WEB DEsign and development

National 5 Computing Science

Alison Cowie

Portobello High School, Edinburgh

Contents

[Analysis 3](#_Toc497026730)

[Design 4](#_Toc497026731)

[Website structure 4](#_Toc497026732)

[User interface design 7](#_Toc497026733)

[Copyright, Designs and Patents Act 1998 11](#_Toc497026734)

[Factors affecting file size and quality 12](#_Toc497026735)

[The need for compression 13](#_Toc497026736)

[Comparing standard file formats 13](#_Toc497026737)

[Prototyping 14](#_Toc497026738)

[Implementation (CSS) 15](#_Toc497026739)

[Implementation (HTML) 21](#_Toc497026740)

[Relative address v Absolute address 26](#_Toc497026741)

[Implementation (JavaScript) 27](#_Toc497026742)

[Testing a website 29](#_Toc497026743)

[Evaluation 29](#_Toc497026744)

## Analysis

The development of a website has several phases. These include:

**1. Analysis**

**2. Design**

**3. Implementation**

**4. Testing**

**5. Evaluation**

For example, Harpreet is a self-employed website developer. Jake runs his own plumbing business called Perfect Plumbers and he has employed Harpreet to make a website for his business

Analysis: Harpreet and Jake meet to decide exactly what Jake wants the website to do. They will create a document called the **website specification** that lists the **end-user requirements** and the **functional requirements** of the website

Design: Harpreet will then use wire-framing techniques to **plan the** **website structure** so that it **matches the functional requirements**. The website structure will include a Home page with links to other pages. Each page is likely to reflect an aspect of Jake’s business e.g. an About page, a Contact page, a Pricing page and so on.

In addition, Harpreet will work with Jake to **plan the user interface** (what is like to interact with the website) so that altogether the website meets the **end-user requirements**.

**Iteration**:Jake will be involved in checking the design and he may ask for changes at the design phase based on what he sees and this could mean going back to the analysis phase with Harpreet. **Iteration** means ‘to do over again’. The website development process is described as **iterative** because **each stage may have to be revisited as a result of new information coming to light**

Implementation: Once the design is complete, Harpreet will **write program code** to implementthe **end-user** and **functional requirements of the website**. To start with, she will create low-fidelity **prototypes** for Jake to see. These will be interactive – rollovers, links etc. - but not show much actual detail. Once he is satisfied with how the website is likely to look and behave, Harpreet will continue to implement the design

Testing: Harpreet **will be testing the website continually** while she is working. When she is finished, Jake will be involved in testing it to make sure that it works as he specified. This may involve going back to previous phases and making changes (iteration). Once it is complete, Harpreet will arrange for the website to be hosted as part of the World Wide Web so that it can be accessed via the Internet

Evaluation: This is to decide whether the website is **fit for purpose**. That means: Does it meet its **end-user** and **functional requirements**?

## Design

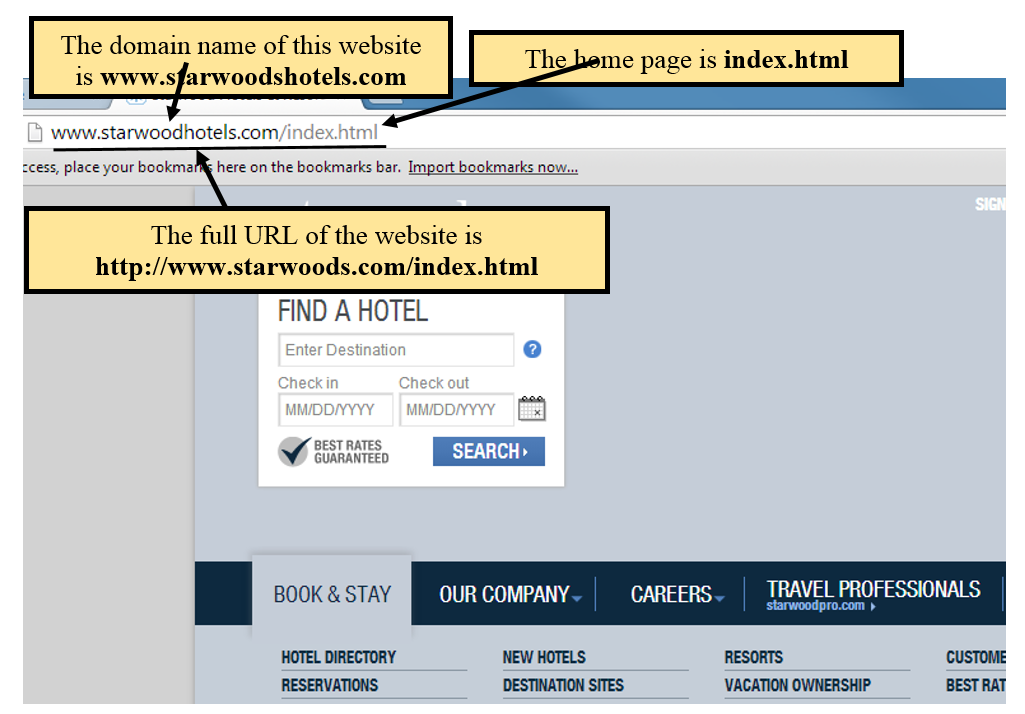
### Website structure

A **website** organises information into **web pages**. A website is a group of web pages usually produced by the same people about the same topic

Websites always have a **home** or **index** page that links to pages (sub-topics) on the website

Every web page on the World Wide Web has a unique address, shown in the address bar of the browser. These addresses or **URLs** (Uniform Resource Locators) can also tell you about the structure of the website

**Example:** Starwoods Hotels Website



When designing a website, it’s useful to draw a **map** of the website showing how all the pages are related and linked to each other. This can be achieved by sketching out the site structure. A partial map of the Starwoods Hotel website may look like this:

This structure is then reflected in the URL of each page, For example, if a user visits the page called alllist.html, then the URL would be

<http://www.starwoodhotels.com/bookandstay/directory/hotels/alllist.html>

**protocol** **domain name** **folder path name of file**

If we break down this URL, we can see that ….

The **protocol** is http

The **server (domain name)** is [www.starwoodhotels.com](http://www.starwoodhotels.com)

The **path** to the file (e.g. the route to the file) is /bookandstay/directory/hotels/

The **folder** that is currently being accessed is hotels

The **name** of the file that is being accessed is alllist.html

Navigation within a website should be **consistent** so that it is easy to use. That usually means that the navigation menu is in the **same place** (quite often at the top of each page or down the left side) with the **same style** on each page. For example:

Navigation structures for webpages within a website can be:

**Linear**

**Hierarchical**

The BBC news website has a hierarchical navigation. Each menu item has a drop-down list of hyperlinks to other web pages.



A website may include **external links** – these are links to websites outside the domain. For example, external links may be to social media sites like Facebook or Twitter

### User interface design

When planning what a web page will look like, it is useful to sketch out a design (a wireframe) either on paper or using one of the software tools available, either application based or web based, that allow you to wireframe a website design

Wireframing tools allow you to design the **layout** of pages and many allow you to create **interactive** designs

Wireframe

Actual website



**Low-fidelity wireframes like the one shown above** are relatively quick to develop and help the designers to communicate ideas.  This type of wireframe tends to be more abstract because they often use simple images to block off space (such as rectangles with crosses through them for graphic elements, plain rectangles for text, etc.). Sometimes Latin (lorem ipsum) text is used as filler for content and labels. Interactive elements such as rollovers, drop-down menus and links will be included to illustrate how the website will work.

**Effective user-interface design**

This means giving significant thought to the **visual layout** and the **readability** of the website and will include consideration of the:

1. navigational links
2. consistency across multiple pages
3. relative vertical positioning of media
4. file formats of media that is included

#### Navigational links

A hyperlink is an element in an electronic document that links to another place in the same document or to an entirely different document. Often websites have a hyperlink on their logo which usually takes you back to the homepage. Inclusion of a **site map** and a **help menu** makes it easier for a user to find their way around a website. Navigation is made easier through well-placed hyperlinks, navigation bars and by grouping hyperlinks

Hyperlinked text is often coloured to show that it’s a hyperlink. Most websites have Navigation Bars offering hyperlinks to other sub-topics or other websites. In addition, hyperlinks can be added to elements on the page. This is often used with photos where the link takes you to a page about that photo.

An **internal hyperlink** is takes the user to a new location within the current page or a page within the current site. This is often used with menu systems at the top of a page

An **external hyperlink** will take the user to a new web page in a completely different website

#### Consistency across multiple pages

Consistency means that the user finds the elements on each page have the **same style** and are **positioned similarly**. This applies to navigation bars, colours, fonts, positioning of objects, audio/visual alerts, etc. The design also needs to consider:

* **Target audience**: If the end-user is likely to be an **adult** (rather than **child**) and **novice** (rather than an **expert**) web user, the web pages need to be straightforward to use with **easy navigation** between pages. To increase the website’s usability, each page needs to have **pleasing**, **readable** and **consistent** layout that is uncluttered and where colour schemes are chosen well. Inclusion of a **site map** and a **help menu** makes it easier for a user to find their way around a website and the use of drop-down lists, radio buttons and auto-complete makes **selection** easier
* **Accessibility** relates to ease of use for people with disabilities. For example, there might be the option to increase the font size, or choose an audio option, for partially sighted people
* **Interactivity** (e.g. feedback on user’s inputs, hotspots that trigger actions, polls, quizzes, …) can improve the user’s experience

#### Relative vertical positioning of media

This considers how media is positioned vertically relative to each other. It could be

1. Bottom
2. Middle
3. Top

#### File formats of media that is included

A standard file format is a file format that is recognised by other computer programs different from the one that was used to create it. This makes it possible to transfer a file to all programs that recognise the standard format

For example, if a text document is created in Microsoft Word and then saved as a Word file, then it may not be opened easily by another word-processing program. However, if it is saved in a standard file format for text, such as RTF, then other programs can recognise and open the file. The disadvantage is that formatting information such as indents and tables can be lost. Examples of standard file formats for different file types:

Text

TXT - a file format that stores the characters in the document but not any formatting information apart from control characters such as RETURN and TAB

RTF - a file format that contains formatting information including font, size and styles such as bold and underline as well as which characters are stored. RTF stands for Rich Text Format since it is ‘richer’ than the TXT file format, which is ‘poorer’ in its formatting information

Graphic

BMP (bitmap) - a file format that uses a binary code to store the colour of each pixel. This format does not use compression, so the file sizes can be very large. BMP files can have bit depth of 24 bits, which produces over 16 million colours

GIF (Graphics Interchange Format) - a file format that uses lossless compression. GIF uses a bit depth of 8 bits, which gives a maximum of 256 colours. Often used for transparent images and logos

JPEG (Joint Photographic Expert Group) - a file format that uses lossy compression which means that the quality of the image may be impaired. JPEG uses a bit depth of 24 bits which provides over 16 million colours

PNG (Portable Network Graphics) is a file format for graphics that supports over 16 million colours. PNG can provide lossless compression

Video

Video files are made up of graphics frames that are typically displayed around 20 times per second to create movement. Since video files can be extremely high-capacity, they are normally stored in a compressed file format to reduce their size

MPEG (Motion Picture Expert Group) is a file format that uses lossy compression

AVI (Audio Video Interleave) is an uncompressed file format for video

Audio

MP3 - stands for layer 3 of the MPEG-1 standard file format. It uses lossy compression, so that the quality is reduced. This format is widely used because files are compressed to about one tenth of their size and so reduce storage requirements and decrease transmission times

WAV (Waveform Audio Format) - a file format for sound that is uncompressed or uses lossless compression

OGG – a container format that can embed audio and video in various forms of compression

Other common standard file formats

CSV (Comma-Separated Values) is a standard file format that can be used to save tabular data such as spreadsheets by using symbols to separate the rows and columns.

XML (Extensible Mark-up Language) is another standard file format that can be used to store spreadsheet files. It is commonly used with HTML documents.

PDF stands for Portable Document Format. It is a multi-platform file format that captures the text, graphics and formatting of documents from a variety of application programs.

### Copyright, Designs and Patents Act 1998

The Copyright Designs and Patents Act makes it illegal to engage in software piracy as follows:

* to make unauthorised copies of software that is copyright protected
* to send software over a network and copy it without the permission of the licence holder
* to run unauthorised copies of software on a computer or network

The Copyright Designs and Patents Act also protects the owner’s rights over web content such as **text**, **graphics**, **video** and **audio**

### Factors affecting file size and quality

Graphics, sound and video files is determined by factors such as resolution, colour depth and sampling size and rate. A computer has a limited amount of storage capacity in main memory and backing store. There is always a balance between improving the quality by increasing the resolution, colour depth and sampling rate and having enough space to store the very large file sizes that result.

Resolution The term resolution is a measure of the number of pixels in an image. High-resolution graphics have a large number of small pixels. Low-resolution graphics have a small number of large pixels. High-resolution graphics have a better quality than low-resolution graphics but have a larger file size since they have to store the colours of more pixels. The resolution of graphics is usually measured in dpi (dots per inch)

Colour Depth Bit-mapped graphics use binary codes to represent the colour of each pixel. Colour depth is the number of bits that are used for the colour code of each pixel. The higher the number of bits, then the higher the number of colours that can be represented. Increasing the colour depth will give better quality colour graphics with a wider range of colours but will also increase the file size

Audio standard file formats (WAV, MP3)

Sampling Rate Digital sound is created by taking a sample of a sound many times a second. The sampling rate is *the number of times that the sound is sampled per second*. A higher sampling rate will result in a better quality digitised sound but will increase the file size, as more sound samples are stored per second. Sampling rates are in the order of many thousands of times per second. A typical sampling rate is 44,100 hertz or 44.1 KHz (used for music CDs) – this means 44,100 samples per second

Sample Size is *the number of bits used to store each sound sample*. The larger the sample size then the better the quality of the sound since the sample represents the sound with higher definition, but the file size will increase. Typically sample sizes of 8 bits and 16 bits are used

Video files

Frame Rate The quality of video data is determined by the number of frames that are captured per second. Each frame has settings for resolution and colour depth. The file size increases for higher frame rates, resolution and colour depth

In general, improving the quality >> increasing storage requirements >> increasing transmission times

### The need for compression

Compression techniques are used to reduce the size of large multimedia files. Lossless compression results in no reduction in quality, but lossy compression reduces the quality of the file

Need for compression Graphic, video and sound files can be very large. These files can be compressed to reduce their file size, but they must be decompressed before they can be used. There are two main advantages of compressing large files on a computer system

1. Compressed files require less storage space than uncompressed files on storage devices such as hard discs

2. Compressed files can be transmitted faster over the Internet or a computer network, since there is a smaller number of bits to be transferred.

Types of compression File compression can be either lossy or lossless. Lossy compression reduces the file size but at the expense of detail and quality. Lossless compression uses mathematical techniques to reduce the file size with no loss of detail or quality

### Comparing standard file formats

Audio

|  |  |  |  |
| --- | --- | --- | --- |
| **File format** | **Compression** | **Quality** | **File size** |
| WAV | Uncompressed | Very high | Large |
| FLAC | Lossless compression | High | Large |
| MP3 | Lossy compression | Reasonable | Relatively small |

Bit-mapped graphic

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **File format** | **Compression** | **Animation** | **Transparency** | **Colour depth** |
| JPEG | Lossy compression | No | No | 24 bits |
| PNG | Lossless compression | No | Yes | 24 bits (32 bits) |
| GIF | Lossless compression | Yes | Yes | 8 bits |

### Prototyping

Once the design is complete, Harpreet will **write program code** to implementthe **end-user** and **functional requirements of the website**. To start with, she will create **low-fidelity** **prototypes** for Jake to see.

A low-fidelity prototype is an implemented version of the wireframe that **does not have that much detail** but which can be **interacted with**. Links and interactivity within the prototype would be coded to function as they would in the finished site e.g. menus, mouse overs, internal hyperlinks.

The prototype can be used with the customer to get feedback on how suitable the proposed design is. Once Jake is satisfied with how the website is likely to look and behave, Harpreet will continue to implement the design

## Implementation (CSS)

Cascading Style Sheet rules are used to set the style of objects on a web page. For example, CSS rules can:

* Set the position of objects on the webpage (left, right, centre, full justified, paragraph, table row/column, etc.)
* Set how the objects look (bold, underline, italics, border, background colour, border colour, etc.)
* Give details regarding a placed object (resolution of graphic, location of sound file)

CSS style rules look like this:

font-family: "Times New Roman", Times, serif;

font-size:40px;

color: red; color:green; (colour of text)

text-align:center; text-align: left; text-align:right;

background-color:blue;

width:96%;

height:auto;

float:left;

padding:1%;

margin-left:1%;

margin-right:1%;

border: 2px solid;

border-radius: 5px;

CSS selectors are used to target specific HTML elements.

For example, the selectors below are p (paragraph) and div (a division/section of the page)

|  |  |  |
| --- | --- | --- |
| **HTML** | **CSS** | **Displayed by a browser** |
| <p>What colour do you like?</p>  <div>I like blue.</div>  <p>I prefer red!</p> | /\* All p elements are red \*/  p {  color: red;  }  /\* All div elements are blue \*/  div {  color: blue;  } | What colour do you like?  I like blue.  I prefer red! |

CSS classes are used to apply the **same CSS rules to** **several elements** of a page. The **classes** below are “center” and “large” (the **.** means a class)

|  |  |
| --- | --- |
| **HTML code with internal CSS rules in the <head> tag** | **Displayed by browser software** |
| <html>  <head>  <style>  p.center { *//paragraphs belonging to the class “center”*  text-align: center;  color: red;  }  p.large { *//paragraphs belonging to the class “large”*  font-size: 300%;  }  </style>  </head>  <body>  <h1 class="center">  This heading will not be affected  </h1>  <p class="center">  This paragraph will be red and center-aligned.  </p>  <p class="center large">  This paragraph will be red, center-aligned, in a large font-size.  </p>  </body>  </html> |  |

CSS IDs are used to **apply CSS rules to a specific element** on a page. The IDs below are “firstname” and “hometown” (the # means an ID)

|  |  |
| --- | --- |
| **HTML code with internal CSS rules in the <head> tag** | **Displayed by browser software** |
| <head>  <style>  p {  color : blue;  }  .intro p {  color : red;  }  #firstname {  background-color: yellow;  }  #hometown {  text-decoration: underline;  }  </style>  </head>  <body>  <h1>Welcome to My Homepage</h1>  <div class="intro">  <p id="firstname">My name is Donald.</p>  <p id="hometown">I live in Duckburg.</p>  </div>  <p>My best friend is Mickey.</p>  </body> | Notice that: the text colour of the paragraphs (p) is blue but that the text colour of paragraphs belonging to the class “intro” (.intro p) is red. This an example of one CSS rule over-riding another |

**Internal CSS:** The style rules are embedded directly into the HTML code (see example 1) – either in the <head> tag or in the actual element tag itself

**External CSS**: The style rules are written in a file that can be linked to the html code (see example 2)

With an external CSS file, the objects (elements) in the webpage HTML code are selected (tagged) and the CSS file that is linked to the HTML file contains the rules that are applied to each tag.

For example, the HTML code <div id = “text”>Web design and development</div>

might have these CSS rules for the element with the ID “text” #text {

color: red;

font-size:30px;

text-align:center;

}

The code needed to link the CSS file to the HTML code is placed in the head tag of the html. For example:

<head>

<title>Artists</title>

<link rel="stylesheet" type="text/css" href="style.css"> *link to relevant style sheet*

<link href='https://fonts.googleapis.com/css?family=Lobster' rel='stylesheet' type='text/css'>

</head>

The **head tag** is where information about the webpage is held for browsers and search engines to use to display the page correctly. The example above shows the **title** that will go on the **browser tab**, the **link to a style sheet** and the Google **font** to be used on the page

**Example 1 - Internal** CSS: The HTML code below shows some CSS rules being used to style the elements on the webpage. They are **inline** in that they are embedded in the HTML

This is how the HTML code displays through a browser

**Example 2 -** External CSS showing objects being grouped in **classes**.

body {

background-color:#E4F0E4;

font-family: 'Ravi Prakash', cursive;

}

.clearboth {

clear : both;

}

.header {

width:96%;

height:auto;

background-color:blue;

float:left;

padding:1%;

margin-left:1%;

margin-right:1%;

border: 2px solid;

border-radius: 5px;

}

.content {

width:96%;

height:auto;

background-color:blue;

float:left;

padding:1%;

margin-left:1%;

margin-right:1%;

border: 3px dotted;

border-radius: 5px;

}

.header #text {

color: black;

font-size:40px;

text-align:left;

}

.content #text {

color: red;

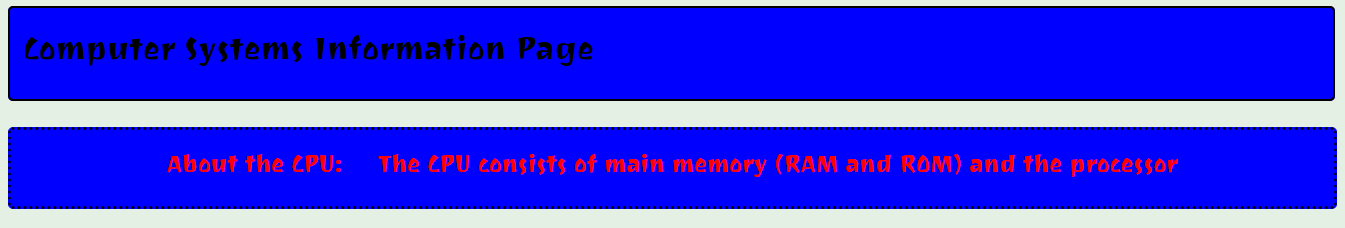
font-size:30px;

text-align:center;

}

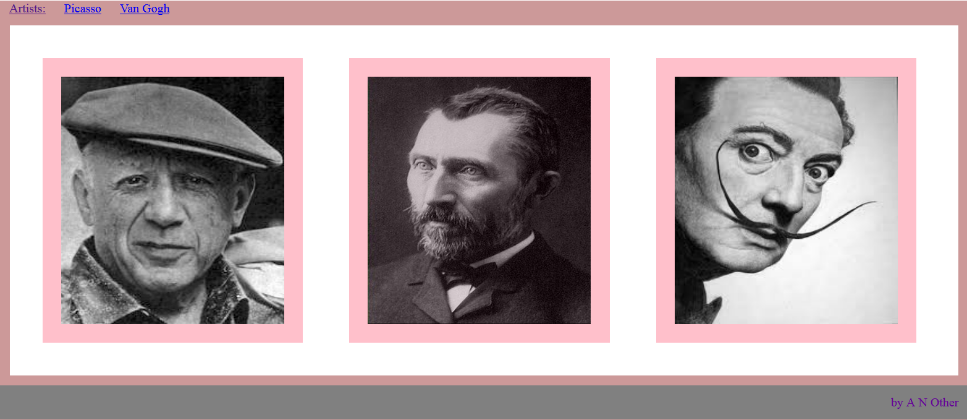
The example below shows text that belongs to a class called **header** and also text that belongs to a different class called **content**. This way we can vary the style of the text depending on its parent class.





## Implementation (HTML)

HTML (Hypertext Markup Language) is used to create the structure of a webpage. HTML uses tags (<> </>) to enclose an object. Each tag describes how the object inside the tags is to be positioned, displayed or behave. Tags have opening angle brackets e.g. <div id = “container”> and a corresponding closing tag to mark the end of that division e.g. </div>.



For example:

<html>

<head>

<title>Artists</title> *Information about the webpage but doesn’t appear on the webpage*

<link rel="stylesheet" type="text/css" href="picassoStyleSheet.css">

</head>

<body>

<div id = "container"> *This starts a division called “container”. It has three divisions inside it called “nav”, “gallery” and “text”*

<div id ="nav">

<a href="index.html">Artists:</a>

<a href="picasso.html">Picasso</a> *Can you see this in the image above?*

<a href="van gogh.html">Van Gogh</a>

</div>

<div id = "clearboth"></div> *This clears both sides of “nav” to move down the page*

<div id = "gallery">

<img class="images" src="Pablo\_picasso\_1.jpg" alt="Pablo Picasso" >

<img class="images" src="van gogh.jpg" alt="Van Gogh" >

<img class="images" src="dali.jpg" alt="Salvador Dali" >

</div>

<div id = "clearboth"></div> *This clears both sides of “gallery” to move down the page*

<div id = "text">

*Some information about the artists could be written here*

</div>

</div>

<div id = "clearboth"></div>

<div id = "footer">

by A N Other

</div>

</body>

</html>

HTML tags have an opening and a closing tag. For example, <p>This is a paragraph</p>

Some common HTML tags:

**<html>**

The <html> tag tells the browser that this is an HTML document

The <html> tag is the container for all other HTML elements (except for the [<!DOCTYPE>](https://www.w3schools.com/tags/tag_doctype.asp) comment tag)

**<head>**

Holds information about the webpage but doesn’t appear on the webpage

**<title>**

The title that appears on the browser tab

**<body>**

Contains the content of the web page – all the html tags, hyperlinks, images etc.

**<h1>, <h2>, … , <h6>**

**Headings** are defined with the <h1> to <h6> tags. They have pre-defined settings primarily to do with font-size and they are bold. <h1> has the largest font-size and <h6> is the smallest

|  |  |
| --- | --- |
| <h1>This is heading 1</h1>  <h2>This is heading 2</h2>  <h3>This is heading 3</h3>  <h4>This is heading 4</h4>  <h5>This is heading 5</h5>  <h6>This is heading 6</h6> |  |

**<p>**

Used to define paragraphs

**<div>**

The <div> tag is a container unit that encapsulates other page elements and divides the HTML document into sections

**<link>**

The <link> tag defines a link between a document and an external resource

e.g. <link rel="stylesheet" type="text/css" href="picassoStyleSheet.css">

**<a>**

The anchor tag is used to define a hyperlink that links one page to another. The page could be internal to the current website or an external reference. The link might contain the full path name (an absolute address) or it might be relative to the current page

*Example (absolute address)* <a href="http://www.w3schools.com">This is a link</a>

*Example (relative address)* <a href="computing/aboutComputing.html">Go To Page 2</a>

**<img>**

**Images** are defined with the <img> tag

*Example* <img src="w3schools.jpg">

The <img> tag has *attributes* that can be used to define the image in more detail.

*Example* <img src="w3schools.jpg" alt="W3 Logo" width="80" height="60">

The alt attribute is used to display the text W3 Logo on the page if the image w3schools.jpg cannot be found

**<audio>**

**Audio** is defined with the <audio> tag

Browser supported audio file formats are MP3, Wav and Ogg

The corresponding media type listed in the HTML code would be “audio/mpeg”, “audio/wav” and “audio/ogg”

For example:

<audio controls>

<source src="Machine Language.mp3" type="audio/mpeg">

</audio>

**<video>**

**Video** is defined with the <iframe> or <video> tag

Browser supported video file formats are MP4, WebM and Ogg.

The corresponding media type listed in the HTML code would be “video/mp4”, “video/webm” and “video/ogg”

For example:

<iframe

src="https://www.youtube.com/embed/p3q5zWCw8J4" width="200" height="180" frameborder="0" allowfullscreen>

</iframe>

<video width="320" height="240">  
   <source src=" https://holidays/japan.mp4" type="video/mp4">  
  </video>

**<ol> <ul> <li>**

These are tags for lists

An unordered (bulleted) list starts with the **<ul>** tag. Each **list item** in the list starts with the **<li>** tag. The list items are marked with bullets (typically small black circles). **<ol>** means ordered list and would result in the list items being numbered instead of just bulleted

<ul> <ol>

<li>Coffee</li> <li>Coffee</li>

<li>Milk</li> <li>Milk</li>  
</ul> </ol>

1. Coffee
2. Milk

* Coffee
* Milk

### Relative address v Absolute address

The pages in a website are usually organised in a **hierarchical** folder structure

An **absolute** address shows the complete pathname for an html file. For example:

<a href="H:\Football \ FootballTeams \ Barcelona \ Players \ Messi.html "> Lionel Messi </a>

is an anchor tag applying an **absolute** hyperlink to the text “Lionel Messi”

A **relative** address is when the link shows the position of the destination file relative to the current page being used.

For example, on the page about Barcelona (Barcelona.html), the **relative** address for “Messi.html” from “Barcelona.html” is

<a href=" Players\ Messi.html "> Lionel Messi </a>

Sometimes the hyperlink tag contains a full URL – for example:

<a href="http://www.bbc.co.uk/”>BBC Website</a>

This link would be called an **absolute** hyperlink

**Note:** Absolute hyperlinks often start with http:// and the http (or Hypertext Transfer Protocol) part of the URL sends a message to a web server to telling it to fetch the requested page

## Implementation (JavaScript)

Programs written in JavaScript **run in the web browser** itself and are **executed on the user’s computer**. JavaScript lets the user **interact** with a webpage and **change the content** of a webpage.

Javascript can:

* load new page content without reloading the web page.
* animate page elements, fading them in/out, resizing them, moving them etc
* provide interactive content: games, audio and video
* allow users to enter data into input boxes in forms and validates the input to ensure its acceptable
* send your web activities to other websites (this may be used to collect and analyse data to optimise web usage)

Exampe 1

The JavaScript code below displays a lightbulb on a webpage and, if you click on the light bulb, it goes on and off

<img id="myImage" onclick="changeImage()" src="pic\_bulboff.gif" width="100" height="180">

<script>

function changeImage() {

var image = document.getElementById('myImage');

if (image.src.match("bulbon")) {

image.src = "pic\_bulboff.gif";

} else {

image.src = "pic\_bulbon.gif";

}

}

</script>

Example 2

<html>

<body>

<img

onmouseover = "bigImg(this)"

onmouseout = "normalImg(this)"

border = "0"

src = "smiley.gif"

alt = "Smiley"

width = "32"

height = "32">

<p>The function bigImg() is triggered when the user moves the mouse pointer over the image.</p>

<p>The function normalImg() is triggered when the mouse pointer is moved out of the image.</p>

<script>

function bigImg(x) {

x.style.height = "64px";

x.style.width = "64px";

}

function normalImg(x) {

x.style.height = "32px";

x.style.width = "32px";

}

</script>

</body>

</html>

## Testing a website

* Functionality – is the website fit for purpose? Does it meet the original specification?
* Links and navigation – the internal and external links need to work correctly
* Matches UI design – the user interface needs to match the original wire-framed design
* Display of media – the testers make sure that all text, images, video and audio display correctly, especially if the website is expected to run on different hardware platforms and under different operating systems

*Reaching ahead to Higher* … many website have databases that sit behind them and the user accesses the information in the database via webpages. For example, shopping websites. Testing a website will include making sure that the database is accessed correctly and that information from the database displays correctly on the webpages

## Evaluation

Finally, the complete website is evaluated in terms of whether it is **fit for purpose**. This means making a judgement as to whether it meets its **specification** by delivering its **functional requirements** (see Analysis). You would also judge whether it meets the **end-user requirements**. You would consider your test results and draw conclusions from them.

For example, you might say “The functional requirements included a navigation bar and my original design shows the navigation bar running down the left side of each webpage. However, I was only able to implement it as running along the top of each page so my completed website meets the functional requirements in terms of having a navigation bar but the website is not correct in terms of the positioning of the navigation.”