Creating a Simple Webpage

Contents

Creating a Simple Webpage

Text structures

In the first chapter, the one stating the Requirements for following this course, a small exercise was printed, in which you created your first simple webpage. If you haven't done that exercise yet, go there now and do it.

A text entered in a text editor and saved to a hard disk, or other form of storage, is often called a 'plain text file'. Plain text files generally provide for two small ways for marking up text: you can separate words using tabs and spaces, and you can separate paragraphs using returns.

People have found creative ways of producing intricate lay-outs using just these small methods of mark-up. In webpages, these lay-outs will generally be discarded: a web browser must collapse all consecutive spaces, tab stops and returns into one single space or soft return, depending on where on the line the word occurs. There is a way around this, which will be discussed later.

When you structure a text, you generally do so to make it easier to digest and to read. By making chapter and section headings more pronounced, you allow the reader to skim over a text until they find an especially interesting part. By using introductions and abstracts, you allow a reader to decide if this text will be interesting to them. You can use illustrations, because sometimes people will much sooner understand what's going on when they can see what's going on.

HTML, the HyperText Mark-up Language for the web, and its successor XHTML, the eXtensible HyperText Mark-up Language, allows you to impose a structure on a plain text document. It replaces the few simple mark-up methods plain text allows you by its own. (Note: the abbreviation (X)HTML is often used when talking about features that are common to both HTML and XHTML.)

The way (X)HTML lets you do all this, is by letting you label certain parts of the text as heading, image, table, list, important et cetera. Some structures are not catered for by (X)HTML; it is a relatively simple mark-up language. For instance, there is no element for introductions, or leads. The reader will have to infer from the position in a text which is the introduction, the lead or the abstract.

Now there was a term you should know: element. (X)HTML lets you structure a text by dividing it up into elements. Elements are separated from each other by a type of label called tags.

An example:

```
<p>This is an <em>important</em> example.</p>
```

What you see above is a paragraph element with an embedded emphasis element. The paragraph element starts with a <p> tag and ends with a </p> tag. The emphasis element starts with an <em> tag and ends with a </em> tag.

Creating a Simple Webpage

A webpage by any other name...

There are several versions of the (X)HTML standard. The more recent ones add a lot of structure verbiage: many elements are obligatory under many circumstances. However, web browsers should also support older versions of the standard, so that examples such as the one given in the Requirements chapter are considered valid HTML documents.

Every webpage must have a title element (under older versions of HTML this was the only required element):

```
<title>My webpage</title>
```

The title is the text by which the browser window will be named. It is the text that appears in your list of bookmarks if you bookmark a page (add it to your Favourites). It is the text by which your page will be listed by the search engine.

Find a descriptive title. The heading of your page will often do just fine. The heading of this chapter is 'Creating a simple webpage', so its title could be the same text. Since this webpage explains to you how to create simple webpage, that would be an excellent title.

Bad titles abound on the web. Companies who feel that their own satisfaction is much more important than their customer's satisfaction (which is very much true, but it's bad marketing to hammer that point home too much) tend to title their web pages as follows:

```
<title>Big Fridge Manufacturer Inc.</title>
```

If you are lucky, they will even add in a bit of information related to the webpage you are visiting, for instance:

```
<title>Big Fridge Manufacturer Inc. - Manual of the Cool 3000 ice box</title>
```

Better for the visitor is:

```
<title>Manual of the Cool 3000 ice box - Big Fridge Manufacturer Inc.</title>
```

After all, it is much more likely that the visitor who gets to this page is searching for the manual, rather than for company info.

So, the lesson is: always put the important information first. Often, you only have a limited amount of characters available to you in your bookmarks menu, in the search engine listing or in the window title bar; utilize that space to the maximum.

An (X)HTML document with only a title element is not very useful. We will now introduce you to a couple of elements that will allow you to make good use of 90% of the power of the web.

Creating a Simple Webpage

A simple linking webpage

title	The name of a page.
h1	The most important heading(s) on a page, often the same as the title.
h2	A sub-heading. There may be multiple of these.
p	A paragraph.
a	An anchor for a link.

With these elements, we can make the following simple web page:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
<html>
 <head>
  <title>Friends and family of Clemence Wylie</title>
 </head>
 <body>
  <h1>Friends and family</h1>
  <p>The following are links to the websites of my friends and family</p>
  <h2>Friends</h2>
  <p><a href="http://www.tomsawyer.us">Tom Sawyer</a></p>
  <h2>Family</h2>
  <p><a href="http://www.tantejeanette.ca">Aunt Jeanette</a></p>
 </body>
</html>
```

Exercise 2-1

Copy the above sample code to your text editor. Save it as exercise2-1.html. Open it in a webbrowser. Does it display like you expected?

Doing the link

The a tags (the name is short for 'anchor') in the previous example have a special form. They contain attributes with attached values. Many (X)HTML tags can actually contain attributes. With attributes you may modify or extend the meaning of an element.

a tags have several possible attributes, of which the most important are href and name. href is the attribute that defines the URL (Uniform Resource Locator, a fancy way of saying the address) a link leads to. With name you can attach a name to an anchor, so that other anchors

can in turn link to it.

(Forward compatibility note. In XHTML 1.1 name is replaced by id.)

A URL takes the following form:

protocol://domain/path#named_anchor

The protocol for web pages is usually http (HyperText Transfer Protocol) or its secure variant https.

However, several of these components are optional. A URL can just be a relative path (for example 'wines/french/red/bordeaux.html' on the website of a wine lover): in that case, the address will be calculated from the page that contains the link.

A URL can also just be a service and domain name: 'http://www.tomsawyer.us/ ' leads to a website with that address; the web server of that site is supposed to figure out which document you want, which will typically be called 'index.html' or 'default.htm'.

Elements deconstructed

An (X)HTML document consists of elements. These elements are constructed as follows:

<tag>contents</tag>

An opening tag may contain attributes, and attributes are often connected with attribute values, like so:

<tag attribute1="value" attribute2="value2" attribute3>

(Note: all attributes must have values in XHTML so attribute3 is valid HTML but invalid XHTML.)

The tag that closes an element is just like the opening tag, but has a forward leaning slash in front of the name, and cannot contain attributes:

</tag>

Not every element can contain every other element. The (X)HTML standards defines per element which other elements it can contain. The permitted combinations vary from version to version.

On a very abstract level, it can be said that elements are either block level elements or character level elements. Block level elements are larger elements: paragraphs, table cells, list items, forms et cetera. They can contain all character level elements and sometimes also other block level elements. Character level elements are generally smaller--links, emphasis, images et cetera--and can often only contain other character level elements.

For instance, the following is valid HTML:

```
<h1><a>Valid HTML</a></h1>
```

But this is not:

```
Invalid: <a><h1>invalid HTML</h1></a>
```

Validity

The term 'valid (X)HTML' was already mentioned a couple of times. Since webpages are supposed to be authored by people, and since people make mistakes, web browsers tend to be extremely forgiving towards those mistakes. They will even try to correct your mistakes.

Still, there are several reasons why you should try and mark up a webpage with valid (X)HTML:

- different browsers may correct your mistakes differently
- future browsers will not be as forgiving - particularly with XHTML.
- valid (X)HTML is easier to read and maintain
- when trying to correct bad mark-up, it helps if you are not side-tracked by other possible errors.

The organization currently responsible for maintaining the (X)HTML standard is the World Wide Web Consortium. It runs a validator service that you can use to check if your (X)HTML is indeed valid. You can find it at http://validator.w3.org. Whilst learning (X)HTML is a good idea to validate every page you write to confirm you have understood the rules correctly.

A common mistake is to forget to start every document with a DOCTYPE. A document without a DOCTYPE is automatically invalid (HTML version information). Note: many texts erroneously state that the DOCTYPE is optional. It is true that all major browsers will forgive the absence of DOCTYPE but this does not make the page valid. The appearance of a page may vary noticeably between different browsers if the DOCTYPE is omitted as each browser has its own peculiarities when rendering such pages.

Exercises

Time to have some fun. The following exercises will let you make some simple webpages and websites. The goal is to teach you the power of the several different ways of linking.

Exercise 2-2

Copy the example web page above to the clipboard and open http://validator.w3.org. Paste the example in to the 'Validate by Direct Input' section and click on 'Check'. Is the example valid?

Exercise 2-3

If you have an anchor <a name="anchor1"></a>, then <a href="#anchor1">link</a> will link to it. That means that when you activate the link, the webpage will be displayed starting at the anchor (rather than as usual from the top).

Make a copy of the webpage you created in Exercise 2-1, and save it as 'exercise2-3.html'. Change this file to include a 'menu' of hyperlinks at the top that link to the headings of the different subsections (Family, Friends).

Exercise 2-4

There is a hybrid form of book and game called Choose Your Own Adventure (CYOA). In such a game-book, you read a bit of text as in a normal book, but after a (usually short) while, you get to make a choice as to how to continue.

For instance:

Sue Walkins was sitting in the tub, soaking, relaxing. Her rubber ducky was chattering away happily when suddenly a pike grabbed it from below and dragged it down. Sue

- dived into to the water to save the ducky at page 89

- pulled the plug to empty the bath at page 24

In this exercise, you will write a short CYOA, in which the choices are represented by hyperlinks that will lead to the text continuing from that choice. Every 'chapter' must be a separate webpage.

Keep it snappy and don't spend too much time on this. Ten to twenty webpages should be sufficient. The story does not need to be good or finished.

Tip: create a template HTML file, on which you base all subsequent chapters.

Exercise 2-5

Create a webpage and save it as 'exercise2-5.html'. The webpage should contain a short, informative text about a subject of your choice. It should contain at least three working links to external websites about the subject.

Creating a Simple Webpage

Images

Including an image on a webpage is done using the img element. In HTML up to version 4.01, this was one of a few special case elements, in that they are empty. That is to say, although you use a tag <img> to include an image on a webpage, you do not use </img> to indicate the end of the element.

The img element has two obligatory attributes: src and alt.

src takes a URL as value. A URL indicates the 'address', the location of the image.

If an image is located in the same folder as the web page that includes it, the URL consists merely of the file name of the image.

The alt attribute contains a textual description that functions as the image when the image cannot be displayed. For instance, if the image is a photo of a lake with a castle, you could have the following code:

```
<img src="lakecastle.jpeg" alt="photo of a lake and a castle">
```

When the purpose of the image is decorative, it can be wise to use an empty alt value. That way, when the page will be displayed, the 'decorative' text will not interrupt the flow of the page's main text.

```
<img src="prettypattern.jpeg" alt="">
```

However, when the image has a function to fulfill on a webpage, the presence of an alt text is very important for visitors who do not get to see the image. For instance, many webpages have menus of links, where the links are represented by images. This allows the page author to clearly distinguish the menu items as such.

```
<a href="family.html"><img src="button-family.png" alt="Family"></a>
```

The img element lets you embed an image on a page. You can of course also link to an image that you do not want to display on the page, because it has no role there. For instance, if you want to offer people the chance to download photos you made, you can offer links to those photos. For that you use the same a element that we have been using to link to other webpages:

```
<a href="lakecastle-large.jpeg">Photo of a lake and a castle (JPEG, 512 kilobyte)</a>
```

Note how you can create links to every file that can be located using a URL. Also note how we provide feed-forward to the user. By indicating that a photo is stored in the JPEG format (a very common file format for photos) and by indicating the file size, we give visitors the opportunity to decide whether they A) can use a file of this format, and B) whether they are willing to download a file of this size.

Pre-formatted text

(X)HTML contains many more elements (for example, HTML 4.01, contains 91 different elements), but for now we will discuss only one more before moving on to the style of webwriting.

The pre element allows you to retain plain text formatting (as discussed shortly at the beginning of this chapter). This means that within the element, consecutive spaces, tabs and hard returns will not be collapsed into a single space or soft return.

There is little use for this element. It stops the text from reflowing neatly when the browser width is reduced or expanded, causing visitors to scroll horizontally, which websurfers generally hate to do. (X)HTML and its companion lay-out language CSS have plenty of options to display line-breaks and indentation. Also, it is pretty meaningless in non-visual browsers.

However, when you wish to copy preformatted text from other documents, it may be handy to use the pre element until you have the time to mark that text up.

Example:

```
<pre>1  2    4        8</pre>
```

Further reading

Later during this course, we will discuss further elements. However, the intention of this course is not to make you fluent in the HyperText Mark-up Language; it is to make you fluent in authoring webpages.

Generally, to fully comprehend something requires that you fully comprehend its form first. You cannot be a successful karate-ka if you cannot perform the various moves. You cannot be a successful French speaker if you have not mastered its grammar and vocabulary first. However, knowing all the ways to hit someone does not make you a good karate-ka. And knowing all the words and rules of the French language does not prevent you from becoming a mumbling baboon the next time you need to speak French.

To fully comprehend authoring webpages, you need to look beyond the language in which you write them. This is what we will do in most of the further chapters of this book.

There are at the time of writing this several Wikibooks under development that aim to teach you (X)HTML. These have been referenced elsewhere in this book. *where???*

The official (X)HTML Recommendations of the World Wide Web Consortium can be found at the HyperText Markup Language (HTML) Home Page. At times, they can be pretty hard to read, but they represent the last word in any discussion of what is valid (X)HTML and what is not.

Further, one the authors of the HTML 4.01 Recommendations, Dave Ragett, has written a couple of handy guides to HTML and its companion lay-out language CSS, which you can find at http://www.w3.org/MarkUp/#tutorials. If there are points in this and later chapters that you do not fully comprehend, you could do worse than study Dave's texts. They are much clearer than the official specifications, and short enough to study alongside this text.

More exercises

The following exercises are optional. You can use them to practice putting images on your webpages.

Exercise 2-6

Download the following images, and put them all on a webpage that you will save as 'exercise2-6.html'. Think of useful alt texts.

To download linked files, a lot of browsers contain a Save Link As function. In graphical browsers using a mouse, this function is often part of the context menu. On the PC this means you have to click you right mouse button, on Mac OS this means you have to press the Ctrl-key and press the mouse button.