HTML

An introduction to creating web documents

Matthew Roberts
Macquarie University
mattr@ics.mq.edu.au

Preparation

You will need to ensure your computer has the following capabilities

- A web browser (any one of the following)
  - Internet Explorer
  - Netscape
  - Opera
  - Other
  - If you have no web browser, there is a copy of Netscape included in this CD.
- A text editor. Text editors are simple programs that can open a file made up of text and allow you to edit it with no formatting applied to the text. Text editors are vital for all forms of computer programming. Most operating systems have a default text editor (in Windows it is Notepad, however, Wordpad can also be used). Having a good text editor is important to productive and enjoyable HTML work. The default editors are often no better than adequate. A great, free, text editor that is suitable for HTML is EditPad Lite. A copy of this program for Windows is included on this CD.

Surfing the Internet

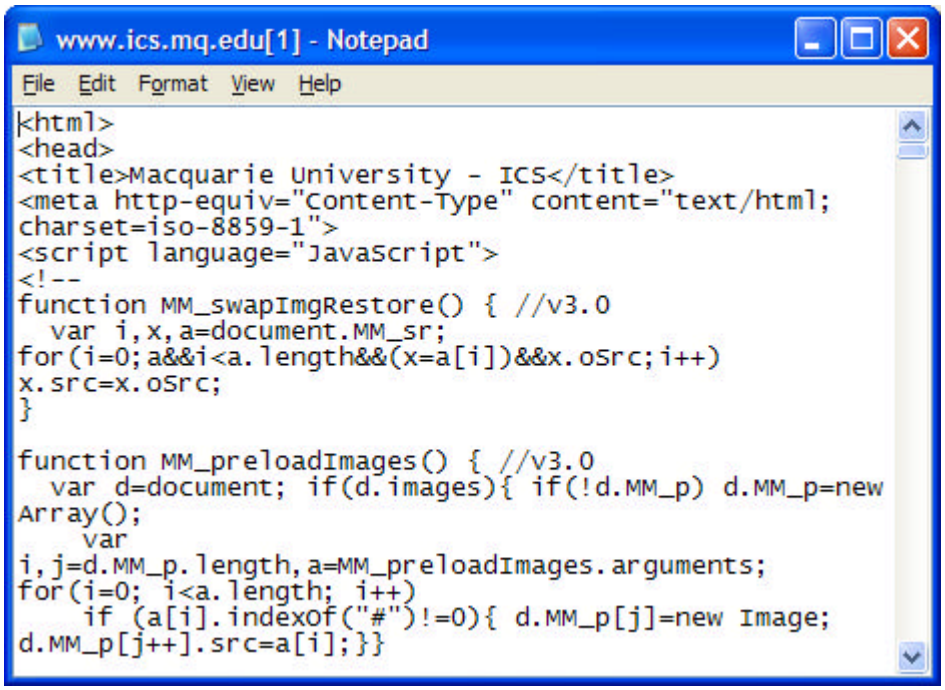
When you visit a web site, your computer requests a web page from the place you are visiting. For example, when you visit

www.mq.edu.au/index.html

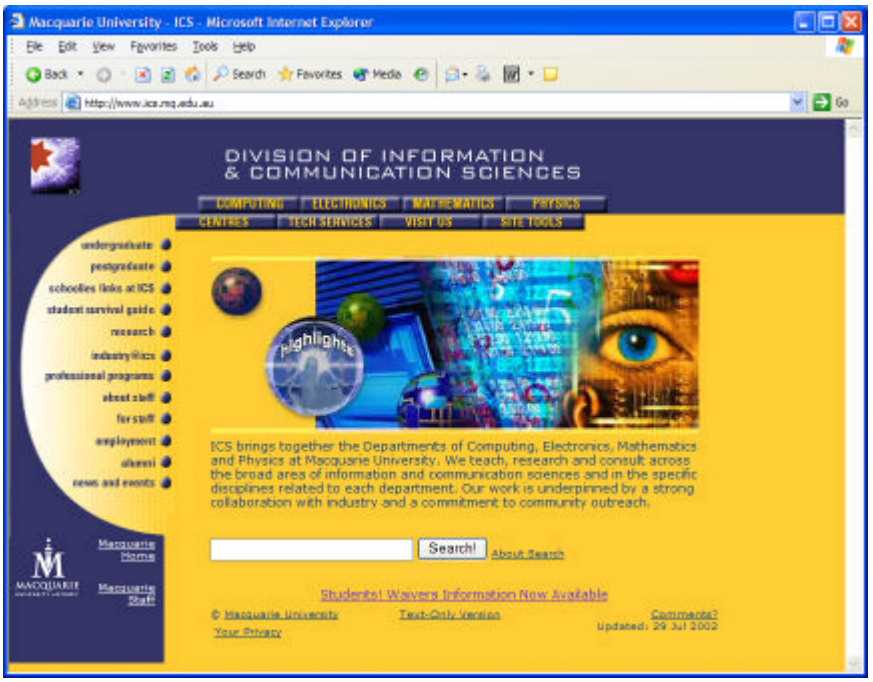
you are asking the computer located at www.mq.edu.au to give you a copy of the page `index.html`. What actually gets sent across the Internet is a text file made up of characters. It is then the job of the web browser on your computer to interpret this file and create the final image on your screen.

HTML

HTML is the **H**yper**T**ext **M**arkup **L**anguage. It is a language for defining the formatting and the layout of a web page. Your web browser (Internet Explorer, Netscape, Mozilla, Opera, etc.) is able to read HTML and to create the final web page from this specification.


www.ics.mq.edu[1] - Notepad
File Edit Format View Help
<html>
<head>
<title>Macquarie University - ICS</title>
<meta http-equiv="Content-Type" content="text/html;
charset=iso-8859-1">
<script language="JavaScript">
<!--
function MM_swapImgRestore() { //v3.0
var i,x,a=document.MM_sr;
for(i=0;a&&i<a.length&&(x=a[i])&&x.oSrc;i++)
x.src=x.oSrc;
}
function MM_preloadImages() { //v3.0
var d=document; if(d.images){ if(!d.MM_p) d.MM_p=new
Array();
var
i,j=d.MM_p.length,a=MM_preloadImages.arguments;
for(i=0; i<a.length; i++)
if (a[i].indexOf("#")!=0){ d.MM_p[j]=new Image;
d.MM_p[j++].src=a[i];}}


/Browser


Macquarie University - ICS - Microsoft Internet Explorer
DIVISION OF INFORMATION
& COMMUNICATION SCIENCES
ICS brings together the Departments of Computing, Electronics, Mathematics and Physics at Macquarie University. We teach, research and consult across the broad area of information and communication sciences and in the specific disciplines related to each department. Our work is underpinned by a strong collaboration with industry and a commitment to community outreach.
Search!

The text file that the browser reads is called the **source file** and the output in the browser is called the **web page**.

Exercise 1 – Getting Started

In this exercise, you will use your web browser and your text editor to view the source and the web page for the file `page1.html` on the CD.

1. Open your web browser (Internet Explorer, Netscape, Opera, etc.)
2. Find the Open command for your browser (often in the File menu)
3. Find the file `page1.html` on the CD and open it.
4. This is the Web page.
5. Find you favourite text editor and open it.
6. Open the file `page1.html` in the text editor.
7. This is the source code.

You will notice that the two are very different, but you may be able to "read" some parts of the source code and work out which parts of the web page they correspond to. We will now learn how to understand the source code, bit by bit.

Understanding the source.

1. View the source of the file `hello.html`

You should see the following in your text editor.

```
<html>

<head>
<title>
Hello world web page
</title>
</head>

<body>
Hello World
</body>

</html>
```

What does it all mean?

The first thing to understand about HTML is that it is a ***markup*** language. This means that it is a way of taking plain text and

formatting it. A HTML source file is made up text that is wrapped in ***formatting tags*** (often shortened to "tags"). All formatting tags are enclosed in '<' and '>' brackets. '<' is used to open the tag and '>' is used to close the tag. There are two different types of tags

1. start tags (eg <body>)
    a. start tags are always in the form `<full_tag_details>`
2. end tags (eg </body>)
    a. there is normally one end tag to match every start tag and it will always be in the form `</tag_name_only>`. Notice the slash after the less than sign to indicate is a closing tag.

A description of each tag

<html>

This designates that this document is a HTML document. This allows any browser reading the file to distinguish it from the other types of files it may be asked to open. The `<html>` tag should be at the start of the document and the corresponding end tag, `</html>`, should be at the end of the document.

<head>

These tags designate the header of the document. The header section includes all information about the document. In this case, only the document's title in given.

<title>

All text within the start and end title tags makes up the title of the document. In most cases this text will appear in the title bar of the browser when is it displayed as a web page.

<body>

Everything within the body tags is the actual body of the document. It is here that all the information you wish to display on the page must be specified.

Exercise 2 - Your first web page.

In this exercise, we will create our first web page. In the process, we will modify a source file and then view it in a browser to see our changes. Note that the browser will not automatically update your web page to reflect changes in your source code. To get changes reflected in the web page you must first save any changes to the source and then press the refresh button on the browser.

1. Open the source for the file `hello.html`
2. Save it as a different name of your choosing (perhaps `myhello.html`)

3. Modify the text in the body of the document to have it include information about a class you are teaching.
4. View your first web page in your browser by opening `myhello.html` (or whatever you named your file) in your web browser.

Tag attributes

Tags can be simple, or you can specify attributes. In the above example, there were no attributes specified (i.e. all tags are simple). Tag attributes are specified by a `name="value"` pairing included in the start tag (the end tag will not contain these pairings). Most tag attributes are optional and can appear in any order. Here is an example of a tag with attributes.

```
<body bgcolor="green">
```

This is the same as the <body> tag in the above example, but with the extra attribute that the background colour of the body of the web page should be green. All attributes have a default value, so if you do not specify anything, the default value of the browser will be used.

Have a look at what happens to the Hello World document when we use some tag attributes.

New source (`hello2.html`)

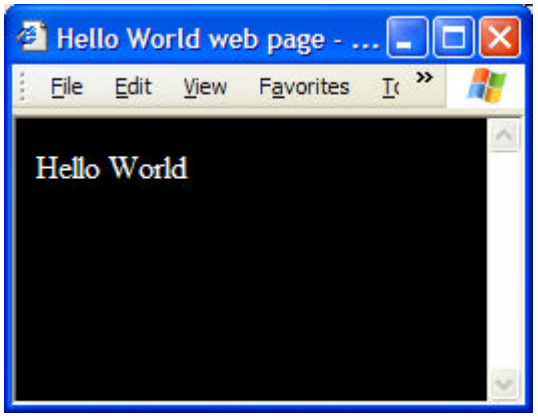
```
<html>

<head>
<title>
Hello World web page
</title>
</head>

<body text="white" bgcolor="black">
Hello World
</body>

</html>
```

New web page



Exercise 3 – Add colour to your page.

1. Open the page you created in Exercise 2 (your page) in your text editor.
2. Add some colour to your page using attributes of the body tag.
3. (advanced) If you want to try some other colour attributes, you can try:
   a. `alink`
   b. `vlink`
   c. `link`

HTML as instructions

A good way to think about HTML is that it is a set of instructions you are giving the browser to help it show the page the way you want it. If you leave something out, browsers are often able to "guess", but you are not guaranteed to get exactly what you wanted.

Exercise 4 – Basic text formatting tags

1. Open the file `basic text.html` in your text editor
2. Open the file `basic text.html` in your browser
3. Read the source and use the web page to see the effect of each of the tags presented.
4. Use some of these tags to improve the formatting of your web page. You may like to add some more content to your page at this stage. Try to get a document with some information that students attending your class may find useful.

Notice that most tags have a start and an end tag. Notice also that attributes in the font tag were used to describe the font and size.

Exercise 5 – Lists

Lists are a great help in structuring a HTML document and they are very easy to use. There are two types of lists, an ordered list (which will be numbered) and an unordered (which will use bullet points).

1. Open the file `lists.html` in your text editor
2. Open the same file in your browser.
3. Read the source and use the web page to see how the lists are shown in the final web page.
4. Add some lists to your own web page. Perhaps a list of the topics covered in that class.

Images

Most web pages have images, however, to this point, we have not seen any. Images can be added to a web page with an image tag

```
<img src="image.gif" alt="text">
```

Image is another of the tags that do not require a closing tag to follow it. It is also unusual in that it has two compulsory attributes. Your browser will still read the page with these missing, but they are required to have correct HTML code. The two required tags are `src` which defines the source file for the image, and `alt`, which specifies text to render in place of the image if the image cannot be rendered. The alt attribute is required because there are some text only browsers as well as some designed for blind people that read the pages aloud. The text you specify in the alt tag will also appear in some browsers if the user hovers their mouse over the image.

Exercise 6 – Adding images

Add some images to you document to make it look better. I have provided images in the image directory of the CD if you wish to use them, or you can source your own.

1. Open the file `images.html` in both your text editor and your browser.
2. Notice how the `img` tag works
3. There are two strange lines in this code, they are there to ensure our HTML satisfies the standard perfectly. You need not understand them, just include them in every file if you wish. If you use frames you will need a different doctype tag.
4. Add some images to your own document.

Hyperlinks

Hyperlinks are the heart and soul of the web. These are the underlined words in web pages that you can click and will take you

to another page. Hyperlinks are sometimes called links or web links, essentially the terms are synonymous.

Anchors

Before you can include a link in a web page, you need to ***anchor*** it to some part of you page so that the web browser knows where to put it. You can anchor links to text or to images. The anchor tag is <a> but it is not much use without the `href` attribute. The `href` attribute describes where to link to. Thus the tag

```
<a href="destination">
```

can be read to mean "anchor a link to *destination* to whatever is inside this tag"
When we say something is inside a tag, we mean that it is between the start and the end tags. Thus the following HTML

```
<a href="http://www.mq.edu.au">Macquarie Uni</a>
```

Will create a link to www.mq.edu.au that is anchored to the text "Macquarie Uni"

The most common way to use a link is to link to another page (as in the above example). However, it is also possible to create a link to a particular place within a document. This can be done by naming anchors so they can be later used as destinations.

```
<a name="contents">contents</a>
```

This location in the file can then be linked to from within the same document with an anchor such as this.

```
<a href="#contents">go to contents</a>
```

Notice the # before the name to indicate it is a named location and not another page.

Exercise 7 – Adding links to your document

1. Open the file `links.html` in your text editor
2. Open the same file in your browser
3. Compare the two and follow some of the links in the browser.
4. Add links to some useful pages to your own web page. Try adding a link to another location in the same file.

5. (advanced) Try out the attribute `target="_blank"` to see what it does.

Tables

Tables can be inserted into a web page using the table tag and the associated table row (tr), table cell (td), table header (th) tags.

```
<table>
  <tr>a table row
    <td>a table element
    </td>
  </tr>
</table>
```

Tags for tables

<table>

The tag that begins and ends the table definition.

<tr>

This tag begins and ends each row of the table. Each`<tr>` tag should be inside a `<table>` tag.

<td>

This tag begins and ends each cell in the table. Each`<td>` tag should be inside a `<tr>` tag.

<th>

This tag operates in the same way as the `<td>` tag, however, it should be used for the cells in the header row of the table. It will result in the text in those cells being shown in a different font to indicate they are a heading (often bold text).

Layout of source

When you begin adding tables to your HTML code, you will quickly loose track of what cell is in which row, etc. To help in this you should follow consistent indenting rules to improve readability. A good way to go is to indent each row by two spaces, to indent each cell by two more spaces and to put each cell on a new line. An example of this can be seen in the code in `tables.html`

Attributes

There are a number of important attributes for the various table tags:

- `border` defines the width of the table border.

- `cellpadding` defines the space between the edge of the cell and the content.
- `cellspacing` defines the space between cells.
- `width` and `height`.

The width and height can be set in terms of pixels or as a percentage. For a table tag, the percentage is of screen size and for the other tags, it is a percentage of the table size.

Exercise 8 – Tables

1. Open the file `tables.html` in both your text editor and your browser.
2. Read the source and see how the attributes effect the tables
3. Play with the attributes to learn how they work.
4. Add a table to your own document.
5. (advanced) Try to use tables to layout your page. The web page `frames_layout.html` is an example of a page that uses tables to carefully position objects on the screen. Try to reproduce the layout of this page. Try not to look at the source. Use only the web page and try and recreate it from that. If you are getting stuck, or you have finished and want to see how it was done, then look at the source. In particular the two-colour title bar is created with careful use of tables and background colours. Also try to reproduce the two column effect using tables.

Colours in HTML documents

Many tags have attributes that allow you to set a colour. Earlier we saw the <body> tag with colour attributes for text, background, and links. There are two ways to define colours in HTML. You can either describe the colour by a name (e.g. black, blue, green, yellow, etc.) or you can describe it with a hexadecimal notation. The hexadecimal notation is a combination of Red, Green and Blue colour values (RGB). The lowest value you can give a light source is 0 (#00 in hex) and the highest is 255 (#FF in hex). Using names is quite convenient, but it does not comply with the HTML standard and may result in your page not being shown correctly in some browsers.

Exercise 9 – Experimenting with colour

1. Open the file `colour.html` in both your web browser and your text editor.
2. Play with the colour values to see what effect you can have.

Frames

Frames are a very useful construct that you can use to help make organising your web page simpler. Frames allow you to split the screen up into rectangular regions (frames) and to load a different HTML file into each frame. This is particularly useful for adding an index to your page.

Exercise 10 – Frames and Indexes

In this exercise, we develop a page using frames that will have an index list on the left and a viewing region on the right.

1. Open `example_frames.html` in your web browser to get a sneak peek at an example of what you will end up with.
2. Click on the different entries in the index to see what that does.
3. Open a new file in your text editor
4. Name it `myframe.html` and save it.
5. Open another new file, call it `people.html` and save it in the same directory as `myframe.html`.
6. Open another new file, call it `companies.html` and save it in the same directory.
7. In `people.html`, create a web page that contains all the names of people you know and perhaps some details about them (an example can be found in `example_people.html`).
8. In `companies.html`, create a web page that contains the details of all the companies or schools you deal with (an example can be found in `example_companies.html`).
9. We must now create the frames page and the index to complete the job.
    a. Add the following text to `myframe.html`

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

<!-- notice that the doctype tag is (slightly)
different       for a page with frames -->

<html>

<head>
<meta http-equiv="Content-Type"
content="text/html; charset=utf-8">
<title>
A frames demonstration
</title>
</head>

<!-- NO BODY TAG!! -->

<frameset cols="25%, 75%">
  <frame name="left" src="index.html">
  <frame name="right" src="people.html">
</frameset>

</html>
```

b. Save `myframe.html`
c. Create a new file called `index.html` in the same directory and add the following text to it.

```
<html>

<body>
<h3>Index</h3>
<p><a href="people.html"
target="right">people</a></p>
<p><a href="companies.html"
target="right">companies</a></p>
</body>

</html>
```

d. View the file `myframes.html` in your browser, click on the two items in your index and you should see the right pane change to reflect this.
e. (further) add more items to your index and create content pages to go with them.

There are a few interesting points to make about frames. Notice that each frame had a name attribute. This allows us to tell the browser which part of the page to send a href to. Notice that in the index file, each anchor tag included the attribute `'target'`. This is telling the browser to open this link in that section of the page. Notice also that the page you ask your browser to open is just the top level frames page. It is able to compose the page from the information in that file only.

Forms

Forms in web pages can be quite useful. They allow you to get information from the people who are visiting your page. In this example we develop a form that can be e-mailed to someone when the user hits a submit button.

Exercise 11 – Getting user information

1. Open the file `forms.html` in both your browser and text editor.
2. Try out different things in the text file and see how that changes the web page.
3. Change the line below to include your email address in the 'action' attribute. Make sure you leave the mailto: part alone.

```
<form method="post" action="mailto:yourname@youraddress.com">
```

4. Hit the submit button.
5. Check your e-mail. You should see a new message containing the post information. This will only have worked if you are able to send e-mails from the computer you are working on.
6. Add a form to your page. The form should ask the user for some feedback about your page and include a submit button. It should be posted to your email address.

Deprecated Tags

Deprecated tags are old tags that you are discouraged from using. There are also some attributes of non-deprecated tags that have been deprecated. Many of the tags and attributes we have been using to this point have been deprecated. For example, the `<font>` tag is deprecated. Often they have been deprecated because style

sheets have taken over their function. However, if you are not comfortable with style sheets or are writing for an audience that cannot use them (i.e. they have old browsers) then you should continue to use deprecated tags and attributes. Deprecated tags have no guarantee for future browser support.

Style Sheets

Once you can properly use style sheets, you should have no need for any of the deprecated font and colour tags and attributes.

Style sheets (or cascading style sheets – CSS) are a fundamentally different, and infinitely superior approach to formatting the text in a document than those we have seen so far.

The basic idea

Instead of formatting each page one by one with font and colour tags, create a file that contains the global "style" of all your pages. For each page, tell the browser to consult a style sheet to get fonts, colours, line spacing, etc. Style sheets end up being separate files and there is normally just one for a whole site full of pages. This makes it easy to maintain a consistent look to all the pages at one site, makes updating formatting simpler, and separates formatting (CSS) from layout (HTML).

A very simple cascading style sheet rule is

```
H5 { color: red }
```

This specifies that everything within a H5 tag (heading level five) should be coloured red.

Style sheet rules are made up of ***selectors*** and ***style definitions***. In the above rule, H5 is the selector and {color: red} is the style definition. The selector indicates which part of the HTML document to apply the associated style definition to.

A style sheet is made up of a number of rules. The following is a complete style sheet.

```
BODY { background-color: #c565Fcc; font-
family: Arial, Helvetica, sans-serif }
P { line-height: 200% }
OL LI { font-size: 85%; colour:blue; }
H1, H2, H5 { color: red }
.green { color: green }
.blue { color: blue }
#large { font-size: 120% }
#to_upper { text-transform: uppercase }
```

When selectors are separated by commas, it specifies that the style definition should be applied to text within *all* these tags. When there are two selectors on the same line that are not separated by commas, it specifies that that style definition should only be applied to text that is within both those two tags.

It is possible to have selectors that are not tag names. These are user defined ***classes*** (starting with a period) or user defined ***IDs*** (starting with a hash). Classes and IDs are largely the same unless you are doing some scripting. Every tag in HTML can have a class or ID attribute, so you can format any part of the document in that style.

To include a style sheet called `style.css` in a HTML file requires the following line between the <head> tags.

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

It is possible to define style sheets within a HTML document and it is even possible to define new rules within a tag. However, these techniques will not be covered here.

This is a very brief overview of Cascading Style Sheets, for a more in-depth introduction to style sheets go to http://builder.cnet.com/webbuilding/pages/Authoring/CSS/ .

Exercise 12 – Style Sheets

In this exercise we investigate style sheets and how they can make creating a whole site much easier.

1. Open the file `style1.css` in your text editor. This is a style sheet
2. Open the file `styles_demo1.html` in your text editor and browser. Notice that this HTML file links to the external style sheet.
3. Open `style2.css` in you text editor. Notice the differences between `style1.css` and `style2.css`.
4. Open the file `styles_demo2.html` in your text editor and browser. This HTML file links to the second external style sheet. Otherwise it is identical to `styles_demo1.html`. Notice what a difference the new style makes. You should see that none of the statements within the file are correct because we have changed the style.
5. Create your own (very simple) HTML file that uses `style1.css` (or `style2.css` if you prefer it).

6. Take all the formatting tags out of your own web page and format it using style sheets instead. Add extra formatting now that you have the power and convenience of style sheets.

Taking it further

This is just a glimpse of HTML. There are many more tags and attributes we have not gone through here. The best way to learn more HTML is to play around with it. You should visit a tag reference site (see resources) and just try out tags and attributes you haven't used before to see how they work.

Resources

These are some vital resources that I am sure you will not be able to live without.

HTML Tag Reference

There are any number of HTML tag reference pages on the Internet, here are a few good ones. Try a few out until you find one that you like.

The Complete HTML 4.01 Reference

http://www.w3schools.com/html/html_reference.asp

Highly recommended, it is clean and easy to use. It also points out tags and attributes that may still work but which have been deprecated.

The official w3c specification for HTML4.01

http://www.w3.org/TR/html401/

This is the place to go for the final word on any HTML related question.

Netscape Tag Reference

http://developer.netscape.com/docs/manuals/htmlguid/contents.htm

This gives all tags and how they work *in Netscape*. Feel free to use this resource if you prefer it, but be aware that some tags Netscape supports are non-standard.