

Introduction to using HTML to design webpages

#HTML is the script that web pages are written in. It describes the content and structure of a web page so that a browser is able to interpret and render the page for the viewer.

It is usually used in conjunction with **CSS (Cascade Style Sheets)** which dictate the style and format of a web page opposed to its content.

HTML is written in angle brackets that enclose tags, often in the opening and closing pairs.

We use software like notepad or Adobe Dreamweaver to write in HTML as shown below. This is then saved as .html to be interpreted by the web browser as a web page. HTTP is used to render HTML in the web browser as a web page.

Key

1. Peach = comments
2. Blue = generic tags/open tags
3. Yellow = text (not code specific)
4. Green = closing a tag

<! Doctype HTML>

<! comments not included -->

<html>

<head>

<link>

<! This is used to link to an external CSS file>

<title> ENTER TEXT </title>

<! This is called the header, we place the name to appear in the window header or browser tab. A title tag is needed to add this (NOTE: this title only appears as the name of the page in the tab). The title may also appear in any script that may enhance your page content>

</head>

<body>

<! The body contains the main HTML content of the page, defining text, images, hyperlinks. >

<h1> </h1>

<h2> </h2>

<h3> </h3>

<! Headlines can be inserted with h1 (1 to 6, 6 being less important headlines i.e. smaller text size) >

<div id="Insert a unique id here for reference" >

<! The div tag facilitates division of the page into separate areas, each of which may be referred to uniquely by name, and styled differently using CSS. >

</div>

<p>

This is used to insert a paragraph of text. Simply write it between the open and close tags of p.

 To get bold text we use the strong tag

 The em tag is used to create "emphasis" by italicising a word

</p>

<! The anchor tag <a> needs to be used with href as it defines a hyperlink with location parameter. Href gives this location. This can be used universally to create a hyperlink to another page; for example, the img tag can be placed within the href tag so the image acts as link to another page. Alternatively we can use text to act as the click on for the hyperlink: Enter text here or instead <h1> Enter text as a heading </h1> >

<! We would probably put a <h2> tag before the list so the list looks like it has a title >

 <! An ordered/numbered list >

 Defines an individual list item. Simply place text here

 <! An unordered / bullet pointed list >

 Defines an individual list item. Simply place text here

 <! The break tag is the only tag we have met that does not require closing. It creates a new line >

<form action="/action_page.php" >

<! A form is useless unless some kind of processing takes place after the form is submitted. The action attribute is used to inform the browser what page (or script) to call once the "submit" button is pressed >

First name: <input type="text" name="fname" >
 <! textbox>

Last name: <input type="text" name="lname" >
 <! textbox>

Submit: <input type="submit" value="Submit" >
 <! Button>

<! A submit button sends the information in the form to another page>

</form>

<script>

<! The <script> tag is used to define a client-side script (JavaScript). This element either contains scripting statements or it points to an external script file through the src attribute. Common uses for JavaScript are image manipulation, form validation, and dynamic changes of content. >

</script>

</body>

</html>

Summary of key tags

HTML Tag	Definition
<html>	All code enclosed within these tags will be interpreted as HTML
<body>	Defines the content in the main browser content area
<head>	Defines the browser tab or window area
<title>	Defines the text that appears with the tab or window heading area
<h1> , <h2>, <h3>	Heading styles in decreasing sizes
<p>	A paragraph separated with a line space above and below
	Self-closing image tag with parameters:
<a>	Anchor tag defining a hyperlink with location parameter: Link text
 , 	Defines an ordered (numbered) or unordered (bulleted) list.
	Defines an individual list item
<div>	Facilitates the division of a page into separate areas, each of which may be referred to uniquely by name/id and styled differently with CSS

The alt attribute in an tag is used to:

- Give a description of an image on screen if the image does not appear
- Allow text to speech readers to describe images to users with visual impairment
- Allow search engines to index web page content (can return images when the text in an alt tag is searched for in a search engine).

CSS – Cascading Style Sheets

CSS is a scripting language used to describe the layout and styles of a webpage. Styles can be applied to existing html elements such as <h1> <p> or <div> .

This is a different language to HTML but is commonly embedded within HTML. To make HTML neater and more concise, we can alternatively link to a CSS file.

1. An external styling sheet

<! The link typically goes at the top of our HTML page inside the head section >

```
<link rel="stylesheet" type="text/css" href="File path .css" >
```

<! If the CSS file is in the same directory as the HTML file then you can just type **name.css** for the href >

An external style sheet can be written in any text editor. The file should not contain any HTML tags. The style sheet file must be saved with a .css extension.

This is the most common technique and it operated on the basis of defining styles in an external file. Linking to an external style sheet has the added advantage that multiple HTML pages within the same website can link to the same style sheet so that the formatting can be applied consistently without the need for duplicate CSS styles.

We use id's or selectors to link divisions or elements to a CSS reference in the CSS file

Identifier and class selectors are named hooks onto which you can hang styles. You can then apply these grouped styles into an HTML element such as a <div> element by adding the class or id name as a parameter. E.g.

```
<div id =“page”>
```

The styles for the id selector called page are listed within curly brackets within the CSS document.

```
#page { max-width: 800px;
        margin: 20px auto;
        padding: 30px;
        background-color: #cc6633;
    }
```

Identifiers are defined with a hashtag (#) preceding the id name, e.g. #header
Identifiers must be unique to every web page.

More generic references can be made in the external CSS file too. For example,

```
body {                                <! Styles the entire section of the body division >
    background-color: lightblue;       <! Sets background colour >
}

h1 {                                   <! Styles all h1 text >
    color: navy;
```

```
margin-left: 20px;    <! Alignment / margin >
}
```

Classes work in a similar way to an identifier but use a full stop as a prefix to the class name e.g. `.list` .
Classes can be used multiple times on the same webpage. E.g.

```
<div class = "list">
    <ol> <! An ordered/numbered list >
        <li> Defines an individual list item. Simply place text here </li>
    </ol>
</div>
```

Note: we could use the same `<div class = "list">` later on in the body but we could not do this with an id selector

And in the CSS file, all `class = list` will be styled as....

```
.list
{
    Color: #ffcc66;    <! A hexadecimal colour value for all items in the list >
}
```

2. Internal CSS

CSS can also be inserted directly into the HTML document `<head>` as internal or embedded CSS between its own `<style>` `</style>` tags. E.g.

```
<head>
<style>
body {
    background-color: linen;
}
```

```
h1 {
    color: maroon;
    margin-left: 40px;
}
```

```
</style>
```

```
</head>    <! We can place the same script as shown in a CSS file within these style tags. However, this is
not neat and it is best to separate content from style. >
```

3. Inline CSS

When CSS is inserted directly into the body of a html document. This is commonly used to make one-off style adjustments that are unlikely to affect other parts of the document.

e.g

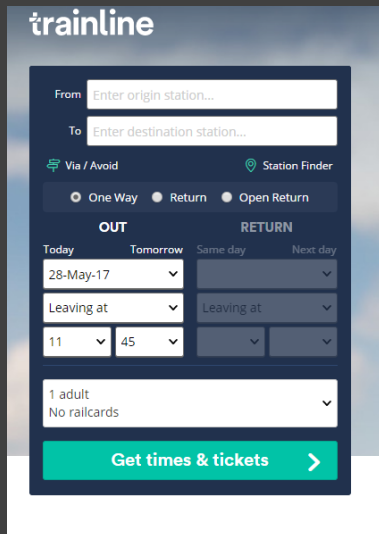
```
<h1 style="color:blue;margin-left:30px;"> This is a heading </h1>
```

CSS script language to be familiar with:

background-color: x;	Sets the background colour for different elements You can use named colours, hexadecimal or RGB. E.g. yellow or #00ff00 or rgb(255,0,255)
border-color: x;	Sets the colour of the surrounding border. This can be used around the text, the body, headings, lists etc. You can add between 1 and 4 colours. E.g. border-colour: yellow light blue red green; Each one will change the individual segments i.e. left, right, top, bottom.
border-style: x;	The style of the four borders. The options are: dotted, solid, dashed or solid double
border-width: x;	The width of the four borders. Enter a value such as 15px
color: x;	Sets the colour of an element .
font-family: x;	This specifies the font of a paragraph or piece of text. E.g. <pre>p { font-family: "Times New Roman", Georgia, Serif; }</pre>
font-size: x;	The size of the font. The default is medium. You could put small, large, x-large, xx-large or you could put 200% which would set the font to 200% of the parent element's font size. You could also put smaller or larger. This would make the font size smaller or larger than that of the parent element's font size. E.g. <pre>p { font-size: smaller; }</pre>
height: x;	Specifies the dimensions of an element e.g. <pre>.myparagraphs { height: 100px; width: 100px; }</pre>
width: x;	

Web forms

Web forms enable websites to collect user input data and selections. Input types include text boxes, checkboxes and radio buttons. For example,

A screenshot of the trainline website's search interface. It features a dark blue header with the 'trainline' logo. Below the logo are two input fields for 'From' and 'To', both with placeholder text 'Enter origin/destination station...'. There are also icons for 'Via / Avoid' and 'Station Finder'. Below these are radio buttons for 'One Way', 'Return', and 'Open Return'. The interface is split into 'OUT' and 'RETURN' sections. The 'OUT' section has a date selector (Today, Tomorrow, Same day, Next day) with '28-May-17' selected, and a 'Leaving at' dropdown menu with '11' and '45' as options. The 'RETURN' section has a similar date selector and 'Leaving at' dropdown. At the bottom, there is a dropdown for '1 adult' and 'No railcards', and a large green button labeled 'Get times & tickets' with a right-pointing arrow.

Inputs can be validated and submitted to the website owner's database or processed as part of a search query to find, for example, train times or your nearest shop branch when you enter a postcode.

Creating web forms using basic HTML form controls

A simple, unformatted web form that uses basic textboxes and buttons to submit data can be created easily. This is, however, functionless until JavaScript is added which is used to add behaviours to a web page or create active web forms.

<h1> Register form </h1>

```
<form action="process.php" method = "post">
```

```
  <label> Enter your email to register: </label> <br>
```

```
<! label>
```

```
  <input type="text"      id="email"      value="" size="40" /> <br>
```

```
<! textbox>
```

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

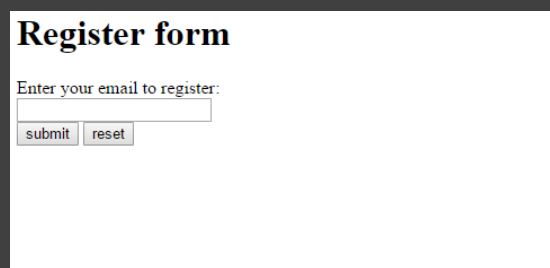
```
<! Button>
```

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

```
<! Button>
```

```
  <! A submit button sends the information in the form to another page>
```

```
</form>
```

A rendered HTML form titled 'Register form'. It has a white background and a dark blue border. The title is in a bold, black serif font. Below the title is a label 'Enter your email to register:' in a dark blue sans-serif font. Underneath the label is a white text input box with a dark blue border. At the bottom of the form are two buttons: a white 'submit' button and a grey 'reset' button, both with dark blue text and borders.

Form handling with reset and submit and reset actions

The button type is specified as an attribute of the button, e.g. type="reset". This will provide some basic functionality in the case of the reset button which clears the form data.

A submit button type will send data to a **form handler** specified in the **action attribute** of the <form> tag. The form handler on the server will then process the form data – in this case, an email address.

JavaScript

JavaScript is a language that uses all the programming constructs that are familiar with languages such as Python and VB. It should not be confused with the language Java.

JavaScript is commonly used to add interactivity to websites, including:

- The manipulation of page objects
- Animations
- Navigation tools
- Form validation

JavaScript is an interpreted language rather than a compiled language.

JavaScript **needs to be translated** into the **object code** for whichever computer the browser is running on, and will be translated by the **interpreter as the web page is displayed**. An interpreter in the browser reads the Javascript code, interprets each line and runs it.

Compilers produce object code that is **specific to a particular type of processor**.

Some of the latest browsers, however, use “**just-in-time**” **compilation** which compiles JavaScript into **executable bytecode** just before execution. This method is known as **compilation on-the-fly**. This significantly increases speeds but it has to be constantly checking for the types of variables and whether functions have been redefined all the time.

Input

JavaScript can be used to process input data on the client’s computer. This may change the local page interactively or post data to a server. The advantages of processing input data before it is posted to a server are that:

1. The local computer validates erroneous data before it is submitted to a database
2. A busy server is relieved from having to process everything itself.

Output

JavaScript can reference and interact with HTML elements to edit, style or move them. For example, a validation script may change a postcode input label to become red if a user has entered invalid data.

```
Document.getElementById(“postcode”).style.color=“red”;
```

Using JavaScript to control webpage functions

JavaScript can be used to make a simulated Captcha form as shown.

The HTML form elements are given ids in order for JavaScript to reference them. Buttons are given `onClick` attributes in order to execute JavaScript functions when they are pressed. Their type has also changed to become “`button`” rather than `submit` or `reset` actions.

There are a number of mouse events that are used in HTML code to trigger the execution of JavaScript functions.

1. `onmouseout` - occurs when the cursor is moved away from an element such as a button or a heading.
2. `onmouseover` - occurs when the cursor is hovered over an element such as a button or a heading.
3. `onmousedown` - occurs when the user presses a mouse button over an element.
4. `onmouseup` - occurs when the user releases the mouse button over an element.
5. `onclick` - occurs when the user clicks on an element such as a button or hyperlink.
6. `onmousemove` - occurs when the cursor is moving over an element such as an image or menu.

Other simple JavaScript elements:

Creating a variable

```
Var yourName
```

This creates a variable called 'yourName'.

Receiving input from the user

```
Prompt("What is your name");
```

Printing text onto a webpage

```
Document.writeln("Hello, this is a web page");
```

Input

Input will be taken in by reading values from a form. *NB learners will not be expected to memorise the method for doing this as focus will be on what they do with that input once it is received.*

Output

By changing the contents of an HTML element

```
chosenElement = document.getElementById("example");  
chosenElement.innerHTML = "Hello World";
```

By writing directly to the document

```
document.write("Hello World");
```

By using an alert box

```
alert("Hello World");
```

Should any other aspects of JavaScript be used they will be introduced and explained in the question.