1

HTML
HyperText Markup Language

Mistakes are: Expected, respected, inspected, corrected.

- Anonymous

Contents

How To Use This Document

Bits of text in red are links and should be clicked at every opportunity. Bits of text in `monospaced green` represent code. Most the other text is just text: you should probably read it.

Some sections are marked "Read-Only": these are sections that are intended to be read through in your own time and will not have a corresponding lecture.

Copying and pasting code from a PDF can mess up indentation. For this reason large blocks of code will usually have a View Code link above them. If you want to copy and paste the code you should follow the link and copy the file from GitHub.

Taking Notes

In earlier cohorts I experimented with giving out notes in an editable format. But I found that people would often unintentionally change the notes, which meant that the notes were then wrong. I've switched to using PDFs as they allow for the nicest formatting and are also immune from accidental changes.

Make sure you open the PDF in a PDF viewing app. If you open it in an app that converts it into some other format (e.g. Google Docs) you may well miss out on important formatting, which will make the notes harder to follow.

I make an effort to include all the necessary information in the notes, so you shouldn't need to take any additional notes. However, I know that this doesn't work for everyone. There are various tools that you can use to annotate PDFs:

- Preview (Mac)
- Edge (Windows)
- Hypothes.is
- Google Drive (*not* Google Docs)
- Dropbox

Do not use a word processor to take programming notes! (e.g. Google Docs, Word, Pages). Word processors have the nasty habit of converting double-quotes into "smart-quotes". These can be almost impossible to spot in a text-editor, but will completely break your code.

Chapter 1

Introduction

1.1 HTML Basic Syntax & Structure

HTML stands for Hypertext Markup Language. It’s the code the describes the content you see on the World Wide Web.
It’s a *markup* language, a data structure based on another language XML.

HTML == Content

HTML describes the content we see on a webpage, CSS is used to *style* it, give it colours and placement. JavaScript is used for interactivity, like changing things when a user clicks.

1.1.1 Let’s take a look at the syntax:

```
<p class="lede">Here is some text</p>
```

Here we write the element `<p>`, by opening angle brackets, writing the element and closing them. We write the content inside and *close* the element by writing it again but adding a forward slash after the first angle bracket.

Closing Elements

It’s really important to close your HTML elements. Modern browsers will try to do this for you, however if there are multiple unclosed elements it can get confused and cause strange behavior.

There are some elements that close themselves, rather than writing the element again at the end:

```
<img src="myimage.jpg" alt="A photo of a girl holding a book" />
```

1.1.2 Attributes

Inside our elements we have *attributes*; `class="lede"`, these give the program more information about the element.
There are lots of these too and are written by writing the attribute name, followed by an equals sign, then opening up quotes and writing what we call the *parameter* inside.
You can have more than one attribute for an element so we separate them with a space.

1.1.3 Nesting elements

You can put elements inside of elements. This is often referred to as nesting.

```
<p class="lede">
    <img src="myimage.jpg" alt="A photo of a girl holding a book" />
</p>
```

Note: We add a tab to the inside element when we do this, so we can easily read it. When we put an element inside another element the one inside is called the child and the one containing the other element is called the parent. We can put as many elements inside as many elements as we like!

1.1.4 Whitespace

This is what we call the spaces in code, like the regular space or tabs at the beginning. Spaces matter when you write HTML elements because you want to make sure attributes don't merge into each other or the element.

1.1.5 Comments

Comments are parts of a code file that are ignored by the program running it. In HTML we write a comment like this:

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

Comments are useful to describe what the code is doing, or to remind yourself of things. They can even be used for documentation.

1.1.6 Basic structure

This is the bare minimum you need in your HTML file:

```
<!DOCTYPE html>
<html lang="en">
    <head>
        <title></title>
    </head>
    <body>
        <!--write the content here -->
    </body>
</html>
```

There's a declaration at the top, then everything is inside `html` tags. The next section describes which we would write in the `head` element and we add all the content to be shown inside the `body` element. There can only be one of each of these per HTML document

1.2 Metadata

```
<head>
    <!-- character set -->
    <meta charset="utf-8">

    <!-- use ie edge if available -->
    <meta http-equiv="x-ua-compatible" content="ie=edge">

    <!-- ignore retina pixels -->
    <meta name="viewport" content="width=device-width,
    ↪  initial-scale=1">

    <!-- title and description -->
    <title></title>
    <meta name="description" content="">

    <!-- icons -->
    <link rel="apple-touch-icon" href="icon.png">
    <!-- Place favicon.ico in the root directory -->

    <!-- CSS files -->
    <link rel="stylesheet" href="css/main.css">

    <!-- JS files, might not be here -->
    <script src="js/vendor/modernizr-2.8.3.min.js"></script>
</head>
```

Here is a https://github.com/h5bp/html5-boilerplate/blob/master/src/index.html which is maintained to the latest standard.

Chapter 2

Element reference

2.1 HTML Content Elements

h1 - h6 `<h3>My Heading</h3>` Headings for different parts of the webpage

p `<p>My sentence here</p>` For paragraphs and lines of text

ol `<ol>...</ol>` Ordered list (numbered)

ul `<ul>...</ul>` Unordered list (numbered)

li `<li>List item content</li>` List item, only direct children of `ol` and `ul`

dl `<dl>...</dl>` Description list

dt `<dt>Term here</dt>` Description term

dd `<dd>Definition here</dd>` Definition

a `<a href="http://webaddress.com">Link Text</a>` Anchor or link; takes you somewhere & needs `href` attribute

button `<button>Do something</button>` Button for interactivity

img `<img src="example.png" alt="A photo of a girl holding a book" />` Image. Don't forget the attributes.

blockquote `<blockquote cite="">...</blockquote>` To add a quote, optional cite attribute

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

details `<details>...</details>` Details

summary `<summary>...</summary>` Summary for details (this is what is shown)

The difference between a button and a link

Lorem Ipsum

Lorem Ipsum is latin. We use blocks of latin text as *placeholders* when we are building websites, before real content is added. It's a good language as it doesn't mean much to us, but is still representative of how westernised languages would appear in blocks & sentences. To get **Lorem Ipsum** you can type *Lorem* followed by enter into vscode, or find a generator online here.

Tables

Tables contain a lot of HTML. Below is an example of a simple table. We only use them in HTML for data which is good to represent as a table, however they are still used to layout emails

```html
<table>
    <tr>
        <th>Dessert</th>
        <th>Calories</th>
        <th>Fat</th>
        <th>Carbs</th>
    </tr>
    <tr>
        <td>Frozen yogurt</td>
        <td>159</td>
        <td>6.0</td>
        <td>24</td>
    </tr>
    <tr>
        <td>Ice cream sandwich</td>
        <td>237</td>
        <td>9.0</td>
        <td>37</td>
    </tr>
    <tr>
        <td>Eclair</td>
        <td>262</td>
        <td>16.0</td>
        <td>24</td>
    </tr>
</table>
```

2.2 HTML Sectioning Elements

header `<header>...</header>` Header of the part of the document

footer `<footer>...</footer>` Footer of the part of the document

main `<main>...</main>` Main unique content, can only be one per page

aside `<aside>...</aside>` Related, but not main, content.

nav `<nav>...</nav>` Site related navigation.

article `<article>...</article>` A standalone piece of content (always has header).

section `<section>...</section>` Part of a piece of content.

div `<div>...</div>` For design purposes only.

2.3 HTML Media Elements

video `<video src="myvideo.mp4" poster="firstimage.jpg" controls></video>` Include a video

audio `<audio src="myaudio.mp3" controls></audio>` Include an audio track

picture `<picture>...</picture>` Use to load different image sizes and types (with source)

source `<source src="/media/examples/flower.mp4" type="video/mp4">` Add inside audio or video to load different file types

figure `<figure>...</figure>` Usually a figure is an image, illustration, diagram, code snippet.

figcaption `<figcaption>...</figcaption>` A caption or legend describing the rest of the figure contents.

2.4 HTML Text Elements

strong `<strong>This is important</strong>` Important text, usually bold style

em `<em>This is emphasised</em>` Emphasised text, usually italic style

b `<b>This is bold</b>` Bold style

i `<i>This is italic</i>` Itlaic style

span `<span>This text has an element around it</span>` Like a div but for text

sup & sub `<sup>...</sup>` `<sub>...</sub>` Smaller raised or lowered text

time `<time datetime="2020-07-21">21st July</time>` Any time or date.

q `<q>Someone said this</q>` A quote but inline with the text

code `<code>Inline</code>` Describes code

pre `<pre>...</pre>` Used for code blocks

Semantics

Semantics is a word used to describe giving something meaning. We use it a lot when talking about HTML as that is what the elements are for. To give the content meaning.

Read More

There's more to all of these elements. It's a good idea when you get used to them to check out the documentation for each. Just google the element with *mdn* after. There's also lots more elements. You can find a comprehensive list of all elements here.

Chapter 3

HTML Forms

Forms are really important on the web, as they are used to pass data from the user to the server. Every time you log in to a service and when you *google* you may not realise it but you are using a form.

3.1 Basic Structure

- `<form method="" action="">` Parent container, encapsulates entire form
- `<fieldset>` Sections within a big form
- `<legend>` Label for a fieldset

3.2 Labels

- `<label for="">` Label for each input (apart from submit) needs for attribute to match id of input

3.3 Input

- `<input type="" name="" id="" />` Self closing. Necessary attributes are type (see below) name (passed with data) and id (links to label for)

3.3.1 Common Inputs Types

- `text` Default, generic text box
- `email` Email address
- `password` Shows dots or stars instead of letters

- `number` integer only
- `tel` telephone number
- `url` web address format
- `checkbox` displays a tick box
- `file` upload a file
- `search` if it's a search
- `hidden` to pass data with no user input

Radio buttons

```
<div>
    <label for="inradio">Radio</label>
    <input type="radio" name="radioin" value="optone" id="inradio"
    ↪   /><span>Option one</span>
    <input type="radio" name="radioin" value="optwo" /><span>Option
    ↪   Two</span>
</div>
```

Date and Time

- `date` Just date
- `datetime-local` Date and time
- `week` Week number
- `month` Month
- `time` Time

3.3.2 Not written as input

Dropdowns

```
<label for="dropdown">Select</label>
<select id="dropdown" name="dropdown">
    <option value="opone">opt one</option>
    <option value="optwo">opt two</option>
    <option value="opthr">opt three</option>
</select>
```

TEXTAREAS

```
<label for="message">Textarea</label>
<textarea id="message"></textarea>
```

3.3.3 Buttons

INPUTS

Use the `value` attribute to change text

- `button` No default behaviour
- `reset` Clears the form, better not use it
- `submit` Submits the form

```
<button type="submit">Also submits form</button>
```

3.3.4 More input types

- `range` Value, allows for min, max and step
- `color` Select a colour

This is not a definitive list, you can find a definitive list and more information here

Glossary

- **HTML**: Hypertext Markup Language. The language you write to describe content being displayed on a webpage
- **XML**: The language HTML is based on, it's a *data structure*.
- **element**: Sometimes referred to as *tag*. The part of the HTML inside angle brackets which describes the content.
- **attribute**: Gives the element more information. Write like `attribute="parameter"`. Add these to the opening element with spaces inbetween them.
- **nesting**: When we put elements inside of other elements
- **parent**: The containing element
- **child/children**: The element(s) inside other elements.
- **whitespace**: The parts of a file that don't contain anything. Like spaces, new lines or tabs.
- **comment**: Comments are parts of a code file that are ignored by the program running it.

Colophon

Created using TEX

Fonts

- Feijoa by Klim Type Foundry
- Avenir Next by Adrian Frutiger, Akira Kobayashi & Akaki Razmadze
- Fira Mono by Carrois Apostrophe

Written by Ruth John

∞

December 1, 2020