



Fast Forward Graphic Design Guide

Table of Contents

Introduction	2
General Design Guidelines	9
Main Site Template.....	11
Advanced Features in the Template	19
Printer-Friendly Version	26
Pop-Up Template	29
Secure Server Template.....	31
CourseBot template.....	34
Multiple Templates	40
Appendix A – Sample Text.....	43
Appendix B – Checklist	45



Introduction

This development guide gives you step-by-step procedures for creating the graphic design templates for a Fast Forward Web site.

Overview of Our System

A typical Web site has a number of pages, all of which have a similar layout (style) but each with different information (content). The idea behind our system is that we separate the style from the content.

This is how it works:

- You design the overall style of the Web site in template files. These templates include the header, footer, menu bar, copyright notice, style sheet, common graphics, and other stuff that's common to all pages.
- The client creates the content for each page. They do this in a simplified format called "Smart Text". It's not necessary to understand how this works; just realise that it's a way of generating HTML automatically.
- A script takes the template file and each of the content pages, and combines them to create the true HTML files that users see.
- The client has a password to go into their site and edit the content of any page. The script then rebuilds the HTML file automatically.

If you have used other content management systems in the past, you will be familiar with this concept.



IMPORTANT NOTE

Note that you are *not* creating the individual pages for the Web site. Rather, you are creating a *template* that shows the overall look of the site. The client will typically add the Web pages themselves.

Overall Design Guidelines

Most of our clients have information-rich Web sites, so they require a clean, attractive and **simple** Web site that matches their existing branding.

A typical site will have these characteristics:

- Loads fast
- Verdana 10 point font
- Black text on a white background
- Full contact information at the bottom of every page
- Pages print correctly
- Works for both Firefox and IE
- Works on at least 800-wide screen (maybe even 640-wide if that's required for printing correctly), and expands for users with wider screens
- External style sheet to control the styles
- No Flash!

For examples of our other clients sites, see www.firststep.com.au/clients.html.



Initial Information Required

You should have the following information provided to you before you begin:

- The Web site URL (e.g. example.com). Note that the "www" is **not** required, and should be omitted for brevity
- The client's full contact information for use on the Web site
- The client's "E-Service user name"
- The E-Service server - usually fsplugins.net, but might be eservice.com.au or fsplugins.com
- List of buttons, and the HTML page names that each button leads to
- Any secondary text areas for the template (see page 19)
- Google Analytics code for inserting into the template

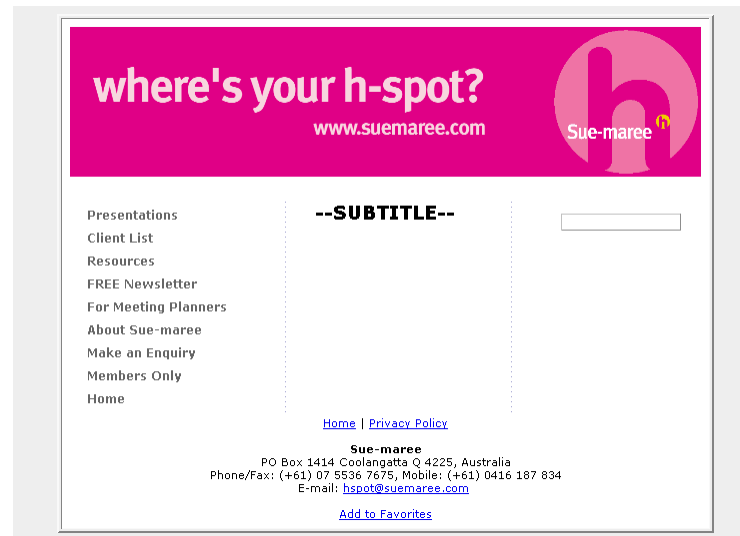
If you are missing any of this information, please ask for it before starting the design.

Files Required

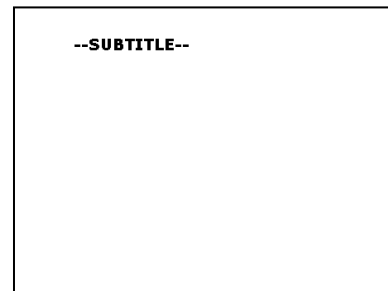
Even though the description above talks about just one template file, in fact you will be creating five separate HTML files:

- The main site template `template_site.html`, which will be used as the basis of most of the pages on the Web site. Here is an example:

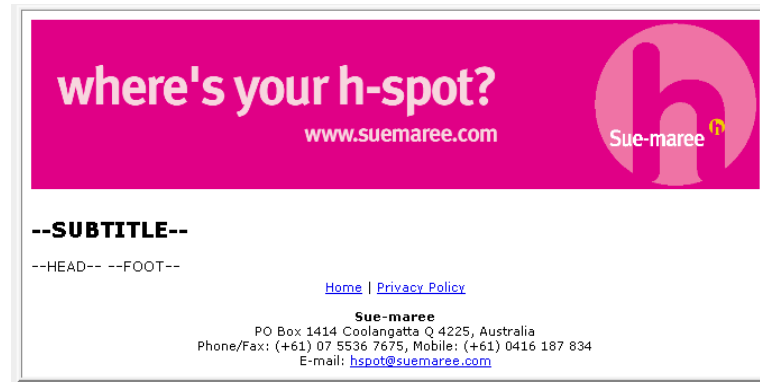




- A small template `template_popup.html`, which will be used as the template for any pop-up windows on the site. It will usually be very plain, and might not have any colours, logo or pictures. Here is an example:



- The secure template `skeleton_secure.html`, which is used for secure order forms. This typically looks like the main template, except without the distractions of a navigation menu and testimonials. Here's an example:



- The CourseBot template `skeleton_coursebot_html.html`, which is used for the e-mail newsletter and on-line courses. It will look very similar to the main template, except that it will be narrower, won't have buttons on the left, won't have testimonials on the right, and will have a Manage Subscription link at the bottom. Here's an example:





Testing

After creating a template, you can test it by inserting sample HTML code into it and inspecting it. See Appendix A for a sample block of text.

Before delivering the final files, complete the checklist in Appendix B.

File Structure

Please deliver the final files to us as a single ZIP file, with the following file structure:

template_site.html	Main site template
styles.css	Style sheet for this template



template_popup.html	Pop-up template
styles-popup.css	(Optional) Style sheet for this template
skeleton_coursebot_html.html	CourseBot template
styles-coursebot.css	(Optional) Style sheet for this template
images/	Image files used by these templates
secure/	Sub-folder for secure server files
skeleton_secure.html	Secure server template
styles.css	Style sheet for secure server template
images/	Image files used by this template

Outline

This guide has the following sections:

- **General Design Guidelines** - things that we have found to be useful to ensure a high-quality design that is compatible across a variety of browsers
- **Main Site Template** - instructions for creating the main template file
- **Design Checklist** - checklist of items to check in your design
- **Pop-Up Template** - instructions for creating the pop-up template
- **Secure Template** - instructions for creating the secure server template
- **CourseBot template** - instructions for creating the CourseBot template
- **Multiple Templates** - special situations where you might create more than one design for a site



General Design Guidelines

We have listed some guidelines here that we think will create a design that is compatible across a variety of browsers and operating systems.

These are only guidelines, and we leave the final design up to you, in consultation with your client.

Corporate colours

If the client already has corporate colours in the rest of their marketing material, use it for your Web site as well.

Light background

It's best to use a plain light-coloured background (white is best) and dark text. If you want a textured background, use a very light texture so it doesn't interfere with the main text. You'll notice that almost all of the best-known Web sites - such as search engines and newspapers - use black text on a plain white background.

No splash page

A splash page is a graphical "front" page for your Web site, usually displaying a large graphic and a "Click here to enter" link. Although splash pages can look very attractive, they are generally not a good idea because they take a long time to download and waste your site visitor's time.



Text links

In addition to any graphical navigation buttons, we recommend that the design has text links to match the navigation buttons. These links usually appear in the footer of the page, and serve two purposes:

- They are convenient to use when the user has scrolled to the bottom of the page; and
- They appear before any graphics have loaded, and in fact appear even if the user has turned off graphics.

No frames

We recommend that you do not use frames, because of the problems they cause with linking and search engine listings.

No Flash

We recommend that you do not use Flash in the design, because it is not available on all browsers.



Main Site Template

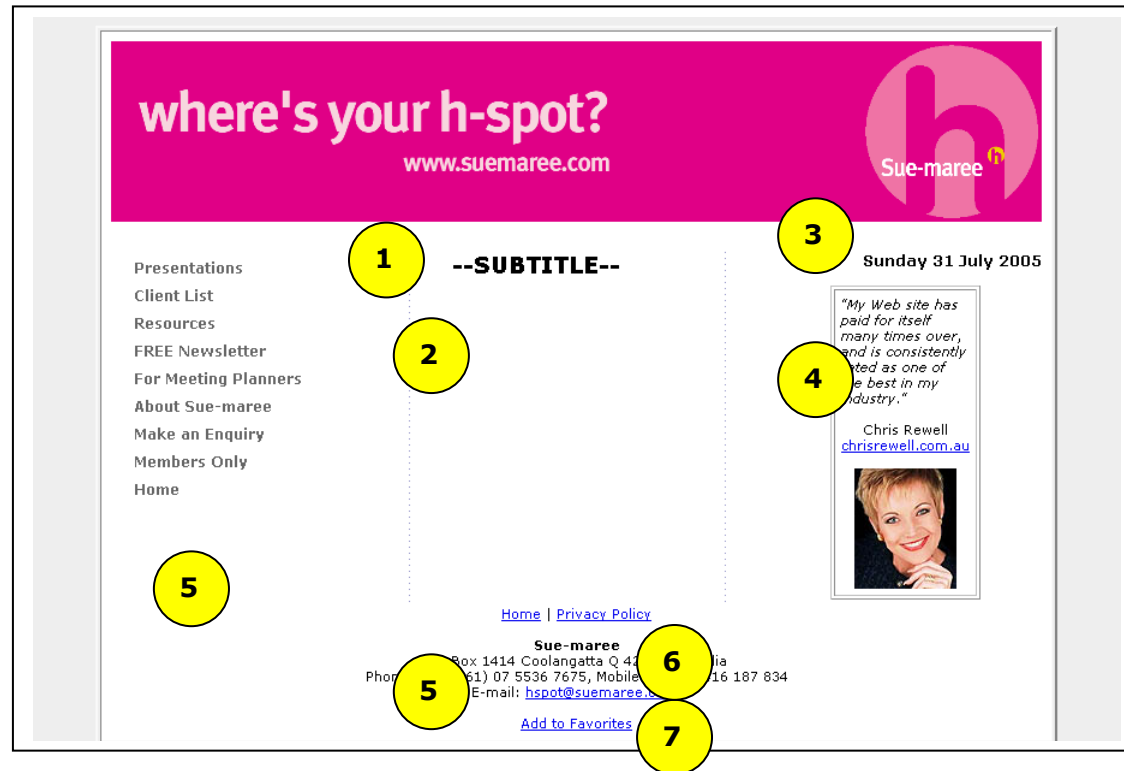
Essentially, your job is to create a standard HTML Web page with all the required colours, graphics, and other design elements; but without any body text - and a few special lines for the content management system.

You will set up these things in the template (see the diagram on the next page):

1. **Page heading:** The text --SUBTITLE-- will be replaced with the page heading.
2. **Page body:** The large blank space below this will be replaced with the body text on the page. You will insert some HTML code here (which doesn't show up in the template above because it's an HTML comment).
3. **Date:** You will insert some JavaScript to display today's date.
4. **Testimonial:** You will insert some JavaScript to display a testimonial at random.
5. **Secondary area:** In some cases, the client might like a secondary area on the page where they can insert some other text. For example, they might choose to insert a special promotion on every page. This is optional.
6. **E-mail address:** When you insert the e-mail address, you will encode it in a special way so that spam robots can't detect it.
7. **Add to Favorites:** You will insert some JavaScript for this link.

The next page shows each of these things visually.





For full details on all of these things, and more, keep reading.

File Structure

The file must be named `template_site.html`.



Put any images you use in an `images` sub-folder.

If you would like a sample file to start from, you can download `template_site.html` from here:

<http://fslib.info/docs/ff/samples.html>

HTML Coding

Most of the template is just normal HTML, which you can create using whatever Web design tool you normally use.

Then make the changes below.

Note: Please be careful when inserting these lines to make sure that you do them exactly as described here. Some Web design programs mangle up lines after you insert them (e.g. by indenting them or splitting them or joining them). Please check carefully that you do end up with the lines exactly as described here.

The very first line should be:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
```

Then these are the special things to add in the HEAD part of the document (i.e. between `<head>` and `</head>`):

- In the `<title>` tag, include the text `--SUBTITLE--`, like this:

```
<title>Your Site Name: --SUBTITLE--</title>
```

The `--SUBTITLE--` will be replaced by the individual page title.

- Insert a `<base>` tag with your site's URL in it, like this:



```
<base href="http://example.com/">
```

Note the trailing slash, which is important!

- Add these two lines for META tags:

```
<meta name="keywords" content="">
<meta name="description" content="">
```

- Add this line, which supports pop-up windows:

```
<!-- POPUP -->
```

- Add this block of code, which supports the Add to Favorites link, replacing "example.com" with the site's domain name (e.g. microsoft.com) and replacing "SITENAME" with the name of the Web site (e.g. Microsoft Incorporated):

```
<script type="text/JavaScript">
<!--
var bookmarkurl="http://example.com"
var bookmarktitle="SITENAME: --SUBTITLE--"
function addbookmark(){
if (document.all)
window.external.AddFavorite(bookmarkurl,bookmarktitle)
}
//-->
</script>
```

- Add this line to support the affiliate program (Note: This is one long line, so if it splits over two lines, join them together):

```
<script language=JavaScript src="http://fsplugins.net/fsc_affiliates.js"></script>
```

Then in the BODY part of the document (i.e. between <body> and </body>), do the following:



- The `<body>` tag looks like this (change the `#ffffff` background colour if required) (Note: This is one long line, so if it splits over two lines, join them together):

```
<body bgcolor="#ffffff" onLoad="fscEpopup();setAffiliate();" onUnload="fscXpopup()">
```

Note: If you already have other instructions in the `onLoad` or `onUnload` attributes, you must combine them with the instructions above, **not** insert additional `onLoad` or `onUnload` attributes. For example, this is **WRONG**:

```
... onLoad="fscEpopup();setAffiliate();" onLoad="blahBlah();"
```

This is correct:

```
... onLoad="fscEpopup();setAffiliate();blahBlah();"
```

- At the place where you would like the page heading to appear, insert this line:

```
<h1>--SUBTITLE--</h1>
```

Use your style sheet to format the `<h1>` tag appropriately.

- Leave out all the main text on the page. This will get replaced by the page content entered by the user. In its place, insert this block of HTML:

```
<!--ADMIN-->  
<!-- BODY -->  
<br clear=all>  
<!--ADMIN-->
```

- Every page on the site will display the current date. At the place where you would like this to appear, insert this line (Note: This is one long line, so if it splits over two lines, join them together):

```
<script language=JavaScript src="http://fsplugins.net/fsc_today.js">  
</script>
```



- Every page on the site will display a random testimonial. At the place where you would like this to appear, insert this lines, replacing "blah" with the client's E-Service user name (Note: This is one long line, so if it splits over two lines, join them together):

```
<script language=JavaScript  
  src="http://fsplugins.net/clients/blah/docs/testimonials.js">  
</script>
```

- For all links on the page, add the text `onclick="xpop=0"` into the `<a>` tag, like this:

```
<a href="index.html" onclick="xpop=0">Home page</a>
```

- At any place where an e-mail address is to be displayed (usually in the footer), "spam protect" it by inserting the line like this (Note: This is one long line, so if it splits over more than one line, join them together):

```
<script language=javascript>var em1="gihan";var em2="GihanPerera.com";document.write("<a  
href="+mail+"to:"+em1+"@"+em2+">"+em1+"@"+em2+"</a>")</script>
```

(In this example, the e-mail address is `gihan@GihanPerera.com`. Of course, you would replace this with the client's e-mail address)

- Add an "Add To Favorites" link (usually in the footer) by inserting this text:

```
<a href="javascript:addbookmark()">Add to Favorites</a>
```

- Add an "Add To Favorites" link (usually in the footer) by inserting this text:

```
<a href="javascript:addbookmark()">Add to Favorites</a>
```

Style Sheet

The use of a style sheet is very important.



As mentioned earlier, the user writes their page content using a simplified form of HTML called "Smart Text", which is then converted automatically into real HTML. The resulting HTML tags are standard HTML - without any formatting embedded in them - so it's up to you to do the formatting in the style sheet.

See Appendix A for a block of sample text that's generated by the system. You can see that it generates fairly simple HTML tags, like `<p>` and ``. It's up to you to define styles for these tags in your style sheet.

You can download a sample style sheet `styles.css` from here:

<http://fslib.info/docs/ff/samples.html>

Store your style sheet in an external file named `styles.css`, in the same folder as the HTML file. Link to it from the template with a line like this in the HEAD of the document:

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

Links

If you have been provided with a list of HTML file names for linking, please use them exactly as provided.

If you haven't been given a set of links, create them yourself, following these guidelines:

- File names end in `".html"`, not `".htm"`.
- Use lower case only.
- If a link has two words (e.g. "Coming Events"), use a hyphen between the words (e.g. `coming-events.html`).



Standard Links

Include these as standard links:

- Home – links to `index.html`
- Privacy Policy – links to `privacy.html`
- Newsletter – links to `newsletter.html`
- Search This Site – links to `search.html`
- Members Only – links to `members-only.html`

You might choose to change the link text in consultation with your client (for example, “Members Only” is sometimes called “Clients Only”). But make sure that you keep the HTML names correct.



Advanced Features in the Template

The previous section described the essential components of the site template. In this section, we'll look at some additional things you can include, which are optional but sometimes useful.

Secondary Text Areas on the Page

As mentioned above, you insert `<!-- BODY -->` to indicate the main body area of the page. In addition, sometimes the client will want other areas of the template that they can customise. For example, they might have a right-hand area of the page where they insert links to special offers, news, etc.

We can do this in the template, but there are two possible scenarios, *and it's very important* to know which one the client wants.

- One scenario is where the client will insert *different* text in this area, for each page of their site. For example, they might have written 10 books, and they want to promote a book on each page of the site, but they will choose which book they want on each page.
- The other scenario is where the client wants the *same* text in this area, for all pages of the site. In other words, they want to make one change, and then that change happens on every page of the site. For example, they might have a different special offer each month, and at the start of the month they want that month's offer to appear on all pages.

Let's look at each of these in turn.



Different text on each page

Insert this into the template, in the place where you want the new text to appear:

```
<!-- BODY 1 -->
```

As with the main `<!-- BODY -->` code, this line must appear all by itself, without any other HTML code on the same line.

Often this part of the template will have its own style - e.g. different font size, in a box, etc. You can use any HTML you like around this line, as usual - e.g.

```
<div style="special">  
<!-- BODY 1 →  
</div>
```

In fact, you can have more areas using `<!-- BODY 2 -->`, `<!-- BODY 3 -->`, and so on (up to `<!-- BODY 9 -->`), but that is rare. They can appear in anywhere in the template, and in any order.

Same text on each page

This is done using an entirely different method, using the Global Replace module from our plug-ins pack. The Global Replace module simply gives you a bit of JavaScript code that displays whatever text the client has provided elsewhere. By inserting it into the template, you allow that text to appear on all pages of the site.

Insert the code like this:

```
<script type="text/javascript">var idi="blah"</script><script language="JavaScript1.1"  
src="http://fsplugins.net/clients/xxx/docs/replace.js"></script>
```

In this code, replace "XXX" with the client's E-Service user name.



You'll also need to replace "blah" with something that identifies this Global Replace block. If this hasn't been provided to you yet, please ask for it.

Often this part of the template will have its own style - e.g. different font size, in a box, etc. You can use any HTML you like around this line.

Newsletter Sign-Up Form

If you would like to put a newsletter sign-up form on the template, copy and paste this code, changing the "XXX" near the end to the client's E-Service user name:

```
<form action="http://fsplugins.net/cgi-bin/allegro.pl" method=get
onSubmit="this.Referer.value = document.referrer; return true">
<input type=hidden name="Referrer">
<table border=0 cellpadding=0 align=center>
<tr><th>First Name:</th></tr>
<tr><td align=center><input type=text name="f_first_name" size=15></td></tr>
<tr><th>Last Name:</th></tr>
<tr><td align=center><input type=text name="f_last_name" size=15></td></tr>
<tr><th>E-Mail Address:</th></tr>
<tr><td align=center><input type=text name="f_email" size=15></td></tr>
<tr><td colspan=2 align=center><input type=submit name=search value="Sign Up">
</td></tr>
</table>
<input type=hidden name=action value=search>
<input type=hidden name="db" value="ac_course_enrol">
<input type=hidden name=context value=enrol>
<input type=hidden name="f_code" value="NEWSLETTER">
<input type=hidden name=account value="xxx">
</form>
```

You can re-format this form any way you like, as long as you keep all the fields (including the hidden fields) intact.



Password Form

If you would like to put a password form on the template, copy and paste this code, changing the "XXX" in the second line to the client's E-Service user name:

```
<form action="http://fsplugins.net/cgi-bin/goto.pl" method=post>
<input type=hidden name=account value="xxx">
<input type=password name="f_key" size=10>
<input type=submit name=go value="Go">
<input type=hidden name=fields value="f_key">
<input type=hidden name="required_fields" value="">
</form>
```

Google Analytics

Insert the Google Analytics code in the usual place, as required by Google (usually at the very bottom of the page, just before the closing </body> tag).

Drop-Down Menus

If you want to add drop-down menus to your navigation structure, download and read this document:

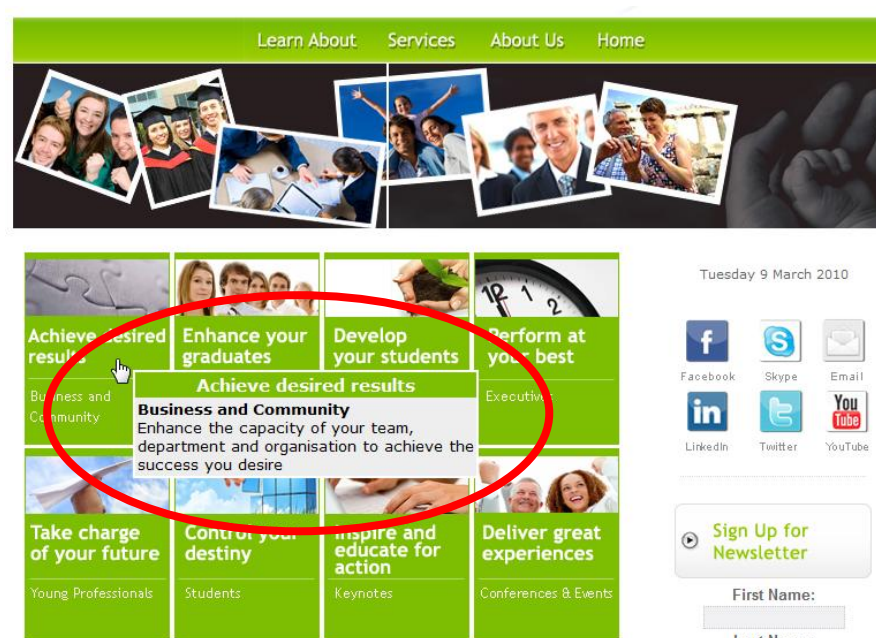
<http://fslib.info/docs/ff/FastForwardGraphicDesignSubmenus.pdf>

It gives you step-by-step instructions on how to do this, in a way that's compatible across different Web browsers.



Pop-Up Help Text

You might want to include a facility to show “pop-up text” when the user hovers their mouse over a link or picture – e.g.



To do this, you need to enhance the template to allow for some special JavaScript code that displays these pop-up mini-windows.

The first step is to copy this code and paste it into the <head> section of your template:

```
<script type="text/javascript">
```



```
        var GB_ROOT_DIR = "./linker/";
    </script>
    <script type="text/javascript" src="http://conferenceonthenet.com/fsc-
    linker/AJS.js"></script>
    <script type="text/javascript" src="http://conferenceonthenet.com/fsc-
    linker/AJS_fx.js"></script>
    <script type="text/javascript" src="http://conferenceonthenet.com/fsc-
    linker/gb_scripts.js"></script>
    <link href="http://conferenceonthenet.com/fsc-linker/gb_styles.css" rel="stylesheet"
    type="text/css" />
    <link rel="stylesheet" type="text/css" href="infobox.css">
    <SCRIPT language="JavaScript" type="text/javascript"
    src="http://conferenceonthenet.com/fsc-linker-infobox.js"></script>
```

(Don't worry about the fact that it contains all these "conferenceonthenet" references – that is simply where the scripts are stored)

You then need to create a file `infobox.css` and place it in the same folder as the template file. This is a style sheet that defines the look and feel of the pop-up boxes. Start by copying this and changing the settings:

```
.boxs {
border: 2px solid #ffffff;
width: 300px;
}
.boxheader {
background-color: #7DBE03;
color: #ffffff;
FONT-WEIGHT: bold;
FONT-SIZE: 11pt;
FONT-FAMILY: verdana;
}
.boxcontent {
background-color: #e0e0e0;
color: #000000;
FONT-WEIGHT: normal;
```



```
FONT-SIZE: 10pt;  
FONT-FAMILY: verdana;  
}
```

That's it as far as the template goes.

Later, when somebody is creating a page of content, they can create one of these pop-up text windows by inserting something like this in their content:

```
<a href="blah.html" onMouseOver="infobox('Title','Subtitle','Description');" onmouseover="exit();">click here</a>
```

```
<a href="blah.html" onMouseOver="infobox('Title','Subtitle','Description');" onmouseover="exit();"></a>
```

```

```

The first example shows a plain text link with pop-up text, the second shows an image link with pop-up text, and the third shows a plain image (i.e. not clickable) with pop-up text.

In all three cases, "Title" is the main title in the pop-up window, "Subtitle" is the sub-title, and "Description" is the descriptive text that appears in the window. Of course, you also have to replace "blah.html" with the appropriate URL for the hyperlinks; and replace "blahblah.jpg" with the URL of the graphics.

Note: The title, sub-title and description must not include double quotation marks (because they confuse the JavaScript coding). And if they have single quotation marks (apostrophes), you have to insert a backslash in front of each occurrence, like this:

```
John\'s Web site - see what he\'s got to offer!
```



Printer-Friendly Version

Most of our clients are information experts, so their Web sites are content-rich rather than purely visual. It's essential that site visitors can print individual pages and still see the content.

Broadly, the printed page should show the essential content of the page, remove the non-essential content, and still provide enough branding to identify the source.

With a CSS style sheet, you can specify that certain elements of your design apply only to the screen version, others apply only to the print version, and the rest apply to both. This means you can incorporate all the printer-friendly factors into your design without sacrificing the visual appeal of the on-screen page.

This section describes the way you create a printer-friendly version of an HTML page. **This applies to the main site template (previous section), the CourseBot template and the secure template (coming up soon).**

Content Requirements

From a content viewpoint, the goal for that version is to show the essential content on the page without any unnecessary information. Typically, this should be **included**:

- A banner, logo or other branding at the top
- The main body content
- Full contact information at the bottom of the page

and this should be **removed**:



- Navigation links – e.g. graphical menus, text menus
- Advertising or other promotion (banners, icons or text).
- Anything else that only makes sense on a “live” Web page, not on a printout (e.g. a newsletter sign-up form, a search box, a password entry box).

Technical Requirements

From a technical viewpoint, keep these things in mind:

- The design should fit a 640-pixel-wide page. Even though most screens are now wider, many printers can only accommodate this narrower size.
- Use black text on a plain white background.
- Use an appropriate font size.

Using the Style Sheet

If you’re already familiar with creating printer-friendly designs, skip these instructions – just go ahead and do it your way, meeting the requirements above. However, if you’re new to this concept, or you’d like some assistance in making it work, keep reading ...

If you look at the sample style sheet we provide (see Style Sheet section earlier), you will see this block of code:

```
.pwidth {width: 777px}
@media print {
    .noprnt { display: none; }
    .pwidth {width: 640px}
}
```



This defines three important things for you:

- Something that lets you set the on-screen page width to 777 pixels.
- Something that will switch this to 640 pixels automatically in the printed version.
- Something that allows you to “hide” certain things in the printed version.

Overall page width

For example, you might use a table around your entire page design to set the initial page size. Instead of specifying the width of the table in pixels, like this:

```
<table border=0 align=center width=777>
```

you use the style sheet definition like this:

```
<table border=0 align=center class=pwidth>
```

Then that table will be 777 pixels wide on screen, and 640 in print.

Hiding page elements

For example, suppose you have a Home link at the bottom of the page, like this:

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

This is useful for the screen version, but unnecessary for the print version. To hide it in the print version, use “class=noprint”, like this:

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

You can use this to hide entire blocks of HTML, such as table columns.



Pop-Up Template

The next step is to create the pop-up window template, which will be used for creating any pop-up pages on the Web site.

This must be named `template_popup.html`.

This is significantly simpler than the main site template. In fact, here is an entire sample, which you can use as your starting point:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head>
<title>Example Site Name: --SUBTITLE--</title>
<base href="http://example.com/">
<link rel="stylesheet" href="styles.css" type="text/css">
</head>
<body onLoad="resizeTo('600', '450')" bgcolor="#ffffff">
<center><table border=0 cellpadding=10 width="90%" align=center>
<tr valign=top><td>
<h1>--SUBTITLE--</h1>
<!--ADMIN-->
<!-- BODY -->
<br clear=all>
<!--ADMIN-->
</td></tr></table></center>
</body>
</html>
```

These are the things to change in this template (each highlighted in **bold** above):

- Replace "Example Site Name" with the name of the Web site (e.g. First Step Communications)



- Replace example.com with the site's domain name.
- If the pop-up template is using a different style sheet than the main site template, name it `styles-popup.css` and replace the name in the `<link>` tag above.
- Replace the pop-up window dimensions (600 and 450) with the dimensions that you would like to set. If you do decide to change these dimensions, keep in mind that they have to fit on smaller windows as well.
- Set the background colour ("`#ffffff`" in the example above).
- The next two lines put the body text in a table that is 90% of the page width, which gives a small left and right margin. You can adjust this if you like, or remove these lines.
- Similarly, the third-last line is the end of that table. If you removed the two lines earlier, remove this as well.



Secure Server Template

The secure server template is used for the e-commerce payment pages on the Web site.

The look and feel of this template is generally very similar to the main site template, except that this is a cut-down version.

Page Content

Typically, the only significant visual difference is that the secure server template does not have any navigation buttons, except for a Home button. This keeps the customer focused on completing the order, rather than distracting them with other options.

The other important consideration is that the secure server template must have at least a postal address and a fax number at the bottom of the page (We recommend a phone number and e-mail address as well). This ensures that customers can print the order pages and send them manually rather than order on-line. **It's important that you provide this information**, even if it's not on the rest of the site, because the shopping cart software does refer to it during the order process.

File Structure

The secure server files do actually reside on a different server than the main template files. So all files associated with this template are kept separate from the main template files.

So create a separate sub-folder called secure, and put the following files in there:

- The template file, which must be named `skeleton_secure.html`.



- The style sheet `styles.css`, which might be identical to the main style sheet.
- The image files, in an `images` sub-folder within here.

HTML Coding

The easiest way to create the secure server template is to edit the main site template.

So start with `template_site.html` and save it as `skeleton_secure.html` in the secure folder.

Then make the following changes:

- Remove all extraneous buttons and links from the design, except for the Home button and Privacy link.
- For the remaining links (i.e. Home and Privacy), use full URLs (e.g. `http://example.com/privacy.html` rather than just `privacy.html`).

- Change the `<base>` tag to:

```
<base href="https://fsplugins.net/clients/blah/">
```

where "blah" is the client's E-Service account name. Note the trailing /, which is important.

- Remove this line:

```
<!-- POPUP -->
```
- Remove the block of code for the Add to Favorites link.
- In the `<body>` tag, remove `onLoad="fscEpopup()" onUnload="fscXpopup()"`.
- Remove the line `<!-- BODY -->` and replace it with these two lines:

```
--HEAD--  
--FOOT--
```



- Remove any JavaScript lines that call up JavaScript code from fsplugins.net – e.g.

```
<script language=JavaScript src="http://fsplugins.net/fsc_affiliates.js"></script>
```

There might be a number of lines like this in the file, but with something else instead of "fsc_affiliates.js".

- Remove the "Add To Favorites" link.

Images

Copy any graphics that are in use into the `secure/images` sub-folder. These will typically be the same graphics as in the main `images` sub-folder, except that there will be fewer graphics because there are fewer navigation buttons.

Style Sheet

Copy `styles.css` into the `secure` sub-folder. Generally, you won't have to make any changes to the style sheet, though you can do so if you choose.



CourseBot template

The final template you're creating is the CourseBot template, which is used for newsletters and on-line courses on the Web site.

The layout of the template is up to you. Typically, it will look similar to your Web site pages, so that the branding remains consistent. However, it doesn't have to match the layout exactly, and it's quite common for this template file to look different - e.g. no buttons, narrower page width.

File Structure

You must create a file named `skeleton_coursebot_html.html`, which is the default template to be used for newsletters and on-line courses.

Optionally, your client might also ask you to create additional templates for specific courses. In this case, give each template a name that matches its use - e.g.

`skeleton_coursebot_html_loyalty.html`.

These templates can access the main images in the `images` sub-folder, and the main style sheet `styles.css`. If they require any additional images, put them in the same `images` sub-folder; and if they require different styles, either add styles to `styles.css` or create a new style sheet `styles-newsletter.css`.

HTML Coding

The template file is a standard HTML file, just like the other template files you're creating.



In many cases, this is very similar to the main site template, so it might be easiest to copy that template and adapt it. Whether you do it that way or not, follow the guidelines here.

Simplify the HTML

Keep the HTML code as simple as possible. People will be getting this by e-mail, and some e-mail clients can only display HTML in a limited way.

Base tag

Because the template arrives in e-mail, not on a Web page, you have to ensure that the e-mail program knows where to find all the graphics and hyperlinks.

Use the `<base>` tag to specify the base directory for locating files - e.g.

```
<base href="http://example.com/">
```

Note the trailing slash at the end of the address, which is important.

If you're copying the main site template, this `<base>` tag will already be set correctly.

No JavaScript

Make sure there's no JavaScript in the file. Some corporate mail systems block e-mail with JavaScript.

This means that you must remove any JavaScript references in the HEAD part, any pre-loading of images in the `<body>` tag, and any JavaScript in the body.

In particular, if you're adapting the main site template, make sure that you:

- Remove the JavaScript for the current date;



- Remove the Add to Favorites code in the head and body;
- Change the spam-protected e-mail address into a normal mailto: link.
- Remove the random testimonial JavaScript.

Heading

Insert the text COURSE-TITLE at the appropriate place to display the course title. This will be replaced with the name of the newsletter or on-line course.

This line sometimes gets deleted by the software, so make sure that it can be deleted safely without affecting the rest of the page structure. For example, if it appears like this:

```
<td><h1>COURSE-TITLE</h1></td>
```

split that into multiple lines like this:

```
<td>
<h1>COURSE-TITLE</h1>
</td>
```

If you're adapting the main site template, you might simply replace --SUBTITLE-- with COURSE-TITLE. Note that --SUBTITLE-- appears **twice** in the main template: Delete it from the <title> tag and replace it with COURSE-TITLE in the main body area.

Main body

Insert the line --BODY-- **on a line by itself** at the appropriate place for inserting the message text.

If you're adapting the main site template, replace:

```
<!--ADMIN-->
<!-- BODY -->
```



```
<br clear=all>
<!--ADMIN-->
```

with this:

```
--BODY--
```

Unsubscribe links

Insert this line to give the user a link to unsubscribe or switch to the text-only option (This is one long line, so if it splits over more than one line when you copy it, join them together):

```
<a href="http://fsplugins.net/coursebot-manage/--KEY--" target="_blank">Manage
Subscription</a> (Unsubscribe, change e-mail, text only)
```

You can change the text of the link, or add additional text after the link. However, as with the COURSE-TITLE line, this line sometimes gets removed automatically by the software. So make sure that it will still make sense when removed. For example, if you say something like "Click the link below to manage your subscription" before this line, put this text on the same line as the link so that it all gets removed together.

Date

You can also insert the date into the message. Typically, insert a line like this:

```
<p align=right>--DD-- --Month-- --YYYY--</p>
```

You can use different date formats if you choose:

- --YYYY-- 4-digit year
- --YY-- 2-digit year
- --MM-- month number



- --Mon-- 3-letter month abbreviation
- --Month-- full month name
- --DD-- day number

So for example, to display the date as "24/02/04", use "--DD--/--MM--/--YY--". And to display it as "24 February 2004" (as in the example above), use "--DD-- --Month-- --YYYY--".

As before, this line sometimes gets deleted automatically, so make sure that it can be deleted without affecting the rest of the page structure.

Correct link format

Make sure that all images and links are correctly referenced, with quotation marks around the addresses – e.g.

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

```

rather than:

```
<a href=index.html>Home</a>           THESE ARE WRONG!  
<img src=images/blah.gif ...>
```

Some browsers will correctly display the second version, even though it is not strictly valid HTML. However, it's important that you do it correctly, because the newsletter software expects it this way.

"Sheet of paper" design

Many e-mail newsletters look like they are sheets of paper on a slightly different coloured background. If you think this effect would look good, here's how to achieve it:



- In the <body> tag, change the background colour to #eeeeee (or another light background colour).
- Insert these two lines directly below that:

```
<p>&nbsp;</p>  
<p><table border width=600 align=center bgcolor="#cc0000" cellpadding=0  
cellspacing=0><tr><td bgcolor="#ffffff">
```

The "cc0000" is the border colour, which you should change to an appropriate colour for this site.

- At the very end of the page, just before the </body>, add:
</td></tr></table>

Sometimes this won't work. If the template has wide graphics along the top or bottom, it might not fit in a 600-wide table. In that case, you might have to make other adjustments to make this work.



Multiple Templates

For most sites, you will do everything described so far, and that will be enough. However, some clients might ask you to create more templates. Typically, this happens for one of two reasons:

- Some pages of the site will have a different design, so they have to use a different template. For example, the front page might look different from the others.
- A client would like different templates for certain newsletters and on-line courses. For example, they might use a “standard” CourseBot template for most of them, but ask you to design a special template for their regular e-zine.

Our content management system allows you to create as many templates as you like for a Web site.

More than one main template

The main site template, as we’ve seen earlier, is named `template_site.html`. If you create other templates, simply replace “site” in the file name with something else. For example, if you’re creating a special template for the home page, you might name it `template_home.html`. The client **does** see this name when selecting a template for a page, so choose an appropriate name.

Apart from the name, to create the new template, follow the same procedure as for the main template. Everything that you can do with the main template applies equally to any new template.



Images

If the new template uses some of the same images as the main template, they can both refer to the same images in the `images` folder. If it requires its own images, put them in the `images` folder as well.

Additional editable fields

As we noted earlier, some sites have more than one body area on the page, where the client can insert text. As we saw earlier, you denote this in the template by inserting `<!-- BODY 1 -->`, `<!-- BODY 2 -->`, and so on.

Your new template can also have these additional fields. However, give them a consistent meaning across all your templates.

For example, suppose that on the main template, `<!-- BODY 1 -->` is used for, say, a special offer. Then if your new template also has a field for a special offer, use `<!-- BODY 1 -->` for that as well. The field doesn't have to look the same on the page, and it doesn't even have to be there at all. But the point is that `<!-- BODY 1 -->` always refers to the special offer field.

More than one CourseBot template

As with the main templates, you can create more than one CourseBot template, and you just name them differently.

The default CourseBot template is named `skeleton_coursebot_html.html`. Any new templates must be named `skeleton_coursebot_html_blahblah.html`, where you replace "blahblah" with an appropriate name describing that template.



For example, if you create a special template for an e-zine, you might name it `skeleton_coursebot_html_ezine.html`.



Appendix A – Sample Text

Use this sample text to test your templates. Take a copy of a template and paste this code directly **and unchanged** into the area where the body text is to appear. Then view the file in your browser and inspect the formatting of this text. If anything is not correct, adjust your style sheet to make it work.

```
<h2>First Level Heading</h2>
<p>No ray of sunlight is ever lost, but the green which it awakes into
existence needs time to grow; and it is not always granted to the
sower to see the harvest.
<p>
All work that is worth anything is done in faith.
<p>
<h3>Sub-Heading</h3>
We believe in:
<ul>
<li>Life
<li>Liberty
<li>The happiness of pursuit
</ul>
<p>
Don't even <strong>think</strong> about it!
<h4>Lower Level Sub-Heading</h4>
You have three choices:
<ol>
<li>Apples
<li>Oranges
<li>Pears
</ol>
<p>The following text appears in a table:
<table>
<tr><th>Country</th><th>Capital</th></tr>
```



```
<tr><td>Australia</td><td>Canberra</td></tr>
<tr><td>New Zealand</td><td>Wellington</td></tr>
<tr><td>China</td><td>Beijing</td></tr>
</table>
<h3>Filling in a Form</h3>
<p>Fill in this form:
<form>
<p>Name: <input type=text>
<p>Choose from a list:
<select>
<option>First option
<option>Another option
</select>
<p>Type into a box:
<textarea rows=3 cols=50>
</textarea>
<input type=submit value="Click Here">
</form>
```



Appendix B – Checklist

As part of our quality assurance program, we require graphic designers to complete a design checklist on their template files. Please check all of the items on this list, unless you have discussed them with your client and agreed that you are not going to comply with some of them.

Main template

Creating a sample page

Because you're creating a "blank" template, you can't predict how it will work when text is inserted into it. To test this, create a sample page as follows:

- First make a temporary copy of the template and place it in the same place as the main template, with an "images" sub-folder as well.
- Remove, or comment out, the `<base>` tag, so that the browser won't try to retrieve the images from the live Web site
- Replace the `<!-- BODY -->` line with the code from the sample block of HTML code from Appendix A.

Then open the page in your browser and inspect it visually, and go through the checklist below.

Visual inspection

- Appropriate font sizes
- Adequate margins for text on all four sides



- Correct formatting for headings and sub-headings
- Correct formatting for tables
- Correct formatting for form input (test by typing into the form fields)
- Other formatting looks correct

HEAD section

- Correct <base> tag
- Appropriate title in <TITLE> tag
- POPUP line has been added
- META tags added
- Style sheet has been included

Body text

- Correct <body> tag
- Background colour **must** be specified
- Block of comments with SUBTITLE/BODY/ADMIN is correct.
- Today's date appears on the page (from inserted JavaScript code)
- JavaScript code for testimonials has been inserted (the testimonials themselves might not appear yet)
- All links have `onclick="xpop=0"` in the <a> tag

Footer Text

- Full contact information included
- E-mail link is working correctly
- E-mail link has been spam-protected
- Add to Favorites link is working
- Google Analytics code included correctly



Graphics files

- Correct ALT text provided for all files
- Correct WIDTH and HEIGHT for all files
- Optimised for small file size

Navigation buttons

- All have been included, as discussed with client
- The order of the buttons is as specified in the design brief
- Correct spelling of text on buttons
- Buttons link to the correct pages
- If JavaScript mouseovers are used, they work correctly

Text links

- All have been included
- Order of links is same as for navigation buttons
- Correct spelling of text
- Same links as buttons (i.e. they lead to the same pages)
- No trailing spaces in links

Window size

Ensure that your design works within a 800x600 window without any need for horizontal scrolling. The design should expand for wider screens.

- Test in 800x600 window
- Test in 1024x768 window



Browser compatibility

Design for HTML 4.0. Any advanced features must degrade gracefully (i.e. the site still works without them).

- Test in Mozilla
- Test in Internet Explorer
- Run the HTML Validator at validator.w3.org to check for any HTML errors.

Test with JavaScript turned off

If your design uses JavaScript for cosmetic effects (such as mouseovers), the design must still work if JavaScript is disabled or the browser doesn't support JavaScript:

- All links work correctly
- All graphics are displayed (but without mouseovers)

Test with images turned off

Some Internet users browse the Web with images turned off, for speed. Ensure that your design allows for easy navigation in this case.

- Navigation is still easy
- ALT tags pop up correctly
- All links work correctly

Printer Friendly

Print a page and check:

- Body text fits on page (doesn't get cut off right-hand side)
- Branding (e.g. banner/logo) appears correctly
- Contact information appears in footer



- Extraneous information removed (menus, navigation links, ads, etc.)
- Black font on white background
- Appropriate font size

CourseBot Skeleton

Repeat all the tests above, as follows:

- Visual inspection – as above
- HEAD section – as above, but simplified. In particular:
 - No POPUP section
 - No META tags
 - `<title>` tag doesn't include `--SUBTITLE--`
- Body text – as above, but simplified. In particular:
 - No JavaScript in `<body>` tag
 - No `onClick` anywhere
 - No JavaScript anywhere in code
- Footer Text – as above, and:
 - Full contact information included
 - Link to privacy policy
 - Link to Manage Subscription
 - E-mail link is `mailto`, not JavaScript
 - No Add to Favorites link
- Graphics files – as above
- Navigation buttons – as above, but:
 - Most navigation buttons removed



- Home button or link retained
- No JavaScript mouseovers
- Text links – as for navigation buttons
- Window size – fits 640-pixel window (i.e. *narrower* than main template)
- Browser compatibility – use the HTML Validator at validator.w3.org to check for any HTML errors.
- Test with images turned off – as above (*extremely important* because modern versions of Microsoft Outlook turn off images by default)
- Printer Friendly – as above

Secure Skeleton

Repeat all the tests above, as follows:

- Visual inspection – as above
- HEAD section – as above, and:
 - No POPUP line
 - META tags added
- Body text – as above, and:
 - Remove JavaScript for the date
 - Remove JavaScript for testimonials
 - Remove all other external JavaScript calls
 - Remove `onclick="xpop=0"` from links
 - Style sheet is a separate local copy
- Footer Text – as above, and:



- No Add To Favorites link
- Graphics files – as above, and:
 - All image files are in a separate images/ folder
 - Only those images are in that folder
- Navigation buttons – as above, except:
 - All buttons removed, except Home
 - Home button links to the **absolute URL** of the home page
- Text links – as above, except:
 - All links removed, except Home and Privacy Policy
 - Home links to the **absolute URL** of the home page
 - Privacy links to the **absolute URL** of the privacy policy page
- Window size – fixed width to fit within a 800x600 window (i.e. does **not** expand/shrink automatically)
- Browser compatibility – as above
- Test with JavaScript turned off – as above
- Test with images turned off – as above
- Printer Friendly – as above





More Help

- Frequently Asked Questions: <http://first-step-support.info/>
- Help Desk <http://firststep.com.au/help-desk.html>
- Other support options: <http://firststep.com.au/support.html>
- Consulting: <http://gihanperera.com/consulting.html>
- Products: <http://gihanperera.com/products.html>
- Other User Guides: <http://www.firststep.com.au/fast-forward-guides>



Subscribe

Subscribe to the e-mail newsletter *Expert's Gold* at www.firststep.com.au.

© Copyright 2010 First Step Communications

About First Step

Briefly, we build Web sites for speakers.

More fully, at First Step Communications www.firststep.com.au, we help information experts – speakers, trainers, consultants, coaches, authors, service professionals and others like them – to leverage your expertise on the Internet.

Disclaimer

The material contained in this report is general and is not intended as advice on any particular matter. The author expressly disclaims all and any liability to any persons whatsoever in respect of anything done by any such person in reliance, whether in whole or in part, on this report. Please take appropriate legal advice before acting on any information in this report.

