

## Basic Structure

---

### <!DOCTYPE>

The DTD (Document Type Declaration), this tells your browser which [version of HTML](#) you're using. Make sure you use the right DTD, or your page may display incorrectly.

### <html>...</html>

Standard opening and closing tags for any HTML page. Enclose everything else in these. **Container tag.**

### <!-- ... -->

A comment — whatever you put here will be skipped over by the browser.

### <head>...</head>

Starts the header part of your document. Everything between these is mainly used to help your browser and search engines classify your page. Using this is optional, but recommended. **Container tag.**

#### <title>...</title>

Whatever is between these tags will appear in the blue bar at the top of the screen.

#### <meta>

A group of tags that give page and creator information specifically to the search engines.

#### <base>

Changes the default link target or relative link URL, useful if the page is read on another server.

#### <link>

Allows you to associate [stylesheets](#) and a [favorites icon](#) to your page.

### <body>...</body>

Everything visible on your page goes between these tags. Everything. **Container tag.**

## Links

---

### <a>...</a>

Makes the enclosed text or image a hyperlink to another file.

## Lists

---

### <ol>...</ol>

Creates an ordered list, where each item is numbered in order. **Container Tag.**

### <ul>...</ul>

Creates an unordered list, with each item bulleted. **Container Tag.**

#### <li>

Each list item begins with an **li**, and they are all placed in either an **ol** or **ul**.

### <dl>...</dl>

Creates a definition list.

#### <dt>

Creates a definition term.

#### <dd>

Creates a definition, which appears below its parent term and indented from the left.

## Multimedia

---

### <img>

Places an image on your page

### <embed>

Adds a multimedia element directly into your page, allowing your browser to play it with a plug-in.

### <script>...</script>

Adds a script, usually a JavaScript into your page.

#### <noscript>...</noscript>

Enclose anything you want displayed by browsers that do not support scripts.

## Tables

---

**<table>...</table>**

Places a table on your page. **Container Tag.**

**<caption>...</caption>**

contains the caption of the table, the title of sorts. It will appear across the top unless specified otherwise. This tag should not be contained in a [tr](#) or [td](#).

**<tr>...</tr>**

starts a new table row. Cells go inside this. Attributes are the same as [td](#)'s.

**<td>...</td>**

encloses a table cell. Content goes in these.

**<th>...</th>**

same as table cells, but with all contents bold and aligned to the centre.

**<thead>...</thead>**

Defines the header part of a large table. Wrap the tags around the rows/cells you wish to define as the header.

**<tbody>...</tbody>**

Defines the main body of a complex table.

**<tfoot>...</tfoot>**

Wrap this around the footer part of your table.

**<colgroup>**

Allows you to set attributes for the entire column. Each column has to be defined sequentially.

## Frames

---

**<frameset>...</frameset>**

Starts a new frame layout. When constructing a frame page, no [body](#) is used. **Container Tag.**

**<frame>**

Defines a single frame within a frameset.

**<noframes>...</noframes>**

If a visitor has an old browser which doesn't support frames you can leave a message or some content between these tags.

**<iframe>...</iframe>**

Places an inline or 'floating' frame. This can be placed anywhere on a normal page, i.e. it doesn't need to be part of a [frameset](#).

## Forms

---

**<form>...</form>**

Begins a form area. Add in any form elements you want to use between these tags. **Container Tag.**

**<input>**

Allows you to add various user input fields, like text-boxes, checkboxes, radio buttons, submit and reset buttons, depending on how you set the [type](#) attribute.

**<textarea>...</textarea>**

Adds a multi-lined text area, suitable for input of a larger amount of information than the single-line text box. Any text added between the tags is placed in the area when the page loads.

**<select>...</select>**

Sets up an empty drop-down selection box. You can add choices with the **<option>...</option>** tag.

**<fieldset>...</fieldset>**

Allows you to group form elements together into logical arrangements.

**<legend>...</legend>**

You can title your [fieldsets](#) with this tag.

**<label>...</label>**

You can make the explanatory text next to a form element into part of the clickable area with this tag, which makes selecting elements much easier.

## Text Formatting

---

`<address>...</address>`

Encloses the signature and address of the author, displayed in italics.

`<b>...</b>`

Makes the enclosed text into bold text.

`<big>...</big>`

Makes the enclosed text one size bigger.

`<blockquote>...</blockquote>`

Indents the text in from both sides.

`<br>`

Stops the current line and goes on to the next.

`<center>...</center>`

Aligns the surrounded objects (anything from text or images to forms etc.) to the center.

`<cite>...</cite>`

Includes a citation, and is usually rendered as italics.

`<code>...</code>`

If you are including either computer or HTML code into your documents wrap these around it. It is rendered in small text.

`<dfn>...</dfn>`

A definition. As usual with these things, rendered in italics.

`<div>...</div>`

Surround distinct sections of your page in divs, primarily to align them, but many other attributes are supported and divs can be used to set up layers too.

`<em>...</em>`

Emphasises the surrounded text, changing it to italics.

`<font>...</font>`

Sets the font properties for the selected text.

`<h>...</h>`

Sets the text as a heading, with values of `h1` (the biggest) to `h6` (the smallest).

`<hr>`

Inserts a grey horizontal line across the page.

`<i>...</i>`

The default tag for italics.

`<kbd>...</kbd>`

Implies that the text should be entered on the keyboard. It is rendered mono-spaced and small.

`<nobr>...</nobr>`

Tells the enclosed text not to wrap at the edge of the screen, but continue on as long as it needs.

`<wbr>`

If you need a line break in a block of text you have set in `nobr`, add this.

`<p>...</p>`

Makes the enclosed text a paragraph, with lines skipped at the top and bottom.

`<pre>...</pre>`

Displays text in fixed-width font and retains the formatting of the original text (i.e. spaces and line-breaks).

`<q>...</q>`

Used to mark up short, inline quotations. Some browsers will add quotation marks around the text.

`<s>...</s>` (or `<strike>...</strike>`)

Displays the text with a strike-through.

`<samp>...</samp>`

Indicates sample output from a form or program. Text is rendered in small font.

`<small>...</small>`

Makes the enclosed text one size smaller.

`<strong>...</strong>`

Creates emphasis for the selected text, rendered in bold.

`<sub>...</sub>`

Renders the text in subscript, which is words under the normal text. For example: `this`

`<sup>...</sup>`

The sister tag to `<sub>` above, this renders text in superscript. Like this

`<tt>...</tt>`

Renders text in fixed-width, mono-spaced font; like an old typewriter.

`<u>...</u>`

Displays the surrounded text underlined.

# Attributes

## CLASS

HTML elements can have one or more classes, separated by spaces. You can style elements using CSS by selecting them with their classes.

### Example

```
<div class="big-box yellow-box">This is a big yellow box.</div>
```

## ID

An HTML element can have an id attribute to identify it. id elements should always be unique to that single element, and each element should never have more than one id.

### Example

```
<div id="my-box">This is my box! Put your text in some other  
box.</div>
```

## HREF

Links tell the browser where to go using an href attribute, which stores a URL.

### Example

```
<a href="http://google.com">Google it!</a>
```

## Basic Formatting

You can easily format text to be bold, italic, or underlined using simple formatting tags.

### Example

```
This text is <b>bold</b>, <i>italicized</i>, and  
<u>underlined</u>.
```

## Body

The body is the container for all of a page's content. Comes after the `<head>` tag, within the overall `<html>` tag.

### Example

```
<html>   <head>       <title>An example of the body tag</title>   </head>  
<body>   This is inside the body!   </body> </html>
```

### Read more

- <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/body>

## USAGE

Almost all content belongs inside the body tag. The main exceptions are script and style tags, as well as the page title tag. As you can see in this example, there is a heading, an image, and a link all inside the body tag. The head tag contains only external files and the page title.

### Example

```
<html>   <head>       <title>My homepage</title>       <link rel="stylesheet"  
type="text/css" href="homepage.css" />       <script type="text/javascript"  
src="homepage.js"></script>   </head>   <body>       <h1>Hello, this is a  
picture of my cat!</h1>              <a  
href="mailto:cat@codecademy.com">Email my cat</a>   </body> </html>
```

# Children

An element that is an immediate descendent of another element or nested within another element is called a child. These become useful when using CSS child selectors and psuedo-elements.

## Example

```
<ul id="parent">  <li id="child">I'm a child of parent!</li> </ul>
```

# Comments

HTML comments are sometimes used in code to explain parts of the markup. They are similar to comments in other languages. Users do not see comments in their browser.

## Syntax

```
<!-- This is an HTML comment! -->
```

# Div

A block level container (or 'division' of the web page) for content with no semantic meaning.

## Syntax

```
<div>This is a div element.</div>
```

# Head

Tag that surrounds important content that is invisible to the user, but is important to the browser. Elements within this tag contain metadata about the page and links to stylesheets, scripts, etc.

```
<html>    <head>    </head>    <body>    </body> </html>
```

# Headings

Heading elements like `<h1>`, `<h2>`, `<h3>`, ... allow you to use six levels of document headings, ranging from largest to smallest, breaking up the document into logical sections. For example, the word 'Headings' above is wrapped in a `<h2>` tag.

## Syntax

```
<h1> This is a header! </h1>
```

# Horizontal rules

This tag creates a black line one pixel thick that runs the all the way across its container. It can be styled to look differently with CSS.

## Example

```
This text is divided <hr> ...from this text!
```

## Read more

- <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/hr>

# HTML

## WHAT IS HTML?

HTML stands for Hyper Text Markup Language. It is the language used to create all websites.

## Read more

- [http://www.w3.org/wiki/HTML/Training/What\\_is\\_HTML](http://www.w3.org/wiki/HTML/Training/What_is_HTML)

**<HTML> TAG**

All HTML files live within an over-arching html tag. This is the basic tag that defines an html document.

### Syntax

```
<html> The rest of your web page goes in here! </html>
```

### Read more

- <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/html>

## Hyperlinks

Hyperlinks (or just links) take the user to another webpage when they click on it. The most common attribute used with links is href, which tells the browser where the link goes.

### Syntax

```
<a href="url this link goes to">Link text</a>
```

### Example

```
The following text is <a href="http://google.com">goes to Google</a>.
```

## Images

The img tag embeds an image into your HTML. Always found with the 'src' attribute, which tells the browser where to find the image. Note that the `<img/>` tag is self-closing.

### Syntax

```
<img src='mylocalimage.jpg'/>
```

## Line breaks

This tag is used in a block of text to force a line break. This is to be used for things which are a single paragraph, but where this formatting is necessary such as poems or addresses. To separate paragraphs, separate each



paragraph into a separate element instead. The resulting element on a web page will look like:

#### Example

```
<p> Some text <br/> that spans two lines </p>
```

## Links

Link elements are used to connect your document to a related resource (very different from hyperlinks, which take you to another webpage when you click on them). Links appear only in the head section of a document so they do not alter the content, but only the presentation. Links are most commonly used to connect to a stylesheet, script, favicon, or alternate format of the page such as an RSS feed or PDF.

#### Example

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

## Lists

HTML supports two kinds of lists: ordered lists and unordered lists. Within lists each individual list item has its own tag.

### UNORDERED LISTS

Unordered lists are just lists whose items are denoted with bullet points.

#### Example

```
Shopping list <ul> <li>Dish soap</li> <li>Kitty litter</li>  
<li>Tomato sauce</li> </ul>
```

#### Read more

- <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/ol>

### ORDERED LISTS

Ordered lists' items are denoted with numbers.

### Example

```
My numbered list <ol> <li>First item!</li> <li>Second item!</li>  
<li>Last item!</li> </ol>
```

### Read more

- <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/li>

## Paragraphs

**<P>**

One of the most common tags in HTML - it denotes a paragraph of text. It often has other elements nested inside of it, such as `<img/>`, `<a>`, `<strong>` and `<em>`.

### Syntax

```
<p>This is paragraph text!</p>
```

## Semantic formatting

These tags are similar to the previously mentioned formatting tags which have fallen out of favor. The difference is that these tags have semantic value (meaning). `<em>` is used for something that you wish to emphasize and `<strong>` is used for something that is important. With both of these elements, you can convey the level of emphasis or importance with nesting. The more times that you nest the element within itself, the higher the magnitude of the text it contains.

### Example

```
<p><strong><strong>Warning:</strong>Acid can cause severe  
burns</strong> </p>
```

## Tables

An element for displaying information in rows and columns. Supports headers and footers for labeling columns. Divides information into rows (denoted by the tr tag) which contain cells (denoted by the td tag).

#### Example

```
<table>  <thead>    <tr>      <th>Item</th>    <th>Price</th>
</tr>    </thead>    <tbody>    <tr>      <td>Banana</td>
<td>$56.75</td>    </tr>    <tr>      <td>Yogurt</td>
<td>$12.99</td>    </tr>    </tbody>    <tfoot>    <tr>
<td>Total</td>    <td>$69.74</td>    </tr>    </tfoot> </table>
```

## Tags & Elements

Tags are basic labels that define and separate parts of your markup into elements. They are comprised of a keyword surrounded by angle brackets `<>`. Content goes between two tags and the closing one is prefixed with a slash (Note: there are some self-closing HTML tags, like image tags). Tags also have attributes, which are properties of the content, such as the "href" attribute of a hyperlink tag "a".

#### Syntax

```
<tag attribute='value'>content</tag keyword>
```

## Title

This tag tells the browser what to display as the page title at the top and tells search engines what the title of your site is. It goes inside `<head>` tags. Try and make your page titles descriptive, but not overly verbose.

#### Example

```
<title> HTML Glossary </title>
```