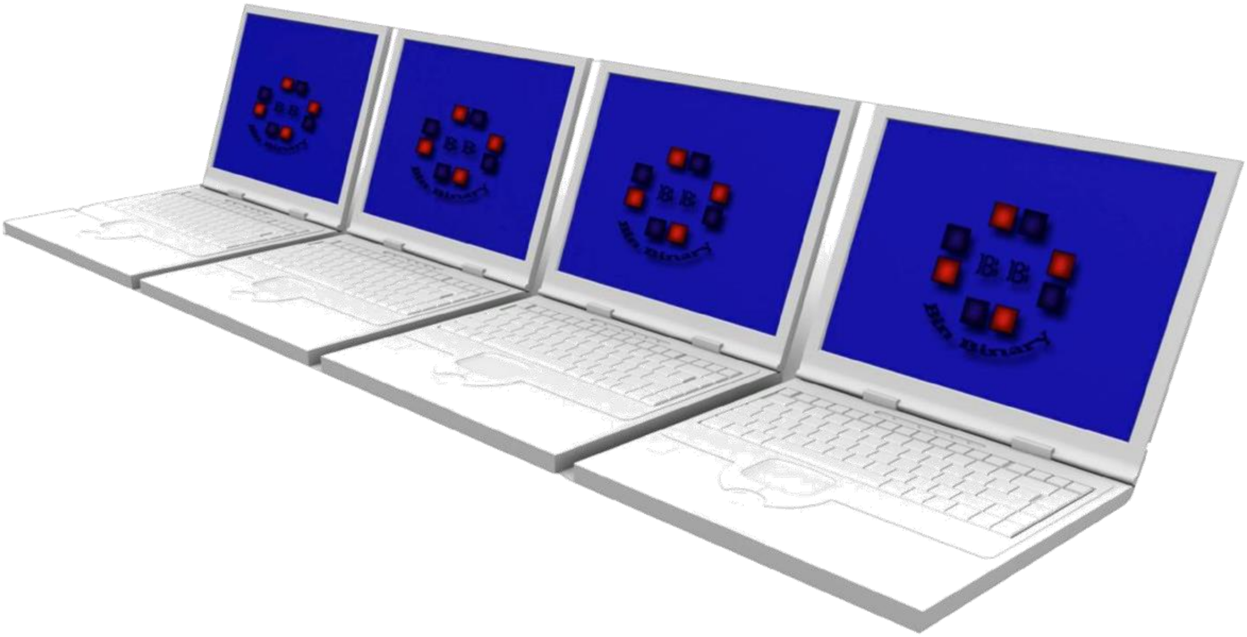
BIN BINARY
TECHNOLOGIES

www.binbinarytechnology.com
info@binbinarytechnology.com
+234 803 807 120 2, +234 905 707 7075



Bin Binary Technologies

www.binbinarytechnology.com

info@binbinarytechnology.com

Website Design Courseware

Website Design Made Understandable by Bin Binary Technologies

About Bin Binary Technologies

Bin Binary Technologies is a leading global IT services provider whose focus is to provide quality IT services to clients and be a fast tracker of client's business patronage, growth and development.

The company was founded with the vision to be recognized as a global provider of innovative IT solutions, services and training.

Chaired by the dynamic entrepreneur Basil Isaac Nzewure, Bin Binary Technologies was incepted in 2010 and was initially known as Bassanio Institute of Information Technology [BIIT]. In 2013, the company was officially rebranded and registered as Bin Binary Technologies with Rc: 2297811. Since then, the company has become a pioneer provider of various IT services ranging from software, web, training, hardware and general IT services.

With our crop of highly skilled personnel, we provide customized business solutions to our clients. We stand-out with our qualified professionals who have passion for professionalism and quality service. We create and implement complete solutions with the newest versions of the latest technologies.

With our management team of highly qualified personnel who have vast experience and in-depth knowledge of software development, Bin Binary Technologies have made a great point in the software development market.

Our services portfolio has in it project management services, web hosting, design and development, software development, mobile apps development, Site Engine Optimization [SEO], Social Media [SMO] creation, management and optimization, blogging, Google apps activation, training, sales and supply of computer and computer related devices. Bin Binary Technologies deliver training on demand and training solutions for individual and corporate organizations. We deliver our projects with the main aim of customer satisfaction within the agreed time frame and client's specification and requirements.

Introduction To Internet

The **Internet** is a global system of interconnected computer networks that use the standard Internet protocol suite (*TCP/IP*) to serve several billion users worldwide. It is a *network of networks* that consists of millions of private, public, academic, business, and government networks, of local to global scope, that are linked by a broad array of electronic, wireless and optical networking technologies. The Internet carries an extensive range of information resources and services, such as the inter-linked hypertext documents of the World Wide Web (WWW), the infrastructure to support email, and peer-to-peer networks. The origins of the Internet date back to research commissioned by the United States government in the 1960s to build robust, fault-tolerant communication via computer networks.

THE WWW

The **World Wide Web** (abbreviated as **WWW** or **W3**, commonly known as **the web**) is a system of interlinked hypertext documents accessed via the Internet. With a web browser, one can view web pages that may contain text, images, videos, and other multimedia and navigate between them via hyperlinks.

In order to understand the history of the World Wide Web it's important to understand the differences between the World Wide Web and The Internet. Many people refer to them as the same thing, but in fact, although the end result is the common perception of most everyday users, they are very different.

The internet is a series of huge computer networks that allows many computers to connect and communicate with each other globally. Upon the internet reside a series of languages which allow information to travel between computers. These are known as protocols. For instance, some common protocols for transferring emails are The Internet Message Access Protocol IMAP is an Application Layer Internet protocol that allows an e-mail client to access e-mail on a remote mail server. The current version, **IMAP** version 4 revision 1 (IMAP4rev1), **POP3** (Post Office Protocol 3) is the most recent version of a standard protocol for receiving e-mail. **POP3** is a client/server protocol in which e-mail is received and held for you by your Internet server and Simple Mail Transfer Protocol (**SMTP**) is an Internet standard for electronic mail (email) transmission. Just as email is a layer on the internet, the World Wide Web is *another* layer which uses different protocols.

HTTPS (**HTTP Secure**) is a protocol for secure communication over a computer network which is widely used on the Internet. HTTPS consists of communication over Hypertext Transfer Protocol (HTTP) within a connection encrypted by Transport Layer Security or its predecessor, Secure Sockets Layer. The main motivation for HTTPS is authentication of the visited website and to protect the privacy and integrity of the exchanged data. In its popular deployment on the internet, HTTPS provides authentication of the website and associated web server that one is communicating with, which protects against man-in-the-middle attacks.

The World Wide Web uses three protocols:

- **HTML (Hypertext markup language)** - The language that we write our web pages in.
- **HTTP (Hypertext Transfer Protocol)** - Although other protocols can be used such as FTP, this is the most common protocol. It was developed specifically for the World Wide Web and favored for its simplicity and speed. This protocol requests the 'HTML' document from the server and serves it to the browser.
- **URLS** (Uniform resource locator) - The last part of the puzzle required to allow the web to work is a URL. This is the address which indicates where any given document lives on the web. It can be defined as <protocol>://<node>/<location>

In 1980 an English man by the name of Tim Berners Lee was working on a project known as 'Enquire'. Enquire was a simple database of people and software who were working at the same place as Berners Lee. It was during this project that he experimented with hypertext. Hypertext is text that can be displayed on devices which utilize hyperlinks. The Berners Lee Enquire system used hyperlinks on each page of the database, each page referencing other relevant pages within the system. Berners Lee continued to develop three major components for the web; HTTP, HTML and the world first web browser. Funnily enough, this browser was also called "the World Wide Web" and it also doubled as an editor.

A Web site

Web site, or simply **site**, is a set of related web pages typically served from a single web domain. A website is hosted on at least one web server, accessible via a network such as the Internet or a private local area network through an Internet address known as a uniform resource locator (URL). All publicly accessible websites collectively constitute the World Wide Web.

Web pages, which are the building blocks of websites, are documents, typically written in plain text interspersed with formatting instructions of Hypertext Markup Language (HTML, XHTML). They may incorporate elements from other websites with suitable markup anchors. Webpages are accessed and transported with the Hypertext Transfer Protocol (HTTP), which may optionally employ encryption (HTTP Secure, HTTPS) to provide security and privacy for the user of the webpage content. The user's application, often a web browser, renders the page content according to its HTML markup instructions onto a display terminal.

The pages of a website can usually be accessed from a simple Uniform Resource Locator (URL) called the web address. The URLs of the pages organize them into a hierarchy, although hyperlinking between them conveys the reader's perceived site structure and guides the reader's navigation of the site which generally includes a home page with most of the links to the site's web content, and a supplementary about, contact and link page.

Choosing A Workspace Layout

When you launch Dreamweaver for the very first time, the program automatically opens a dialog box asking you to choose which file extensions should be associated with the project, and then displays the newly updated default Designer layout. After the program is open, you can switch to another available layout option by choosing Window➪Workspace Layout, or by clicking the Workspace Switcher menu on the Application bar.

✦ **App Developer:** Select this layout to have the CSS Styles, AP Elements, Databases, Bindings, Server Behaviors, Files, Assets, and Snippets panels docked on the left with the Document window displayed in the center and no Properties inspector.

✦ **App Developer Plus:** Select this layout to have most of the App Developer panels mentioned in the preceding bullet docked on the left; the Document window set to Split Code view (code on top) in the center; the Properties inspector below the Design view; and the iconic view of the Insert, Databases, Bindings, and Server Behaviors panels docked along the right. Click the double-arrows at the top of either panel dock to expand and collapse the panels as needed.

✦ **Classic:** This layout most closely mirrors the default Designer layout setup from previous CS versions of Dreamweaver. The right edge of the screen displays the most used docked panels, while the rest of the workspace is composed of the old tabbed Insert panel across the top, the Document window in Split Code view below that, and the Properties inspector along the bottom.

✦ **Coder:** Select this layout to have the CSS Styles, AP Elements, Files, Assets, and Snippets panel groups on the left—similar to several popular programming software applications—with the Document window in Code view in the center of the screen.

✦ **Coder Plus:** This layout shows the Files, Assets, and Snippets panel group on the left; the Document window in Code view in the center of the screen; and the iconic view of the Insert, CSS Styles, and AP Elements panels docked along the right.

Designer: Select this layout to have the Insert, CSS Styles, AP Elements, Business Catalyst, Files, and Assets panels docked on the right, with the Document window in Split Code view in the center, and the Properties inspector along the bottom.

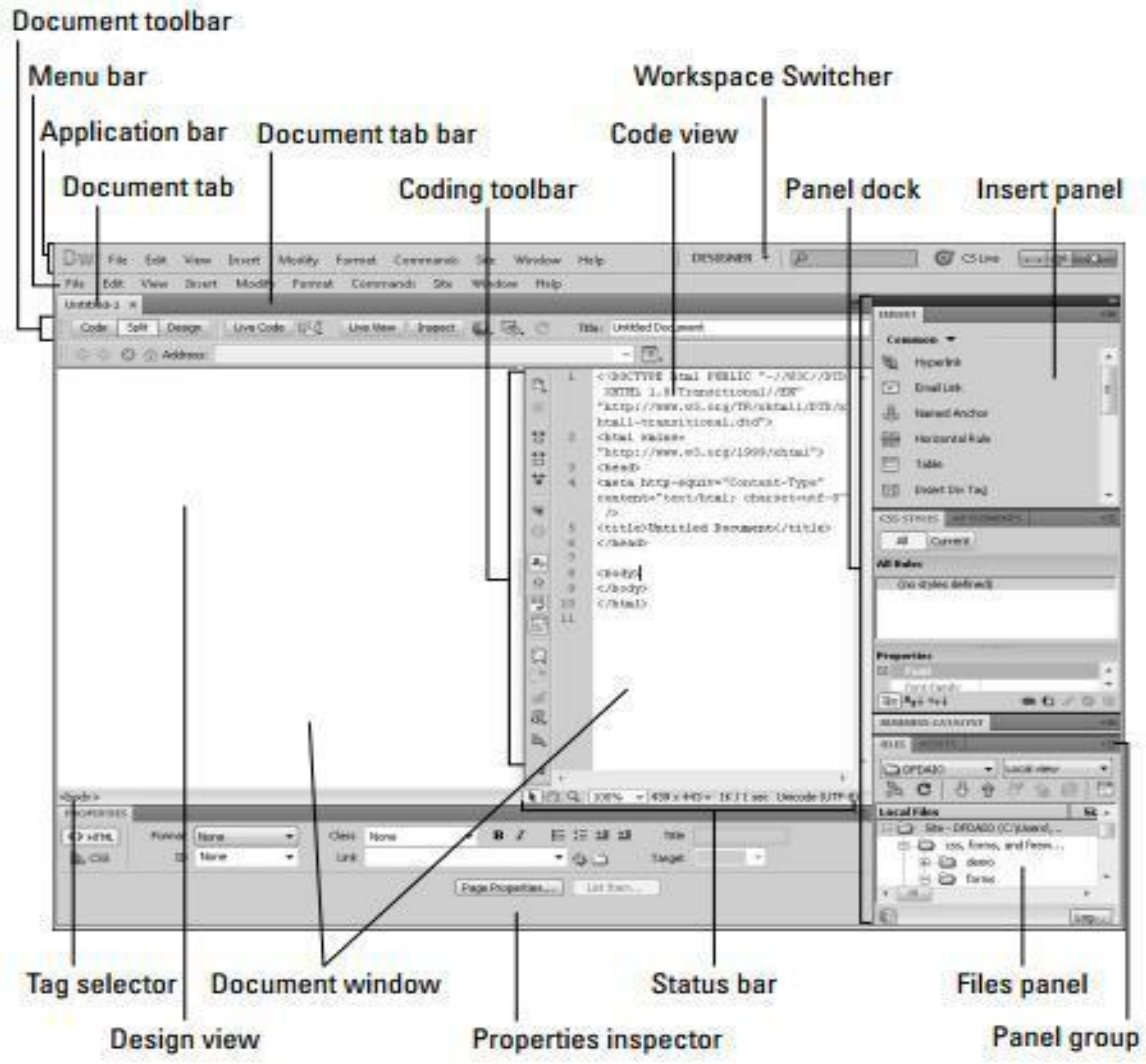
✦ **Designer Compact:** Select this layout to have the iconic Insert, CSS Styles, AP Elements, Business Catalyst, Files, and Assets panels docked along the right, with the Document window in Split code view in the center, and the Properties inspector along the bottom.

✦ **Dual Screen:** Select this layout if you have a secondary monitor to the left or right of your primary monitor. Initially, the panel dock is displayed on the left monitor and the Document window and code view are displayed on the right; however, you can customize this setup if you prefer working with a different configuration.

Each workspace layout is designed to assist you with a particular task. Choose an Application Developer layout if you'll be developing applications, select a Coder option if you'll be working only with code, select a Designer layout if you'll be working with the WYSIWYG editor (Design view) and Code editor, pick the Classic option if you're not quite ready to start using one of the other layouts, and choose the Dual Screen layout when working with two monitors. If you are unsure of which layout to use, we recommend that you select the Designer option. Alternatively, feel free to open and arrange any of the panels to your liking and save your customized workspace by clicking the Workspace Switcher menu and choosing New Workspace.

Understanding Your Dreamweaver Workspace

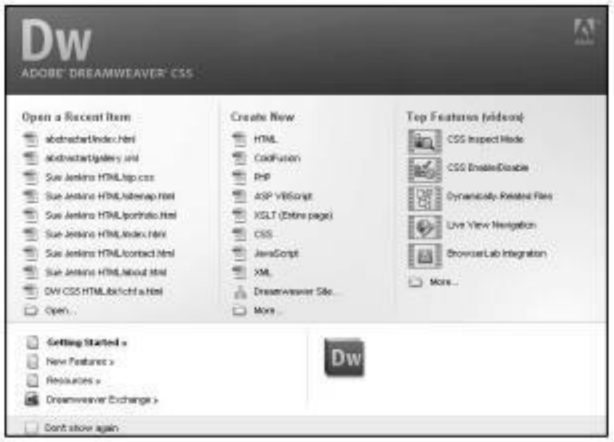
Because most Dreamweaver users are designers, we'll explore the workspace using the Designer layout, which consists of the Application bar at the top of the screen, the Document window in the center, the panel dock and panels on the right, and the Properties inspector along the bottom, shown in Figure 1-1 All of these layout elements work in harmony to assist you with adding and editing the content in any open document.



```
<!DOCTYPE html PUBLIC "-//W3C//DTD
 XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/x
html1-transitional.dtd">
<html xmlns=
"http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type"
content="text/html; charset=utf-8"
 />
<title>Untitled Document</title>
</head>

<body>
</body>
</html>
```

✦ **Welcome Screen:** When you launch Dreamweaver, the Welcome Screen, which is automatically enabled, appears in the open workspace area any time no files are open. The Welcome Screen (shown in Figure 1-3) contains quick links to open recent files, create new documents, and access Web links to Dreamweaver Exchange and a library of top-featured Adobe training videos. In addition, you'll find links to "Getting Started," "New Features," and "Resources," all of which are helpful for users seeking quick details about what's new. Hide and show the Welcome Screen by selecting or deselecting the Show Welcome Screen check box in the General category of the Preferences dialog box.



✦ **Application bar:** The Application bar includes the Dreamweaver logo ("DW"), menus, a Layout menu for choosing different code and design views, an Extend Dreamweaver menu for accessing Dreamweaver Exchange, a Site menu for quickly managing new Dreamweaver sites and editing existing ones a Workspace Switcher menu for toggling between the workspace layout options, a search field for finding help in Dreamweaver, and a CS Live Services button that offers quick links to services such as BrowserLab, Acrobat.com, and CS News and Resources.

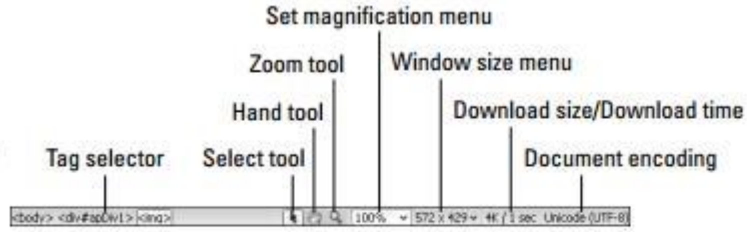
✦ **Insert panel:** The Insert panel, when visible, appears as its own panel at the top of the panel dock. The Insert panel displays a variety of icons for quickly adding objects such as images and tables into an open file.

✦ **Document window:** This window shows the active document as you design and edit it. View the document in Code, Split (half code and half design), or Design view. When more than one document is open, you can toggle between them by clicking their document title tabs.

✦ **Document toolbar:** This toolbar is located at the top of every open document just below the document's title tab. The toolbar includes the Code, Split, and Design view buttons as well as quick links for other features such as the Live Code and Live View settings.

The Status Bar

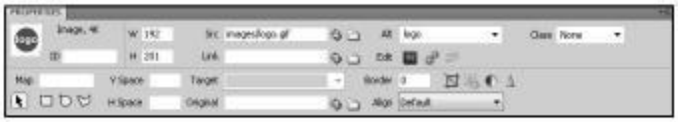
Dreamweaver's Status bar includes many handy features to improve your productivity. The Tag selector is on the bottom left edge of the Document window and allows for easy tag and tag content selection. On the bottom-right edge, the Status bar shows the current size (in pixels) of the Document window, as well as a file size and estimated file opening time reflecting the file preferences for projected site visitors. For example, an open document may display 1100 x 435 and 35K/5 sec, meaning that the current page has a 35K file size and would take 5 seconds to load in a browser on a computer using a 56K modem. Keep in mind that the size of the Document window is not the same as the fixed width of your Web site layout; the numbers presented here merely represent the pixel dimensions of the Design view portion of the open Document window.



<body> <div#apDiv1> <img> 100% 572 x 429 4K / 1 sec Unicode (UTF-8)

Using The Properties Inspector

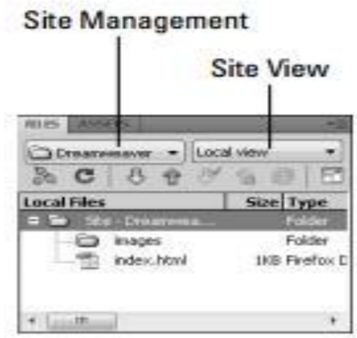
The Properties inspector is the panel to use for adding HTML and CSS formatting and other attributes or properties to selected objects in your document. The most versatile of all the panels, the contents of the Properties inspector change according to the object or text selected in Code or Design view. For instance, when you select text in your document, the Properties inspector displays options for adding properties to text; when you select a graphic, the inspector displays options for adding properties to images. This works for most object selections. The main thing to remember is that you must select the desired object or content before adding properties



Image, 4K
ID
W 192
H 201
Src images/logo.gif
Link
Alt logo
Class None
Map
V Space
H Space
Target
Original
Border 0
Align Default

Working With The Files Panel

Use the Files panel to manage (organize, select, and open) all your site files and folders, view both remote server and local file listings, access other files on your hard drive, and manage sites within Dreamweaver. Rather than relying on your Explorer window to find and open files, create a “managed site” for each project you work on in Dreamweaver and use the Files panel to locate and open all your site files. It is easier when you centralize this task to the Files panel. The Files panel is displayed in its collapsed mode in the Designer workspace, and when displaying a “managed” site, it lists all the files and folders of the specified directory. You can also expand the panel into two panes to show both local and remote server views in one window; see the following list for details. At the top of the Files panel, two drop-down menus assist you with site-management tasks.



- **Site Management:** This menu lists all the managed sites you’ve created in Dreamweaver. A site becomes managed when you define a folder on your local computer for the HTML files of a specific project, and tell Dreamweaver where to find that folder. To get the most out of Dreamweaver’s features, you need to create a managed site for each project you work on. To view a list of managed sites, scroll to the bottom of this menu and choose Manage Sites to open the Manage Sites dialog box.

- **Site View:** When the Files panel is collapsed, use the Site View menu to toggle among four views:

- **Local view:** Select this view to see the file structure of the local managed site. You may also use this view to see both local and remote server sites with the Files panel split into two panes. Use the Expand/Collapse icon to toggle between viewing one and two panes. By default, the local site appears in the right pane and the remote server appears in the left, but you can modify this in the Site category of Dreamweaver's Preferences if you'd prefer having the local pane appear on the left, as in many older FTP programs.
- **Remote server:** Select this view to see the file structure of the remote server site. A remote server site is a version of your local site sitting on a remote hosting server accessible from the Internet. You must set up a remote server site in advance to see the remote site files.
- **Testing server:** This view shows a directory listing of both the testing (staging) server and local site files. You must set up a testing server in advance to see the testing server site.

Planning And Designing Your Site

Putting a Web site on the Internet involves more than figuring out how to build Web pages. Creating a site that people enjoy using requires careful planning and design, based on site visitors' expectations and your (or your client's) Web site goals. Planning a good Web site involves conducting market research, defining business needs, making decisions about Web-based technology, discussing search engine optimization and accessibility strategies, gathering and organizing content, setting a budget and signing contracts, purchasing a hosting plan, designing and optimizing graphics, constructing the site in HTML, CSS, JavaScript, or some combination thereof, and adding the required dynamic features and functionality.

Understanding The Web Design Workflow

Building a Web site has a logical workflow. The different stages of the Web
design workflow are as follows:
Define > Design > Build > Test > Launch

Each stage requires cooperation from all parties involved — including the client, designers, and programmers — because each is responsible for different parts of the process. Table 2-1 shows who should be involved with each step of the workflow process.

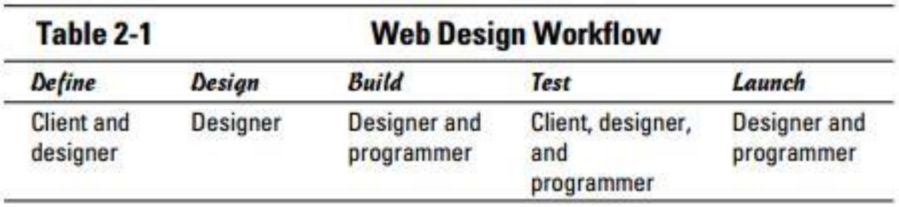


Table 2-1 Web Design Workflow

Define	*Design*	*Build*	*Test*	*Launch*
Client and designer	Designer	Designer and programmer	Client, designer, and programmer	Designer and programmer

Typically, the client provides content, the designer (you) creates the design and builds the site, the programmer adds dynamic functionality, everyone gets involved in testing, and finally, the designer or programmer or both launch (publish) the site on the Internet.

Exploring the audience's expectations Market research is one of the best ways to determine site visitors' expectations. Here's the information you need to assemble:

✦ The target demographic. Be specific. Who are they?

✦ The target visitors' Web browsing habits. Determine whether they are seeking products, information, or a consultation, or have information to share with others.

✦ The target visitor's bandwidth capabilities. Not all Internet users have access to high-speed Internet, especially visitors from across the globe. If this is a consideration, a fast-loading Web page will be one of your main priorities.

✦ The target visitor's Internet browsing equipment. Some of your visitors may want to view your site on mobile devices in addition to using an Internet browser, which means you may need to develop separate mobile content or mobile-ready Web pages.

✦ Whether the target audience prefers to purchase online or shop in a brick-and-mortar store.

Collecting this information can help you determine the site's design direction, layout, and structure. For instance, if the site is aimed at a worldwide audience, you may need to consider language and cultural issues. You can find Web sites for every conceivable industry in the world, so a good place to start the design process when creating a new Web site is to review what's already on the Internet. Do a general online search for companies in the same industry worldwide or countrywide. Then do additional searches for competition in the same general geographic region, such as statewide, countywide, and citywide. Think about design, usability, budget, time frame, and technical issues. Look at the competitors' sites and take notes about what works and what doesn't with regard to content and layout. After gathering data about competitors' sites, draw diagrams and sketch design ideas, write copy, and make a list of key points that the site should address. You may even want to generate a preliminary site map, or wireframe, of all sections of your new site based on your research of competitor's sites, client feedback, and so on. A clear understanding of the target audience helps define the site's organization and structure.

Examining Issues That Affect The Site Design

Several issues affect the design of your site, such as cost, timeframe, marketing, target audience, accessibility issues, content, and browser optimization. The following sections introduce you to these concepts and provide ideas for making the most of your site. Balancing time, cost, and scope. When you're designing a Web site, whether for yourself or for a client, keep in mind these three main components to any Web project:

✦ **Time:** Decide when you can deliver the job.

✦ **Cost:** Determine how much the project is going to cost.

✦ **Scope:** Determine what you hope to achieve.

Most projects begin with specific limitations with regard to these project components. One frequent limitation has to do with budget; a client may specify that he or she hopes to keep their cost within a specific budget for a new web site that achieves the defined scope and is delivered by a specific time. Although that may be an ideal goal, achieving all three components of the web project may not be feasible. For example, if a project needs to be delivered in a short time frame, the project cost may need to increase, or if the client only has *X* Naira to pay for the project, only parts of the scope may be met. Obviously, the three Web project components have a strong dependency between them; if the project cost, time frame, and scope are in sync, all three can be achieved. However, if limits are placed on any one of the components, a compromise may need to be reached.

Website Content

Before building a Web site, you (or your client) need to create and gather content and other site assets. Content gathering includes writing copy for every page and creating or licensing image, sound, and video files. It's important to gather these assets in advance so that you don't have to stop site development repeatedly to create or find any missing content. Gathering content is a big undertaking, even for the smallest sites, so unless you're also being compensated by your client as the project or content manager, this process should be the client's sole responsibility before the project begins; otherwise, you may be blamed for significant delays, even though you're not the one holding up the project. Before you sign contracts, make sure you reinforce the idea that it's the client's responsibility to gather the content before you start the project time clock.

After you’ve gathered the content, organize everything electronically in a place that’s easily accessible when it comes time to build the site. For example, you may decide to create a folder on your computer called Websites, and in that folder create a subfolder for the client. Inside the client’s folder, you may create several additional subfolders for all the different assets.

Planning The Site Layout

You can save time by planning and designing the site layout before working in Dreamweaver. A consistent layout and design helps create a good user experience. Site layout applies to the look of all the pages on the site, as well as how the pages are logically arranged and how they interact with each other. This phase is where you’re creating the site’s *architecture*, or structure. With regard to the layout of the pages themselves, consider designing a mock-up that has fixed as well as editable areas. Dreamweaver allows you to create templates and library items for page layouts and elements to make them consistent on every page. For instance, the navigation element may be at the top of every page, with an area for sub-navigation on the left margin and page-specific content in the center of the page below the navigation. As you create the design, think about the site visitors’ experience:

✦ Visitors should be able to move from page to page with ease. Therefore, navigation should be consistent throughout the site.

✦ Visitors should know where they are inside the site and how to return to the home page. Use indexes and sub-navigation to assist visitors with finding information. Also provide a method for contacting the company in case a visitor wants to communicate by telephone, fax, mail, or e-mail. After you gather and organize your data, you may want to create some HTML sitemap or wireframe pages to help organize the site’s structure. A wireframe is a tree diagram or flowchart of a Web site that includes all its pages. Each wireframe indicates links between pages but doesn’t typically include any reference to the design of the site or the content on any specific pages. You can create additional wireframes for the individual pages to assist with the organization of content on the page, before you have real content.

After creating a wireframe and before building the site in Dreamweaver, you may also want to create a mock-up, or *comp*, of the site design on paper or in a graphics program such as Adobe Photoshop, Adobe Illustrator, or Adobe Fireworks. A key benefit to designing a mock-up in a graphics program is that after the client approves the mock-up, you can use it to generate many if not all of the graphics. A mock-up differs from a wireframe in that the mock-up is a design of the site that contains all the graphical information for the Web site layout including company identity, navigation, headers, text, and other graphics. In other words, the mock-up should have all the elements that the client has requested for the site. For instance, clients may tell you that they "want the logo on the top of the page, the navigation below that, an area for links to frequently accessed pages, and a section for a photo gallery." In response to these needs, you plan the layout of the page and the site. Then you show the mock-up to the clients to make sure the design meets their needs.

Designing And Optimizing Graphics

Before adding graphics to Web pages, you need to compress them, because in their native format (PSD, PDF, AI, EPS, TIFF, and so on), the file sizes are much too large to download quickly over the Internet. When compressing images for the Web, you can choose from three graphics formats: GIF, JPEG, and PNG, as shown in Table 2-4. Web browsers have widely supported GIF (Graphics Interchange Format) and JPEG (Joint Photographic Experts Group) for years: however, the PNG (Portable Network Graphic) format has more recently gained in popularity and is now also widely supported. Nonetheless, due to some older browser incapability issues, you may still want to be mindful when using PNG files. For example, IE6 interprets transparent PNG files into a wonderful shade of baby blue!

Creating And Managing Sites

Web site is a group of pages that are linked and share common features such as design, navigation, content, and purpose. Dreamweaver enables you to organize all the pages and assets of your site in one convenient location. To take advantage of Dreamweaver's great site-management features, such as uploading files and managing links, first you need to create a managed site in Dreamweaver, which is really just a way to tell Dreamweaver where to create and save files locally for each Web site project. When managing your site, you'll likely adopt a general Web site structure and opt for either a root-level or document-level organization for your files.

Understanding General Website Structure

Before you manage your first Dreamweaver Web site, you need to be familiar with the general structure of a Web site and have a basic understanding of the different types of root-level organization, as discussed in the sections that follow.

Website Construction Folders

When building a Web site, you need to work in two or three locations:

✦ **The local folder:** This folder, also called the local root folder, holds all the files, images, and other assets of a managed site. The root level, simply put, is the top level or starting point a browser uses for finding objects in a Web site. The local folder typically sits somewhere on your computer's hard drive, though it may also be on a mapped drive or a network server. Wherever it resides, you must specify the location of the local root folder in the Site Setup dialog box when defining a managed site.

✦ **The live server folder:** This folder is where you publish your site, typically on a remote live Web server. For example, when you purchase a web hosting plan, you're essentially renting a parking space for your web site in the virtual parking lot of the Internet. That space is on the host's remote live Web server. You'll be transferring files to the remote live server folder from the local folder to ensure that the published web site is functional and up to date. You can choose from several types of live servers.

✦ **The testing server folder:** This folder is where Dreamweaver processes dynamic data to create dynamic content and connect with a database while you create and test your site. Your testing server can be on the local computer, a development or testing server, or a production server. We don't recommend that you use the live server folder for the testing server folder because that can cause confusion and create other problems. For most simple, non-dynamic sites, you will build your site in the local folder and publish your site to the remote live server folder. For dynamic sites, you will build in the local folder, test in the testing server folder, and publish to the live server folder.

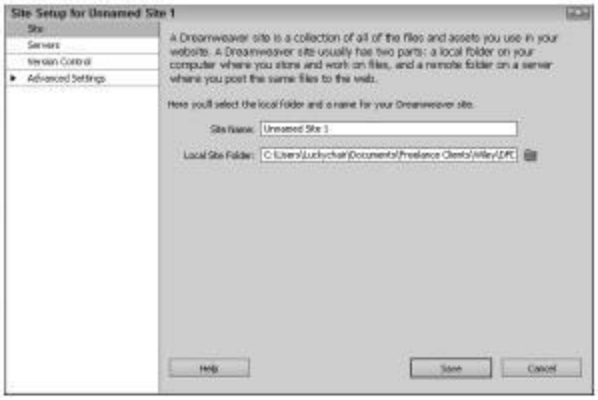
Setting Up A Managed Site

Setting up a managed site simply means defining a local site on your computer so that Dreamweaver knows where to save documents and find files related to that site. You'll want to create a managed site for each project that you work on so that you can open and edit site-specific files from the Files panel. Always try to define or manage a site before you start development to ensure that the site takes advantage of Dreamweaver's great site-management features, such as site wide filename change support and the site wide link checker.

To get your site up and running locally, follow these steps:

1. **Choose Site➪New Site.**

The Site Setup dialog box appears to the Site category, which displays
fields for the Site Name and Local Site Folder, as shown:



A Dreamweaver site is a collection of all of the files and assets you use in your website. A Dreamweaver site usually has two parts: a local folder on your computer where you store and work on files, and a remote folder on a server where you post the same files to the web.

Here you'll select the local folder and a name for your Dreamweaver site.

Site Name: Unnamed Site 1

Local Site Folder: C:\Users\Luckychair\Documents\Freelance Clients\Wiley\DFC

2. In the Site Name field, enter the name of your site.

Choosing a name that matches the name of the client or indicates your site's purpose is best, such as ABC Company or My Blog.

3. In the Local Site Folder field, verify that the path points to the folder on your computer that you want to use for this site.

For example, if you have a folder on your computer dedicated to local web sites (local sites), and you want to create a folder inside that to house the files for a new ABC Company Web site, your Local Site Folder path might be something like

Users\YOURUSERNAME\Documents\local_sites\ABC Web

4. Click Save to close the dialog box.

Your new managed site opens, displaying all the existing files and folders (if any) in the root folder in the Files panel. In the Server section that follows, you find out how to configure the Server category of the Site Setup dialog box to transfer files to a remote server. You can easily adjust these site settings at any time by choosing Site➪Manage Sites.

If your server or work computer is not backed up regularly, seriously consider making a local backup copy of your site before you modify it each time. Having backups is a great practice, one that can serve you well when mistakes are made, data is lost, or you need to revert your site to a previous backup version.

SEO

Search Engine Optimization (SEO) refers to the collection of techniques and practices that allow a site to get more traffic from search engines (Google, Yahoo, Microsoft). SEO can be divided into two main areas: off-page SEO (work that takes place separate from the website) and on-page SEO (website changes to make your website rank better). This tutorial will cover both areas in detail! Remember, a website is not fully optimized for search engines unless it employs both on and off-page SEO.

SEO Techniques Are Classified Into Two Broad Categories

1. Techniques that search engines recommend as part of good design referred to as White Hat SEO.
2. Techniques that search engines do not approve and attempt to minimize the effect of search referred to as Black Hat or spamdexing.

White Hat SEO

An SEO tactic, technique or method is considered as White Hat if it follows the following:

- If it conforms to the search engine's guidelines.
- If it does not involve any deception.
- It ensures that the content a search engine indexes and subsequently ranks is the same content a user will see.
- It ensures that web page content should have been created for the users and not just for the search engines.
- It ensures the good quality of the web pages
- It ensures the useful content available on the web pages

Always follow a White Hat SEO tactic and don't try to fool your site visitors. Be honest and definitely you will get something more.

Black Hat Or Spamdexing

An SEO tactic, technique or method is considered as Black Hat or Spamdexing if it follows the followings

- Try to improve rankings that are disapproved of by the search engines and/or involve deception.
- Redirecting users from a page that is built for search engines to one that is more human friendly.
- Redirecting users to a page that was different from the page the search engine ranked.
- Serving one version of a page to search engine spiders/bots and another version to human visitors. This is called **Cloaking** SEO tactic.

- Using Hidden or invisible text or with the page background color, using a tiny font size or hiding them within the HTML code such as "no frame" sections.
- Repeating keywords in the Meta tags, and using keywords that are unrelated to the site's content. This is called **Meta tag stuffing**.
- Calculated placement of keywords within a page to raise the keyword count, variety, and density of the page. This is called **Keyword stuffing.**
- Creating low-quality web pages that contain very little content but are instead stuffed with very similar key words and phrases. These pages are called **Doorway or Gateway Pages**
- Mirror web sites by hosting multiple web sites all with conceptually similar content but using different URLs.
- Creating a rogue copy of a popular web site which shows contents similar to the original to a web crawler, but redirects web surfers to unrelated or malicious web sites. This is called **Page hijacking**.

Always be away to adopt any of the above Black Hat tactic to improve the rank of your site. Search engines are smart enough to identify all the above proprieties of your site and ultimately you are not going to get anything.

SEO – Website Domain

When you start thinking of doing a business through internet, first thing which you think about is your web site domain name. Before you choose a domain name you should consider the followings:

- Who would be your target audience?
- What you intend to sell to them. Is it a tangible item or just text content?
- What will make your business idea unique or different than everything else that is already on the market?

Using keywords in your domain name give you a strong competitive advantage over your competitors. Having your keywords in your domain name can increase click through rates on search engine listings and paid ads as well as make it easier to using your keywords in get keyword rich descriptive inbound links.

Keyword Placement

Where your keywords are placed on a page is very important. For example, in most engines, placing the keywords in the Title of the page or in the Heading tags will give it more relevance. On some engines, placing keywords in the link text, the part that is underlined on the screen in a browser, can add more relevancy to those words.

Best Places to Put Keywords:

Here is a list of places where you should try to use your main keywords.

- Keywords in the <title> tag(s).
- Keywords in the <meta name="description">
- Keywords in the <meta name="keyword">
- Keywords in <h1> or other headline tags.
- Keywords in the <a href="http://yourcompany.com">keywords</a> link tags.
- Keywords in the body copy.
- Keywords in alt tags.
- Keywords in <!-- insert comments here> comments tags.
- Keywords contained in the URL or site address, e.g., http://www.keyword.com/keywordkeyword.htm.

How Meta Tags Looks Like

You can add following in the head section of the web page:

<meta name="keywords" content="KEYWORD1 KEYWORD2 KEYPHRASE1 etc. about 30 to 40 unique words">

<meta name="description" content="An accurate, keyword-rich description about 150 characters">

Meta Description Tag Tips:

There are few important tips for good meta description tags:

- Use keywords in your meta description tag.
- Try not to repeat the words overly often, but try to use multiple syntaxes of your key words.
- There should not be more than 150 characters in a description meta tag of a single web page.
- Use a different meta description tag for each page, as each page is different and stands a better chance of being found if you place a good title and description on it.

SEO – Title Optimization

Hope you understand the meaning of *HTML TITLE* tag which is put inside head tag. The page title (not to be confused with the heading for a page) is what is displayed in the title bar of your browser window, and is also what is displayed when you bookmark a page or add it to your browser Favorites.

This is the one place on a web page where your keywords MUST be present. Correct use of keywords in the title of every page of your website is extremely important to Google - particularly for the home page. If you do nothing else to optimize your site, remember to do this!

The title shouldn't consist of much more than about 9 words or 60 characters, with your keywords used at the very beginning of the title. Since Google is looking for relevant keywords in the title, this means you should NOT include your company name in the title unless your company name is very well known.

Improper or nonexistent use of titles in web pages will keep more websites out of top rankings on Google than any other factor except perhaps for a lack of relevant content on a page or a lack of quality links from other websites that point to your site.

Best Practices for Creating Titles:

Here are some best practices you should follow for creating titles on pages:

- Each page should have a unique title.
- If practical, try to include your Primary Keyword Phrase in every title of every page.
- Begin the title of your home page with your Primary Keyword Phrase, followed by your best Secondary Keyword Phrases.

- Use more specific variations to your Primary Keyword Phrase on your specific product, service, or content pages.
- If you must include your company name, put it at the end of the title.
- Use the best form, plural or singular, for your keywords based on what Word Tracker says is searched on more often.
- Don't overdo it - don't repeat your keywords more than 2 - 3 times in the title
- Make sure the <title> tag is the first element in the <head> section of your page - this makes it easier to find by Google.

HTML Basics

Before I get started, you should know that HTML code almost always uses beginning and ending tags. These tags surround the text that will be affected by the code.

A beginning tag is generally a word surrounded by brackets. The closure tag is surrounded by the same brackets but with a forward slash right after the opening bracket.

For example, if you want to bold a portion of a sentence, then you would use <b> for the opening tag and </b> for the closing.

Let's say you want to bold the word "Hello!" in the sentence below. Then your HTML code would look like this:

<b>Hello!</b> My name is Basil.

The output would be:

Hello! My name is Basil.

Only the word "Hello!" is bolded because the tags surround that word. If you wanted to bold the entire sentence, then you would have put the closure tag, </b>, after the word "Basil". Be sure to always include your closing tag because if you forget, your entire page will be affected by the tag.

You can apply this same concept to many other HTML codes. Here are several of the basics...

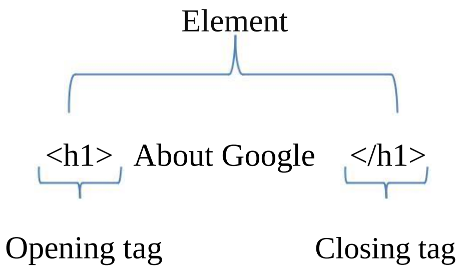
```
< html >
    < head >
        < title >Popular Websites: Google< /title >
            < /head >
                < body >
                < h1 >About Google< /h1 >
                < p > Google is best known for its search engine, although
                Google now offers a number of other services. < /p >
                < p > Google’s mission is to organize the world’s information and make it
universally
                accessible and useful. < /p >
            < p > its founders Larry Page and Sergey Brin
        started Google at Stanford University. < /p >
        < /body >
    < /html >
```

Tags And Elements

If you look at the first and last lines of the code for the last example, you will see pairs of angle brackets containing the letters “html” Starting on the first line, the first angled bracket looks like a less - than sign, then there are the letters “html” followed by a second angled bracket, which looks like a greater- than sign. The two brackets and all of the characters between them are known as a *tag*.

In this example, there are lots of tags and they are all in pairs; there are ***opening tags*** and ***closing tags***. The closing tag is always slightly different from the opening tag in that it has a forward slash after the first angled bracket: < /html >.

A pair of tags and the content these include are known as an *element*. In Figure 1 - 2, you can see the heading for the page of the last example.



The opening tag says “This is the beginning of a heading” and the closing tag says “This is the end of a heading.” Like most tags in XHTML, the text inside the angled brackets explains the purpose of the tag — hereh1 indicates that it is a level 1 heading (or top - level heading). As you will see shortly, there are also tags for subheadings (<h2>, <h3>, <h4>, <h5>, and <h6>). If we had not put tags around the words “About Google,” it would just be another bit of text; it would not be clear that these words formed the heading.

Now look at the three paragraphs of text about the company; each one is placed between an opening <p> tag and a closing </p> tag. And, you guessed it, the p stands for paragraph.

> The basic distinction between tags and elements is very important to understand: a tag is made up of a left - and right - angular brackets and letters and numbers between those brackets, whereas elements are the opening and closing tags plus anything between the two tags.

As you can see, the tags throughout this example actually describe what you will find between them, creating the structure of the document. The text between the <h1> and < /h1 > tags is a heading, and between the opening <p> and closing </p> tags make up paragraphs. Indeed, the whole document is contained between opening < html > and closing < /html > tags. You will often find that terms from a family tree are used to describe the relationships between elements. For example, an element that contains another element is known as the *parent* , while the element that’ s between the parent element ’ s opening and closing tags is called a *child* of that element. So, the < title > element is a child of the < head > element, the < head > element is the parent of the < title > element, and so on. Furthermore, the < title > element can be thought of as a grandchild of the < html > element.

Separating Heads From Bodies

Whenever you write a web page in XHTML, the whole of the page is contained between the opening < html > and closing < /html > tags, just as it was in the last example. Inside the < html > element, there are two main parts to the page:

- **The** < head >**element:** Often referred to as the head of the page, this contains information *about* the page (this is not the main content of the page). For example, it might contain a title and a description of the page, or instructions on where a browser can find CSS rules that explain how the document should look. It consists of the opening < head > tag, the closing < /head > tag, and everything in between.
- **The** < body >**element:** Often referred to as the body of the page, this contains the information you actually see in the main browser window. It consists of the opening < body > tag, closing < /body > tag, and everything in between. Together, the < html >, < head >, and < body > elements make up the skeleton of a HTML document — they are the foundation upon which every web page is built.

  Inside the < head > element of the first example page, you can see a < title > element:

  ```
  < head >
  < title >Popular Websites: Google< /title >
  < /head >
  ```

Between the opening < title > tag and the closing < /title > tag are the words Popular Websites: Google , which is the title of this web page**.** The title is where browsers such as Internet Explorer, Firefox, and Safari display the title of a document; it is also the name they use when you save a page in your favorites, and it helps search engines understand what your page is about.

The real content of your page is held in the < body > element, which is what you want users to read, and this is shown in the main browser window.

> The< head > element contains information about the document, which is not displayed within the main page itself. The< body > element holds the actual content of the page that is viewed in your browser.

You may have noticed that the tags in this example appear in a symmetrical order. If you want to have one element inside another, then both the element ’ s opening and closing tags must be inside the containing element. For example, the following is allowed:

<p> This paragraph contains some <em> emphasized text. </em> </p> whereas the following is wrong because the closing </em> tag is not inside the paragraph element:

<p> This paragraph contains some <em>emphasized text. </p> </em>
In other words, if an element is to contain another element, it must wholly contain that element. This is referred to as **nesting** your elements correctly.

Attributes Tell Us About Elements

What really differentiates web documents from standard documents are the links (or hyperlinks) that take you from one web page to another. Let’s take a look at a link by adding one to the example you just looked at. Links are created using an <a> element (a stands for anchor). Here we will add a link from this page to Google in a new paragraph at the end of the document. There is just one new line in this example and that line is highlighted:

```
< html >
< head >
< title >Popular Websites: Google< /title >
< /head >
< body >
< h1 >About Google< /h1 >
< p > Google is best known for its search engine, although Google now offers
a number of other services. < /p >
< p > Google’s mission is to organize the world’s information and make it
universally accessible and useful. < /p >
< p > Its founders Larry Page and Sergey Brin started Google at Stanford
University. < /p>
```

```
< p > <a href=”http://www.Google.com/”> Click here to visit Google’s
Web site. < /a > < /p >
< /body >
< /html >
```

Inside this new paragraph is the <a> element that creates the link. Between the opening <a> tag and the closing </a> tag is the text that you can click on, which says “Click here to visit Google’s Web site.”

If you look closely at the opening tag of the link, it carries something called an *attribute*. In this case, it’ s the href attribute; this is followed by an equal sign, and then a pair of quotation marks which contain the URL for Google’ s web site. In this case, the href attribute is telling you where the link should take you to. Attributes are used to say something about the element that carries them, and they always appear on the opening tag of the element that carries them. All attributes are made up of two parts: a *name* and a *value*:

- The *name* is the property of the element that you want to set. In this example, the <a> element carries an attribute whose name is href, which you can use to indicate where the link should take you.

- The *value* is what you want the value of the property to be. In this example, the value was the URL of the site that the link should take you to, so the value of the href attribute is http://www.Google.com.

  The value of the attribute should always be put in double quotation marks, and it is separated from the name by the equal sign. If you wanted the link to open in a new window, you could add a target attribute to the opening <a> tag as well, and give it a value of_blank: < a href=”http://www.Google.com” target=”_blank” >

This illustrates that elements can carry several attributes, although an element should never have two attributes of the same name.

Basic Text & Font Tags

New Paragraph: <p> Starts a new paragraph and creates a blank line between your new paragraph and the one above it.

The closing tag is </p>, but is not mandatory.

Line Break: <br> This will break your text to the next line. Two <br> tags is equivalent to one <p> tag. There's no closing tag needed for this one.

Bold: <b> Closing tag is </b>

Underline: <u> Closing tag is </u>

Italics: <i> Closing tag is </i>

Centering text: <center> Closing tag is </center>

Left aligning text: <p align="left"> Just use </p> for the closing tag

Right aligning text: <p align="right"> Just use </p> for the closing tag

Change text color: <font color="red"> The ending for any font tag is </font>

Changing font face: <font face="Arial">

Change font size: <font size="3"> (choose between 1 and 7)

Blinking Text: <blink> </blink> (only works in Netscape)

Scrolling Text: <marquee> </marquee> (only works in Internet Explorer)

Basic Structure Of An HTML Page

Here you will see a sample HTML page with the basic structure.

```
<html>

<head>

<title>Title that is displayed at the top of your web browser and also used as the title by many search engines</title>

<meta name="description" content="10-15 word description of your site read by some search engines">

<meta name="keywords" content="main keywords of your site separated by commas. Read by some search engines">

</head>

<body>
```

```
<p align="left">

This is my new web page. I hope you like it. Please come back and visit again. If you need help
creating your web site visit <a href="http://www.binbinarytechnology.com"> To Create a
Website.com </a>.

</p>

</body>

</html>
```

The <html> tag just tells the browser where the HTML starts. It is not necessary to include this tag to get your page to show.

The <title> tells your browser the title of the page and you’ll see this text at the very top of your web browser. This is also used by most search engines when indexing your page. Whatever text you have here will be the title of your site when displayed in the search engines.

The <meta name> information is also somewhat useful for some search engines. They may use whatever is in your "description" tag to describe your site. Others may randomly take an excerpt of the <body> of your page for a description of your site. The keyword tag may also be helpful with your ranking in some engines. Insert 3 or 4 of your main keywords or keyword phrases separated by commas here.

A few years ago, the <meta name> information was quite crucial in getting a top listing with the search engines. However, things have changed drastically with the explosion of so many new sites and the fact that many people abused it. I would still recommend using these tags but don’t expect to get a top ranking because of them.

The body of your site should be included inside the <body> tags.

Inserting Hyperlinks

Hyperlinks are links that take you to another page or web site. You create them by using the code below:

```
<a href="http://www.thepage.com">Name of link</a>
```

The link would appear as, Name of link

Open Links in a New Browser Window

If you don’t want people to leave your site completely when they click on links to other sites, you can set the link to open up a new window. The "target" attribute allows you to do this:

```
<a href="http://www…… "target="_blank">
```

Absolute vs Relative URLs

URL is another name for a web site address and stands for Uniform Resource Locator.

There are two different types of URLs you can use to link to various pages, absolute and relative.

Absolute URLs

Absolute URLs include the complete path to the file’s location, including the names of all the directories and subdirectories.

Let’s say you have a folder inside your web site's root directory called "music" and you want to link to a page inside the "music" folder called ay.html.

The absolute URL is:

```
<a href="http://www.yoursite.com/music/ay.html">Ay</a>
```

Relative URLs

If you don’t want to ever have to worry about going back and editing your hyperlinks if your site structure changes then relative URLs are the way to go.

Relative URLs are more or less like shorthand that tells the browser to go backward one or more directories to find the file.

Let's say you're on the page we referenced above, ay.html (located in the "music" folder) and you want to link back to the home page: (http://www.yoursite.com/index.html)

Using a relative URL, you would tell the browser to go back one directory by using the dot-slash method.

```
<a href="../index.html">Home</a>
```

The two dots followed by a slash instructs the browser to go up one more level to get to the main (root) directory.

Changing the Hyperlink Colors

The default color for hyperlinks on an HTML page is blue, but you can change it to whatever color you'd like by using the link code inside the <body> tag. Here's an example:

```
<body link="green" vlink="yellow" alink="purple">
```

In the above example, hyperlinks will be green, links that have already been visited will be yellow and active links will be purple. (An active link is one that has just been clicked, so for a split second the link will change colors as the mouse activates it).

Creating Email Links

Creating email links are just as simple. All you need is the "mailto" function to get this to work properly: <a href="mailto:bchimself23@aol.com">Mail Us</a>

```
<a href="mailto:youraddress@email.com">Email Me</a>
```

Anchor Links

If you want to create a link that will take the visitor to another section of the same page (rather than a new page or site), then you can create an anchor link. There are two steps to this process:

(1) First, go to the place in your HTML code where you want the anchor to go. This is the spot on the page that the browser will move to when a person clicks on the link.

Insert the code <a name="name1">This is the Text Where the Anchor Will Land</a> "name1" is just the name of the anchor I chose. You can it anything you want.

(2) Now to link to that section of the page, use the hyperlink

code: <a href="#name1">click here</a>

Now when your visitors click on that link, they will be taken to that section of the page.

Creating Tables In HTML

Understanding tables, in my opinion, is one of the important lessons you can learn. Tables are such a fundamental part of web page layouts, and you can do so much with your page design if you understand how they work.

Basic Table Tags

The three most important tags for tables is the opening table tag, <table> and the table row and table data tags - <tr> and <td> respectively.

The <tr> tag represents a row for the table

The <td> tag represents a cell inside the row.

Now, with that in mind, let's create a simple table:

```
<table>

<tr>  <td>A</td>  <td>B</td>  <td>C</td>  </tr>

<tr>  <td>X</td>  <td>Y</td>  <td>Z</td>  </tr>

</table>
```

And this is what the table would look like published:

A B C

X Y Z

Notice that by looking at the code, you can tell how many rows and columns are included just by looking at the code. The two opening <tr> tags indicate two rows and the three opening <td> tags on each line represents three data cells (or three columns).

Adding Table Borders

Adding a border simply involves inserting the border attribute to the opening table tag. So in the above table the code would be adjusted accordingly:

```
<table border="2">

<tr><td>A</td> <td>B</td>  <td>C</td> </tr>
```

```
<tr><td>X</td> <td>Y</td>  <td>Z</td> </tr>

</table>
```

Notice the "2" represents the thickness of the border. If you had set it to "0" then there would have been no border at all. If you wanted it very thick then you could set it to 8, for example. So now your table will look like this:

A B C

X Y Z

Changing a Table's Border Color

You can change the color of a table border by simply adding the bordercolor attribute.

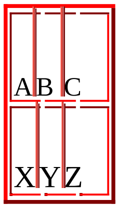
```
<table border="2" bordercolor="red">

<tr><td>A</td> <td>B</td>  <td>C</td> </tr>

<tr><td>X</td> <td>Y</td>  <td>Z</td> </tr>

</table>
```

And here's what it would look like...



A	B	C
X	Y	Z

Adjusting Table Cell Spacing and Cell Padding

You can increase the space within the table cells and the space between the cells by using the cellpadding and cellspacing attributes.

The cellspacing attribute adjusts the space between the cells and cellpadding adjusts the space within (around) the cell.

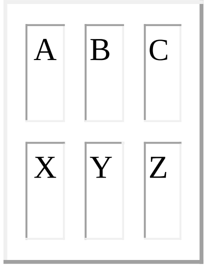
```
<table border="2" cellspacing="10" cellpadding="3">
```

```
<tr><td>A</td> <td>B</td>  <td>C</td> </tr>

<tr><td>X</td> <td>Y</td>  <td>Z</td> </tr>

</table>
```

This is what the table would look like now...



A	B	C
X	Y	Z

Specifying a Table Width

You can specify the width of a table by using either a percentage or a pixel width.

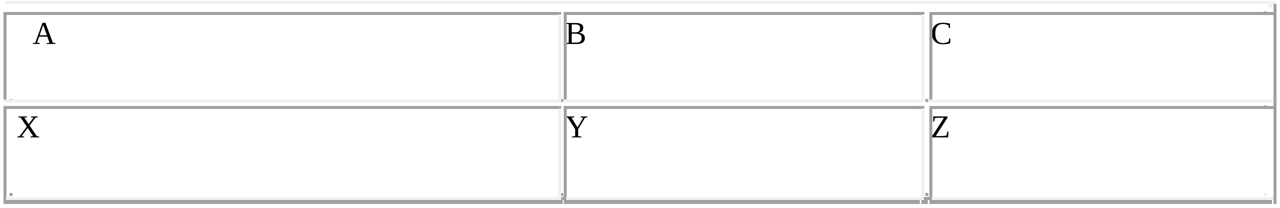
```
<table width="100%" border="2">

<tr><td>A</td> <td>B</td>  <td>C</td> </tr>

<tr><td>X</td> <td>Y</td>  <td>Z</td> </tr>

</table>
```

Since the width is set to 100% that means the table will take up 100% of the screen and the columns in the table will be adjusted evenly. Here's what it would look like.



A	B	C
X	Y	Z

As we mentioned, you can also set the table width using pixels instead of percentages. So instead of setting it to 100%, you could set it to 300 pixels:

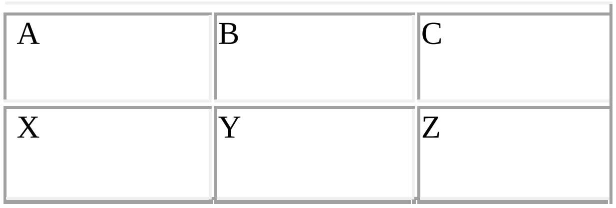
```
<table width="300" border="2">

<tr><td>A</td> <td>B</td>  <td>C</td> </tr>

<tr><td>X</td> <td>Y</td>  <td>Z</td> </tr>
```

```
</table>
```

The table would look like this:



A	B	C
X	Y	Z

Setting Column Widths

Sometimes you may not always want your columns to be the same size. If this is the case, you need to set values on your table data <td> cells. Again, you can set them by using percentages or pixel widths.

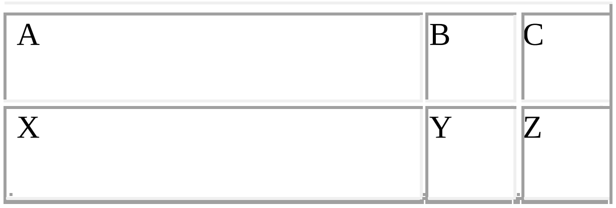
```
<table width="300" border="2">

<tr> <td width="70%"> A</td> <td>B</td> <td>C</td> </tr>

<tr> <td width="70%"> X</td> <td>Y</td> <td>Z</td> </tr>

</table>
```

This is what this table would look like.



A	B	C
X	Y	Z

See how the column width for the first column in both rows is set to 70%. Notice there is no value set for the other 2 columns. If you do not set a value for the remaining columns, their width will automatically be adjusted to take up the remaining space and they'll share it equally.

Since the table width is set to 300 pixels, and the first column is instructed to take up 70% of those 300 pixels (roughly 210 pixels), the other 2 columns divide the remaining 30% of the table (roughly 45 pixels a piece).

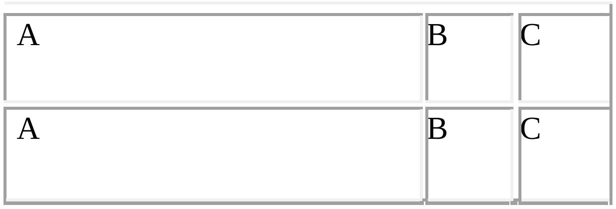
You could also have expressed the column widths of this table in pixels instead of percentages. The code would have looked like this:

```
<table width="300" border="2">

<tr>< td width="210" >A</td> < td width="45" >B</td> < td width="45" >C</td> </tr>

<tr>< td width="210" >A</td> < td width="45" >B</td> < td width="45" >C</td> </tr>

</table>
```



A	B	C
A	B	C

See how the width of the columns in each row adds up to 300 (210 + 45 + 45) -- which is the width of the table.

What’s The Difference Between Using Percentages and Pixel Widths

Many people prefer to express their table width and column widths in percentages because that will ensure that the table takes up the same amount of screen no matter how big or small the screen resolution is.

If someone is using a 21 inch monitor to view your site and you have a table width set to 300 pixels, the table will show up very small on their screen. However if you set the table width to 70%, it will take up 70% of the screen no matter what size the person is using. So it's really up to you to decide what the best layout for your tables is.

Specifying a Table's Height

You can also set the table height by adding the height tag to the table code.

```
<table height="250" width="300" border="2">

<tr><td width="210">A</td> <td width="45">B</td> <td width="45">C</td> </tr>

<tr><td width="210">A</td> <td width="45">B</td> <td width="45">C</td></tr>
```

```
</table>
```

This will produce the following table:

A	B	C
A	B	C

Horizontally Aligning the Content Inside Tables

The content inside a cell is left aligned by default, but you can also center or right align the text as well by using the align attributes.

```
<table width="300" border="2">

<tr><td width="210" align="center" >A</td> <td width="45">B</td> <td
width="45">C</td> </tr>

<tr><td width="210" align="center" >A</td> <td width="45">B</td>
<td width="45">C</td></tr>

</table>
```

A	B	C
A	B	C

See how the first column is aligned to the center? You can also right align the columns by using the align="right" inside the <td> cells.

Vertically Aligning the Content Inside the Table Cells

So far we've kept the table cell alignment at the default, which is the middle. Notice in the above table that the A, B, and C are all three aligned in the middle of their cells. Well you can change their alignment to either top, bottom, or middle by using the valign (which stands for vertical align) tag:

```
<table height="250" width="300" border="2">
```

```
<tr><td valign="top" width="210">A</td> <td width="45">B</td> <td
width="45">C</td> </tr>

<tr><td valign="top" width="210">A</td> <td width="45">B</td>
<td width="45">C</td></tr>

</table>
```

A	B	C
A	B	C

The table height was set to "250" so the alignment would be more noticeable. Notice that the A in both rows are aligned to the top. You can also align to the "bottom" or the "middle".

Creating a Left Navigation Layout With Tables

As we mentioned earlier, most left and right navigations are created using tables. All you do is create a table with one row, two columns and no border. Then align both of your columns to the top (using the valign tag) so your text will start in the top of the columns, not the middle. Depending on if you're going to have a right or left navigation, you'll make one column significantly smaller than the other.

Here's a simple left navigation layout:

```
<table width="100%" border="0">

<tr><td valign="top" width=" 25% ">Left Nav Links Here</td> <td valign="top" width="
75% ">Body Here</td> </tr>

</table>
```

And here's what it would look like:

Left Nav Links Here	Body Here

Notice The border was set to "0" but it's still showing in the example. I just did that to show how the layout would look. If you set your border to "0" you won't see any lines around your table.

So there you have it! That's a general overview of tables. There's so much more you can do with them, but if you can understand the basic layout, you'll be able to do so much with the design of your web site.

Inserting Images

Once you have the image you want to use you can insert it into your web page. Next you'll need to do is upload the graphic to your web server. Your web host will either provide the environment for you to upload your images or you'll have to use an FTP program.
When you upload the graphic, make sure you pay attention to what folder you're putting it into.
Let's say you upload the graphic called "apple.gif" to your "images" folder on your web server.
The image folder is located inside your "root" directory.

Your HTML code will look like this:

```
<img src="images/apple.gif">
```

Now let's say you have uploaded the graphic to the "fruit" folder/directory that is located inside of the images folder then the code would appear as:

```
<img src="images/fruit/apple.gif">
```

The Alt Tag

If you want text to pop up when you run the mouse over the graphic, then you need to add the alt tag.

```
<img src="images/apple.gif" alt="This is my apple">
```

Specify Height and Width

If you want to adjust the height and width of the image then you need to use the height and width tags:

```
<img src="images/apple.gif" alt="This is my apple" height="100" width="150">
```

It is recommended that you always include the height and width of your images because it makes your pages load faster.

Adding a Border

To add a border to your image, you need the "border" attribute. It’s very simple:
<img src="images/apple.gif" border="5"> The 5 represents the thickness of the border. The higher the number, the thicker the border. The number 0 is equivalent to no border. Or you could just leave the border attribute out if you do not want one.

Creating a Clickable Image

Linking images is helpful if you have buttons or banners on your site and you want the visitor to be taken to another web page or site when they click on the image. To accomplish this, use the following code:

```
<a href="http://www.the_linked_site.com"><img src="images/apple.gif" border="0"></a>
```

The first part of the code tells the browser which site to go to and the second part, of course, tells it where the image is located.

Inserting A Background

If used properly, backgrounds can really enhance your site. If used incorrectly, they can make your site very difficult to read.

If you plan to use a "loud" background that you think is cool, be sure to keep your visitors in mind. Even though you may love it, it may be distracting for the average visitor and you certainly do not want a background that takes away from your text.

In general, try to use light backgrounds on dark text.

To insert a background color on your web page, use the following code INSIDE your
<body> tag:

```
<body bgcolor="red" >
```

Inserting A Background Image

If you want an image for your background then you would use the background and image tag together:

```
<body background="images/weave.gif">
```

Keeping in mind that "images/weave.gif" is the location of the graphic. So you’re telling the browser that the graphic is located in your images folder off the root/main directory. You can also use the absolute path to your image. For example, http://www.yourdomain.com/images/weave.gif

Inserting Bullets

You can insert bullets into your pages by simply using the <ul> (unordered list) and <li> (list item) codes.

```
<ul>

<li> Bullet 1 </li>

<li> Bullet 2 </li>

<li>Bullet 3 </li>

</ul>
```

The above code will create the standard, round bullet that looks like this:

- Bullet 1
- Bullet 2
- Bullet 3

If you would like an open-circle bullet, you can change the code slightly by adding an attribute to the <ul> tag. See below:

```
<ul type="circle" >

<li> Bullet 1 </li>

<li> Bullet 2 </li>

<li> Bullet 3 </li>
```

</ul>

This will create an open-circle bullet. See below:

- Bullet 1
- Bullet 2
- Bullet 3

Indenting Text

To indent your text about an inch to the right, simply use the open and closing <i> tag around the text you want to indent.

<i> This text will be indented indented </i>

This text will be indented indented

Special Characters

Code	**Symbol**	**Description**
&trade;	™	Trademark
&amp;	&	Ampersand
&reg;	®	Registered trademark
&copy;	©	Copyright
&dagger;	†	Dagger
&raquo;	»	Right pointing double angle quotation mark
&laquo;	«	Left pointing double angle quotation mark
&#151;	—	Em-dash
&deg;	30°	Degree
&frac14;	¼	Quarter
&frac12;	½	Half
&frac34;	¾	Three quarters
&middot;	·	Middle dot
&iexcl;	¡	Inverted exclamation mark

Horizontal Lines / Dividers

Horizontal lines are great for breaking up paragraphs of text or separating sections of your page. To create the standard/default horizontal line, you'll just simply insert the <hr> code. This will create a line that looks like this:

Setting Horizontal Line Width, Size And Color

To set the width of your line, simply add a "width" attribute. For example, <hr width="250"> will create a line that is 250 pixels wide. See below...

You can also set the size of the line by inserting a value. So if you wanted a line that is 250 pixels wide and 6 pixels thick, you'd use the following code:

<hr width="250" size="6">

This will produce the line below:

Lastly, you can control the color of the line by inserting the color attribute:

<hr width="250" size="6" color="blue">

Creating Email/Feedback Forms

Probably one of the easiest forms to setup is the FormMail form. However, in order to make this work your site must be hosted with a host that allows you to upload files to your cgi bin.

HTML Code for Forms

Once the FormMail file is in place on your server, all you have to do is create a page with the form code to make it work.

Below you'll find the code for creating a form in which your visitors can fill out and the results will be emailed to your email address.

The form has the following fields:

- Text box
- Password Field (hidden text)
- Check Box
- Radio Buttons

Here is what the form will look like and below you will find the HTML code. You will need to change some variables (in red) to customize it to your own needs.

Your Name:

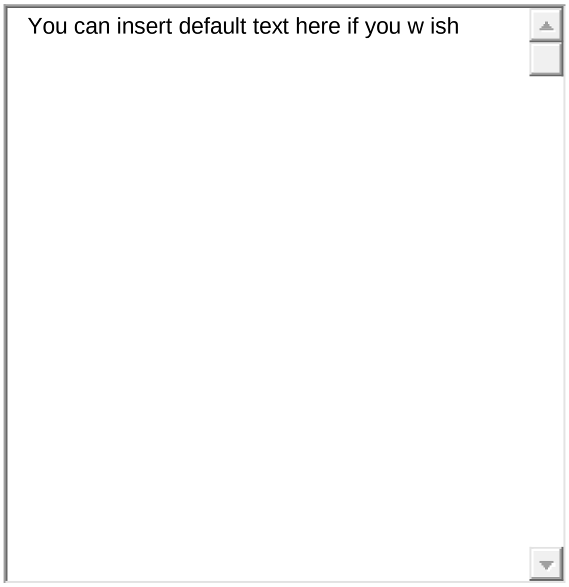
Password:

Please Place me on your mailing list: ☑

What's Your Gender?

How Old Are You? ● 0-25 ○ 26-50 ○ 51 and Over

Comments:



You can insert default text here if you w ish

Here's the code for the form you can copy and paste. Feel free to remove the HTML code for fields you don't need or adjust the various values to fit your liking.

Notice the red text. This is what you need to edit to customize the form.

1) The 1st line should be changed to the address of the FormMail script on your server.

2) The 2nd line should be the address of the page the user views after they submit the form. You may need to create this page.

3) The 3rd line is the email address the form results will be sent to.

4) The 4th line is the subject of the email when the form is sent to your address.

```
<p>Password:
<input TYPE="password" NAME="password" SIZE="10" MAXLENGTH="10">
</p>


<p>Please Place me on your mailing list:
<input TYPE="checkbox" NAME="mailing-list" VALUE="Yes"
checked> </p>
```

```
<p>What's Your Gender?
<select NAME="Gender">
<option VALUE="Male">Male
<option VALUE="Female">Female
</select>
</p>


<p>How Old Are You?
<input TYPE="radio" NAME="Age" VALUE="0-25" checked>0-25
<input TYPE="radio" NAME="Age" VALUE="26-50">26-50
<input TYPE="radio" NAME="Age" VALUE="50 and over">51 and
Over </p>


<p>Comments: <br>
<textarea NAME="comments" ROWS="10" COLS="50"
wrap="virtual"> You can insert default text here if you wish </textarea>

</p>


<p>
<input TYPE="submit" NAME="Request" VALUE="Submit This
Form"> <input TYPE="reset" NAME="Clear" VALUE="Clear Form">
</p>


</form>
```