

HTML

Objectives

- Construct an HTML page with common elements and attributes
- Describe what the purpose of each HTML tag is
- Name the two HTML tags that function as containers
- Describe what it means for an HTML element to be a 'child' of another HTML element
- Identify the limitations of what can be done with HTML alone

Cool Stuff using HTML/CSS/JavaScript

- [Tearable Cloth](#)
- [3D Solar System](#)
- [Fun with Primes](#)
- [Snake Grid](#)

HTML stands for **H**yper**T**ext **M**arkup **L**anguage

We're going to step through a HTML document, dissect the parts, and view some examples. But first, let's explore the syntax of HTML elements.

HTML elements

HTML elements consist of a start tag, an end tag, and content in-between. Example:

```
<h1>Here is some text</h1>
```

Elements can be nested.

```
<div>
  <h1>Here is <strong>some</strong> text</h1>
</div>
```

Elements can also contain attributes, which are key-value pairs.

```
<div data-attr="5"></div>


```

Note that the last element is an example that does **not** need a closing element. These are known as **void elements**.

Some of the more important attributes are `class` and `id`, which we will see later in CSS. Just know that class names can be used in more than one element, but ids must be unique. See below.

```
<div class="general-container"></div>
<div class="general-container"></div>
<span class="general-container"></span>

<div id="specific-container"></div>
```

Going through a HTML document

What is `<!DOCTYPE html>`?

This statement specifies the markup rules for the page. There are other DOCTYPEs that define [other markup standards](#), like XHTML.

You won't need to worry about the other document types for now, so make sure to always use `<!DOCTYPE html>` at the top of a HTML document.

The `<html></html>` tags

These tags tell the browser we're beginning a HTML document. Put everything inside these tags.

The `<head>` tags

The `<head></head>` tags is where hidden information about the document goes. This includes metadata, the title of the webpage (which appears in the browser), links to CSS and possibly JavaScript files. The head goes at the top of the page, and is declared only once.

Meta Tags

Metadata is data (information) about data. The tag provides metadata about the HTML document. Metadata will not be visible on the webpage, but will be machine parsable. Meta elements are typically used to specify page description, keywords, author of the document, last modified, and other metadata. The metadata can be used by browsers (how to display content or reload the page), search engines (keywords), or other web services.

Some common meta tags you will see are charset, content, author, and description (for SEO).

The `<body>` tags

The `<body></body>` tags denote the content of the document. This content is rendered and displayed in the browser.

Divs + Spans

HTML provides for us two 'empty' containers to store whatever content we want. One is a div (block element) and the other is a span (inline element)

See the Pen [Divs vs Spans](#) by Brian Hague ([@bhague1281](#)) on [CodePen](#).

Text Tags

See the Pen [Text Tags](#) by Brian Hague ([@bhague1281](#)) on [CodePen](#).

Lists

See the Pen [Lists](#) by Brian Hague ([@bhague1281](#)) on [CodePen](#).

Tables

See the Pen [Tables](#) by Brian Hague ([@bhague1281](#)) on [CodePen](#).

Before CSS became mainstream, websites were designed using tables. Although they are much less frequently used, building tables is still a very useful skill to know. The tags for tables are such:

- `<table></table>` - create a table
- `<tr></tr>` - create a table row
- `<th></th>` - create a table heading
- `<td></td>` - create a table cell
- `<tbody></tbody>` - create the body of the table (newer tag)
- `<thead></thead>` - create the head of the table (newer tag). No matter where this is located, whatever is in it will be the first row
- `<tfoot></tfoot>` - create the foot of the table (newer tag). No matter where this is located, whatever is in it will be the last row

Practice Creating Tables

Clone the repo and open instructions.html in your browser. Edit the basic data provided in skeleton.html to create a webpage that looks like the solution picture shown in the instructions.

https://github.com/WDI-SEA/html_top_ten_movies_table

Images and Links

See the Pen [Images and Links](#) by Brian Hague ([@bhague1281](#)) on [CodePen](#).

Forms, Labels, Input Types

Forms

One of the most common ways to send data to a server is by using a form. A form has two essential attributes, action and method.

- **Action** - This specifies a route where you are going to. For example an action of '/test' will take you to the /test route (something you have probably configured in your server side code)
- **Method** - The HTTP Verb that this form will be using (HTML only knows GET and POST, but there are ways to override this default which we will see when we use Node and Rails. The default method is GET so if you are making a GET request you can leave this empty.

Labels Labels are text you place before/after inputs to tell the user what the input is for. The for attribute is for screen readers and if the ID of the input matches the ID of the for attribute then you can click on the label and have it automatically focus/check the input.

Input Types

By default, input elements will allow users to type in text. There's also a plethora of different input types, specified by a `type` attribute. Take a look at the Codepen below for some examples.

See the Pen [Form Elements](#) by Brian Hague ([@bhague1281](#)) on [CodePen](#).

Documentation on input types: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/input>