, before working with HTML, think about the layout structure you would like to have. You can draw it out on a piece of paper or in your mind. It will help a lot.

Nesting

HTML elements can be nested

```

<div class="my-list">
  <h4>My list:</h4>

  <ul>
    <li>Apple</li>
    <li>Orange</li>
    <li>Banana</li>
  </ul>
</div>

```

My list:

- Apple
- Orange
- Banana

Tip: Draw the layout structure with rectangle boxes on a piece of paper or on your mind

What are HTML Attributes?

Elements also have attributes, which contain extra information about the element that will not appear in the content.

```

```


In the example above, the `` element has 2 attributes: `src` or source to specify the path of the image, and `width` to specify the width of the image in pixels.

Attributes


Attributes contain extra information about the element that won't appear in the content.

```

```



Attribute



Attribute

- Attributes should have spaces between attributes and element name
- Attributes are added in the opening tag
- Elements can have many attributes
- Attributes usually have a name and value: `name="value"`

With this example, you can see the following characteristics of attributes:

- There is a space between attributes and the element name
- Attributes are added in the opening tag
- Elements can have many attributes
- Attributes usually have a name and a value: `name="value"`

But not every attribute has the same pattern. Some can exist without values, and we call them Boolean Attributes.

```
<button onclick="alert('Submit')" disabled>Button</button>
```

In this example, if we want to disable the button, all we have to do is pass a `disabled` attribute without any values. This means that the presence of the attribute represents the true value, otherwise, the absence represents the false value.

Common HTML elements

There are in total more than 100 elements. But 90% of the time you will only use around 20 of the most common. I have put them into 5 groups:

Section elements

6/15

`<div>`, ``, `<header>`, `<footer>`, `<nav>`, `<main>`, `<section>`

These elements are used to organize the content into different sections. They are usually self-explanatory, for example, `<header>` usually represents a group of the introduction and navigation section, `<nav>` represents the section that contains navigation links, and so on.

Text content

`<h1>` to `<h6>`, `<p>`, `<div>`, ``, ``, ``, ``

These elements are used to organize content or text blocks. They are important to accessibility and SEO. They tell the browser the purpose or structure of the content.

Forms

`<form>`, `<input>`, `<button>`, `<label>`, `<textarea>`

These elements can be used together to create forms that users can fill out and submit. Forms might be the trickiest part of HTML.

Images and Links

``, `<a>`

These elements are used to insert an image or create a hyperlink.

Others

`
`, `<hr>`

These elements are used to add a break to the webpage.

You can find all the elements on developer.mozilla.org. But for beginners, you just need to know the most common ones.

Block-level vs inline HTML elements

By default, an element can be either block-level or an inline element.

Block-level elements are the elements that always start on a new line and take up the full width available.

Inline elements are the elements that do not start on a new line and it only take up as much width as necessary.

Block-level vs Inline elements

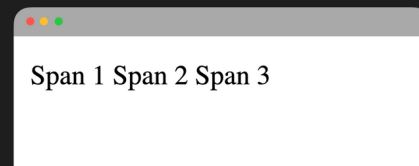
Block-level element **always starts on a new line** and takes up the full width available

```
<div>Div 1</div>
<div>Div 2</div>
<div>Div 3</div>
```



Inline element **does not start on a new line** and it only takes up as much width as necessary

```
<span>Span 1</span>
<span>Span 2</span>
<span>Span 3</span>
```



Block level vs. Inline HTML elements

Two elements that represent block-level and inline elements, respectively, are `<div>` and ``. In this example, you can see that the `<div>` elements takes 3 lines, whereas the `` element only takes up 1 line.

But the question is: how do we know which ones are block-level elements and which ones are inline elements? Well, unfortunately you need to remember them. The easiest way is to remember which are inline elements – and the rest are block elements.

If we look back at the most common HTML elements, inline elements include: ``, `<input>`, `<button>`, `<label>`, `<textarea>`, ``, `<a>`, `
`.

How to comment in HTML

```
<p>This is a paragraph.</p>
```

```
<!-- <p>I am not showing.</p> -->
```

The purpose of comments is to include notes in the code to explain your logic or simply to organize your code.

HTML comments are wrapped in the special markers: `<!--` and `-->` and they are ignored in the browser.

How to use HTML entities

What if you want to show the text: `the <p> tag defines a paragraph.` , but the browser interprets `<p>` as an opening tag for a new element? In this case, we can use HTML entities like in the following example:

```
<p>the <p> tag defines a paragraph.</p>
```

```
<p>the &lt;p> define a paragraph.</p>
```

How to use emoji in HTML

In the modern web, we can display emoji in HTML pretty easily, like this:

```
<p> Grinning Face.</p>
```

```
<p> Birthday</p>
```

Common beginner mistakes in HTML

1. Tags/Element names

Tags/Element names are cAse-inSensitive. This means that they can be written in lowercase or uppercase, but it is recommended that you write everything in lowercase: `<button>` not `<ButTon>` .

2. Closing tag

Failing to include a closing tag is a common beginner error. Therefore, whenever you create an opening tag, immediately put in a closing tag.

3. Nesting

This is wrong:

```
<div>Div 1 <span> Span 2 </div></span>
```

The tags have to open and close in a way that they are inside or outside one another.

4. Single quotes and Double quotes

This is wrong:

```

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>

</body>
</html>
```

This is the minimal code that an HTML document should have to make up a website. And here we have:

1. `<!DOCTYPE html>` : First we have Doctype. For some weird historical reason in HTML we have to include the doctype for everything to work correctly.
2. `<html lang="en"></html>` : The `<html>` element wraps all the content on the page, also known as the root element. And we should always include the `lang` attribute to declare the language of the page.
3. `<head></head>` : The `<head>` element is a container for everything you want to include, but not content that you show to your users.
4. `<meta charset="UTF-8" />` : The first meta element is used to set the character set to be UTF-8, which includes most characters from written languages.
5. `<meta name="viewport" content="width=device-width, initial-scale=1.0" />` : The second meta element specifies the browser viewport. This setting is for a mobile-optimized site.
6. `<title>Document</title>` : This is the `<title>` element. It sets the title of the page.
7. `<body></body>` : The `<body>` element contains all the content on the page.

How to build a pancake recipe page

Alright, now that we have the starter code, let's build a pancake recipe page. We are going to use the content from this [AllRecipes Page](#).

First, let's give the `<title>` element content of the pancakes recipe. You will see the text on the web page tab change. In the `<body>` element, let's create 3 elements: `<header>`, `<main>` and `<footer>` representing 3 sections.

1. Build the header section

In the header, we want to have the logo and the navigation. Therefore, let's create a `div` with the content `ALL RECIPE` for the logo.

For the navigation, let's use the `<nav>` element. Within the `<nav>` element, we can use `` to create an unordered list. We want to have 3 `` elements for 3 links: Ingredients, Steps, and Subscribe. The header code looks like this:

```
...
<header>
  <div>ALL RECIPE</div>
  <nav>
    <ul>
      <li><a href="#ingredients">Ingredients</a></li>
      <li><a href="#steps">Steps</a></li>
      <li><a href="#subscribe">Subscribe</a></li>
    </ul>
  </nav>
</header>
...
```

2. Build the Main Section

In the main section, first, we want to have a title and an image. We can use `h1` for the title and `` for the image (we can use an image from [Unsplash](#) for free):

```
...
<main>
  <h1>Good Old Fashioned Pancakes</h1>
  
</main>
...
```

Next, we want to list all the ingredients. We can use `` to create an ordered list and `<input type="checkbox" />` to create a checkbox.

But before that, we can use `<h2>` to start a new content block. We also want to add the `id` attribute for `<h2>` so that the link in the navigation knows where to go:

```

...
<main>
...
  <h2 id="ingredients">Ingredients</h2>
  <ol>
    <li><input type="checkbox" /> 1½ cups all-purpose flour</li>
    <li><input type="checkbox" /> 3½ teaspoons baking powder</li>
    <li><input type="checkbox" /> 1 teaspoon salt</li>
    <li><input type="checkbox" /> 1 tablespoon white sugar</li>
    <li><input type="checkbox" /> 1¼ cups milk</li>
    <li><input type="checkbox" /> 1 egg</li>
  </ol>
</main>
...

```

After the ingredients, we want to list all the steps. We can use `<h4>` for the step heading and `<p>` for the step content:

```

...
<main>
...
  <h2 id="steps">Steps</h2>

  <h4>Step 1</h4>
  <p>
    In a large bowl, sift together the flour, baking powder, salt and sugar.
    Make a well in the center and pour in the milk, egg and melted butter;
    mix until smooth.
  </p>

  <h4>Step 2</h4>
  <p>
    Heat a lightly oiled griddle or frying pan over medium-high heat. Pour
    or scoop the batter onto the griddle, using approximately 1/4 cup for
    each pancake. Brown on both sides and serve hot.
  </p>
</main>
...

```

Alright, now that we are done with the main section, let's move on to the footer section.

3. Build the Footer Section

In the footer, we want to have a subscribe form and copyright text.

For the subscribe form, we can use the `<form>` element. Inside it, we can have an `<input type="text">` for text input and a `<button>` for the submit button.

For the copyright text, we can simply use a `<div>`. Notice here, we can use the HTML entity `©` for the copyright symbol.

We can add `
` to add some space between the subscribe form and the copyright text:

```
...
<footer>
  <h6 id="subscribe">Subscribe</h6>
  <form onsubmit="alert('Subscribed')">
    <input type="text" placeholder="Enter Email Address" />
    <button>Submit</button>
  </form>
  <br />
  <div>&copy; dakota kelly at Allrecipe.com</div>
</footer>
...
```

Alright now we are done! Here is the full code for reference:

```

<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Pancake Recipe</title>
  </head>
  <body>
    <header>
      <div>ALL RECIPE</div>
      <nav>
        <ul>
          <li><a href="#ingredients">Ingredients</a></li>
          <li><a href="#steps">Steps</a></li>
          <li><a href="#subscribe">Subscribe</a></li>
        </ul>
      </nav>
    </header>
    <main>
      <h1>Good Old Fashioned Pancakes</h1>
      
      <h2 id="ingredients">Ingredients</h2>
      <ol>
        <li><input type="checkbox" /> 1½ cups all-purpose flour</li>
        <li><input type="checkbox" /> 3½ teaspoons baking powder</li>
        <li><input type="checkbox" /> 1 teaspoon salt</li>
        <li><input type="checkbox" /> 1 tablespoon white sugar</li>
        <li><input type="checkbox" /> 1¼ cups milk</li>
        <li><input type="checkbox" /> 1 egg</li>
      </ol>
      <h2 id="steps">Steps</h2>
      <h4>Step 1</h4>
      <p>
        In a large bowl, sift together the flour, baking powder, salt and sugar.
        Make a well in the center and pour in the milk, egg and melted butter;
        mix until smooth.
      </p>
      <h4>Step 2</h4>
      <p>
        Heat a lightly oiled griddle or frying pan over medium-high heat. Pour
        or scoop the batter onto the griddle, using approximately 1/4 cup for
        each pancake. Brown on both sides and serve hot.
      </p>
    </main>
    <hr />
    <footer>
      <h6 id="subscribe">Subscribe</h6>
      <form onsubmit="alert('Subscribed')">
        <input type="text" placeholder="Enter Email Address" />
      </form>
    </footer>
  </body>
</html>

```

```
        <button>Submit</button>
    </form>
    <br />
    <div>&copy; dakota kelly at Allrecipe.com</div>
</footer>
</body>
</html>
```

Conclusion

You can build a simple website with just HTML. But to be able to build beautiful and functional websites, you need to study CSS and JavaScript.

You can follow me on social media or Youtube for future updates on these topics. But meanwhile, you can check out the [freeCodeCamp Curriculum](#) to practice HTML by solving small tasks.

Otherwise, stay happy coding and see you in future posts .

_____ About me _____



Thu Nghiem

Creator of devchallenges.io