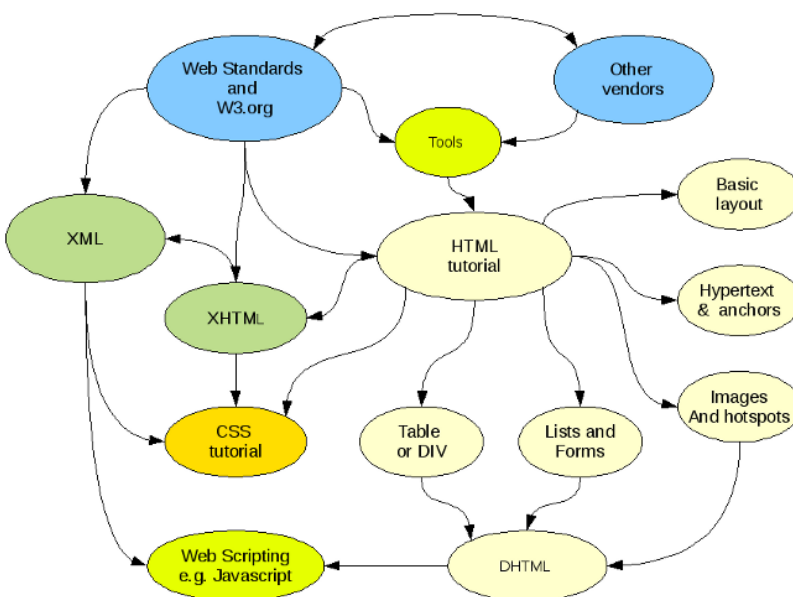


GETTING STARTED WITH HTML

Introduction to HTML

HTML is the language of the World Wide Web (WWW). This short course will enable you to *read* a HTML document and understand broadly what is going on. By the end of the course you will have the basic knowledge needed to create your own web pages, If it has wetted your appetite for more, there are numerous Web sources and books which cover both HTML and JavaScript. Its not possible to talk about Mark-up languages and the Web without providing a plug for the WWW consortium whom manage the web standards and development.

The mind-map below provides an overview of the basic web technologies and indicates the links between the various topics. XML, Javascript and PHP will be touched on, insofar as they impinge on HTML and CSS (Cascading Style Sheets).



HTML

The HTML tutorial itself provides a series of examples along with the code and is designed to be self paced. You are encouraged to experiment, and to build on the basic examples. An attempt has been made to keep the coding to a minimum, so some coding may be possible during a session. Apart from specialist software (Dreamweaver, Quanta Plus etc.) some browsers include an editing facility (e.g. Sea Monkey composer and the W3's own Amaya which is used as a test bed for new developments. However all you need to get started is a simple text editor, such as (windows) notepad or Linux (gedit).

The accompanying notes provide tips and things to note along the way, for example relating to accessibility and stopping your hypertext links becoming a bowl of spaghetti.

Warnings are also given as appropriate, e.g. on compatibility issues, or just plain bad technique such as the use of the tag.

Images can be used in both foreground and background and are crucial to the look and feel of a web site, as are colour schemes. What constitutes a good (or bad) layout? after all the purpose of a web site (commercial at least) is to attract customers. Even if the purpose of the site is to inform or educate, layout plays a crucial role. Hotspots provide an alternative (and perhaps more interesting) way to present lists.

Tables and Dividers (or layer tags) are fundamental tools when presenting information, the latter is very closely associated with CSS and Javascript. Some comparisons are made between the two techniques.

Dynamic HTML

While forms and their associated tags are covered, they are of little use without some form of scripting, which leads to Javascript and adds the D(ynamic) in DHTML Applets or Flash or just raw (Javascript) code, you may be surprised at what you can do with just a few lines of code!

Scripting languages

Scripting can be either **Client side**, that's on your (the users) machine or a **server side** script, where (pre or post) processing is carried out. In particular server side scripting allows websites to be *hooked up* to databases. While Javascript can be used in both situations, **PHP** is becoming increasingly popular, especially server side. ASP (Active Server Pages) are Microsoft's answer to Javascript/PHP and, being proprietary, will lock you into that vendors products.

Cascading Style Sheets - CSS

What can be achieved in terms of layout with just HTML is limited. **Cascading Style Sheets** provide a much more flexible alternative, as they make the various HTML tag properties accessible to coding. A comparison between the CSS way and HTML way to layout web pages. Cascading Style Sheets can also be used with XML (e Xtensible) Mark-up Language. Like HTML Cascading Style sheets also have their limitations. XSL is a style sheet (just for XML) that is tackling some of those issues.

XHTML

With different browser vendors competing with one another, compatibility went out of the window and a lot of time is still being wasted ensuring web pages work on different browsers. By adding (non standard) **features** to browsers, under the guise of assistive technology, had the effect of locking you into proprietary products, and promoted sloppy coding. HTML is moving towards XML and a halfway house is XHTML as the world attempts to clean up its code.

XML

XML is a subset of SGML and so to is XHTML. which is where we started this session. XHTML is a transitional stage between (untidy in terms of standards) HTML and XML. It is a major attempt to clean up the web. Browsers will not render XML code that contains errors which do not conform to the W3 standards. So as well as being *well formed* XML documents can be further validated against **schemas**. The DOM (**Document Object Model**) provides a need way to parse XML documents to extract the content needed. Important for business and an EDI (Electronic Data Interface).

XML can be styled with either CSS, or it has it's own style

language **XSL**. Finally have you ever wondered what you missed when Google turns up a million results which you can't be bothered to browse through? The web was intended for humans not machines. Meta tags, (data about data), keywords etc. are an attempt to address that problem, but XML will surely make that task easier. Some vendors are slower than others when it comes to providing support for new features, which gives rise to incompatibility between browsers, so you should test your web site an a range of browsers.

Source : <http://webstyle.soslug.org/content/html-getting-started>