

KXT105/205 - Dynamic Web Development

Assignment 1

Due: 03:00PM 23rd August, 2006 (Wednesday of Week 6)

Task One

For this assignment you will be required to produce a website using content that is provided in this document. Your website will have to conform to a basic design which will also be provided; however the particular style (colour scheme, subtleties of the layout etc.) you choose to apply to your site will be up to you.

Task Two

Doing the advanced assignment involves producing a second stylesheet, which is completely interchangeable with your original stylesheet but which changes the look and feel of the website considerably. It is not considered enough to just change the font of your paragraphs, you must change every element in at least one significant way in order to produce a website which, although has the same content, looks entirely distinct from the original (see the CSS Zen Garden for good examples of this <http://www.csszengarden.com>).

A total of 75% of all marks will be allocated to the successful completion of Task One, with the remaining 25% allocated to Task Two. See the sample marking scheme at the end of this specification for more detail.

Technical

- Your web site must adhere to XHTML strict guidelines.
- The directory structure of your web site should be as follows
 - o A top level directory called 'dwd_ass1' (without the quotes). This must contain the files index.html and form.html
 - A subdirectory that contains the file xhtml.html
 - A subdirectory that contains the file css.html
 - A subdirectory that contains any graphics files you use in the website
 - A subdirectory called 'styles'(without the quotes) that contains the stylesheet(s) that you use in the website.
 - The basic style sheet must be called 'basic.css' (without the quotes).
 - The advanced style sheet (if it is included) must be called 'advanced.css'

Hints

- It is suggested that in order to do well in this assignment you will have to define at least 10 selectors that have 3 or more properties, and you will probably want quite a few more than that! These are not firm guidelines, but it is considered unlikely that most students will find it possible to produce a site which both looks good and demonstrates your full comprehension of XHTML and CSS without this level of detail
- I am not looking for people to *only* use styles and techniques outlined in the Self Studies, I will look more favourably upon those who use elements discovered *from their own personal searching and of course from reading the extra reading material linked in the Self Studies*
- The content provided is just raw text. It is up to you to mark all this text up appropriately. There will be parts which should be headings, parts which should be in paragraphs, parts which should be emphasised, parts which should be in a table, etc. etc. etc. and it is entirely up to you to determine which parts are which and how best to mark up the text. Marks will be awarded so long as it seems reasonable and sensible
- The layout provided is just a basic description. It is up to you to bring this layout to reality, and make something of it. Marks will be awarded as long as your implementation of this design is practical, effective, and extensible
- This unit is not about design of Web pages, but do *try* and make your site attractive and all text easily readable
 - o Sans Serif (i.e. ARIAL) fonts are generally more easily read on computer monitors than Serif fonts (i.e. TIMES NEW ROMAN)
 - o These sites might help you pick appropriate colour schemes
 - http://www.steeldolphin.com/color_scheme.html
 - <http://gmazzocato.altervista.org/colorwheel/wheel.php>
- For an excellent example of how CSS should be done visit <http://www.csszengarden.com/>
 - o You can change the styles displayed on the site by clicking the various links
 - o You can also view the CSS used for each style, a GREAT place to get ideas for how to do things in your site
 - o We will not be amused if you directly copy a style from this site
- Try and stick to CSS1 elements, and when using CSS2 elements make sure they're widely supported, or else your page may display wrong on other browsers (which will probably lose you marks)
 - o Site should render fine on both Safari and Firefox
- Avoid using XHTML markup for style when you can be doing it in the stylesheet
- You can test if your site complies to XHTML strict standards by using the !DOCTYPE for XHTML strict, instead of transitional
 - o `<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">`
- There is also a HTML validator available from W3C or WDG that will check your HTML files and provide useful feedback on their conformance levels and any errors they may have
 - o <http://validator.w3.org/>

Submission

OffLine Components

Assignment cover sheets

- You should submit an assignment cover sheet signed by you. Submission box is next to the school office (in Hobart) and help desk (in Launceston).

OnLine Components

Submission Method :

- Following files and folders should be copied into the “kxa105submit” or “kxa205submit” folder in your UNIX home directory. ‘M’ drive is UNIX home in PC and students have to mount the UNIX home in Mac Lab (In the finder: Go-> Connect to Server-> and type smb://alacritas/<username> or smb://lawson/<username> in the Server Address field.)

Due Date & Time

Due: 03:00PM 23rd Aug. 2006 (Wednesday of Week 6)

Content

The title for each page is “The Joys of CSS based Web design”

You should have 4 pages:

- A main page (index.html)
- A “Form” page (form.html)
- A “CSS” page (css.html)
- An “XHTML” page (xhtml.html)
 - o The CSS and XHTML pages should be stored in appropriately named sub-directories

General

Each page will have a Menu area which contains the links to all four (4) pages.

Each page will have a Footer area which contains the following links to validators:

Validity - [xhtml](#) [css](#)

Each page will have a Heading and a Sub-heading which contain text as defined below.

Each page will have a “catch phrase” which contains text as defined below.

You are entirely responsible for making up the sections of text with appropriate tags, classes and ids. If it seems to be a heading, it probably is, so make sure it is marked up as a heading! If it seems to be a table, mark it up as a table! If you think it’s a good place to link to a site, then mark it up as a link to that site!

Main (index.html)

Heading: CSS Based Web design

Sub-heading: The Beauty of CSS Design

Catchphrase: Providing insight into what can be achieved through CSS-based design.

Sweet, Sweet Interchangeability

Typing in a search on Google for "div vs table" is a sure way to find yourself a whole world of ignorance, although I suppose the ability to sift through the ignorance and find the facts are an important part of what makes us Computing professionals.

If you take a look at the CSS Zen Garden you'll see a perfect example of one of the largest benefits of CSS based Web design, the endless interchangeability of it.

Going back to the div vs tables argument, we can argue that using CSS to arrange our Web layout gives us the flexibility of being able to completely alter our websites layout easily. On top of this it has benefits when we come to making our webpages user-friendly and interactive, but we'll come back to that when we move on to Javascript and Assignment 2.

XHTML (xhtml.html)

Heading: XHTML

Sub-heading: Well defined HTML

Catchphrase: XHTML, better defined and more powerful than HTML4.01.

Ambiguity? No thanks

Never forget to define the right !DOCTYPE when writing in XHTML. The penalties can be ugly.

Previously with every new incarnation of a Web browser a whole range of new tags have become available to the Web developers, along with a completely different view on how things display. The best known and worst culprit for this kind of behaviour is generally thought to be Internet Explorer, although no browser to date is exempt.

By introducing the !DOCTYPE we are able to define to the browser how we want our pages displayed, instead of the browser just doing as they please. If we forget to define a valid !DOCTYPE for our page, the browser will just go back to its old tricks, assuming it can display every element however it pleases.

This can be noticed easily when using CSS, as common CSS methods of doing things (such as centering an element) are often misinterpreted by many browsers if they are not made aware that we are using XHTML vs HTML4.01.

Using Tools as Intended

A hammer is used for banging in nails, a screwdriver is used for turning in screws. We can use a hammer to bang in a screw, but it's harder and doesn't work as well. Similarly we can use a screwdriver to bang in a nail, but it might take you a while! This same analogy extends into the world of web-design.

Consider the div vs table argument that was mentioned on the main page. Using a table to design our layout is counter-intuitive and not as intended. Tables were just designed to display tables of data such as the one in Figure 1. CSS on the other hand was designed to define the appearance of our site.

	Site A	Site B	Site C	Site D
Time to load table layout(s)	5	8	13	3
Time to load CSS layout(s)	4	8	11	3

Figure 1: Time taken to load various sites with table layout vs. CSS layout (not real statistics)

Similarly, when we want to make an ordered list we don't write out the numbers, we use the ol tag. The great thing about using an ordered list here, is that it's an excuse for me to make the page text far too long so I can test how your page responds to this. Normally it is undesirable if a page requires scrolling to read, although sometimes it's unavoidable.

An ordered list!

1. The first item
2. The second item
3. The third item
4. The recurring item
5. The recurring item
6. The recurring item
7. The recurring item
8. The recurring item
9. The recurring item
10. The recurring item
11. The recurring item
12. The recurring item
13. The recurring item
14. The recurring item
15. The recurring item
16. The recurring item
17. The recurring item
18. The recurring item
19. The recurring item
20. The recurring item
21. The recurring item
22. The recurring item
23. The recurring item
24. The recurring item
25. The recurring item
26. The recurring item
27. The recurring item
28. The recurring item
29. The recurring item
30. The recurring item
31. The recurring item
32. The recurring item
33. The recurring item
34. The recurring item

Want to return to the top?

CSS (css.html)

Heading: Cascading Style Sheets

Sub-heading: The answer to many design problems

Catchphrase: Making pages prettier than the ones on [THIS](#) site since 1997.

Phenomenal Cosmic Power

You should be getting the idea by now that CSS allows you an incredible level of control over the design and layout of your page, and indeed your entire site. Combining these skills with programming languages such as Javascript, PHP, JSP, ASP and others it is possible to make entire incredible Web applications.

Form (form.html)

Heading: A Form

Sub-heading: Good form that

Catchphrase: Senselessly added to the site to give you experience with forms.

Your form should look like this: (“Good Form That” is outside the form as a heading)

Good Form That

Name:	<input type="text"/>
Do you like green eggs and ham?	<input type="radio"/> Yes <input type="radio"/> No
What year were you born?	1980 <input type="button" value="v"/>
<input type="button" value="Submit"/>	<input type="button" value="Reset"/>

But the styling and colouring of text etc. is still up to you. Make it suit your site.

Layout

For the basic CSS implementation you are to produce a fairly standard 3 column affair. The Menu on the left, body of text in the middle, and the catchphrase off to the top right. The headings should be at the top-middle, and the footer should be at the bottom-middle.

For the advanced implementation you are allowed to create any layout you choose, so long as it is reasonably different to the basic layout. You can put the catchphrase in the middle of the text if you want, the menu across the top, the headings down the side. It's up to you.

Sample Marking Scheme

	<i>Out Of</i>	<i>Awarded</i>
BASIC		
Content Markup	20	
The provided content is used	2.5	
Divs used correctly	2.5	
Spans used correctly	2.5	
no HTML styling	2.5	
Links present where appropriate	2.5	
Headings used appropriately	2.5	
Other markups	2.5	
Formatting and comments	2.5	
CSS	40	
Reasonable number of different styles defined	15	
Reasonable number of different rules defined	15	
Layout correct	10	
Requirements Met	15	
index.html in a directory called <code>dwd_ass1</code>	1	
form.html in a same directory as index.html	1	
xhtml.html in its own subdirectory	1	
css.html in its own subdirectory	1	
graphics in graphics subdirectory	1	
stylesheets in styles subdirectory	1	
Successfully validates CSS	1	
Successfully validates as strict XHTML	2	
Renders fine on both Safari and Firefox	2	
Basic site is reasonably attractive	4	
<i>SUB-TOTAL:</i>	75	
ADVANCED		
Each element is sufficiently altered	10	
Site looks significantly different	10	
Advanced site is reasonably attractive	5	
<i>SUB-TOTAL:</i>	25	
<u>GRAND TOTAL</u>	100	