N5

National 5

Web Design & Development

Learning Intentions

The following provides details of skills, knowledge and understanding that can be covered in the course assignment and the exam.

Analysis – Page 3-4

Identify the end-user and functional requirements of a website problem that relates to the design and implementation at this level.

Design – Page 5- 14

Describe and exemplify the website structure with a home page, a maximum of four linked multimedia pages, and any necessary external links.

Describe, exemplify and implement, taking into account end-user requirements, effective user-interface design (visual layout and readability) using wire-framing:

- navigational links
- consistency across multiple pages
- relative vertical positioning of the media displayed
- file formats of the media (text, graphics, video, and audio)

Describe and identify the implications for individuals and businesses of the Copyright, Designs and Patents Act 1988 relating to:

- web content (text, graphics, video, and audio)

Compare a range of standard file formats:

- audio standard file formats WAV and MP3 in terms of compression, quality, and file size
- bit-mapped graphic standard file formats JPEG, GIF, and PNG in terms of compression, animation, transparency, and colour depth

Describe the factors affecting file size and quality, relating to resolution, colour depth, and sampling rate.

Describe the need for compression.

Describe, exemplify and implement prototyping (low-fidelity) from wireframe design at this level.

Implementation – HTML Page 15 -21

Describe, exemplify and implement HTML code:

- HTML
- head
- title
- body
- heading
- paragraph
- DIV
- link
- anchor
- IMG
- audio
- video
- lists — ol, ul and li

Describe and implement hyperlinks (internal and external), relative and absolute addressing. Read and explain code that makes use of the above HTML.

Implementation – CSS Page 22 - 26

Describe, exemplify and implement internal and external Cascading Style Sheets (CSS): selectors, classes and IDs and properties

— text:

- `font (family, size)`
- `color`
- `alignment`

— background colour

Read and explain code that makes use of the above CSS.

Implementation – JavaScript Page 27 - 28

Describe and identify Javascript coding related to mouse events:

- Onmouseover
- Onmouseout

Testing - Page 29

Describe and exemplify testing:

- matches user-interface design
- links and navigation work correctly
- media (such as text, graphics, and video) display correctly
- consistency

Evaluation - Page 29

Evaluate solution in terms of:

- fitness for purpose

Analysis

In the analysis stage of developing a website it is important to consider the functional requirements of the website. The functional requirements specify exactly what the website should do and should take into account the purpose, features, functions and users of the website.

Purpose

The purpose of an information system is to contain information and present it in a useful way to the user.

For example, the purpose of a school information system is to take care of all the administrative tasks in a school whereas the purpose of an online shop is to sell items to their customers.

Features

The main features of an information system are:

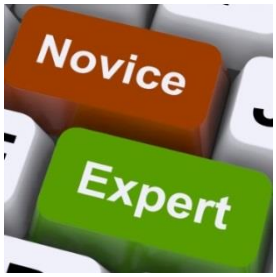
- **Hardware** – stores the data and provides backup facilities
- **Software** - the programs that the computer runs. The software can process, search and sort the data and it may be a package such as Microsoft Access.
- **Storage –** All information systems need to have storage and this may be hard and solid state disks or on the cloud.
- **Networks/Connectivity –** Allows for information systems to be connected to allow for data transfer or communication.

Functions

The main functions of an information system are:

- **Collecting –** Taking in and gathering together information.
- **Organising –** Managing the information by using software such as a database or web page application.
- **Storing –** Saving and backing up the information so that it can be used again.
- **Processing –** Performing operations on the information including searching, sorting and carrying out calculations.
- **Outputting** – Displaying or communicating the information to the user in a suitable format.

Types of Users



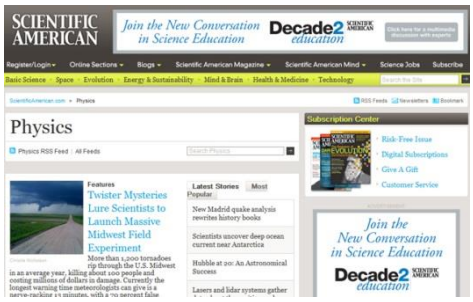
Experts – A person who is familiar with the features and functions of the system and can use it with ease. They do not require instructions and can navigate their way around the system.

Novice – A person who is unfamiliar with the features and functions and requires support or instructions to allow them to use the system.

Age Range of Users

Typical users (target audiences) of information systems can include young children, teenagers and adults.

When designing the user interface of an information system, developers need to take the target audience into consideration.



Design

Webpage and Website

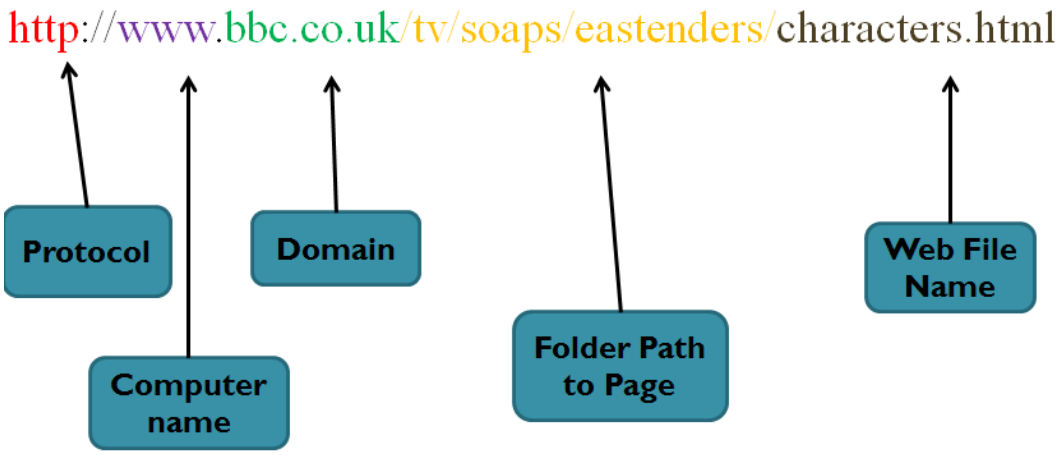
A **webpage** is a single document containing information accessed via the World Wide Web.

A **website** is a set of related **web pages** owned by a single individual, company or organisation.

Each **webpage** on the World Wide Web is identified through its unique address or URL.

URL

A **Uniform Resource Locator** is used to uniquely identify every web page on the World Wide Web.



If you only type in the Protocol, Computer Name and Domain, you will normally be directed to a **home page**.

Web Browsers

A browser is a **program** that is used to display and navigate between webpages.

There are many different web browsers available that all do the same job.

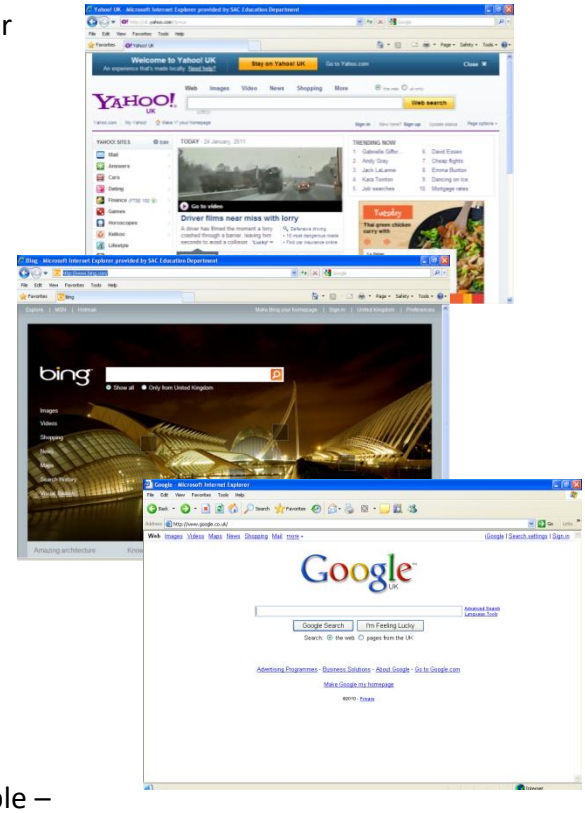
All browsers translate **HTML** tags in order to display web pages correctly, so no matter the browser used webpages should appear the same.

Web browsers also contain many features to make life easier such as:

- **Bookmarks** to allow users to save and organise a library of interesting and frequently used pages
- **History** to keep a record of pages visited
- **Tabs** to allow several pages to be opened within a single browser session
- **Plug-ins** which improve the functionality of the browser to allow it to display different types of content.

Search Engines

A search engine is a web page that is used to find other web pages.



By entering **keywords** related to the topic you are interested in, the search engine will find and display **links** to **relevant web sites**.

Again, there are many different search engines available – some better than others.

Hyperlinks

Hyperlinks are used for linking web pages together.

A hyperlink can be added to text or images and must be **clicked** to take the user to another web page.

When the mouse pointer moves over a hyperlink on a page, it will change to a **pointing finger**.

Internal and External Hyperlinks

An internal hyperlink will take you to **another page** within the **same** website.

An external hyperlink will open a page from a **different** website altogether.

Relative and Absolute Addresses

Internal hyperlinks use **relative addresses**. A relative address only needs to include the path and page name, not the full domain.

/music/katy-perry.html

External hyperlinks must use **absolute addresses**. An absolute address **must** include the full domain name as well as the path and page name.

http://www.mysite.com/music/katy-perry.html

User Interface

The user interface is where interaction between humans and machines occurs.

Visual Layout

User interface design should consider the following:

- Pages should look **appealing** to all users
- Avoid information **overload**
- Ensure a good **balance** between graphics and text
- Pages contain **Accessibility** features
- Appropriate for target audience
- No clutter
- Good use of colour and white space

Consistency

All of the site pages should have similar appearance such as:

- Text style, font, colour
- Use of colour
- Page balance
- Navigation features (button size/style)

Readability

The design should consider how easy it will be for the user to read and understood text and instructions.

Is the **text large enough** to read? Is it easy to pick out the required information?

Navigation

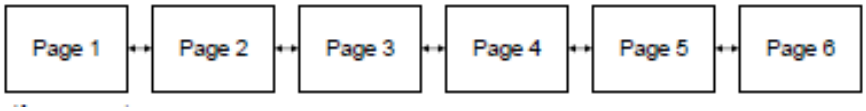
Well-designed navigation on a website makes it easy for users to find their way around – especially if the site contains a large number of pages.

Some useful methods of improving site navigation include:

- Navigation bars
- Back/Forward Buttons
- Home button
- Graphic and text links
- Highlighted selection
- Consistency
- Site map
- Breadcrumbs

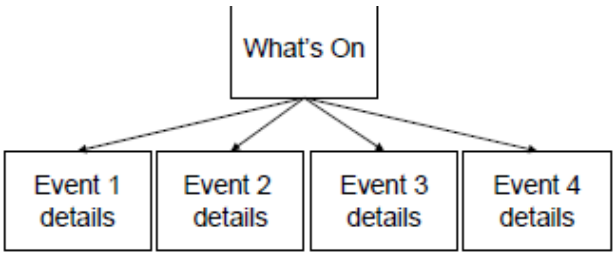
Linear Navigation

Each page is visited in order, one after the other.



Hierarchical Navigation

Hyperlinks allow pages to be visited in any order

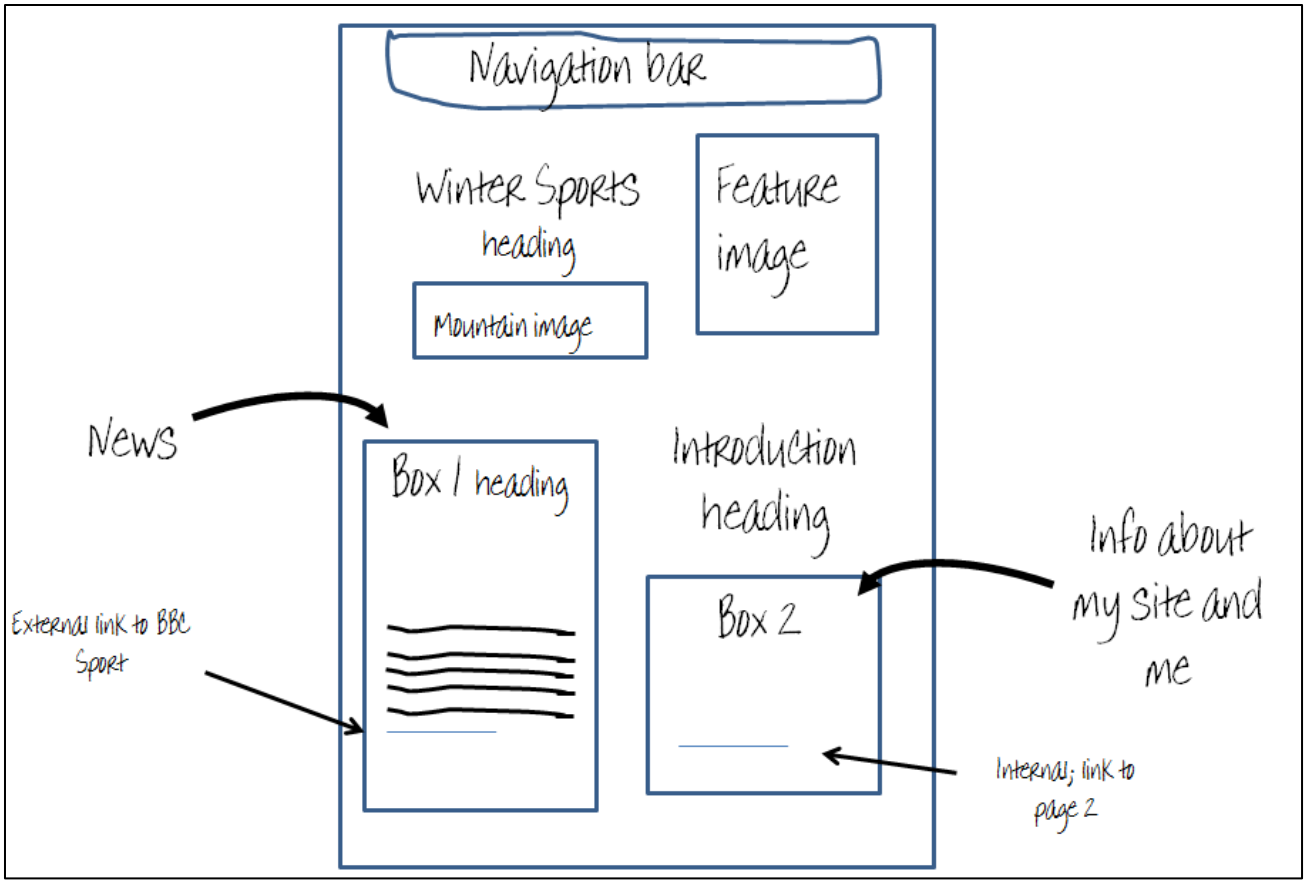


Wireframe

The user-interface planning can be illustrated using **wireframes**. A separate wireframe is needed for each page on a website. Each wireframe indicates the intended layout of the page and shows the position of:

- all text elements on the page
- any media elements (images, audio clips and video clips)
- elements that allow the user to interact with the page
- intended position and type of all hyperlinks on the page

An example is shown below:



Low-fidelity prototyping

A prototype is used to show the intended user interface for any software product.

Once developed, a prototype forms a critical component of testing, along with a detailed description of typical end users.

Low-fidelity prototypes are paper-based. They can be created quickly and give potential end users of the finished product an indication of how the product will look and feel as they interact with it.

During usability testing, a selection of end users (or testers who adopt the personas) are asked to complete the task described in each of the test cases using the prototype.

Developers will be on hand to 'change screen' (show a new page of the interface) or 'update the content of the screen' (for example, replace page content, show the results of a calculation or perform an interactivity) as the user interacts with the widgets used on the prototype. By listening to user feedback and observing any difficulties users have as they perform the specified tasks, the developers can make changes and improvements to the user interface at an early stage in the development of the software product.

Prototypes should be based on the layout indicated in the wireframes. However, unlike wireframes which are created to ensure consistency and share details with members of the development team, the intended audience of prototypes is end users of the finished product.

For this reason, a prototype should show more details for the content and the screen widgets that will be used to perform tasks.

Creating low-fidelity prototypes

Low-fidelity prototypes can be created in a number of ways. For example:

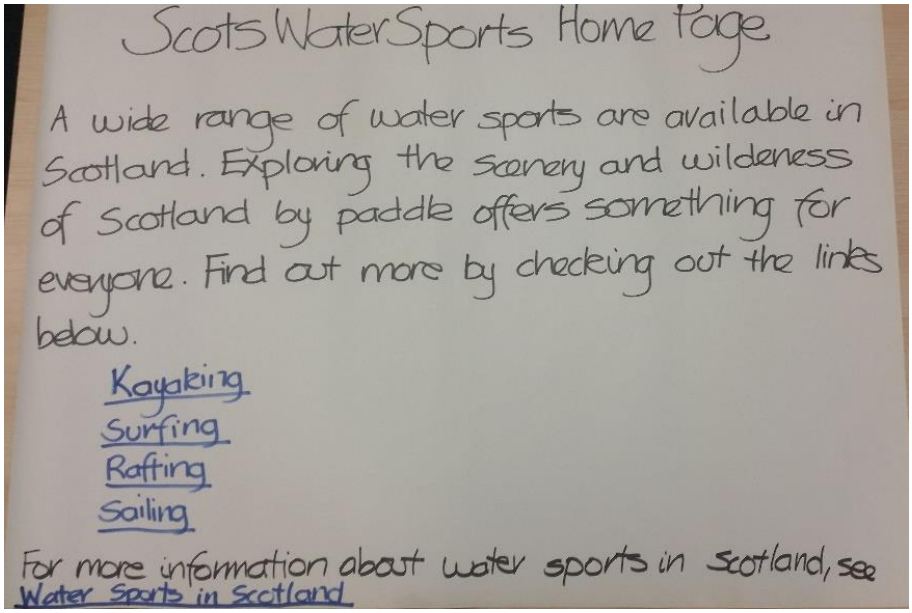
- A simple hand-drawn sketch of the proposed interface is one of the easiest and cheapest ways of creating a paper-based prototype. Coloured pencils, felt pens and markers can be used to add colour; quick hand drawings of images and widget icons will give end users a good idea of what is intended. Any size of paper can be used but flip-chart paper can be easier for candidates to handle and facilitates collaborative group work.

- 'Pencil' software by 'Evolus' provides free prototyping tools. A number of in-built templates are provided and additional Android and iOS templates are available for download. 'Pencil' templates can be used to create realistic-looking interfaces that can be exported as PNG files and printed to generate prototypes for usability testing.

- Graphics packages such as ‘Paint’ can be used to create the intended layout using tools provided in the toolbox.

Example
A new website for ScotsWaterSport is being developed. The wireframes for each of the pages have been created and prototyping of the Home Page and Kayaking Page of the ScotsWaterSports website can now take place.

Version 1
These prototypes were created with marker pens and flip-chart paper.



Version 2
These prototypes were created using ‘Pencil’ templates.

Copyright, Designs and Patents Act

Copyright, Designs and Patents Act main purpose is to protect inventors, authors, artists, composers, film makers, designers etc. (the copyright holder) if someone tries to copy or steal their work.

Passing off someone else's work as your own is known as **plagiarism**.

People can be prosecuted under the Copyright, Designs and Patents Act for:

- **Downloading or copying** music, movies, software without paying or getting permission.

Piracy is the illegal copying, downloading or usage of someone else's work.

The act protects a wide range of work including computer based work such as software and web content.

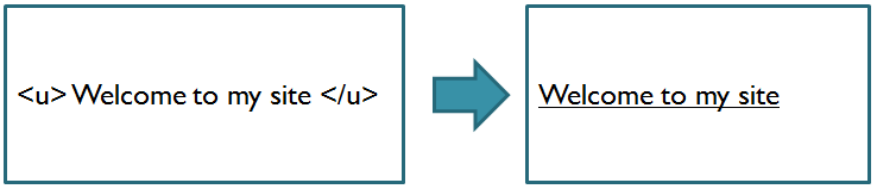
Copying images or photographs or sections of text from a website without permission and pretending it is your own work is not allowed and is covered but the act.

Implementation - HTML

HTML stands for **Hyper Text Markup Language.** It is the language used to **create web pages**. HTML uses **tags** which surround plain text. The text between the tags is then **translated by the web browser** and displayed as **formatted text**.

HTML tags normally come in pairs e.g. and .

The first tag is the **opening** tag and the second tag is the **closing** tag. Closing tags always have a forward slash inserted before the tag name.



HTML code will always start and end with a <Html> tag and within these tags there will always be a <head> tag and a <body> tag. HTML code is laid out as shown below:

```
<html>
<head>
        <title>
                My Example Website
        </title>
</head>
<body>
        <p> My first website </p>
        <p> <b> By J Bloggs</b> </p>
        <p> <i> from Glasgow</i> </p>
        <p> <big> <u> Ayr </u> </big> </p>
</body>
</html>
```

The main content of the page is included within the body section.

<html>

The page always begins with the start tag of the **<html>** element and always terminates with the end tag of the **</html>** element as shown below.

The html element basically tells the browser that this is an HTML document. All other element tags are 'nested' within the start and end html tags.

```
<html>

…web page…

</html>
```

<head>

This tag is used to define information about the web page such as the title. The head section begins with the **<head>** opening tag and terminates with the **</head>** closing tag.

<body>

The body tag will contain all the contents of the web page such as text, hyperlinks, images etc. All content should go inside the opening tag <body> and closing tag </body>.

```
<html>

<head>

</head>


<body>

</body>

</html>
```

Head Section

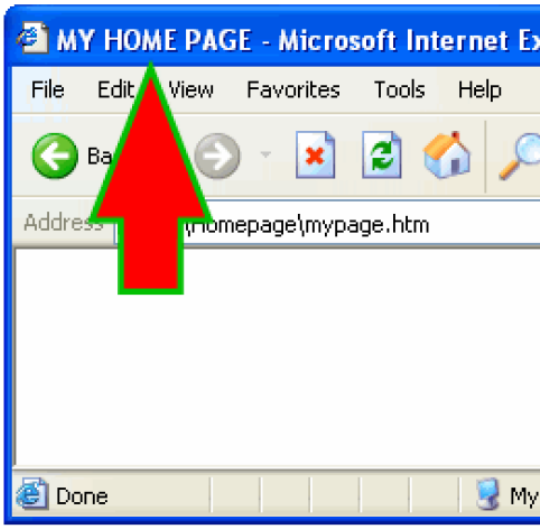
The Head can contain:

- Title
- Link

<title>

Web pages usually have a title that appears in the title bar or tabs across the very top of the web page. All text appearing after the **<title>** start tag and before the**</title>** end tag will be displayed as your web page title.

```
<html>

<head>

     <title> My Home Page </title>

</head>

<body>

</body>

</html>
```



Important note

No other elements can be used inside the title element. (e.g. you cannot make the title contents bold)

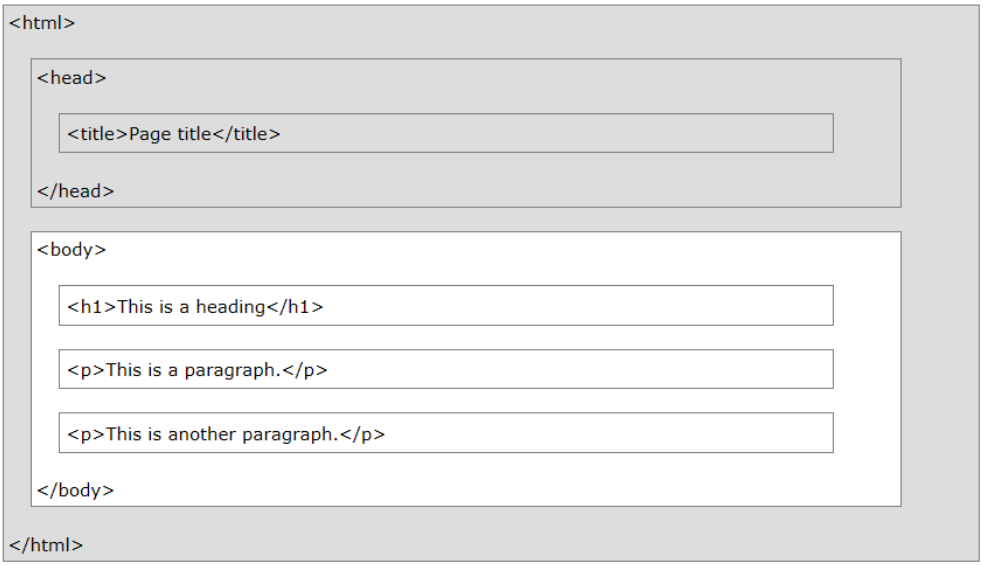
<title> My Home Page </title>

<link>

Used to link to other documents, most often an **external style sheet** that provides instructions about how to style the various elements on the page but also used to link to JavaScript files that add interactivity and/or functionality to the web page.

Body Section

This is where the **content** of the page is contained. Everything that you see in the browser window is contained inside this element including paragraphs, lists, images, tables and more.



HTML **elements** are written with a start tag, an end tag and with the content in between:

<tagname> *content* **</tagname>**

Start tag	Element content	End tag
<h1>	My First Heading	</h1>
	My first paragraph.	

HTML **attributes** provide additional information about an element. They are always specified in the start tag and come in name/value pairs.

<tagname ***attributename***=**"*attributevalue*"**> *content* </tagname>

```
<p id="detail">Here is a paragraph with an id</p>
<img src="w3schools.jpg" alt="W3Schools.com" width="104" height="142">
<a href="http://www.belmont.com">Click here</a>
```

<h1> <h2> <h3> <h4> <h5> <h6>

These tags are used to define a heading on your web page. H1 is the largest size of heading with H6 being the smallest. All heading tags must be closed used the corresponding closing tag. E.g.<h3> Welcome to my website </h3>

** **

This tag is used to define a paragraph. This is some text in a paragraph.

** **

This tag is used to define a section (division) within your web page. It is often used alongside CSS to change the layout of a web page.

<a>

This tag is the anchor tag and is used to create an internal hyperlink or external hyperlink. This tag is used with the href attribute to allow you to indicate the links destination.

External hyperlink: BBC News

Internal hyperlink: Page 2

<img>

This tag is used to define an image. The <img> tag has two attributes: src and alt. src is used to specify the URL of the image and alt specifies an alternative text for the image.

<img src = “cat.jpeg” alt=”Cat”>

<audio>

This tag defines sound items to be included on your webpage.

```
<audio>
    <source src="horse.mp3" type="audio/mp3">
</audio>
```

<video>

This tag defines video items to be included on your webpage.

```
<video>
    <source src="movie.mp4" type="video/mp4">
</video>
```

Lists

<li>

This tag is used to define an item within a list.

<ol>

This tag is used to define an ordered list. The list can be ordered numerically or alphabetically.

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

1. Coffee
2. Tea
3. Milk

<ul>

This tag is used to define an unordered bulleted list.

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

- Coffee
- Tea
- Milk

Several other html tags exist. A complete html reference guide can be found at

https://www.w3schools.com/tags/default.asp

Below is an example of a HTML page:

```
<html>
    <head>
        <title>My first website</title>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">


    </head>
    <body>
        <h1> Welcome to my website </h1>

        <p> Here is a paragraph of information.</p>

        <p> Here is another paragraph of information </p>

        <a href ="http://www.bbc.co.uk"> This is a hyperlink </a>

        <br>

        <img src ="animal.gif">


    </body>
</html>
```

Implementation - CSS

A Cascading Styles Sheet (CSS) is a way to style and present HTML content. Whereas HTML code defines the content of the page, a style sheet is the presentation of that content.

The use of style sheets gives developers more control over how pages are displayed.

Using CSS, designers can create style sheets that define how different elements, such as headers, paragraphs and links, appear.

These style sheets can then be applied to any web page within the site. In this way, consistency can be easily achieved.

HTML was never intended to contain tags for formatting a document. Instead, HTML was intended to define the **content** of a document.

For example, content is contained within the <h1> and <p> tags as shown below:

```
<h1> This is a heading </h1>
<p> This is a paragraph </p>
```

Notice that there is nothing in the elements above to tell the browser what colour or font to use for this content.

When **<font> tags** and **colour attributes** were added to the HTML 3.2 specification, it started a nightmare for web developers.

Development of large web sites, where fonts and colour information were added to every single page, became a long and expensive process.

To solve this problem, the World Wide Web Consortium (W3C) created CSS. In HTML 4.0, all formatting could be removed from the HTML document, and stored in a separate CSS file. Today, all browsers support CSS.

Style Sheet Rules

CSS Rules consist of a selector and a declaration.

Declarations for each selector are contained within curly brackets. Each selector can have numerous declarations separated by semi-colons.

```
h1{
    background-color:red;
    color:blue;
    text-align: center;
    font-family: sans-serif;
}
```

All **Heading 1** elements will have a red background, blue text. They will be centred and use san-serif font.

```
a {
    display:inline-block;
    text-decoration: none;
    background-color: lightblue;
    color: black;
}
```

All **Hyperlink** elements will be displayed in line with no underline (decoration). Their background will be light blue.

Any tag can be used as a selector. You will find a complete selection of CSS rules on the following website:

http://www.w3schools.com/css/default.asp

ID Selectors

ID attributes are used in HTML to allow us to distinguish between elements of the same type.

```
<p id="first"> This is the first paragraph</p>
<p id="second"> This is the second paragraph</p>
```

These elements can now have different CSS rules applied

```
#first {
    background-color:red;
    color:blue;
    text-align: center;
    font-family: sans-serif;
}
```

```
#second {
    background-color:blue;
    color:red;
    text-align: left;
}
```

Class Selectors

Class attributes are used in HTML to allow us group elements together.

```
<p class="main"> This is the first paragraph</p>
<p class="main"> This is the second paragraph</p>
```

Any element that is part of this specific class can now have different CSS rules applied.

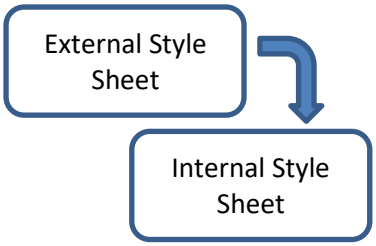
```
.main {
        background-color:red;
        color:blue;
        text-align: center;
        font-family: sans-serif;
       }
```

Types of Style Sheet

Style definitions can be applied in two ways:

- Internal Style Sheet
- External Style Sheet

Cascading refers to how styles are applied and the order of precedence.



Inline styles override internal style sheets

Internal style sheets override external style sheets.

example:

If an external style sheet defined all paragraph text to be red but an internal style sheet defined all paragraph text to be blue, the paragraph colour would be blue.

External Style Sheet

An external style sheet is ideal when the style is applied to many pages.

With an external style sheet, you can change the look of an entire Web site by changing just one file.

Each page must link to the style sheet with the <link> tag. The <link> tag goes inside the head section as shown below.

```
<head>
        <link rel="stylesheet" type="text/css" href="mystyle.css">
</head>
```

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. An example of a style sheet file is shown below.

```
h1 {color:red;}
p {margin-left:20px;}
body {background-image:url ("images/background.gif");}
```

Internal Style Sheet

An internal style sheet should be used when a single document has a unique style.
Internal styles are defined in the head section of an HTML page, inside the **<style>** element, as shown below.

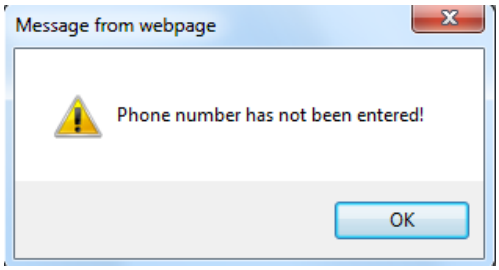
```
<head>
<style>
        h1 {color:red;}
        p {margin-left:20px;}
        body {background-image:url ("images/background.gif");}
</style>
</head>
```

Implementation - JavaScript

Javascript is a Scripting Language**.**

It is used to add **additional functionality,** interactive features and **dynamic** content to web sites.

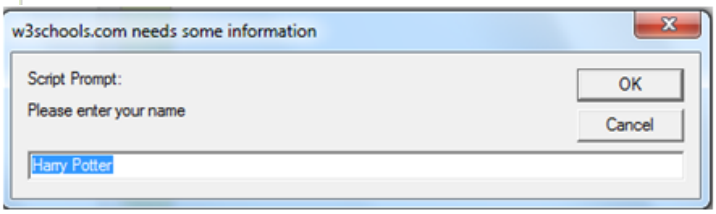
Some examples of Javascript uses are to:

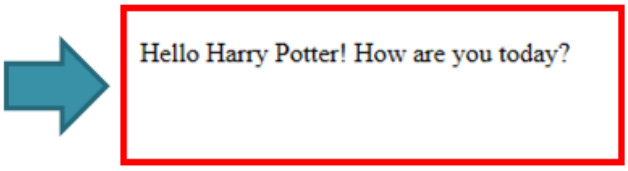


- **Validate data** entered into online forms
- Display warning and confirmation **messages**
- **Add check boxes** or **buttons** to a webpage.
- Control drop-down menus
- On screen animation such as a clock

Example: User Input

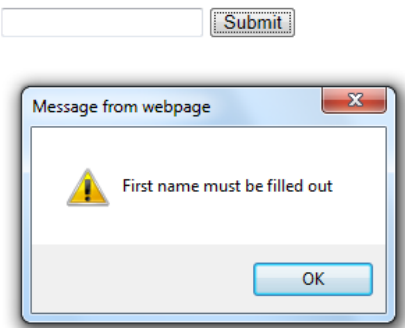
```
<!DOCTYPE html>
<html>
<body>

<p>Click the button to demonstrate the prompt box.</p>

<button onclick="myFunction()">Try it</button>

<p id="demo"></p>

<script>
function myFunction()
{
var x;

var person=prompt("Please enter your name","Harry Potter");

if (person!=null)
  {
  x="Hello " + person + "! How are you today?";
  document.getElementById("demo").innerHTML=x;
  }
}
</script>

</body>
</html>
```





Example: User Input Validation

```
<!DOCTYPE html>
<html>
<head>
<script>
function validateForm()
{
var x=document.forms["myForm"]["fname"].value;
if (x==null || x=="")
  {
  alert("First name must be filled out");
  return false;
  }
}
</script>
</head>

<body>
<form name="myForm" action="demo_form.asp" onsubmit="return validateForm
()" method="post">
First name: <input type="text" name="fname">
<input type="submit" value="Submit">
</form>
</body>

</html>
```



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

onmouseover

The onmouseover event occurs when the mouse pointer is moved onto an element.

onmouseout

The onmouseout event occurs when the mouse pointer is moved out of an element.

Examples of javascriot mouse events can be found at:

https://www.w3schools.com/js/js_events_examples.asp

Testing and Evaluation

Testing an information system involves checking that all the links and navigation work correctly and that the solution matches user interface design.

Links

Do all the internal and external hyperlinks work and take you to the correct location?

Navigation

Is the information system easy to navigate around to be able to do a certain job? Do all the navigation features take you to the correct location?

Multimedia

Images displayed on the page should be checked to make sure they are not pixelated and load properly.Sounds and Videos should be checked to make sure they play properly.

Javascript Issues

Any interactive and dynamic content on the page that has been programmed using Javascript should be tested to ensure it works correctly

Browser Compatibility

Web pages should be tested to ensure they work and are displayed correctly on a range of different browsers.

Matches Original Design – Fit for purpose

Before creating an information system in industry, there will be a specific design agreed with the client. When the information system is complete it is tested against this to ensure it is fit for purpose and does everything it should.