

Basic HTML 5

(Last revision: Jan 18 2013: updated to include multimedia tags)

What follows is a very basic introduction to some elements of HTML. For more details, see any book (on HTML 5), or the website <http://www.w3schools.com/tags/default.asp>

Simple page example.

Note the overall format: whole HTML document starts with a DOCTYPE which tells the browser that the document is html (default is HTML5), and the rest of the file is bracketed by `<html> ... </html>`. The document is divided up into a head part (bracketed by `<head> ... </head>`) and a body part (bracketed by `<body> ... </body>`).

```
<!DOCTYPE html>
<html>
<head>
  <title> A Simple HTML Page </title>
</head>
<body>
  <p> A short paragraph with <b> bold</b> <i>italic </i>
and <u>underlined</u> text and even some that is
<b> <i> <u> all three. </u></i></b>
</p>
</body>
</html>
```

Note that HTML is free-format (so newlines and spaces are ignored).

HTML types

There's a need to tell the rendering system which sort of HTML is being used. There are several varieties of this (see http://www.w3schools.com/tags/tag_DOCTYPE.asp for more detail). We suggest you use

```
<!DOCTYPE html>
```

which says that the document is in HTML5.

For *XHTML 1.0 transitional* use

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-
transitional.dtd">
<html>
...
```

and for *HTML4.01 loose* use

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01
Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
```

Note that this is not strictly HTML at all – simply a description of what type of HTML follows. There are several varieties of this (see http://www.w3schools.com/tags/tag_DOCTYPE.asp for more detail).

We strongly recommend using *HTML 5*.

Basic tags

Title: this is one of the very few tags used inside the head of the document. Use it to provide a title for your document. This title is usually displayed in the browser window title location. Example

```
<title> My nice new web page </title>
```

Headers: HTML supplies six levels of headers for use in the body of the HTML document, using the `<h1>`, `<h2>`, ... `<h6>` tags. These are intended to be used to mark different levels of header inside a document (and not intended to be used simply for highlighting text). Example

```
<h3> The third-level heading of this document </h3>
```

Paragraphs: paragraphs are introduced with the `<p>` tag, and terminated by the `</p>` tag. Example

```
<p> Although this paragraph is relatively short, it still
will be rendered as a paragraph </p>
```

Lists: HTML supports a number of types of lists, with various details within them. Numbered lists are introduced with the `` tag (and terminated with ``), and un-numbered (bullet-point) lists are introduced with the `` tag (and terminated with the `` tag). The actual elements of the list are introduced in both cases by the `` tag, and terminated by the `` tag. Example:

```
<ul>
<li> first bullet point of list
<li> Second bullet point of list
</ul>
```

HTML also has definition lists, introduced by `<dl>`. For these, the `<dt>` tag is used to introduce the item being defined, and the `<dd>` tag to introduce the item description. Example:

```
<dl>
<dt>garlic</dt>
<dd>pungent member of the onion family</dd>
<dt> leek </dt>
<dd> member of the onion family used as a vegetable</dd>
</dl>
```

Note that lists may be nested. There are lots of ways of changing the detailed presentation of the list using a style sheet.

Highlighting text: `` introduced bold text, `<i>` introduces italic text, `<u>` introduces underlined text, `` introduces emphasised text (often rendered as italic), `` introduces strong text (might be rendered as bold). Each is terminated by the appropriate tag (``, `</i>`, `</u>`, ``, ``). They can be mixed as desired. An example is given on the first page of this document. Note that the intention is that headers are used where more appropriate – these are better used for single words that need highlighted, or for italicising paragraphs, for example.

Breaks: Since HTML is free-format, taking a new line in the text has no effect. To get a new line use `
`. (There is also a tag `<wbr>` which provides a *word break opportunity*, used by most (not IE) browsers to insert a break, allowing you to avoid breaking a line at an inappropriate place.)

Horizontal lines: `<hr>` produces a thematic break (often implemented as a horizontal line).

There are lots of other basic formatting tags as well, see <http://www.w3schools.com/tags/>.

Images

Images may be included in HTML pages. Images come in many different forms, as well as many different sizes. Not all browsers support all formats. It is important to note that you should always include an optional text to be displayed if the page is being viewed using a screen reader (or indeed, on a text-only browser). The tag for

image insertion is `` and it has a number of attributes. The `src` attribute describes where the image file is to be found, and the `alt` attribute allows alternative text to be displayed. The `` tag allows the setting of the height and width of the image, either as a number of pixels, or as a percentage of the overall browser window. Example:

```

```

will display an image of the file `nicepicture.jpg` (in the same directory as the file containing the tag), with a height and width of 150 pixels, and with the stated alternative text.

Anchor (hyperlink) tags.

Anchor tags are the mechanism whereby HTML becomes hypertext, rather than linear text. They are used both to define links, i.e. the places that take you somewhere else when you click on them.

The basic format for a link is

```
<a href = "location_to_go_to">The hot text</a>
```

and the effect is that *The hot text* becomes highlighted and when clicked on (or pressed) the browser will jump to the "location to go to". Note that both images and text may be "hot". The "location to go to" may be

- An identifier (`id`) in the same document: Example: `link to inside this document`
- A URL (in which case the jump will be to the start of another document) Example `link to another document`
- A identifier in another document. Example `link to inside another document<a/>`

The href attribute and the location of the document.

In the section above, we used only the name of the file in the `href` attribute. This only works if the both the HTML file with the link and with the destination are in the same directory as each other. More generally one may use the full URL

```
<a href="http://www.cs.stir.ac.uk/courses/CSC941/index.html">l
ink text</a>
```

and in addition, one may use two dots (`..`) to point to the directory above, so that

```
<a href="../anotherfile.html">link text</a>
```

will point to the (beginning of the) file `anotherfile.html` in the parent directory of the directory in which file containing the link is placed.

*The identifier (*id*) attribute*

Virtually all tags can have an `id` attribute. This can be used for several purposes, including providing a target location inside a document for a hyperlink to jump to. (Another usage is as a style sheet (CSS) selector).

For example consider the HTML

```
<h4 id="section3">The title of an important subsection</h4>
```

If this tag were inside an HTML document called `document1.html`, then a link in the same file of the form `link text` would jump to this location, and a link in a different file in the same directory (folder) of the form `link text` would jump to the same location.

Canvasses

Canvasses are new in HTML5: they enable the browser to draw directly¹. The `canvas` tag itself simply defines where the image/picture/diagram will be drawn: actual drawing uses Javascript (see Javascript hand-out). The `canvas` tag has the form

```
<canvas id="mynicenewcanvas" height="200" width="400">
</canvas>
```

Because canvasses are new in HTML 5 there may be browsers that do not support them: your code should take that into account. One method of doing this is to put HTML between the `<canvas ...>` and `</canvas>` tag: this will be executed only if the browser does not support canvasses.

```
<canvas id="mynicenewcanvas" height="200" width="400">
The browser you are using does not support canvasses.
</canvas>
```

¹ Previously, other technologies (such as Flash, or Java Applets) were required to achieve this. Putting it into HTML5 means that browsers that support HTML5 will also be able to render images generated using the `canvas` tag, making it more likely that images will render correctly across a range of browsers.

For more information on using the canvas tag effectively, and on drawing, see both the CSS and Javascript hand-outs, and also http://www.w3schools.com/tags/tag_canvas.asp and http://www.w3schools.com/tags/ref_canvas.asp.

Tables

Tables are not only used for tabular data, but also to lay out text and images in a rectangular array. Tables are introduced by the `<table>` tag (and terminated with the `</table>` tag). Tables are made up of rows, each introduced by the `<tr>` tag and terminated by the `</tr>` tag. Each table row is made up of

- An optional table row header, introduced by the `<th>` tag, and terminated by the `</th>` tag
- Zero or more table row data elements, each introduced by a `<td>` tag, and terminated by a `</td>` tag.

Tables may optionally have a head, body, and foot, introduced using the `<thead>`, `<tbody>` and `<tfoot>` tags: see <http://www.w3schools.com/tags/default.asp> for details.

In HTML5, the only supported attribute is the `border` attribute: this specifies whether the table has borders or not. Tables without borders are useful for simply setting out text and images on the page. The nature of the border is specified in the CSS file.

Simple example:

```
<table border="1" >
  <tr>
    <th> Males </th>
    <td> 10 </td>
    <td> 99 </td>
  </tr>
  <tr>
    <th> Females </th>
    <td> 20 </td>
    <td> 1430 </td>
  </tr>
</table>
```

Multimedia support

HTML5 supports multimedia through the use of the `<audio>` and `<video>` tags. These allow a number of audio and video formats to be directly embedded in web pages.

Both tags have a small number of attributes:

`autoplay` allows the clip to be played immediately the page is loaded (as opposed to being played when the play button on the player inside the page is pressed)

`controls` allows controls (e.g. volume, play etc.) to be displayed

`loop` means that the clip will be played repeatedly.

In addition, the `<video>` tag has more attributes:

`height` sets the height of the embedded video player

`width` sets the width of the embedded video player

The `<audio>` and `<video>` tags are used in conjunction with the `<source>` tag which determines where the multimedia clip to be played should be found. Its format is (nearly!) straightforward: it has two attributes:

`src` shows where the clip is to be found

`type` describes the particular format for the clip: this uses the MIME encoding for particular formats, but only a small number of these are actually used. For audio, these are `type="audio/ogg"` and

`type="audio/mpeg"`, and for video `type="video/mp4"`,

`type="video/webm"`, and `type="video/ogg"`.

For example,

```
<audio controls autoplay>
  <source src="drums1.mp3" type="audio/mpeg" >
  <source src="drums1.ogg" type="audio/ogg" >
  Your browser does not support the audio tag.
</audio>
```

will embed one audio file into the page: note that two possible sources have been supplied. The browser will attempt to load these in order (and stop once it has loaded one). It uses the type of the source to determine whether it can load each source. For example, Safari can load `audio/mpeg` sources, but Firefox cannot; however, Firefox can load `audio/ogg` sources. A useful guide to converting videos can be found at http://commons.wikimedia.org/wiki/Help:Converting_video.

```
<video width="320" height="240" controls>
  <source src="movie.mp4" type="video/mp4">
  <source src="movie.ogg" type="video/ogg">
  Your browser does not support the video tag.
</video>
```

will embed one of the video files: the same comments apply as for audio about formats. For both `<audio>` and `<video>` tags, the text before the closing tag will be displayed if these tags are not supported.

Character entities

Character entities are introduced by & and terminated by ; and may not have any spaces inside them. These are useful for characters that have special meanings in HTML, or for (e.g.) diacritical marks in languages.

To get a < use < ; (note that the semicolon is necessary)

To get a > use > ;

To get & use & ;

There's lots of others: see <http://symbolcodes.tlt.psu.edu/web/codehtml.html>

Finally: some of the tags above have attributes that can be changed: see <http://www.w3schools.com/tags/default.asp> for details. However, in HTML5, many of these tags are no longer available, and the attributes are adjusted using Cascading Style Sheets (CSS), which are the subject of another handout.

(© LSS October 2010, January 2013.)