What’s HTML?

Hyper Text Markup Language

HTML basics

Tags

Attributes

Tags

```
<p>I’m a paragraph</p>

<div>I’m a meaningless container</div>

<a href="http://google.com">I’m a link</a>
```

Tags

doctype, html, head, title, base, link, meta, style, script, noscript, body, article, nav, aside, section, header, footer, h1-h6, main, address, p, hr, pre, blockquote, ol, ul, li, dl, dt, dd, figure, figcaption, div, table, caption, thead, tbody, tfoot, tr, th, td, col, colgroup, form, fieldset, legend, label, input, button, select, datalist, optgroup, option, textarea, keygen, output, progress, meter, details, summary, command, menu, del, ins, img, iframe, embed, object, param, video, audio, source, canvas, track, map, area, a, em, strong, i, b, u, s, small, abbr, q, cite, dfn, sub, sup, time, code, kbd, samp, var, mark, bdi, bdo, ruby, rt, rp, span, br, wbr ...

Attributes

```
<img src="images/logo.png" alt="Company">

<h1 id="header">Hello, world!</h1>

<div class="container">
   <p>I can be styled using CSS!</p>
</div>
```

Tags + Attributes

- Load assets like images, css, javascript...
- Wrap and structure content
- Add meaning to the wrapped content
- Allow CSS styling
- Communicates browsers explicit features of your page.

Comments

```
<!-- I’m an HTML comment -->

<!-- Use me with caution -->
```

Basic HTML file

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <title>Page title</title>
  </head>
  <body>
    <img src="doge.jpg" alt="Doge">
    <h1 class="hello-world">Hello, world!</h1>
  </body>
</html>
```

Basic HTML file breakdown

```
<!DOCTYPE html>
```

Enforce standards mode and more consistent rendering in every browser possible.

Basic HTML file breakdown

```
<html lang="en">
  <head>
    <!-- Non graphically represented stuff -->
  </head>
  <body>
    <!-- Your awesome content -->
  </body>
</html>
```

The <html> tag declares the start of the document, the <head> is mostly used to call external css files, and declare properties of the current page. The <body> contains everything represented in the viewport.

Basic HTML file breakdown

```
<html lang="en">
```

A lang attribute on the root html element defines the document language. Mostly for accessibility (a11y) reasons.

Basic HTML file breakdown

```
<meta charset="UTF-8">
```

Ensure proper rendering of your content by declaring an explicit character encoding.

Basic HTML file breakdown

```
<meta name="viewport" content="width=device-width,
initial-scale=1">
```

Put this inside your <head>, it’ll be useful in a while.

Basic HTML file breakdown

```
<img src="doge.jpg" alt="Doge">
<h1 class="hello-world">Hello, world!</h1>
```

The <img> tag represents an image and the <h1> is a header of level 1 (there’s six levels of headers).

CodePen

Frontend playground

Check it out

Basic HTML file

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <title>Page title</title>
  </head>
  <body>
    <img src="doge.jpg" alt="Doge">
    <h1 class="hello-world">Hello, world!</h1>
  </body>
</html>
```

Exercise

Create a basic skeleton HTML file:

1. Name it so it’s accessible from the root domain.
2. Place the text ‘Hello world!’ inside some element of your choice.
3. Congrats you’re a front-end developer.

HTML forms

Basic HTML form

```
<form>
  <div>
    <label for="hi">Say hi</label>
    <input type="text" name="hi" id="hi">
  </div>
  <button type="submit">Send</button>
</form>
```

HTML best practices

CSS includes

```
<!-- External CSS -->
<link rel="stylesheet" href="style.css">

<!-- In-document CSS -->
<style>
  /* ... */
</style>
```

The CSS is included inside the <head> tag.

JavaScript includes

```
<!-- External JavaScript -->
<script src="app.js"></script>
```

The JS files should be appended just before the closing of the </body> tag.

Use double quotes on attributes

```
<!-- Don’t, just don’t -->
<input type='file' name='file'>

<!-- Much better -->
<input type="file" name="file">
```

It might mess with your text editor’s capability to highlight attribute values.

Avoid the trailing slash in self-closing elements

```
<!-- Do -->
<img src="image.jpg">

<!-- Don't -->
<img src="image.jpg" />
```

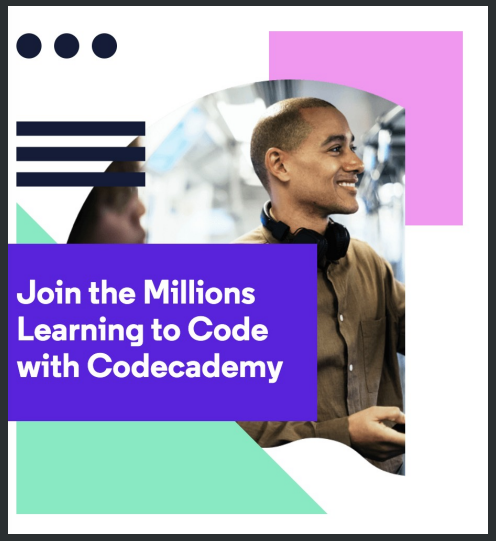
The HTML5 specification says they're optional.

Reduce markup

```
<!-- Not so great -->
<span class="avatar">
  <img src="...">
</span>

<!-- Better -->
<img class="avatar" src="...">
```

Avoid superfluous parent elements when writing HTML. Usually this requires iteration and refactoring, but produces less HTML, which is good for maintenance.



Learn to Code - -> CodeAcademy