Getting Your Message Out There - Notes

The Crux of the Problem

As activists, we are up against the huge and powerful marketing machines of companies and institutions with almost infinite resources at their disposal. Their capacity for getting people's attention is much higher than ours in the mainstream media. However, with the Internet, we come somewhere near a level footing with them in terms of opportunity to grab people's attention, engage with them, and with a bit of luck, change how they think. **All it takes is knowing how!**

The nature of the World Wide Web

We can make websites (and to some extent, printed material) that compete with those of the companies and institutions. Because most web usage is based on people searching for things, with some effort, the right techniques and a bit of luck, we can get our search results to appear alongside (or even above if we're really lucky) theirs!

Structure of a fairly simple no-frills web page

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html>
<head>
  <title>Home - My First Website :: Teaching you the basics FAST!</title>
</head>
<body>
<div id="container">
  <h1 id="main_heading">My First Website</h1>
  <h2>Getting my hands dirty</h2>
  <p>I decided to make a mini website to see what I could do. If you'd like to
  read more about that, take a look at <a href="getting_started.html">getting
  started</a> to see what I did.</p>
  <h2>Some of my other interests</h2>
  <ul id="interests">
    <li>Bike riding and repairing</li>
    <li>Climate change activism</li>
    <li>Low impact houses</li>
    <li>Music</li>
  </ul>
  <h2>Getting in touch with me</h2>
  <img src="images/me.jpg" alt="A photo of me" width="320" height="240" />
  <!-- This is a comment which doesn't appear when the site is viewed through a
  web browser, but which is helpful to remind me of something important. Notice
  that the image tag above has a closing slash at the end? This is because
  it's a "singleton tag" (one that doesn't enclose anything). -->
  <p>You can <a href="mailto:blah@example.com">email me here</a>.</p>
  <h3 class="footer">Thanks for visiting my site – check back soon!</h3>
</div>
</body>
</html>
```

The basics of writing good HTML and getting good search engine ratings:

- The <title> tag is the most important on any page – it should contain the title of the page, the name of the website, and a very brief description of what the site is about.

- Getting lots of other people to link to your site is the best way to get higher ratings – but the “quality” of the incoming links is also very important. Search for sites that cover the topics your site covers, and contact those that appear high in the results (first three pages) to see if they will link to you. Use your judgement to pick relevant sites – a site that is directly relevant to yours is more likely to link to you. One link from an established site that already has a good rating will boost your rating much more than lots of links from your mates' MySpace pages!

- Content is king – remember that computers can't read text contained within graphics files (it's not text – it's a picture), so it's usually better to use the HTML heading tags <h1> to <h6> (though I almost never go beyond <h4>) to keep the most important text on your pages as something that search engines can understand. There are more advanced techniques you can use to both use a heading tag and replace it with a graphic, but it is perfectly possible to make regular HTML text look nice using Cascading Style Sheets (CSS).
- Keep your HTML clean and well structured – avoid using obsolete tags that are concerned purely with appearance, since they don't add any meaning to the content. Such tags include: <font>, <b>, <i>, <u>, <blink> and so on. Instead, use heading tags, <strong>, <em>, etc.
- Validate your pages at http://validator.w3.org/ - many layout headaches are caused by simple things such as forgetting to close a tag. The first step to getting a page to appear consistently and correctly (and so search engines can understand it) is making sure the markup is valid.
- Don't be tempted by the lure of Flash – although it's improved a bit in recent years, it still suffers from the same problems that images do: any text appearing in a Flash file cannot be read by a search engine, so in terms of getting relevant visitors from a search engine based on the contents of your site, it might as well not be there. Your time is better spent learning to make “real” web pages (in plain HTML) look good.
- Avoid “WYSIWYG” (“What you see is what you get”) editors such as Frontpage and to some extent Dreamweaver too. The web is not a piece of paper, so the WYSIWYG way of working really doesn't apply very well, as every web browser works differently. It's far too easy to end up with bloated pages full of unnecessary, virtually meaningless markup. Writing HTML by hand isn't really that hard when you use it as intended – as a markup language for adding meaning to text. The best development tool for a website is a text editor.

Making it look nice

To begin with, don't worry too much about the appearance of your pages – that will come in time. The most important thing in making an **effective** website (that will get people coming in and reading it via search engines) is to keep the content clean and well structured, with the important things such as headings marked up properly (using heading tags), rather than simply made to “look right” in the browser (the old fashioned way using meaningless <font> tags).

In the past, websites with anything more than a basic layout were always done using tables, but this is also a bad idea. Table-based layouts can be a nightmare when you want to move things around later, as they are rather inflexible and unwieldy to work with. They also add lots of extra bloat to your pages, making your site slower, and reducing the ratio of content to markup.

From an academic point of view, it's incorrect to use tables for anything other than tabular data. There is some middle ground in combining tables and CSS to make “hybrid” sites, but speaking from bitter experience, I don't think it makes life easier in the long term – quite the opposite!

Always remember the “KISS” philosophy – **Keep It Simple, Stupid!**

You are better off to make a simple site with really clean, structured markup that you can later go on to improve the appearance of using Cascading Style Sheets.

Cascading Style Sheets (CSS)

Unfortunately it's a bit beyond the scope of these notes to cover CSS in any depth, other than mentioning it so you can look it up later! But the good news is that when you have a web page with nice clean markup, it's a breeze to start making it look good using CSS.

To begin with, you don't need to make use of any CSS, because web browsers come with a standard set of rules that will make any cleanly marked up page look reasonably OK (albeit a bit plain). Starting to use basic CSS will immediately improve things – you can change fonts, add colour (though for CSS purposes, that's spelled “color”), add borders, underline text, add space between things, and then go on to do more advanced things like positioning things on the page.

Unfortunately, older web browsers can be a little inconsistent with their handling of more advanced CSS, so your best bet is to cut your teeth on the simple things such as colours and text formatting, before moving on to the harder stuff.

Tips:

- Items in HTML can be assigned a “class” or an “id” (sometimes both). Classes tend to be used where there are lots of something (eg. lots of photos on a page, all requiring the same formatting), while IDs tend to be used to identify an individual object (eg. the main heading or the navigation menu). CSS makes use of classes and IDs to identify what formatting to apply to each item. Classes in CSS begin with a dot, like “.news_item” or “.photo”, while IDs begin with a hash, like “#main_heading” or “#self_portrait”. Items with no class or ID take the formatting for their type, so “h1”, “p” and so on, without any prefix.
- Once you understand a bit about how HTML works, it becomes relatively easy to look at the code from other people's sites, and borrow bits of their CSS to make your site look nicer, without really understanding how it all works. By identifying the class or ID (or none) that applies to each item, you can find the corresponding code from the CSS, and borrow it. The best way is to play around and try it out!
- The best way of checking your CSS is working is to... check it's working, in a web browser, and preferably a few web browsers. (Firefox, Internet Explorer, Opera, Mozilla, Netscape – all available free.)
- When developing CSS, it helps to use a web browser with the fullest support, and then worry about the “broken” browsers (IE6, IE5, older versions of things) later, once you have it working properly in something like Firefox, Opera or Safari (on Apple Mac).
- You can validate CSS at http://jigsaw.w3.org/css-validator/
- The “Web Developer Toolbar” for Firefox from http://chrispederick.com/ is also infinitely useful.

Graphics as a whole

Graphics is a huge field, and again, something that goes way beyond the scope of these notes. However, with a few pointers in the right direction, it's possible to practise and get a lot better.

First things first – keep a scrapbook! Whenever you see a website you really like the look of, take a screenshot (or photo, if it's a poster) of it and save it for later reference. That way, when it comes to thinking up a design for your own work, you will have plenty of things to look at for inspiration.

Examining the work of other designers and working out what you like or dislike is one of the best ways to improve your “design eye” and start coming up with the goods yourself.

I often borrow design ideas (though never whole designs) from websites I like the look of. Trying to rip-off someone else's style wholesale is an interesting exercise in itself – by the time you've tried that a few times, chances are you'll have gained some (bitter-ish) experience, come up with something completely different, and probably taken one step closer to forming your own style!

Tools

The following bits of software are cross-platform, available free, and are all I use for graphics:

- Inkscape: http://www.inkscape.org/
- (Excuse the weird name) The GIMP: http://www.gimp.org/

I usually start out in Inkscape (which is a vector graphics program – everything is a mathematical curve, so if you draw something nice and then need to change the size, you don't lose any quality), and then move over to The GIMP (which is a bitmap graphics program – everything is coloured, fixed size square dots, which is what it needs to be for the web) to finish things off. There is a fair bit to learn for both programs, but really the best way is to have a play and practise to creating things (anything you like) regularly. Be patient, explore the options, and have fun doing it.

There are no hard and fast rules for designing a website, but I usually start my designs at 760 pixels wide, and try to get the most important things within the top 430 pixels of the page. This is because on an 800 x 600 screen, that's about how much space you have when someone looks at your site in a fully maximised web browser. Most people use screens larger than that, but catering for the lowest common denominator and working up from there isn't a bad way to start.

Once you get better at CSS, it becomes possible to make “elastic” designs that stretch to fill whatever width screen someone has. This can be a bit tricky, and over the years (after having spent ages perfecting it), I've come to realise that one major problem is that extremely long lines of text are hard to read, so elastic designs can be a bit of a disaster if you aren't careful.

According to various sources, the optimal line length for readability is around 2.5 lower case Roman alphabets (65 letters) in any given font – for example:

abcdefghijklmnopqrstuvwxyzabcdefghijklmnopqrstuvwxyzabcdefghijkl

I often exceed this, but not by too much!

Typography and Fonts

Again, typography is a massive field, and there are entire books written about it, as well as obsessive typo geeks who spend forever engaging in conversations about “kerning”, “tracking”, “leading”, “point size” and other obscure terms! I can't go into all that here – thank goodness!

The general rule with fonts is “no more than 3 per design” as more hampers readability. It's OK to use slightly different variations of the same font (eg. bold, italic). For the rest of the time, when choosing fonts, you should pick fonts that are obviously different to each other, rather than using fonts that are quite similar (such as Arial and Helvetica on the same document).

Exporting graphics for the web

Once you have your design more-or-less ready, you'll need to “cut it up” (divide it into sections) for the web. Header, menu, side boxes, footer, in-page graphics – it takes a bit of practise to get the hang of where to slice things!

The GIMP is very good at optimising graphics for the web. I use JPEG for photographs, and 8-bit (256 colour) PNG files for other graphics. 24-bit (full colour) PNG gets an occasional look-in, but the files tend to be rather large, and can have their own problems in IE6 and below. 8-bit PNG generally comes out smaller than equivalent GIF files when optimised, and is widely supported.

Further reading / Useful links

The 7 Essentials of Graphic Design – Allison Goodman, ISBN 1581801246

Designing With Web Standards – Jeffrey Zeldman, 0735712018

Stock Xchange – Royalty free stock photographs: http://www.sxc.hu/

AlistApart – Lots of articles on web development and CSS: http://alistapart.com/

W3Schools – Plenty of easy to understand stuff about web development: http://w3schools.com/

Steel Dolphin colour scheme designer: http://www.steeldolphin.com/color_scheme.html

Web Developer Toolbar for Firefox: http://chrispederick.com/work/webdeveloper/