Web Development Languages

For A-Level Computer Science

Thomas Boxall

February 2021 (Reviewed in April 2022)

Introduction

This was originally produced as part of a directed study week in 2021. During revision for A-Levels in 2022, this document has been reviewed with additional notes added. At this time, it was also typeset into LaTeX.
In the examples used in this document, . . . indicates where other code would be inserted.

Contents

Chapter 1

HyperText Markup Language

This is the standard markup language used to develop web pages. It comprises of elements which have a `<startTag>` and a `</endTag>`. Some elements do not have end tags, for example, the `<br>` tag. These are known as empty elements.

1.1 Elements

Elements are the building blocks of a HTML document. It is through different elements that a web page is designed and built.

1.1.1 Definition of a HTML document

The `<!DOCTYPT html>` tag must appear at the top of every HTML document. It tells the web browser that the document is a HTML document so that it knows how to output it correctly. The element should appear only once, at the top of every HTML file in the website.

1.1.2 Root HTML Element

```
<html>
...
</html>
```

These tags are used to encapsulate the HTML code which is used to build the web page.

1.1.3 Headings

There are 6 heading tags, the text gets smaller as the number in the tag gets bigger.
`<h1> <h2> <h3> <h4> <h5> <h6>`
These are used in a pair, a start tag and end tag.
`<h1>...</h1>`
Browsers will automatically add some white space before and after the heading. Search engines use heading tags to build indexes of what is on pages, this is used in their page rank algorithms. For this reason, do not use heading tags to make text big or bold unnecessary.

1.1.4 Paragraphs

The `<p>` tag is used to define normal text. It is used in the same way as most other tags `<p>...</p>`. A paragraph will always start on a new line and browsers will automatically add some white space before and after the text. The browser will also strip any extra spaces from before and in between the text.

1.1.5 Horizontal Lines

The `<hr>` tag is used to include a horizontal line. They are often used to define a thematic change in the HTML page and separate the page content. This doesn't need an end tag.

1.1.6 Line Break

The `<br>` tag is used to break the line, it is used within other tags (for example, a `<p>`) and doesn't need an end tag.

1.1.7 Hyperlinks

Hyperlinks are fundamental to websites. They allow pages to be linked to other pages, meaning users can move around the website without needing to know the exact URL of each page. The `href` attribute of the `<a>` tag defines where the link should take the user. More information in the Links section.

1.2 Styles

The HTML `style` attribute is used to add styles to an element such as colour, size or font. The syntax is as follows: `<tagname style="property:value;"`. The property is a CSS property and the value is a CSS value.

1.2.1 Background Colour

This defines the background colour of a HTML element. For example, to set the background colour of a `<h1>` element saying ”Cheese” to powder blue:
`<h1 style="background-color:powderblue;">Cheese</h1>`
Using this syntax, the background colours can be different for different elements.

1.2.2 Text Colour

This defines the text colour of a HTML text element. For example, to set the background colour of a `<h1>` to blue with the text ”Apple”:
`<h1 style="color:blue;">Apple</h1>`

1.2.3 Fonts

This defines the font of a HTML text element. For example, to set the font of a `<h1>` element to Veranda with the text ”Milk”:
`<h1 style="font-family:verdana;">Milk</h1>`

1.2.4 Text Size

This defines the font size of a HTML text element. For example, to set the font size of a `<h1>` element to 300% with the text ”Lemon”:
`<h1 style="font-size:300%;">Lemon</h1>`

1.2.5 Text alignment

This defines the alignment of a HTML text element. For example, to set the alignment of a `<h1>` element to the centre with the text ”Orange”:
`<h1 style="text-align:center;">Orange</h1>`

1.3 Text Formatting

Tag	Description
`<b>`	Bold Text
`<strong>`	Important Text
`<i>`	Italic text
`<em>`	Emphasised text
`<mark>`	Marked text (highlights it)
`<small>`	Smaller text
`<del>`	Strike through text
`<ins>`	Inserted text (usually underlined)
`<sub>`	Subscript text
`<sup>`	Superscript text

Table 1.1: Text formatting options

1.4 Colours

These are specified with predefined colour names / RGB / HEX / HSL / RGBA / HSLA values. The style (section 1.2) attribute is used to include colours.

1.5 Links

Links are found in nearly all web pages, they allow users to click from page to page. HTML links are hyperlinks, you can hover over one and the mouse arrow will turn into a little hand. A link doesn't have to be text, links can be images or any other HTML elements. The link colours can be altered using CSS.

1.5.1 Syntax

The `<a>` tag defines a hyperlink. It has the following syntax:
`<a href="url">link text</a>`
The `href` attribute indicates the link's destination. The `link text` is the part which is visible to the reader. Clicking the links text will send the reader to the specified URL address.

1.5.2 Target Attribute

By default, the linked page will be displayed in the current browser tab. To change this, you must specify another target for the link. The target attribute can have one of the following values

Target	Description
`_self`	Default. Opens the document in the same tab as it was clicked in
`_blank`	Opens the document in a new tab
`_parent`	Opens the document in the parent frame
`_top`	Opens the document in the fully body of the window

Table 1.2: Target options

For example, to open the document in a new tab, you would use the following syntax:

```
<a href="url" target="_blank">link text</a>
```

1.5.3 Using an image as a link

Images can be used as a link. The `<a>` tag should wrap around the `<img>` tag.

```
<a href="url">
<img src="imageLocaion" style="width:42px;height:42px;">
</a>
```

1.5.4 Linking to an Email Address

To link to an email address use the `mailto:emailAddress` parameter in the `href` attribute.

```
<a href="mailto:someone@example.com">Send an email</a>
```

1.5.5 Using a button as a link

JavaScript can be used to point to the link rather than using plain HTML. JavaScript is looked at further later in this document.

```
<button onclick="document.location=targetURL">text</button>
```

1.5.6 Link Titles

The title attribute specifies extra information about an element. The element is often shown as a tooltip text when the mouse moves over the element.

```
<a href="linkURL" title="toolTipText">text</a>
```

1.5.7 Bookmarks

Bookmarks can be useful if a web page is very long; they allow users to move up and down the page, jumping directly to an element. To create the bookmark, you have to specify the `id` attribute of the element which you want to bookmark to.

```
<h2 id="C4">Chapter 4</h2>
```

Then, you can add a link to the bookmark

```
<a href="#C4">Jump to Chapter 4</a>
```

1.6 Images

The `<img` tag is used to embed an image in a web page. The images are not technically inserted into a web page, rather they are linked to the web pages. The `<img>` tag creates a holding space for the reference image. The `<img>` tag is empty, it contains attributes only and doesn't have a closing tag.

```
<img src="url" alt="alternativeText" style="width:500px;height:600px;">
```

The `alt` attribute provides alternative text for the browser to display if the image cannot be displayed. The `src` attribute specifies the path(url) to the image. The `style` attribute can be used to change the size of the image.

1.7 Tables

The `<table>` tag defines a table. Each table row is defined with a `<tr>` tag. Each table header is defined with a `<th>` tag, the text is bold and centred. Each table data cell is defined with the `<td>` tag, text is aligned to the left; most other HTML elements can be contained within the cell. Tables can have many different styles, and borders, which can be defined in CSS.

1.8 Lists

There are two types of lists.

1.8.1 Unordered List

These are defined by the `<ul>` tag. Each list item is wrapped in a `<li>...</li>` tag. The `style` attribute can be used to change the style of the bullet. By default, it is set to disc.

```
<ul>
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ul>
```

1.8.2 Ordered List

These are defined by the `<ol>` tag. Each list item is wrapped in a `<li>...</li>` tag. The `type` attribute can be used to change the numbering system.

```
<ol>
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ol>
```

1.9 Block and Inline Elements

To build a HTML web page, different elements can be used to structure the page.

1.9.1 Block-Level Elements

A block-level element always starts on a new line. A block-level element always takes up the full width available (stretches out to the left and right as far as it can). A block-level element has a top and bottom margin, whereas an inline element doesn't. The `<div>` element is a block-level element.

```
<div>Hello World</div>
```

1.9.2 Inline Elements

An inline element doesn't start on a new line. An inline element only takes up as much width as necessary. The `<span>` element is another element. Inline elements cannot contain a block level element.

```
<span>Hello World</span>
```

1.9.3 The div element

This is often used as a container for other HTML elements. It has no required attributes, but `style`, `class` and `id` are common. When used with CSS, the `<div>` can be used to style blocks of content.

```
<div style="background-color:black;color:white;padding:20px;">
    <h2>London</h2>
    <p>London is the capital city of England. It is the most populous city in the
     ↪  United Kingdom, with a metropolitan area of over 13 million inhabitants.</p>
</div>
```

1.9.4 The span element

This is an inline container used to mark up part of a text, or part of a document. It has no required attributes but `style`, `class` and `id` are common. When used with CSS, the `<span>` element can be used to style part of the text.

```
<p>My mother has <span style="color:blue;font-weight:bold">blue</span> eyes and my
 ↪  father has <span style="color:darkolivegreen;font-weight:bold">dark
 ↪  green</span> eyes.</p>
```

1.10 Class Attribute

This is used to specify the class of a HTML element. Multiple different HTML elements can share the same class. It is often used to point to a class name in a style sheet. It can also be used by JS to access and manipulate elements with specific names.

1.10.1 Syntax

Shown below is the use of a class attribute to set CSS in the `<style>` tags in the `<head>` part of the HTML document.

```
.className{
background-color: tomato;
color: white;
}
```

Then, in the `<body>` section of the document, you would use the following syntax to set the class of an element to `className`. `<h2 class="className">Heading2</h2>`

1.10.2 Multiple Classes

HTML elements can belong to more than one class. To define multiple classes, spearate them with a space
`<h2 class="className1 className2">Heading2</h2>`
The HTML element will take styles from both classes.

1.11 Including JavaScript in HTML

JavaScript (JS) makes pages more dynamic and interactive.

1.11.1 The script tag

This is used to define a client-side script (written in JS). It will either contain script statements or point to an external script file through a `src` attribute. Common uses of JavaScript are image manipulation, form validation and dynamic changes of content. To select a HTML element, JavaScript most often uses the `document.getElementById()` method. The following example writes ”Hello JavaScript” into a HTML element with the `id="demo"`.

```
<script>
  document.getElementById("demo").innerHTML = "Hello JavaScript!";
</script>
```

1.11.2 The noscript tag

This defines an alternate content to be displayed to users that have disabled scripts in their browsers or have a browser that doesn't support scripts.

1.12 The head element

The `<head>` element is a container for metadata (data about data) and is placed between the `<html>` and `<body>` tags. HTML metadata is about the HTML document and is not displayed. Metadata typically defines the document title, character set, scripts and other meta information.

1.12.1 The title element

The `<title>` element defines the title of the document must be text only and is shown in the browsers title bar of the pages' tab. It is required in HTML documents and is used in SEO therefore it should be made as accurate and meaningful as possible.

1.12.2 The link element

The `<link>` element defines the relationship between the current document and an external resource. It is most often used to link to external style sheets (CSS documents).

```
<link rel="stylesheet" href="myStyles.css">
```

Chapter 2

Cascading Style Sheets

Cascading Style Sheets, CSS, describes how HTML elements are to be displayed on screen, paper and in other media. It saves a lot of work as it can control the layout of multiple web pages at once. External style-sheets are stored in `.css` files.

2.1 CSS Solved a Problem

HTML was never intended to contain tags for formatting a web page. HTML was intended to describe the content of a web page. HTML 3.2 introduced formatting tags (eg, `<font>`) this caused major problems for web developers so the World Wide Web Consortium (W3C) created CSS. CSS removes the style formatting from HTML and saves a lot of work.

2.2 Syntax

```
p {
  color: red;
  text-align: center;
}
```

The `p` is a selector in CSS. It points to the HTML element that you want to style. `color` is a property, `red` is the property value. `text-align` is a property and `center` is the property value.

2.3 CSS Selectors

These are used to find (or select) the HTMl elements that you want to style. They can be divided into categoried.

- Simple selectors - select elements based on name / id / class
- Combinational selectors - select elements based on a specific relationship between them
- Pseudo-class selectors - select elements based on a certain statement
- Pseudo-elements selectors - select and style part of an element
- Attribute selectors - select elements based on an attribute or attribute values

2.3.1 Element Selector

The element selector selects HTML elements based on the element name. For example, here all the `<p>` elements on the page will be centre-aligned with a red text colour.

```
p{
  text-align:center;
  color:red;
}
```

2.3.2 Id selector

The id selector used the id attribute of a HTML element to select specific element. The id of an element is unique within a page, so the id selector is used to select one unique element. To select an element with a specific id, write a hash (#) character, followed by the id of the element. For example, the code below will be applied to all the HTML elements with the id `para1`.

```
#para1 {
 text-align: center;
 color: red;
}
```

2.3.3 Class selector

The class selector selects all HTML elements with a specific class attribute. To select elements with a specific class, write a period (.) character followed by the class name.

```
.center {
  text-align: center;
  color: red;
}
```

2.3.4 Universal Selector

The universal selector - asterisk (*) selects all HTML element on the page.

```
* {
  text-align: center;
  color: blue;
}
```

2.3.5 Grouping Selector

The grouping selector selects all the HTML elements with the same style definitions. To group selectors, separate their selectors with a comma.

```
h1, h2, p {
  text-align: center;
  color: red;
}
```

2.4 How to add CSS

There are three ways to add CSS to a HTML document. When there are multiple styles specified for a HTML element, all the styles in a page will be cascaded into a new virtual style sheet, in the following order where number one has the highest priority.

1. Inline style (inside a HTML element)
2. External and internal style sheets (in the head section and external CSS document)
3. Browser default

2.4.1 Inline

This is used to apply a unique style to a single HTML element. Inline CSS uses the style attribute of a HTML element.

```
<h1 style="color:blue;">A blue heading</h1>
```

2.4.2 Internal

This is used to define a style for a single HTML page. Internal CSS is defined in the `<head>` section of a HTML document, within a `<style>` element.

```
<head>
  <style>
    body {background-color: powderblue;}
    h1   {color: blue;}
    p    {color: red;}
  </style>
</head>
```

2.4.3 External

This is used to define the style for many HTML pages. External style sheets are linked to in the `<head>` section of each HTML page which uses it.

webpage.html

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

styles.css

```
body {
  background-color: powderblue;
}
h1 {
  color: blue;
}
p {
  color: red;
}
```

2.5 Colours

Colours are specified using any of the following:

- Predefined colour names
- RGB code
- HEX code
- HSL code
- RGBA code
- HSLA code

2.5.1 Background Colours

The background colour of a HTML element can be set as follows

```
<h1 style="background-color:DodgerBlue;">Hello World</h1>
```

2.5.2 Text Colours

The colour of text can be set as follows

```
<h1 style="color:Tomato;">Hello World</h1>
```

2.5.3 Border Colours

The colours of borders can be set as follows

```
<h1 style="color:Tomato;">Hello World</h1>
```

2.5.4 Colour Values

When declaring colours, unless you are using a named colour, you have to define the method which you are declaring the colour with. In all the examples below, the same colour is being declared, in three different ways.

```
<h1 style="background-color:rgb(255, 99, 71);">...</h1>
<h1 style="background-color:#ff6347;">...</h1>
<h1 style="background-color:hsl(9, 100\%, 64\%);">...</h1>
```

2.6 Box Model

This is very important in designing the layout of webpages.
All HTML elements can be considered as boxes. In CSS, the term *box model* is used when talking about design and layout. The CSS box model is essentially a box that wraps around every HTML element. It consists of the following:

- Content - the contents of the bow where images and text appear;

- Padding - the clear area around a box, the padding is transparent;

- Border - the border that goes around the padding and content;

- Margin - the clear area around the outside of the border, it is transparent.

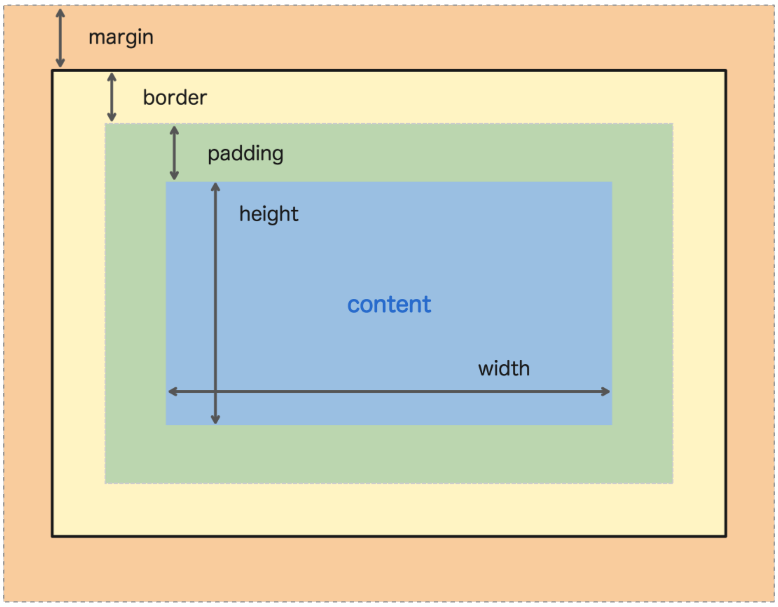

Figure 2.1: The CSS box model

2.6.1 Borders

The CSS `border` property allows you to specify the style, width and colour of an elements border.

Property	Description
`dotted`	Dotted border
`dashed`	Dashed border
`solid`	Solid border
`double`	Double border
`groove`	3D grooved border, effect depends on the `border-color` value
`ridge`	3D ridged border, effect depends on the `border-color` value
`inset`	3D inset border, effect depends on the `border-color` value
`outset`	3D outset border, effect depends on the `border-color` value
`none`	No border
`hidden`	Hidden Border

Table 2.1: CSS border options

Border Width

The `border-width` property specifies the size of the four borders. The width can be set as a specified size (in `px`, `pt`, `cm`, `em`, etc) or by using on of three pre-defined values `thin`, `medium` or `thick`. Different sides are also able to have different thicknesses. For example, for a solid 5px wide border:

```
p.one {
  border-style: solid;
  border-width: 5px;
}
```

Border Colours

The border-color property is used to set the colour of the four borders. If this isn't set then the border inherits the colour of the element.

2.6.2 Margins

The CSS margin properties are used to create space around elements, outside any defined borders. With CSS, you have full control over the margins. There are four properties for setting the margin for each side of an element (top, right, bottom and left).

2.6.3 Padding

The CSS padding properties are used to generate space around an element's content, inside of any defined borders.

2.6.4 Height and Width

The height and width properties are used to set the height and width of an element. The height and width properties do not include padding, borders or margins. It sets the height and width of the area inside the padding, border and margins of the element.

Property	Description
auto	Default, browser calculates height and width
length	Defines the height/ width in px, cm etc
% Percent Symbol	Defines the height/ width in percent of the containing block
initial	Sets the height/ width to its default value
inherit	The height/ width will be inherited from its parent value

Table 2.2: CSS height and width options

For example

```
div {
  height: 100px;
  width: 500px;
  background-color: powderblue;
}
```

Maximum Width

The max-width property is used to set the maximum width of an element. It can be specified in length, values or percent of the containing block. For example

```
div {
  max-width: 500px;
  height: 100px;
  background-color: powderblue;
}
```

2.7 Text

2.7.1 Text Colour

The colour property is used to set the colour of the texti, it is specified by a colour name.

Text Alignment

The text-align property is used to set the horizontal alignment of a text element. A text element can be left or right justified, centred or justified. For example, to centre align a <h1>:

```
h1 {
  text-align: center;
}
```

When the text-align is set to justify, each line is stretched so that every line has an equal width therefore the left and right margins are straight.

2.7.2 Text Direction

The direction and unicode-bidi properties can be used to change the direction of an element. For example

```
p {
  direction: rtl;
  unicode-bidi: bidi-override;
}
```

2.7.3 Vertical Alignment

The vertical-align property sets the vertical alignment of an element.

2.7.4 Text Decoration

The text-decoration property is used to set or remove decorations from text. The value none is often used to remove underlines from links.

2.7.5 Text Transformation

The text-transform property is used to specify uppercase and lowercase letters in a text element. It can be used to turn everything into uppercase or lowercase letters, or capitalise the first letter of each word.

2.7.6 Text Indent

The text-indent property is used to specify the indentation of the first line of text.

```
p {
  text-indent: 50px;
}
```

2.7.7 Letter Spacing

The letter-spacing property is used to specify the space between the characters in a text element.

2.7.8 Line Height

The line-height property is used to specify the space between lines.

2.7.9 Word Spacing

The `word-spacing` property is used to specify the space between the words in a text element.

2.7.10 White Space

The `white-space` property is used to specify how white space inside an element is handled.

2.7.11 Text Shadow

The `text-shadow` property adds shadow to text. For example,

```
h1 {
  text-shadow: 2px 2px 5px red;
}
```

2.8 Fonts

It is important to choose a good font for websites.

2.8.1 Introduction to Fonts

- Serif fonts have small strokes at the edges of each letter. They create a sense of formality and elegance.
- Sans-Serif fonts have clean lines (no small strokes). They create a modern and minimalistic look.
- Mono space fonts have equal spacing for all letters. They create a mechanical look.
- Cursive fonts imitate human handwriting.
- Fantasy fonts are decorative or playful fonts.

2.8.2 Font Families

The `font-family` property specifies the font of a text. The `font-family` should hold seceral font names as a “fallback” system, to ensure maximum compatibility between browser/operating systems. Start with the font you want and end with a generic family (this lets the browser pick a similar font in the family if no other specified fonts are available) the font names should be separated with a comma. If the font name contains more than one word, then it must be in quotation marks. For example

```
.p1 {
  font-family: "Times New Roman", Times, serif;
}
```

2.8.3 Web Safe Fonts

Web safe fonts are fonts that are universally installed across all browsers and devices. However, there are no 100% completely safe web fonts. There is always a chance that a font is not found or installed properly. The best web safe fonts for HTML and CSS are shown below.

- Arial - sans serif;
- Veranda - sans serif;
- Helvetica - sans serif;
- Tahoma - sans serif;

- Trebuchet MS - sans serif;
- Times New Roman - serif;
- Georgia - serif;
- Garamond - serif;
- Courier New - mono space;
- Brush Script MT - cursive.

2.8.4 Font Style

The `font-style` property is mostly used to specify italic text. It has three values: `normal` (text is shown normally); `italic` (text is shown in italics); and `oblique` (text is leaning).

2.8.5 Font Weight

The `font-weight` property specifies the weight of a font.

2.8.6 Font Variant

The `font-variant` property specifies whether or not a text element should be displayed in a SMALL CAPS font. In a small-caps font, all lowercase letters are converted to uppercase letters. However the converted uppercase letters appear in a smaller font size than the original uppercase letters in the text.

2.8.7 Font Size

The `font-size` property sets the size of the text. Being able to manage the text size is important in web design. However, you should not use font size adjustments to make paragraphs look like headings or headings look like paragraphs.
The `font-size` value can be absolute or relative size. Absolute size sets the text to a specified size, doesn't allow the user to change the text size in browsers (bad for accessibility); it is useful for when the physical size of the output is known. Relative size sets the size of the text to be relative to the surrounding elements and allows a user to change the text size in browsers. For example, to set a font size using pixels

```
h1 {
  font-size: 40px;
}
```

2.8.8 Google Fonts

If you do not want to use any of the standard fonts in HTML, you can use Google Fonts[1]. Google Fonts are free to use and have more than 1000 styles. You link them to HTML documents as you would any other style sheet

```
<link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Sofia">
```

[1] https://fonts.google.com/

2.9 Links

Links can be styled in different ways. They can be styled with any CSS properties. For example

```
a {
  color: hotpink;
}
```

In addition, they can be styled differently depending on what state they are in

State	Description
`a:link`	unvisited
`a:visited`	visited
`a:hover`	when a mouse hovers over the link
`a:active`	the moment it is

Table 2.3: Different states which a link can be in

Chapter 3

JavaScript

JavaScript (*JS*) is a widely used programming language that can be used to add interactivity to web pages. It can be used to gather and process data from the user, to create dynamic page elements and to do useful things such as to validate data. The code is interpreted and is mainly used for client-side processing.

3.1 Including JavaScript on Webpages

There are a number of different methods which can be used to include JavaScript in a webpage.

3.1.1 Script Tag

Small amounts of code can be included in the `<script>` tags. If a time to run (eg, on button press) is not specified then it will run as soon as the webpage is opened. For example, the code below will display an alert box containing 'hello world' when the webpage is opened.

```
<script>
alert("Hello World!");
</script>
```

3.1.2 In Separate Files

This is a more advantageous method of including JS in a webpage as you can use the same JS across multiple pages; HTML documents are less cluttered and easier for the developer to read and the webpage may be faster to load as the linked files can be cached. You include the JS file in the same way that you would include a CSS file as seen below

```
<head>
<script src="my_scripts.js"></script>
</head>
```

3.2 Running JavaScript code

Using the onclick attribute of HTML elements, it is possible to trigger JS functions. For example, the HTML below

```
<button onclick="say_hello()">Press me!</button>
```

will run the function below

```
function say_hello() {
  alert ("Hello World!");
}
```

when pressed.

3.3 Syntax

For the A-Level exam, it is only required to know the syntax for the basic programming constructs. Anything further will be introduced in the question. JavaScript lines end with a semi colon `;` .

3.3.1 Comments

These are started with a double slash - `//`

3.3.2 Variables

These are declared with the keyword var then the variable name then the value can be assigned to it. `var age = 4;` JavaScript is a dynamic typed language, this means all the different types of data types are declared using the keyword `var`. To change the value in the variable is like in any other programming language
`age = 6;`

3.3.3 Selection

JavaScript has if, else and else if statements which can be used to control the flow of the program. An example is seen below.

```
if (age >= 18) {
  alert("You are allowed to drink beer");
} else if (age > 16) {
  alert("Nice try. You are not allowed to drink beer.");
} else {
  alert("Here is some lemonade.");
}
```

The code doesn't have to be indented but this can make it more readable whereas the brackets and curly brackets are needed.

3.3.4 Iteration

For Loops

These are written much the same as in other languages. Notice how the count control variable, `i`, is declared within the loop and that `i++` can be used to increment `i`.

```
for (var i = 0; i < 5; i++) {
  document.write("Woohoo");
}
```

While Loops

These are written using the following syntax.

```
var i = 0;
while (i < 5) {
  document.write("Woohoo");
  i++;
}
```

3.3.5 Arrays

These are declared in the following way

```
var colours = ["red", "orange", "yellow"];
```

The arrays are based on a zero-based index and do not have fixed size. The length of an array can be obtained by the following

```
colours.length
```

Notice that the code above does *not* require brackets after the method name.

3.3.6 Subroutines

All subroutines in JavaScript are called functions, they have the option to return a value but don't have to. They are declared with the function keyword and look like the following

```
function multiply(num1, num2){
  // multiplies together two numbers
  var total = num1 * num2;
  return total;
}
```

Parameters can be passed into functions and if there is a value to be returned it can be done so using the `return` keyword.

3.3.7 Output Methods

There are a number of different ways JavaScript can be outputted to the user.

Alert Boxes

This generates a pop-up box within the browser displaying the specified text. Usually, the box will have a single OK button which lets you close the box.

```
<script>
alert("Hello!");
</script>
```

Document Write

This outputs the contents of the brackets at the point at which the `<script>` tag appears in the webpage. This means if you have the `<script>` tag at the top of the page, then the output will be at the top of the page.

```
<script>
var result = 2 + 3
document.write(result);
</script>
```

You can also include HTML tags to format the output.

```
<script>
var result = 2 + 3
document.write("<h1>",result,"</h1>");
</script>
```

Changing The Value Of An Element

This allows you to directly change the value inside a HTML element. This works by using the id of the HTML element to set the contents of the tags within the JS code. This can be seen in the following example.

```
<h1 id="example">Page heading</h1>
<script>
chosen_element = document.getElementById("example");
chosen_element.innerHTML = "Hello World";
</script>
```

3.3.8 Input Methods

HTML forms are widely used across all webpages. The following example shows the HTML for an entire webpage which contains JS in the `<head>` tag and a form in the `<body>` tag.

```
<!DOCTYPE html>
<html>
  <head>
    <script>
        // Define a function which will be called when the button is clicked
        function greeting(){
          // Retrieve the value of the 'firstname' text box
          alert("Hello " + document.getElementById("firstname").value);
          }
    </script>
   </head>
<body>
  <form>
    <p>What is your name?</p>
    <input type="text" id="firstname">
    <input type="submit" onclick="greeting()">
  </form>
</body>
</html>
```

This works by when the button is pressed, the `greeting()` function is run. This displays an alert box with the text entered by the user in it.

Chapter 4

References

- W3Schools HTML Tutorial -
  https://www.w3schools.com/html/default.asp
- W3Schools CSS Tutorial -
  https://www.w3schools.com/css/default.asp
- Isaac Computer Science JavaScript Notes -
  https://isaaccomputerscience.org/concepts/prog_gui_javascript
- Figure 2.1 -
  https://sabe.io/classes/css/css-box-model-padding-border-margin