

# Tools and Techniques for web page design

*Sebastian Rahtz*

November 2000

## 1. Tools 1: Editor and Templates

The simplest way to make good web pages is to *plan* your strategy, and implement it consistently using a good HTML editor. Do a design, perhaps using tables or frames if you need them, and carefully prepare template HTML files. Good editors will validate HTML as you write it, offer 'boiler-plating', and help you make consistent HTML.

Web page templates:

- Can help in web site management
- Provide consistent style
- Help information providers

There are some examples at <http://www.netskills.ac.uk/resources/examples/website/histhome.html>

The big problem is that you cannot automatically change the style of pages after they been created.

## 2. Benefits of using templates

- 'Shell' documents
  - information common to all files
- Consistent format
  - logo
  - link to home page
  - email contact
- Simple HTML file
- May be open to change by information provider

## 3. Example template file

```
<HTML><HEAD> <TITLE>University of Oldsville Template</TITLE> </HEAD>
<BODY> <H2>Oldsville University</H2> <IMG SRC="http://www.netskills.ac.uk/resources/
examples/website/images/shield.gif" ALT="Oldsville Logo" Height="150" Width="130">
<P><HR> <!--Content starts here--> <!--Content ends here--> <HR><P> <A
HREF="http://www.netskills.ac.uk/resources/examples/ website/histhome.html">University
of Oldsville Home Page</A> </BODY> </HTML>
```

## 4. Tools 2: external template files

In this technique, the web server inserts standard text templates into your Web page. This can be used to automate complex designs. The advantages are:

- The template can be provided by an external file

- It cannot be changed by information providers
- You have only one file for many web pages
- Ease of management
- The web file is delivered via Server Side Includes (SSIs)

Note that you can only see the effect when you view the file through the real web server. More details about SSI's in `ssi-slides0.html`.

## 5. Tools 3: style sheets

One of the quickest ways to achieve a consistent *look* is with colour, white space, and fonts. Cascading Style Sheets are excellent for this. We consider this in `css-slides0.html`.

## 6. Tools 4: web site managements tools

If you have more than a few pages, you need a tool to manage them. You should track the history of your pages, know where you have used 'New' flashes, be able to trace links if you move pages. Don't forget that consistency in design is internal (for you) as well as external. If you use metadata tags, you need systems to look at them, update them, etc.

As an example, HotMetal supports

- Managing the 'publishing' of pages to an FTP server
- Drag and drop creation of sites and links in pages
- Friendly editing of tables
- Creation and editing of CSS files
- Database import
- Changing URLs from 'local' to 'public' when you publish

Other products to consider are FrontPage, Dreamweaver, Home Site etc.

## 7. Tools 5: writing web pages in a higher-level language

HTML is a primitive language for describing pages. If you step back and write your material in an entirely logical way (using, eg, XML), you can then *generate* HTML pages (or other formats) on demand. This makes it easy to be consistent, and eg automate standard links. The look of the pages is determined by style sheets which depend on the content, not embedded in the page. The transformation from a logical markup language (HTML) to a presentation one (HTML) can be done

- On your own computer, before you put the pages upon the web site
- On the web server, whenever the pages is requested
- In the browser, when the reader downloads the file

## 8. The Light and The Dark

**Bad** `<p><b>Perec, Georges</b></I>Life - a users manual. Collins, 1988. Translated from the French </I> [La vie mode d'emploi] </I> by David Bellos. xviii+581 pp. 841.941</I> Literature - French - 20th century</p>`

1. what is the author's name?

2. how many books did Collins publish in 1988?
3. have we got any books of 20th century French literature?

**Good** <author>Perec, Georges</author> <title>Life - a users manual</title>  
 <publisher>Collins</publisher> <publDate>1988</publDate> <note>Translated  
 from the French [<title>La vie mode d'emploi</title>] by <translator>David  
 Bellos</translator></note> <pages>xviii+581</pages> <ddc>841.941</ddc>  
 <keywords><term>Literature</term><term>French</term> <term>20th  
 century</term></keywords>

## 9. XSL basic concepts

An XSL specification consists of a set of *templates* which match elements in the XML, and produce a *result tree*, which is itself (almost) always XML. Thus, from <address><name>Sebastian Rahtz</name></address> this stylesheet <xsl-template match="address/name"> <h1>Name: <xsl-apply-templates/></h1> </xsl-template> would generate <h1>Sebastian Rahtz</h1>

## 10. XML displayed on the Web

1. XML to HTML pre-processed translation (using, eg, XSL)
2. XML to HTML run-time translation (in the Web server; cf Apache Cocoon project)
3. Direct display of XML decorated with CSS (IE5)
4. Direct display of XML with XSL stylesheet mapping to HTML (IE5)
5. Direct display of XML with XSL stylesheet mapping to native 'formatting objects' (incomplete implementations)

## 11. Tools 6: dynamic web pages on the server

There are many ways to make your web pages more dynamic, respond differently to different users, and fetch information from databases. These usually involve embedding commands in the HTML file (in languages like Python, TCL, Perl, or special-purpose languages like those in ASP or PHP). Like server-side includes, these depend on support in your web server, and can be quite dangerous. Most of these techniques are *not* supported on our standard public web servers.

If you plan to delve into this, look first at what your web server (both politically and technically) supports.

## 12. Tools 7: dynamic web pages on the client

The alternative to making the *web server* work for you is to deliver scripts to the *browser* and have it do the work. The commonest example of this technique is Javascript. It has the great advantage of efficiency, because the user does not have to keep referring back to the web server.

The downside of this technique is that you cannot be sure what happens on the client machine.

## 13. Tools Summary

In ascending order of complexity:

**Everyone** Plan your web site

**Everyone** Use CSS to set the look of your page

**Do not want to see code?** Use web site building tools

**Want standard headers/footers?** Use server-side includes

**Want to maintain lots of text?** Mark it up in XML

**Want to access databases?** Use ASP or PHP, but don't expect it to be easy

**Want plain HTML to be delivered?** Script your pages in PHP or ASP

**Know what browser your readers have?** Do the fancy work in Javascript