

Introduction to HTML, CSS and web frameworks

Programming (Gold)

Dr Ryan Crosby

Autumn 2024

Week aims

- Give an introduction to HTML

HTML

About I

- **Hypertext Markup Language**
 - Tim Berners-Lee
 - language describing the structure of a Web page
 - providing instructions for displaying content in Web browsers



About II

- Version HTML 5
 - <https://developer.mozilla.org/en-US/>
 - Combined with
 - CSS + scripting language



HTML elements / building blocks

<u><!DOCTYPE></u>	Document type
<u><HTML></u>	Opening and closing tags of HTML Docs
<u><header></u>	Header for a document or section
<u><body></u>	Document's body
<u><p></u>	Paragraphs
<u>
</u>	Single line break
<u><a></u>	Hyperlink
<u><form></u>	HTML form for user input
<u><div></u>	Section tags

Structure

Everything (excluding the doctype declaration) should be enclosed within a HTML tag

Typically, HTML tags have an opening tag and a closing tag

```
<tag open> content </tag close>
```

Within the HTML tags you should have two main sections

Head

Body

Header

Remember that from this point on, all content should be within the HTML tag

The header section is where you store information about the document.

Header tags

`<head> </head>` Opens and closes the header section

`<title> </title>` Gives the webpage a title

`<meta>` Display meta data

`<script> </script>` JavaScript goes here

`<style> </style>` CSS goes here

Body

The body defined by the tags <body> </body> is where the content of the webpage goes.

This information is processed by the browser and displayed to a user
May be influenced by <head> data

Body Tags

- Block level tags `<p> </p>`
 - Define blocks of content
 - Browser will add new lines above and below
 - Default width/height automatically
- Inline level tags ` `
 - Semantics and organisation within block
 - Not associated with new lines
 - Must be nested inside a block level element
- Replaced tags ``
 - Browser calculates dimensions and replaces with embedded or drawn objects
 - Nesting as for inline tags
 - Used for specific page elements such as images and form controls

Example

```
<!DOCTYPE html>
<html>
  <head>
    <title>Title</title>
  </head>
  <body>

    <h1>Heading</h1>
    <p>Paragraph.</p>

  </body>
</html>
```

Attributes

- Name / Value pair

```
<p attribute="value">Text</p>
```

- Specify additional properties and behaviour and are declared in the **opening** tag

```
<div id="sectionOne">Text</div>
```

- This includes self closing tags

Common Attributes

- These attributes can be applied to any <body> tag
- Typically used to improve the framework for enhancements to the user experience

Attribute	Purpose
class	Associates an element with a CSS class
id	Uniquely identifies element for CSS/scripting
style	Provides inline CSS style rules for an element
title	Describes an element and its content. Creates tool-tips in browsers & used by screen readers

Lists

- Lists make up a large part of HTML content. They are made up of two main tags, the list type tag and the list element tag.

```
<listTypeTag>  
  
<listElement></listElement>  
  
</listTypeTag>
```

```
<ul>
```

```
<li></li>
```

```
</ul>
```

```
<ol>
```

```
<li></li>
```

```
</ol>
```

```
<dl>
```

```
<dt></dt>
```

```
<dd></dd>
```

```
</dl>
```

Unordered List

- Used when there is no order to your list. E.g. bullet points.

```
<p> I teach the following modules. </p>
<ul>
<li>Programming Gold</li>
<li>Computer Science into Schools</li>
<li>Introduction to Computer Science</li>
</ul>
```

I teach the following modules.

- Programming Gold
- Computer Science into Schools
- Introduction to Computer Science

Ordered List

- Used when there is an order to what you are trying to display

```
<p> Content in this module will be taught  
in the following order. </p>
```

```
<ol>  
<li>Git</li>  
<li>HTML</li>  
<li>CSS</li>  
</ol>
```

Content in this module will be taught in the following order.

1. Git
2. HTML
3. CSS

Ordered Attributes

- Reversed
- Start
- Type: (a, A, i, I ,1)

```
<ol reversed> </ol>
```

```
<ol start = 5> </ol>
```

```
<ol type = “a”> </ol>
```

Ordered Attributes

- a: lowercase letters
- A: uppercase letters
- i: lowercase Roman numerals
- I: uppercase Roman numerals
- 1: numbers

```
<p> Content in this module will be taught  
in the following order. </p>  
<ol reversed, type = “a”>  
<li>Git</li>  
<li>HTML</li>  
<li>CSS</li>  
</ol>
```

What happens here ?

```
<p> Content in this module will be taught  
in the following order. </p>  
<ol start = 5, reversed>  
<li>Git</li>  
<li>HTML</li>  
<li>CSS</li>  
</ol>
```

Content in this module will be taught in the following order.

- 5. Git
- 4. HTML
- 3. CSS

Unordered Attributes?

- Type
- Compact

```
<ul type = “circle, square, disc, triangle”> </ul>  
<ul compact> </ul>
```



Warning: Do not use this attribute, as it has been deprecated: use [CSS](#) instead. To give a similar effect as the `compact` attribute, the CSS property `line-height` can be used with a value of `80%`.

Description List

- Used when you need to give definitions to things. Used less than the other two as there are other ways to display this information.

```
<p> Content in this module will be taught  
in the following order. </p>  
<dl>  
<dt>Git</dt>  
<dd>This will be taught in week one </dd>  
<dt>HTML</dt>  
<dd> This will be taught in week two</dd>  
</dl>
```

Content in this module will be taught in the following order.

Git

This will be taught in week one

HTML

This will be taught in week two

Nesting lists

- Lists can be nested within a list element tag

```
<p> My favourite Authors Are: </p>
<ul>
    <li> Brandon Sanderson
        <!-- Don't close the tag here if
you want things to be in the same list
element-->
        <ol>
            <li>Mistborn</li>
            <li>Way of Kings</li>
        </ol>
    </li>
    <li> Leigh Bardugo </li>
</ul>
```

My favourite Authors Are:

- Brandon Sanderson
 - 1. Mistborn
 - 2. Way of Kings
- Leigh Bardugo

Hyperlinks

- Hyperlinks are the navigation system of the web. They can be used for external navigation or internal navigation. Internal navigation can be to either another page on your website or a different section of the current web page.

```
<a href = “ ”> </a>
```

```
<p> I am creating a link to <a href=“www.google.com”>google</a>. </p>
```

Relative links to a part of your page.

- To create a link to somewhere in a page, you may do this with an id.

```
<h2 id= "Heading_two"> Heading Two</h2>
```

```
<p>
<a href="contacts.html#Heading_two">This will take a user to a
specific part of a different page </a>.
</p>
```

```
<p>
  The <a href="#Heading_two">This will take the user to the
part of the current page with the id Heading two.</a> </p>
```

Links to files.

- You can create a hyperlink to a file. The path has to be relative to where the html file is.
- E.g Same folder

```
<a href = “document.pdf ”> </a>
```

- E.g. Folder below

```
<a href = “ folderName/document.pdf”> </a>
```

- E.g. Folder Above

```
<a href = “../document.pdf ”> </a>
```

Absolute Vs Relative URLs

- **Absolute URL:** Points to a location defined by its absolute location on the web, including protocol and domain name. An absolute URL will always point to the same location, no matter where it's used.
- **Relative URL:** Points to a location that is *relative* to the file you are linking from, more like what we looked at in the previous section. -> Be careful with these links as changing the file location can make the link invalid.

Accessibility

- Need to make sure that your links are accessible.
 - Screen readers often don't have the context of your links.
So randomly placing a link is not great for those.
 - Including key words in your links allows search features and search engines to accurately locate them.
 - Helpful when reading pages to see if the content is relevant.

Attributes

- Title: Tells the reader (and screen readers) the context of the links when hovered.

```
<a href = "document.pdf" title= "A link to an important document"> </a>
```

- Download: Gives a standardised file name to something you want someone to download

```
<a href = "document.pdf" download = "document.pdf"> </a>
```

Special Characters (version dependent)

Character	Entity Name	Description
"	"	quotation mark
'	'	apostrophe
&	&	ampersand
<	<	less-than
>	>	greater-than

```
<p>&quot; This is how you include quotation marks &quot; </p>
```

" This is how you include quotation marks "

Multimedia – Images.

- To display images on a webpage we need to use the `` tag. The image tag must have two attributes the ‘src’ which is the source of the image (file path location or URL) and the alt, a text-based description for the image.

```
<img src = “image.png” alt= “Image Description” />
```

- Width and height are also possible attributes.

Tables

- Tables like description lists have two main tags.
- Table data <td> </td>
- Table row <tr> </tr>

```
<table>
  <tr>
    <td>Emil</td>
    <td>Name</td>
    <td>SirName</td>
  </tr>
  <tr>
    <td>ryan.crosby@durham.ac.uk</td>
    <td>Ryan</td>
    <td>Crosby</td>
  </tr>
</table>
```

Tables

- Tables like description lists have two main tags.

```
<table>
  <tr>
    <td>Emil</td>
    <td>Name</td>
    <td>SirName</td>
  </tr>
  <tr>
    <td>ryan.crosby@durham.ac.uk</td>
    <td>Ryan</td>
    <td>Crosby</td>
  </tr>
</table>
```

Emil	Name	SirName
ryan.crosby@durham.ac.uk	Ryan	Crosby

HTML Forms

- The majority of inputs within HTML come in the shape of forms. The most basic form capture is the form tag. Most elements of a form are created via the use of the input tag.

```
<form>  
<input type = “text”/>  
</form>
```



Warning: It's strictly forbidden to nest a form inside another form. Nesting can cause forms to behave unpredictably, so it is a bad idea.

Attributes

- Autocomplete : can the browser autofill this element
 - Address forms.
- Action: How the form is processed. (URL, PhP file, JavaScript function)
- Method: HTTP method to submit the form (POST/GET)

Fieldset and Legend

```
<form>
  <fieldset>
    <legend>Fruit juice size</legend>
    <p>
      <input type="radio" name="size"
id="size_1" value="small" />
      <label for="size_1">Small</label>
    </p>
    <p>
      <input type="radio" name="size"
id="size_2" value="medium" />
      <label
for="size_2">Medium</label>
    </p>
  </fieldset>
</form>
```

- Field set groups together form widgets that share the same purpose.
- Legend gives a label to all of these widgets.

Fieldset and Legend

Fruit juice size

- Small
- Medium
- Large



Input

- Input changes depending on what you put in the type attribute. The options are as follows

— Text	— Email	— Radio	— Tel
— Button	— File	— Range	— Time
— Checkbox	— Hidden	— Reset	— Url
— Color	— Month	— Search	— Week
— Date	— Number	— Submit	
— Datetime	— Password		

```
<input type="" />
```

Label

- The label tag gives a caption for an item as a user interface. They are used with form element tags. E.g. input, option, list etc.

```
<div class="preference">  
    <label for="cheese">Do you like cheese?</label>  
    <input type="checkbox" name="cheese" id="cheese" />  
</div>  
<div class="preference">  
    <label for="peas">Do you like peas?</label>  
    <input type="checkbox" name="peas" id="peas" />  
</div>
```

Code from MDN developer

Label

- Visually and programmatically associated with the corresponding form element. This can help with accessibility devices such as screen readers.
- Increased the hit box for the form element.
- To use the label feature you need to give the element its own ‘id’ value.
- button, input, meter, output, progress, select and text area

Select

- Represents a control that allows for a selection of multiple objects.

```
<label for="pet-select">Choose a pet:</label>
<select name="pets" id="pet-select">
    <option value="">--Please choose an option--</option>
    <option value="dog">Dog</option>
    <option value="cat">Cat</option>
    <option value="hamster">Hamster</option>
    <option value="parrot">Parrot</option>
    <option value="spider">Spider</option>
    <option value="goldfish">Goldfish</option>
</select>
```

Select Attributes

- Disabled
- Multiple
- Name
- Size
- Autocomplete

```
<label for="pet-select">Choose a pet:</label>
<select name="pets" id="pet-select" size="5" multiple>
  <option value="">--Please choose an option--</option>
  <option value="dog">Dog</option>
  <option value="cat">Cat</option>
  <option value="hamster">Hamster</option>
  <option value="parrot" disabled>Parrot</option>
  <option value="spider">Spider</option>
  <option value="goldfish">Goldfish</option>
</select>
```

Text area

- Text area is similar to the text feature, the main difference is the ability to have multiple lines in the text area, for larger user inputs.

```
<textarea name="textarea" rows="5" cols="30" placeholder="This is a text area."></textarea>
```

Button

- Interactable element on the screen which will perform an action when clicked.

```
<button type="button"> Button </button >
```

```
<input type = "submit" value = "submit button" />
```

- A special type of input that turns into a button. Once clicked will call the action element of a form.

Get vs Post

- Get and post methods are ways of sending data – usually from forms.
- The main thing to remember is that GET appends the data to the end of the URL, whereas Post sends data to the server using environment variables. Post is slightly more secure as the user can't see the data in the browser. It is not however secure on its own.