

Foundation Issues of “Designing” Web Sites:

Designing Web Sites is like a hybrid of graphic design, interface design, desktop publishing and advertising . It's more like designing a house than a document. The appearance must be accounted for but also the plumbing, electrical wiring, and insulation. There is no designing without engineering. The issue is to make it coherent, easy, well organized and attractive. There are a lot of pretty sites that are incoherent jumbles of information. And there are technically proficient sites that repel with their aesthetics.

The Uncertainty Principle:

Building sites on the Web involves some very strange problems that most people would never guess. There are several important technical variables that play a part:

1. **BROWSER:** A browser is a program used to view web sites. The best-known browsers are Firefox , Microsoft's Internet Explorer and Google Chrome. but there are quite a few others. The most popular browser on Macs is called Safari. And there are another half dozen semi popular ones besides. ALL THESE BROWSERS BEHAVE A LITTLE DIFFERENTLY. A web page is code and a browser interprets that code. So when different browsers choose to interpret code differently, your pages will look (and behave) a bit differently in all of them and if you don't know how to account for these differences, your pages may not even show up in certain browsers!
2. **CONNECTION SPEED:** We tend to assume that high speed has become global but it hasn't and it is far from perfect OR consistent. Even in an age of fast connections we have to consider the impact on download time of the various things we put on a web page. If a bunch of badly prepared graphics slow the loading of your page to 20 or 30 seconds most visitors will give up on you.
3. **OPERATING SYSTEM:** Different operating systems currently in use include: Multiple versions of Windows. Various Apple systems. UNIX, Linux, etc, etc. Users with these different Operating systems will see the same pages at least somewhat differently. HTML is supposed to be unrelated to platform, but there are specific issues that arise. Animation, sound, color and fonts for example may appear differently. Certain programming codes for example may work well in recent versions of Windows but not older versions. It may not work at all on the Macintosh. Code must be chosen very carefully.
4. **THE INDIVIDUAL USERS' SETUP:** Examples: Different screen resolutions have a huge impact on the look of your site. If you design a page for a certain screen size or resolution other sizes may cause your page to undergo a disastrous change of appearance, stretching out or squashed in and looking very strange. If you require a media plug-in that they don't have installed you will sideline them. If the user doesn't have the font on his system that you chose to use in the web pages he just won't see that font and his system will substitute another, which may look bad on your page.

Under the Skin

At the surface, a web page looks rather simple, like a word processing file, it seems to be a collection of some text and graphics. But a web page is actually a series of levels of code, and the more the page is expected to do, the more complex the code must be.

1. The first level is the content, which falls into two basic categories, text and graphics. Text can come in from a word processing file or be typed directly in. Graphics must be inserted into the document through references in the code.
2. The next layer down is XHTML (Extensible Hypertext Markup Language), the language that web pages are written in. This is the code which accounts for the layout of the page including font size, orientation, indenting, tables, headings, etc. This is also where links are created, allowing the visitor to jump to new locations.
3. If the page is supposed to do things like check passwords or validate credit cards there must be another level of code, often called CGI (Common Gateway Interface) code. This may be written in any of a number of scripting languages. Such as Perl, visual basic, JavaScript or Ajax.
4. If the page is supposed to display changing variables to the customer, like products in the current catalog, there must be another level. This is the level of the database. The database must include records of all the information we need and it must be able to run through this information and gather up just what the customer needs (and leave out everything else). This choice only applies to so called "Dynamic" web sites.

Basic XHTML Tags

(X)HTML is the language that Web Pages are created in. It stands for: Hyper Text Markup Language. It takes a little getting used to but soon you will notice that it mostly consists of fairly simple messages about formatting.

This little bit of code is the basis for all the web pages you have seen:

<html>	Top and bottom you see "HTML" beginning and ending the page
<head>	the area marked as "Head" is out of plain sight and contains some technical things
<title></title>	Anything between the brackets that say "Title" is what will appear up on the title bar.
</head>	This ends the head section, see the forward slash?
<body>	Here's where the visible part of your page starts
</body>	And here's where it ends!
</html>	End of the web page

Within the "Body" tag is where we place the information visitors will actually see on our site.

Absolute and Relative Linking:

Absolute links (also known as remote or external LINKS) are very straightforward. This is a link that includes the COMPLETE address to reach the site in question. Here are a couple:

`http://www.google.com`

`http://www.amazon.com`

They can be MORE detailed, included pathways to documents in sub folders but the key thing is that they start with the HTTP and stretch all the way to wherever you want them to go.

And here's the most fundamental part...if it's a link to someone else's stuff, it got to be an absolute link.

Relative links (AKA: local or internal links) can be harder to get right and often have a more dramatic impact on the functioning of your site if they don't work. These links address the files within your site by listing their location RELATIVE TO the current HTML page. If your pages are all together in the same folder, all you need to do to link is list the accurate filename of the page you are trying to go to. If the file is in a subfolder, that folder name has to precede the doc name. You need to understand your own folder structure (file management, essentially) and some people just have trouble with this. If you have this problem, it's an important one.

Set up a local site

Setting up your site properly within Dreamweaver is a very important step on the road. It allows Dreamweaver to offer many helpful functions you would otherwise do without. The most important of these is that Dreamweaver will note the location of all files and links within your site and help you to keep everything working the right way.

The most common approach to creating a website using Dreamweaver is to create and edit pages on your local disk, and then upload copies of those pages to a remote web server to make them publicly available. It's possible to use Dreamweaver in other ways (such as running a web server on your local computer, or uploading files to a staging server, or using a mounted disk as if it were your local disk), but the lessons in this guide assume that you're working locally and then uploading to a remote server. In Dreamweaver, the word site is used as shorthand to refer to any of the following things:

- A website: a set of pages on a server, to be viewed by a visitor to the site using a web browser.
- A remote site: the files on the server that make up a website, from your (the author's) point of view rather than a visitor's point of view.
- A local site: the files on your local disk that correspond to the files in the remote site. You edit the files on your local disk, then upload them to the remote site.
- A Dreamweaver site definition: a set of defining characteristics for a local site, plus information on how the local site corresponds to a remote site.

Normally, you start creating a website by planning it: figuring out how many pages to create, what content appears on each page, and how the pages are connected to each other.

You'll create a site definition using the Site Definition dialog box. You can fill in this dialog box in either of two views: Basic or Advanced. The Basic approach guides you through site setup step by step. But it asks you many questions you may not yet know the answer to.

The following procedure describes how to set options in the Advanced version of the dialog box. For details of how to set options in the Basic version, click the Basic tab and then click the Help button.

To define a site:

Choose Site > New Site. (That is, choose New Site from the Site menu.)

The Site Definition dialog box appears.

On the line that says "Site Name" In the text box, enter a name to identify the site within Dreamweaver. The name can be anything you want. This is simply a nickname.

The next choice is "Local site folder" and this choice is VERY IMPORTANT to a successful outcome! The text box allows you to specify a folder on your local disk where Dreamweaver should store the local version of the site's files. It's easier to specify an accurate folder name if you browse to the folder rather than typing the path, so click the folder icon next to the text box.

In the Choose Local site Folder for Site dialog box, start by navigating to a folder on your local disk where you can store all of your files for this web site. Don't click OK yet.

You can set up information about your remote site later for now, the local site information is all you need to start creating a page.

Click SAVE to allow Dreamweaver to create the site definition.