

Web Wizard Secrets



Version 1.1

Web Wizard Secrets

by Shano DeLeon

Contents

Introduction to Source Code	3
Free Webmaster Tools	8
Easy Page Templates	9
Easy Page Wizard	13
More Web Page Tricks	21

Click on the Bookmarks Tab on the left and use the navigation features of Adobe Reader to jump to any section. You can click on the blue web links to reach the web sites when you are connected to the Internet.

Your Resale Rights

You get resale rights to the Web Wizard Secrets Package as a free bonus. Our reseller package includes the sales page, email capture page, fulfillment page, supporting files, and instructions. [Click Here](#) to reach the download page.

Legal Notice

© Copyright 2007 Shano DeLeon. All rights reserved.

The author, publisher, and vendor assume no liability for the reader's use of the information contained herein or for his/her use of the accompanying files. Though we strive to present accurate and useful information, we do not warrant this book to be free from errors or omissions. The reader assumes full responsibility for the use of this information and for his own business decisions. The full extent of liability of the author, publisher, and vendor consists of the price paid to the vendor for the ebook and accompanying files. We encourage readers to use common sense and to consult with legal and business professionals when appropriate.

Introduction to Source Code

These instructions along with the accompanying html, css, and js files will make it easy for you to create better looking web pages with more features and neater HTML coding than 99% of the pages that are sold with resale rights or private label rights.

Even if you deal purely with web pages that are already constructed, having come with resale rights or private label rights, you are unlikely to be able to customize them successfully without knowing something about HTML. In many cases you have to know a lot about HTML because the coding is in such a muddle that you can't even fix up obvious page errors without getting tangled in the mess.

That's because those so-called "professionally designed" sales pages were really constructed by rank amateurs who can barely tell HTML from a BLT on toast. And I might add that they commonly show little knowledge of correct grammar or punctuation as well.

HTML is actually easy to learn and apply. It looks scary when you look at the mess you get with resale rights products, and it sounds scary when everyone keeps assuring you that you don't have to know any HTML, but it's basically simple.

Someone who knows HTML can construct a web page using nothing but a simple text editor like MS NotePad. Doing so, however, is cumbersome because the source code page looks quite different from the page displayed by the browser. To see the effect of his coding, the web designer would have to frequently shift to the browser view and back again.

Web Construction Programs

Web construction programs provide a way to work on your web page while seeing it almost the same as it appears in a browser. They also give you point and click access to the formatting instructions you insert in your page, saving you from having to remember the codes for it.

I believe Adobe Dreamweaver to be the best one. Microsoft Front Page is the other high-priced, full-featured leader, being about as easy to use, but having the drawback of not producing clean HTML code. It adds tags and does a few other things that are used in the functioning of Front Page without affecting the final display by the browser.

Side Note: Front Page has some features that work only if the web host enables those functions on the server. Almost all hosting services offer such an accommodation. However, enabling Front Page features can disable some standard features that

other sales pages might be designed to use. It's best not to have Front Page features on a sales page that is distributed with resale rights.

If you do not already have a web page construction program, I recommend "My Free Website Builder". It's free and it's easy to learn and use, even having videos to help you. It includes the ability to upload your pages to the server. You can get it from the "Free Webmaster Tools" section that comes after this one.

Naturally, it lacks some features that are in the \$399.00 Dreamweaver application, but that's the point of learning a little HTML. You can insert whatever is needed into the source code to display the effect you want on your web page. In fact, working directly with the source code is the only way to get some of the web page tricks that I cover in this book.

In the following sections I give you a broad outline of what HTML is about. With that understanding, you can look at and edit an HTML page without being intimidated by it. Everything else about HTML is just detail.

What Tags Are

HTML is not a programming language in the usual sense. The letters stand for hypertext markup language. The name derives from the days when newspaper type was set by hand. The writers and editor would put together the text for the newspaper. Then they would mark it up with tags and codes that indicate to the typesetter how the type is to be arranged to produce the desired printed page. The tags and codes were called the markup language.

HTML is simply the rules for tags that are added to the text to tell the browser (Internet Explorer, Netscape, etc.) how to display the page on the screen. The text itself is not coded in any way. When people talk about HTML code, they are referring to the codes inside the tags and in places removed from the page text. The term "source code" refers to all of the type that makes a page, the text and tags and other coding. It's what you view with a plain text editor.

Tags come in pairs that enclose the text or images they apply to. The first tag can be called the starting tag or opening tag. The second one is the ending tag or closing tag. For example, a paragraph starts with a <p> tag and ends with a </p> tag. The starting tag can contain codes that indicate "attributes" that apply to the text between the tags.

An ending tag is just like the base starting tag (before attributes are added) except that it has a forward slash (/) as the first character inside the pointy brackets.

You have to close tags in reverse order from the way they were opened. As an analogy, imagine walking onto your property, then into your house, and then sitting on your sofa. To reverse it, you have to get off the sofa, leave your house, and then leave your property. It's property, house, sofa going in; and sofa, house, property going out. It doesn't work if you mix up the order.

The Head and Body

The source code of a web page is divided into a head section and a body section. The head contains general page information and possibly formatting information. The body is the part containing what is displayed on the screen. The tags that define a basic page look like this:

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
<title>Page Title</title>
```

If desired, include the following information for search engines:

```
<meta name="robots" content="index, follow">
<meta name="description" content="Describe what your page is about.">
<meta name="keywords" content="Keywords you'd like your site to be listed under.">
```

Additional page information goes here if needed.

```
</head>
<body>
Page text and images here
</body>
</html>
```

Though the simple html tag above is sufficient if you use just common tags on your page, a web page construction program may refer to a specific version of HTML and may put more information at the top of the page before the html tag. Go with what your program has.

Today's leading web construction programs use an extended version of HTML called XHTML. If your page specifies XHTML, then you need to have a slightly different end on your tags. Instead of ending with a simple ">", they have a space and a slash in front of the pointy bracket.

Examples: `<link rel="shortcut icon" href="images/favicon.ico" />`
`
`

Follow the examples that your web construction program generates.

Tables and Cells

A table is a rectangular area with rows and columns of cells. Such a construction is used to position text and images on a web page. This ebook is about constructing sales pages. Their basic structure can be as simple as a single table containing a single cell. Below I have examples of a little more than that to show how the tags work.

Here are the tags for a single table with a single cell.

```
<table>
  <tr>
    <td>
      Content
    </td>
  </tr>
</table>
```

I use indentation to help keep track of things. The first tag opens a table. The second tag, <tr>, opens a row. The third tag, <td>, opens a cell. Then the tags close the cell, the row, and the table.

Below to the left are the tags for a table with two cells side-by-side, which means they are in the same row. The cells are in the same row because I close the first cell and open a new one without closing the row first. Below to the right are the tags for a two-celled table with one cell under the other.

```
<table>          <table>
<tr>            <tr>
  <td>          <td>
    Content     Content
  </td>        </td>
  <td>          </tr>
    Content     <tr>
  </td>        <td>
</tr>          Content
</table>       </td>
               </tr>
               <table>
```

Tags must be in the proper order, but they do not have to be placed on the page in a particular way. For example, they can be arranged this way:

```
<table><tr><td>Content</td></tr></table>
```

If you close a table and then open a new one, you get one table under the other. They do not come out side by side. To get two tables side by side, you need to have two cells side-by-side and then put a table in each cell. Tables can be put inside cells that are inside other tables.

Special Tags and Characters

Paragraphs start with a `<p>` tag and end with a `</p>` tag. Web browsers automatically put a vertical space between paragraphs like I do here.

Word wrap consists of automatically starting a new line when the words get to the end of a line. Word wrap is handled by the browser. You actually do start new lines in your source code, but those line breaks are ignored by the browser. No matter how many spaces or tabs or line breaks you put between two words in your source code, the browser counts it as no more than a single space between the words.

The browser then adds line breaks (carriage returns) as necessary to fit the words into the available space in the screen display.

There may be a place where you want to force a new line to start without starting a new paragraph. In a word processing program and in the design mode of a web page construction program, you can do that by holding down the Shift key when you press the Enter (carriage return) key.

In source code, that comes out as the tag `
`. It is one of the few tags that do not go with an ending tag.

If you start a new line purely to make some vertical space on your page, you won't get that space from a lone `
` at the end of a paragraph. You have to put something on that new line to have the browser make room for that line and give you the vertical space. You can accomplish that by putting a non-breaking space on that line. (I'll explain what that is next.) Most people, however, do it by simply putting two `
`'s in succession.

In some cases, you may want to have additional spaces between two words. You cannot do it by adding spaces in your source code because nothing over one space matters to the browser. A non-breaking space is a space that acts in most ways like a letter. You can put as many as you want in a row, and each one takes up its allotted space. It gets its name from the fact that if it is used as a space between two words, the browser will not put a line break between the two words when it does word wrap. It will look for another place to put the line break.

You get a non-breaking space in a word processing program or a web construction program (in design view) by holding down the Shift and Ctrl keys while you press and release the space bar. In source code view, a non-breaking space is represented by ` `.

Span tags are used to apply attributes to a partial line of text. The starting tag is `` and the ending tag is ``. Attributes (such as background color) are specified in the first tag.

Free Webmaster Tools

Web construction programs and graphics programs can be very expensive—and they can be free. It's your choice. Spending a lot of money on software does not translate into better web pages. You can do it just as well with the files that come with this ebook and the free resources and tools listed here.

To have a web site, you need web hosting. You can get it free, but I do not recommend free web hosting for a sales page. Adequate web hosting, however, can cost as little as a few dollars per month. See my web hosting page for suggestions: <http://easylifemarketing.com/hosting.html>

Note: You must be connected to the Internet for the links on this page to work. Unless otherwise noted, all programs referenced on this page are for the **Windows operating system only**.

For web page construction, I prefer Dreamweaver, but it costs \$399.00 and takes two to eight weeks to learn how to use. You can make equally good web pages with “My Free Website Builder”. It costs nothing, and you can learn how to use it in half an hour. Just watch the videos, work with the program a little, and you have it. To download it, go to MyFreeWebsiteBuilder.com.

Color Spy 7 is a handy little program for getting the color number of any color you see on the web. It also has a chart on which you can vary just the brightness, producing numbers for colors that differ only in brightness. That's a way to get a set of colors for your custom scroll bar. Download Color Spy 7 at <http://amazingplanet.com/colorspy7.htm>. After installing it, click on the options button, click on “Output Format”, and select HTML.

ColorPic also can give you the number of any color you see on the web. It includes a web color chart from which it's easy to select colors for your web page. Download it at <http://iconico.com/colorpic/gettingStarted.aspx>.

Free image editing software is available at: <http://www.freeserifsoftware.com/software/PhotoPlus/default.asp>

You can make the bookmark for your site have a little picture in people's favorites or bookmark list. It's called a favicon. Your favicon also is displayed in the address bar of Internet Explorer 7 when someone views your web site with it.

There are some web sites where you can upload an image, and they'll turn it into a favicon for you. Here's one: <http://favicon.co.uk/>

The Easy Page Templates

Included are two sales page templates with the tables and cells already set up for you. Before I talk about the templates themselves, let me discuss the factors considered in setting up the page that way.

On a sales page, it is desirable to have the reader start at the beginning and read straight through your sales message. The first step in accomplishing that is to eliminate distractions. Don't have adjacent columns of other information. Don't have a navigation menu leading to other places. And certainly, don't have ads or banners on the page.

Column Width and Margins

Your sales message should be in a single column that's as long as it takes to say what you have to say. The column of text should be no more than 600 to 650 pixels wide for two reasons. One is that some people still have 800 x 600 pixel screens and possibly have their browsers set to put columns of something on one or both sides of the web view area. You don't want to make them have to scroll back and forth to read each line.

More importantly, for people with the usual 1024 x 768 pixel screens, very wide lines of type are less inviting and more of a chore to read. It's hard to go to the next line without losing your place. An effective sales page is inviting and easy to read.

Most sales pages are set up as a white column against a colored background that shows on both sides. The idea is fine, but in sales pages for resale rights products, it usually is poorly executed. Some say they get the best results with a plain white background. I can see how a plain white background can beat the ugly page setup they're probably comparing it to, but I find plain white backgrounds to be unattractive.

The most common mistake made with colored backgrounds is to make the white margins on each side of the text too small. I think they should be 25 to 35 pixels wide. You get that by specifying cell padding in your table tag or your cell tag. As an alternative, you can specify margins in your cell tag. You'll see what I'm talking about when you start working with my page templates.

A common width of the header and main table is 650 pixels. I'd use cell padding of up to 35 pixels on that, leaving a 580-pixel wide column for content. For a 600-pixel wide column, I'd recommend a table width of 660 or 670 pixels. These are wider margins than you normally see. The wider margins are more pleasing to my eye, but nothing more. I have not tested its affect on sales.

Background Color

The colored background on each side of the text area should not be distracting. The color should not be so intense that it pulls at the reader's attention. Putting big, fat words in the background area is even worse.

Use a toned down color that is closely related to the base color in your header or product image. If you want to use a background tile with the product name on it, make the lettering a slightly different shade of the background color. Nothing in the background should stand out and draw the eye away from the sales message. Also, the reasons for not having the product name in the header apply to the background tile as well.

The Header

A sales message should start with a compelling headline that captures the attention and interest of the kind of readers that the product is targeted to. It is most effectively done by presenting the main benefit that the reader will learn about by reading the text.

But that's not what you see at the top of the sales pages that come with resale rights products. No. You see a colorful, eye-catching header that screams out, "Look Here! I have a product I want to sell to you, so read my sales pitch below!"

Naturally, the header doesn't use those exact words, but everyone knows exactly what it means the instant they see it. Some marketers report that they tested their sales page with and without a header, and the version without the header does much better. I can see where that makes sense.

But that doesn't settle the issue. When I see a sales page with no header, it looks kind of naked to me. And it still looks like a sales page. Plus, for some products, people expect the page to be a sales page.

Purely informational sites do indeed have headers, and a good tactic for getting a sales letter read is to start it out as if it's an informative article. So it seems that a header that makes the site look like an informational site would be a benefit. In this case, the header should not include anything that looks like a product name.

A header can be purely decorative and say nothing at all. Or the main headline could be put in the header. I just wanted to point out these various factors to consider. Your header or lack of one will depend on your product, your market, and your sales approach. My sales page template is designed so a header can be easily added or removed.

The Page Templates

If someone with a dial-up connection goes to your site and sees a blank screen for more than fifteen seconds, there's a good chance he'll give up and leave. You have less chance of losing impatient web surfers if you can get something displayed on their screen quickly. The best way to do that is to have the top part of your web page separate from the rest so it loads fast.

PageTemplate_1.html is set up with the header in its own table, then a main table, then a second main table, then a footer table, and finally a disclaimer table at the bottom. The idea is to make the top main table just big enough along with the header to fill the screen to the bottom. Then that part loads fast, and the visitor can start reading while the rest loads in due time.

To make the two main tables fit together seamlessly, the margins and padding are handled differently than usual. Put only enough content in the top table to extend to the bottom of a common sized screen.

PageTemplate_2.html is set up in the common way with one big table. If you are not concerned with loading time, you can use Template 1 and delete one of the main tables or you can use Template 2.

I wrote a considerable amount of commentary on the page templates themselves. Read them for more details on the layout and for explanations of several features that I included.

The templates have a smaller table inside the sales letter area. That's to show how you can display a rectangle with a border and its own background color. They are often used on web pages for testimonials. Wherever you want one, simply copy the code for the table, including the beginning and ending table tags. Then paste it into the source code wherever you want such a rectangle. Adjust the border, width, padding, and background color to suit. The length will automatically extend to accommodate the content.

Adding Images

The header has an image specified and a background color specified. The background will be covered by the image, but it's good to have a color specified for the header because that's what the viewer sees between the time the text is loaded and the images are loaded. The included header and footer are for demonstration purposes.

In both templates, I set the footer image as a background image. That makes it possible to put text in the footer. Now that you see how it's done, you may want to consider setting up your header the same way. Then you can type your own text in the header and change it any time you want.

Normally when a background image is used, it is much smaller than the space it goes into. The browser then displays a sufficient number of copies of it to fill up the space.

The image tag with its attributes looks like this:

```

```

You can see that the file is in a folder named “images”. It’s common practice to put all images for a web site in such a folder. Web page images are normally jpg files or gif files.

On resale rights sales pages, complicated tables and cells are often used to position images. They are seldom necessary. You can position images using specifications in the image tag. The align attribute tells on which side of the text the image is to appear. hspace=“12” means put a space of 6 pixels on each side of the image. Choose a number that produces a pleasing looking space between the image and the text.

vspace=“16” means put a vertical space of eight pixels above the image and another eight below the image. You can make the vertical space large enough to keep text from appearing near the top or bottom of the image. You can thereby get the same effect as having the image in its own cell.

The “alt” attribute is the name displayed in the space before the image is loaded or instead of the image in the case of a viewer who has his image display turned off, possibly to save loading time. It also pops into view for a short time when the mouse pointer is put over the image.

Rollover Images

A rollover image is an image that changes when the mouse pointer is over it. I like to use it for order buttons that change color or brighten up when pointed to. It is done by having two separate images of the same dimensions and then having a script that switches from one to the other when the mouse pointer is over the image.

It requires some HTML coding on the page and a JavaScript in the page head or in a separate js file with a link to it in the page head. The best way to understand it is to look at the code for the example on PageTemplate 1.

The Easy Page Wizard

The easy page wizard is my wizard.css file. Technically, it's called a cascading style sheet. But there's no cascade to see and no sheet to hold, so I call it a page wizard.

A style is a set of attributes that normally might appear in a tag. But instead of packing the tag with a bunch of attribute information, the set of attributes can be put in a style sheet and given a name such as "xy". Then the tag just needs to contain the statement: class="xy"

The format for specifying attributes in a style sheet is slightly different from the format used in tags. Tags use equal signs and quotation marks. Style sheets use colons and semicolons.

Styles can be defined in the page head. They are put between the style tags like this:

```
<style type="text/css">  
the style statements  
</style>
```

Or styles can be defined in a separate file that has the css extension and is linked to the web page by this tag in the page head:

```
<link href="filename.css" rel="stylesheet" type="text/css">
```

It's handy to have a separate css file for a multi-page web site because you just have to make it once; then link all of your pages to it and have a site that looks coordinated.

A style sheet can be made with a plain text editor such as NotePad. Big website construction programs have point and click ways to generate a style sheet. However, I prefer to edit the style sheet directly. Dreamweaver lets me do it. If your program doesn't let you do it, you can still edit it outside the program using a plain text editor.

I recommend that you turn off Dreamweaver's feature of automatically defining a style for each set of attributes you put in a tag. Because of that feature, I've had to wrestle with resale rights web pages that had over 50 defined css styles with none of them having descriptive names and less than ten of them used more than once in the completed page.

With my page wizard, you have most of the work of designing a style sheet already done for you. Make a copy of the wizard.css file and give the copy a name that goes with your web site. Then modify some items to suit your web page and delete the ones you don't use on that page. Insert the link in your page head, and your web construction program will recognize the style sheet.

If you have an already-constructed web page that does not use styles and you simply want to add some effects, you can add a style sheet (either in the head or in a separate file) that contains only the items you want to use.

You can look at my sales letter for examples of the styles in use. The style sheet for my sales letter has mostly the same styles as the wizard.css style sheet.

The basic format of a style is this:
name { attribute1:value; attribute2:value; attribute3: value }

You'll notice, however, that instead of putting all of the attributes in a line, I gave a separate line to each one and then put the closing bracket on a line of its own. That's standard practice to make it easier to see what you have and to make changes. A semicolon is optional after the last value before the closing bracket.

The styles that have regular tag names apply every time that tag is used. The styles with names that start with period are custom styles. They are applied by putting a `class="stylename"` statement inside a tag.

Below I've copied the styles from wizard.css and added commentary. I have blue headings in places to help you find items of particular interest.

Custom Scroll Bar

```
body {  
  scrollbar-base-color:#99C4E5;           0  
  scrollbar-track-color:#B0D1EB;         +20  
  scrollbar-face-color:#99C4E5;          0  
  scrollbar-highlight-color:#C5DDF0;     +40  
  scrollbar-3dlight-color:#B0D1EB;       +20  
  scrollbar-darkshadow-color:#6DAADA;    -40  
  scrollbar-shadow-color:#83B6E0;       -20  
  scrollbar-arrow-color:#DCEBF6;        +60  
  background-color:#B0D1EB;  
  background-image:url(images/ltbluebg.gif);  
  margin-left:0px;  
  margin-right:0px;  
  margin-top:20px;  
  margin-bottom:20px;  
}
```

The first eight attributes above describe colors on the vertical scroll bar. You can omit those lines and just have the default scroll bar on your page. This works with Internet Explorer, but not all browsers. When a style attribute is not recognized by a particular browser, it is simply ignored.

I like to make the scroll bar match my page, and I prefer it to be subtle so it doesn't draw attention away from the sales copy. I get the colors by using

different shades of the background color. If the background color is very light, I generally use a darker shade of it as the base color for the scroll bar.

You can get the colors using the text color picker in a web construction program or by using the color chart in ColorSPY7. (See the webmaster tools section.) Start with your base color and vary just the lumens or brightness to get the other shades. The numbers I have off to the right indicate how many lumens I increase or decrease the brightness to get the desired shade.

The next lines are the body background color and background image. If you have a background image, it's still good to specify a background color because that's what shows before the image is loaded.

If you do not specify any body margins, a margin of about 15 pixels will outline your web page by default. For a regular web site, it is desirable to have no margin at all. A sales page is different because it looks more like a letter set against a background. Set the right and left margins to zero so someone viewing the page on an old 800 x 600-pixel monitor gets the benefit of the full width.

Webmasters normally set the top margin to zero when they use a header. But, since the background shows on both sides of the text part anyway, sometimes it can look better to have some top margin even with a header. When there is no header, I think a top margin of at least 20 pixels definitely looks better than running the page right up to the top. Likewise, I think a bottom margin of at least 20 pixels looks better than running the page all the way down to the bottom. I like to put the disclaimer outside the sales letter area set against the body background below the footer. In that case, I think there should be an additional 20-pixel margin below the disclaimer.

Default Typestyle

```
table {  
    font-family:Georgia, "Times New Roman", Times, serif;  
    color:#000000;  
    font-size:15px;  
}
```

I put the specifications for the default type style in the table tag because all text will be inside tables. The style specified there is what you will get automatically when you don't specify anything in the cell or paragraph tag. I chose a serif style of type because people in the United States are accustomed to a serif typestyle for the text in newspapers, magazines, and books. It is also customary to use sans serif type for headings. So that's the way I do it. You can change it to anything you want when you format your own page.

Definition: "Serifs" are the little lines on the ends of the main lines in lettering. At the bottom, they look like little feet. "Sans" means without.

Black color, normal font weight, left alignment, and no background color are the default values if not otherwise specified.

Text display is handled by the browser, and it can display only the fonts that are available in the viewer's computer. That's why, in specifying a font, backup font names are included after the desired one.

```
.maincell {
  border-left:1px solid #000000;
  border-right:1px solid #000000;
  padding-left:35px;
  padding-right:35px;
  padding-bottom:20px;
}
```

This style's name starts with a period, indicating that it's a custom style. Its purpose is to have a convenient way to add the borders and padding needed in the two main cells.

Link Colors

```
a:link {
  color:#0000DD;
  text-decoration:none;
  font-weight:bold;
}
a:visited {
  color:#0066CC;
  text-decoration:none;
  font-weight:bold;
}
a:active {
  color:#CC0000;
  text-decoration:none;
  font-weight:bold;
}
a:hover {
  color: #DD0000;
  background-color:#DDDDDD;
  text-decoration:underline;
  font-weight:bold;
}
```

I put in the colors I like to use, which is a red that is a little darker than straight red and also a darker blue. I made the visited link a little lighter blue. It needs to be different from the basic link color only on pages where there are several links and users want to see which ones they did or did not visit (or download from) yet. I put a background color on the hover link mostly to show that it can be done. Another thing you can do is put a line across the top of the link. Here's how to specify both an underline and an overline:

```
text-decoration:underline overline;
```

Side Note: If you have just the customer service text link on your page and you want to put it in the footer, choose your link colors to work against that dark background. I'd use a white link that turns to #FFCC00 when the pointer is over it. If you have other links in the white part of the page, it is possible to set the style colors to work with a white background and then have custom colors for the link in the footer. Setting that up involves a detailed explanation, which I put on Template 1.

Here are two easier solutions. One way is to specify `background-color:#FFFFFF`; on all four link versions. Then the links have their own white background when used in the footer and look normal when placed on the rest of the white page. The other way is to place a font tag on each side of the link text in the footer and use it to set the text color so it contrasts with the background. Then the link doesn't change color when the mouse is over it, but it gets the other effects that are specified in the style sheet, such as an underline or overline. If you use my cs.js JavaScript for putting a customer service link in the footer, the font tags have to be in the JS file to be on each side of the text.

```
h1 {
  font-family:Verdana, Arial, Helvetica, sans-serif;
  font-size:30px;
  color:#DD0000;
  font-weight:bold;
  text-align:center;
}
h2 {
  font-family:Verdana, Arial, Helvetica, sans-serif;
  font-size:24px;
  color:#DD0000;
  font-weight:bold;
  text-align:center;
}
h3 {
  font-family:Verdana, Arial, Helvetica, sans-serif;
  font-size:18px;
  color:#0000DD;
  font-weight:bold;
}
```

These are the headline styles. Six are allowed, but I've read that some browsers get confused if you actually use more than three on a page. Red tests out to be the best color for headlines. I prefer a slightly darker version of it. I made h1 and h2 automatically centered. h3 does not have alignment specified, so it will default to left if nothing is specified in the tag. That gives the webmaster some options in aligning it. I made h3 blue. Of course, you can change them to suit yourself.

Automatically Space List Elements

```
li {  
  margin-bottom:1em;  
}
```

This is the tag used for each element in a list. I specified a margin so they would be vertically spaced in a more eye pleasing way. For a smaller vertical space, change em to ex.

Automatic List Bullets

```
ul {  
  list-style-image:url(images/reddot.gif)  
}
```

This is the tag for an unordered list. That means a list with bullets instead of numbers. I specified a default bullet image. If a list is made with nothing else specified, it gets the bullet images specified here.

Alert: Don't miss the dots. They came out tiny with the typestyle I picked, so let me point out that all the styles from here down begin with a period. That means they are custom styles that are named as classes in the appropriate tags. The initial period appears only in the style sheet, not in the tags.

```
.redcheck {  
  list-style-image:url(images/redcheck.gif)  
}  
.redchecksquare {  
  list-style-image:url(images/redchecksquare.gif)  
}  
.bluediamond {  
  list-style-image:url(images/bluediamond.gif)  
}  
.bluearrowhead {  
  list-style-image:url(images/bluearrowhead.gif)  
  margin-left:70px;  
  margin-right:70px;  
}
```

The above custom styles can be used as a class in an unordered list tag. Example: `<ul class="redcheck">` Then that list displays with red check marks for bullets. It makes things easy when you name the images for what they are and then name the styles the same as the images. These are just examples. Make your own to suit your sales page. You can also put margins in this tag to set your list in a narrower column. I did that in the last one above just to show you an example.

Side Note: What's the difference between margins and padding? Margins are the distance in from the border or edge while padding is space added from the text out. When both are specified, padding is placed within the margins. Any text background color extends through the padding.

```
.bigtext {
  font-family:Verdana, Arial, Helvetica, sans-serif;
  font-size:16px;
  font-weight:bold;
}
```

This is for the big black text that normally is used in the top part of a sales page. You can make up any custom styles you want for text in various places on your page. It's smart to assign a descriptive name to it.

```
.smalltext {
  font-family:Verdana, Arial, Helvetica, sans-serif;
  font-size:11px
}
```

I used this text for the disclaimer. Black color and left alignment is assumed when nothing else is specified. If you use a style just once on the page, it's hardly worth putting it in a style sheet. You can just specify it in the tag. On the other hand, putting it in a style sheet encourages a restraint that is lacking in some makers of web pages for resale rights products. A page is more pleasing to view when it uses a limited number of different type styles.

```
.footertext {
  font-family:Verdana, Arial, Helvetica, sans-serif;
  font-size:10px;
  color:#FFFFFF;
  font-weight:normal;
  padding-left:25px;
  padding-right:25px;
  vertical-align:middle;
}
```

This is designed for the footer cell because it contains the padding and vertical alignment. Though it's used in just one place, it keeps the page neater to make a style for it rather than put all that information in the cell tag. Because the font is specified in the cell tag, any text I put in the cell is automatically formatted correctly. That's handy for when I use text created by a JavaScript.

```
.courier {
  font-family:"Courier New", Courier, monospace;
  font-size:14px;
  text-indent:2em;
  color:#000000;
}
```

This custom style uses the old typewriter font. It's good for testimonials. I included **automatic indentation** of the first line of paragraphs. To make the indentation smaller, use 2ex. You can set automatic indentation as a style in the paragraph tag itself. Then all paragraphs are indented unless their tag has a style class with indentation set to zero. If you center a line of text that has automatic indentation of the first line, the text will not actually be centered. You must set the indentation to zero to get it centered properly.

Text Highlighting

```
.highlight {  
  background-color:#FFFF33;  
  font-weight:bold;  
}
```

This is handy for highlighting text. I made it bold because I always use bold text with highlighting, so it's convenient to include it. You can make styles for various colors of highlighting. This one is yellow that is toned down slightly by having a little white in it.

Drop Capitals

```
.drop {  
  float:left;  
  color:#0065BD;  
  font-size:46px;  
  line-height:36px;  
}
```

You can use this style to easily make drop capitals without having to use images. It's sized to drop one line below the present one. You can change the font size and line height to fit what you want to do. And pick a color to suit your page.

Custom Indentation

```
.indent90 {  
  margin-left:90px;  
  margin-right:90px;  
}
```

This shows how to give custom margins to a section of type to set it off for special attention. Block quote tags have some default margin. If you put this style class in a paragraph tag or a block quote tag, you get exactly the margins you want.

A Box With No Table

```
.bluebox {  
  margin-left:80px;  
  margin-right:80px;  
  padding:20px;  
  border:4px double #000000;  
  background-color:#DCEBF6;  
}
```

When this style class is used in a paragraph tag, it puts a bordered box (rectangle) on the page. Top and bottom margins can be specified as well. Border styles can be solid, dashed, dotted, double and some types with a three-dimensional effect. It can only be used for one paragraph at a time. If used in a blockquote tag, you get unwanted extra top and bottom padding.

More Web Page Tricks

Background Info on Scripts:

Scripts are little programs. Two kinds are commonly used in web pages. JavaScript is a programming language for scripts that are inserted in the page source code and are then executed by the viewer's browser. In other words, the browser reads the JavaScript, executes it, and displays the results. PHP is a programming language for scripts that run on the server and produce results that then can be used by the browser. In other words, the browser never sees PHP source code; it only sees the results after the PHP script is executed on the server. (The server is the big computer that puts your web site on the Internet.)

Link Cloaking

When you have a recommendation at a web site, people are more inclined to click on a link if it looks like a domain name, not an affiliate link. In some cases, it may help to have the domain name for the text link instead of saying "Click Here". Either way, people can see the actual link in the status bar, unless you use this trick to show them what you want them to see. Here's the source code for such a link:

```
<a href="http://your_actual_link_code" target="_blank"
onMouseOver="window.status='What_displays_in_the_status_bar'; return true"
onMouseOut="window.status=' '>What_displays_on_the_web_page</a>
```

After `window.status=` the characters are an apostrophe, a space, an apostrophe, and quotation marks. Omit `target="blank"` if you want the current web page to close when the new one opens.

When you promote a ClickBank affiliate link in an email or on a web site, people will go to the sight with their own ClickBank nickname in the link if you reveal the link code. For links given in emails, even when people can't hijack a link, they may use it with the affiliate ID number omitted or they may accidentally type in the wrong ID number. These problems are solved by setting up a script on your web site that you refer people to and that then redirects them to your affiliate link.

The most secure way to accomplish that is with a PHP script. There is no way for visitors to view or download a PHP script from your site. I recommend that you set up a folder or subfolder that holds the PHP script as `index.php`. In the absence of `index.html` or `index.htm`, `index.php` is the default file for a folder. If you name your folder "WebWizardSecrets", for example, then you can refer people to "www.yourdomain.com/WebWizardSecrets".

Use a plain text editor such as Notepad to make your php script. The script is very simple:

```
<?php
header("Location:http://YourLink.com");
exit();
?>
```

Save it as index.php or change the name to that after you save it. Then upload it to the folder you made for it. It's simple and crack-proof.

Image Protection

It's easy to get free graphic art. Aside from the places that give it away, you can steal images, including buttons and bullets, off other web sites. Just right-click on the image and select "Save Picture As" or whatever the appropriate option is with your browser.

But you may not want your own graphic images showing up on other people's web sites. There are JavaScript scripts for disabling the right click function, but images can be stolen without using that function. Furthermore, all JavaScript protection is disabled by simply turning off the browser's JavaScript long enough to steal the images.

An image at a web site is viewed by downloading it into the viewer's web browser. If it can be viewed, it can be stolen. You can't get around that.

You can, however, fool image thieves into downloading a decoy image instead of the image you want to protect. If you give it a name that fits your image, the thief won't realize that he didn't get your image until he looks at the images in his folder, which could be days or weeks later. Here's how to do it.

Make a table for your image, setting the height and width equal to that of the image. Look at the footer table of Web Page Template 1 to see an example of the html tags for such a table. In that example, I have the image set as the background image. In the same way, make the image you want to protect be the background image in your table. Then use a transparent gif as the content image for the table. Look at the header table of Page Template 1 to see an example of how the image is put in the table as content. When you do it, however, omit the height and width specs from the image tag. Give the decoy gif (transparent.gif in the image file that came with this ebook) a name that would be appropriate for the protected image.

Anyone who looks at the source code can immediately see what you did and can get the name of the real image and download it. To prevent that, you can get the background image name out of the table tag by moving it to your style sheet. Move the height and width attributes into the style sheet as well to hide the fact that the table height and width are not determined by its content.

As an example, let's call the style tt and say the width and height of the image are 150 x 200 pixels. Here's the style:

```
.tt {  
  background-image:url(images/image_name.jpg);  
  width: 150px;  
  height: 200px;  
}
```

Then in your table tag, add `class="tt"`.

Someone who catches on to the trick can still right-click and download the background image. You can't do anything about that. (A no-right-click JS script tells visitors that you are taking protective measures and might make them watch out for the decoy trick.)

On most web servers, anyone can get an index of your image files simply by calling the folder like this: `http://yourdomain.com/images`. To prevent that, put an index file in such folders. Make it one of those `index.php` files described in the previous section and send snoopers to a sales page.

Keeping Your Download Pages Out of Search Engines

Protecting your download page from unauthorized access requires a specialized solution that depends on exactly how you deliver your product. There is, however, one download page security leak that has a very simple solution.

An amazing number of Internet marketers allow anyone to be able to find their paid product download pages simply by entering pertinent information in a search engine. To keep your download pages out of search engine indexes, put this line in your page head:

```
<meta name="robots" content="noindex,nofollow">
```

That tells the search engine robots, "Don't index this page, and don't follow the links on this page to other pages."

Email Address Protection

I think it's reassuring to the potential customer to have a customer service link on the sales page. Do not have it link to a web page, because that will distract him from your message. Instead have it bring up his email program for sending an email to your customer service department.

However, if you have your email address in the source code, web roaming robots will seek it out and grab a copy for sale to spammers. There are various encryption schemes for hiding one's email address from spambots, but my

favorite technique is to have the email address in a separate js file and to call the js file at the spot where I want the email link on the web page. Included is a cs.js file that displays a customer service link. You can use it with your own email address in. Notice that it also puts the subject line on the email.

Insert `<script language=JavaScript src="cs.js"></script>` in the source code at the spot where you want the customer service link to appear on the page. See the example in Page Template 1.

Display Today's Date

You can use a js script to insert today's date on your web page. The whole script can be put in the source code, but I prefer the neater method of having it in a separate js file that is called at the spot where I want the date to appear. There's an example of it on Page Template 1.

The file that displays the date is named date.js. At the spot where you want it to appear insert `<script language=JavaScript src="date.js"></script>` in the source code.

A Self-Updating Copyright Notice

A copyright notice can help make your page look more professional, but you don't want one that gives the impression you're promoting an out-of-date product. So you want it to display the current year. There's an easy way to make it do that automatically and save you the effort of editing it once a year.

Simply put a little JS script in the source code of your copyright notice like this:

```
&copy; Copyright <script language=JavaScript>var d=new Date()  
document.write(d.getFullYear())</script> Your Name
```

"©" is the HTML code for the copyright symbol. Having the word "Copyright" after it is optional. Line breaks don't count in HTML, but they do count in JavaScript. The line break after "Date()" is required.

Give Your Visitors a Favicon Bookmark

A favicon is a little picture that goes with a listing on the prospect's favorites or bookmark list. Your favicon also is displayed in the address bar of Internet Explorer 7 when someone views your web site with it.

If you have a web page that you want visitors to bookmark, it makes sense to get all the advantage you can by giving them a bookmark that stands out. To make that happen, you first need a favicon file. The file type is "ico." You can find free ones on the web, but you are encouraged to make your own

because the point is to have a distinctive image that represents your company in the same way that logos do.

There are web sites where you can upload an image and they'll turn it into a favicon. An example is on the Free Webmaster Tools page. Also on that page is a link to a favicon creator that you can download and use on your computer. (Windows only.) It turns any jpg or gif image into an ico image of the proper size, which is 16 x16 pixels.

Once you have your image, name the file favicon.ico and upload it to your image folder, or wherever you want to put it. Then put this tag in the head of your web page: `<link rel="shortcut icon" href="images/favicon.ico">` (If you are using XHTML, close the tag with `</>`)

Then people automatically get your favicon with your page listing in their bookmarks or favorites list.

The Exit Popup Trick

A Decoy Close Button

If someone leaves your sales page without buying, it's worthwhile to open a page that displays another offer—or perhaps the same product at a lower price or with an additional bonus. You can do that with an exit popup, except for the fact that most people have a popup blocker operating, which blocks such popups or popunders from showing. One way to overcome that is to use some advanced JavaScript that defeats popup blockers. There is a program that can generate the code needed to send visitors to any page you specify when they close your page without buying your product. It is not quite unblockable, but close to it. [Click Here](#) for details.

If you do not have such a popup generator, you can use the decoy close button trick to catch at least some of the visitors who leave the page without buying. Simply put a button at the bottom of your page that looks like a close button but actually is a link to another sales page. It cannot be blocked by a popup blocker.

Insert this in the source code after the page text and before the `</td>` tag.

```
<p>&nbsp;</p> This makes a blank line. Use it if needed for proper spacing.  
<form><center>  
<input type=button class="smalltext" value="Close" onClick="ClosePage()">  
</center></form>
```

The value “Close” is what appears on the button. You could make it “Close Page” or “Close Window” instead. The class=“smalltext” phrase formats the button text according to a style in the stylesheet.

The “ClosePage()” part is a function. You have to define the function and

include your desired destination URL in it. Here's how to define the function in the head part of the page.

```
<script language="JavaScript">
<!--
function ClosePage() { window.location = "http://destination_url.com"; }
//-->
</script>
```

The `<!--` in the second line and the `//-->` in the fourth line serve to hide the script from old browsers that would otherwise display the script on the page. Since it is extremely unlikely that anyone still uses a browser that old, I omit it from the other examples.

To hide this script's secret from people who look at the source code, put the function in an external js file instead. If you already have a js file of functions linked to the page, (for example, the `rollover.js` file) you can add the following function to the existing js file either above or below the other defined functions.

```
function ClosePage() { window.location = "http://destination_url.com"; }
```

If you don't have a js file to add this function to, make a new js file for it using a plain text editor such as Notepad. Put that line in the file and save the file as "close.js" (or whatever name you like) in the same folder as your web page. Then add this line to the page head to link the page to the js file. The part shown in green should be the name you gave to your js file:

```
<script language=JavaScript src="close.js"></script>
```

Side Note: js files and css files do not have to be in the same folder as the page that uses them provided their links from the page specify where they are.

An Easy Alert Message

You can make all kinds of fancy popups with popup makers, but here's an easier way to convey a simple message to the viewer. You can make a simple dialog box containing a message pop up when your page opens. The page opens and the box floats in front of it. The viewer has to click on the OK button before he can scroll the web page.

Put this code in the head of your page or add the function part of it to a file of javascript functions (such as `rollover.js`) that is linked to the page.

```
<script language="JavaScript">
function loadalert ()
{alert("Message goes here.")}
</script>
```

Next, you need to call that function when the page opens by putting code in the body tag like this:

```
<body onLoad="loadalert()">
```

You can announce such things as when the offer will expire, when the price will change, the fact that an extra bonus has been added, etc.

Here's an example of how you can take advantage of the non-commercial look of the dialog box:

Attention: This is a brand new offer, and I apologize for not having the web page completely finished yet. To make up for it, I'm temporarily letting early visitors have the product at half price.

When your page is "completely finished", it just might promote an entirely different offer.

Easy PayPal Buttons and Links

You apply PayPal payment processing to a single-product sales page simply by inserting a PayPal Buy-Now button on the page. (You need a PayPal merchant account first. It's free. [Click Here](#) for details.) PayPal has a page in their "Merchant Services" section for making such a button using a PayPal button image and encrypting the code so people can't look in your source code and see the URL for the fulfillment page.

Details: Click on the Merchant Services tab after logging in. Under "Create Buttons" click on "Buy Now". After completing the first page, select "Add More Options" and specify "No Shipping Needed" on that page.

To get a link URL along with an unencrypted button, select "No" for Button Encryption. The URL you get is huge. To use it, you need to apply link cloaking to make the link short and to hide the actual code. Then you can use the cloaked link in emails and with a button image at your web site. You then can apply a rollover effect.

Instead of getting the "Buy Now" link code from PayPal, you can use my easy payment script. There is one complication. PayPal doesn't get along with things like periods and slashes in the link code. You need to substitute a 3-character code starting with "%" for each of the offending characters.

The payment link file, which came with this ebook, is named PayPalLink.php. To use it, just make a copy named order.php or whatever you like. Then edit order.php with a simple text editor such as NotePad. Only the top part needs to be edited. Fill in your own information using the character substitutions shown at the top of the file. On the next page is a copy of that top part with colors that aid in understanding it.

substitution codes: space %20 @ %40 - %2d . %2e / %2f : %3a

```
$merchant_email = "name%40domain%2ecom"; // Your PayPal email address.  
$product_name = "name%20morename%20morename";  
$product_code = "code"; // Use " " if you don't have a product code.  
$price = "27%2e00"; // It is set up for US dollars.  
$fulfillment_url = "http%3a%2f%2fdomain%2ecom%2fsubdir%2fpage%2ehtml";  
$cancel_url = "http%3a%2f%2fdomain%2ecom%2fpage%2ehtml";
```

The fulfillment URL is where the customer lands when he clicks on “Return to Merchant” after completing his payment. The cancel URL is where the customer lands if he cancels before completing his payment at PayPal. You can make a special page for that or just have the customer return to the sales page. The supplied script is for a product that does not require shipping.

Upload the file to your web site and have your buy button or buy link go to that file. Visitors cannot view or download php files, so your PayPal information is safe.

See how I used such a link named “order.php” with a button rollover image on PageTemplate 1.

The page templates contain additional tips and techniques.

For Resale Rights, see Page 2.