

AutoTheme HTML Theme System

Tutorial: Theme Creation - Part 1

Creating Theme Ready HTML from a Graphic Template

Author: Adam Docherty (a.k.a. Lobos)
<http://www.webvida.com>

Editor: Shawn McKenzie (a.k.a. AbraCadaver)
<http://spidean.mckenzies.net>

Table of Contents

INTRODUCTION	1
PREPARING A GRAPHIC.....	1
SLICING AND EXPORTING	2
PREPARING THE HTML.....	3
THE HEADER COMPONENT	3
THE BODY COMPONENT	4
<i>Block Preparation.....</i>	<i>5</i>
THE FOOTER COMPONENT.....	6
THE HEADER COMPONENT	6
READYING THE HTML	7
<i>logoArea.....</i>	<i>7</i>
<i>headerNav.....</i>	<i>7</i>
<i>bannerArea</i>	<i>7</i>
<i>Let's make it 100% width!</i>	<i>8</i>
BLOCK COMPONENTS	9
READYING THE HTML	9
THE FOOTER COMPONENT	12
READYING THE HTML	12
CSS STYLES	13
THE HEADER COMPONENT	14
BLOCK COMPONENTS	15
THE FOOTER COMPONENT	16
SUMMARY.....	16
COPYRIGHT	17

Introduction

This tutorial focuses on creating an AutoTheme theme from a graphic template such as the ones available at Spidean or Webvida. Though all graphic templates are different, this is a general approach that can be applied to any that you may acquire or create on your own.

For this tutorial we will be using Macromedia Fireworks MX (referred to as FW) as the image editor. FW integrates very well with Macromedia Dreamweaver MX (referred to as DW), which is the HTML editor used in this tutorial. Some may prefer to use Adobe PhotoShop and Adobe GoLive or other tools, but the basic concepts here are the same.

The HTML and graphic image files used in this tutorial are located in the files/ directory. The working theme is located in the themes/AT-Tutorial/ directory.

Preparing a Graphic

Slices are a way to easily create the table layout and structure of your template while still in an image editor, but beware, due to the dynamic nature of information that needs to be displayed in a theme, slices can be more of a hindrance than a help. I have found that the best way to work with sliced output is to take a step by step approach from the top (header) down to the bottom (footer). This makes it easier to visualize what is going on and you also have more control of how your table cells are created, which is very important!

When creating your page design be sure to think about how it will function as a theme, i.e. will all the block areas be able to expand and contract vertically without breaking the layout? Advanced theme designers have to take into account horizontal movement as well. The most important thing is a sound understanding of HTML, tables, rows and cells – if you don't know about them, please do a Google search for some tutorials. I can't stress enough how important tables are to theme design!

Slicing and Exporting

Open “template.png” with FW. You can see I have already sliced the images. Your page layout could be created in numerous other ways, but we will use this one for the example. Basically what I will be doing here is showing you how to export the slices and edit the resulting sliced HTML in such a way that it will function as a theme.

- ✓ From the **File** menu, select **Export**
- ✓ In the Export dialogue, click the **Options** button
- ✓ In the HTML Setup dialogue, click the **Table** tab and make sure all the settings are the same as the figure below. This is done because we do not want spacer images. Why? Because they suck...
- ✓ Click **OK**
- ✓ Now back in the **Export** dialogue, navigate to the directory you will be using for the theme
- ✓ Enter template in the **File name** field
- ✓ Click the **Put images in subdirectory** check box
- ✓ Click **Save**

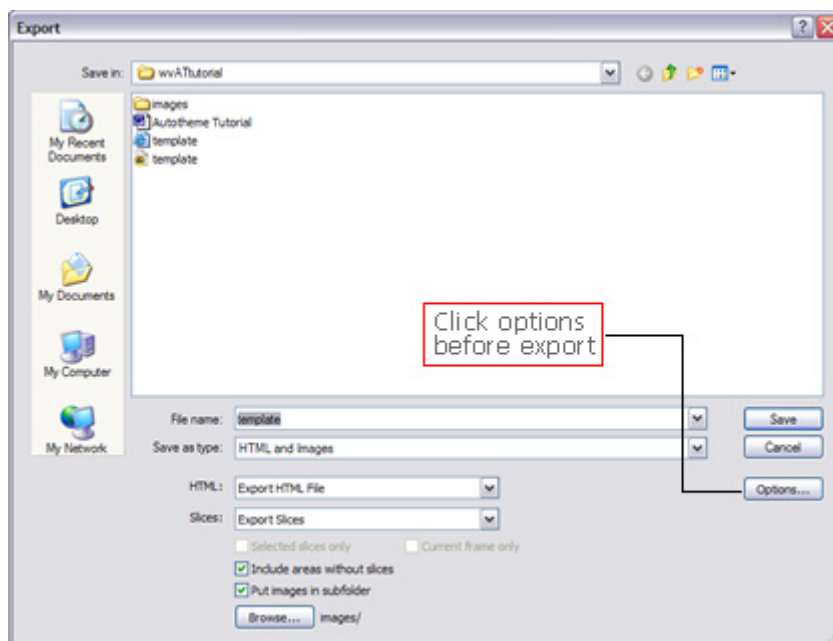


Figure 1 - Export dialogue

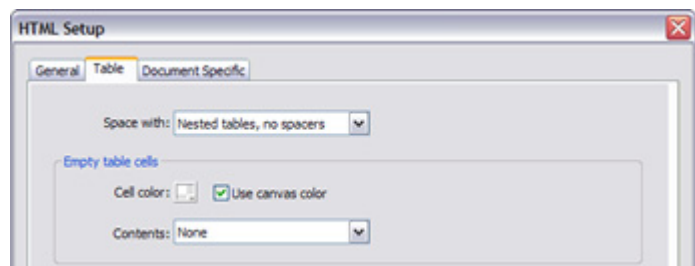


Figure 2 - HTML Setup dialogue

Preparing the HTML

If you navigate to the directory where you exported the slices you will find that you now have a “template.htm” file and a directory called “images”. We now have a web page that you can see this by opening the file in your browser. Don’t worry about the table borders. This is due to a quirk in FW and is easily fixed.

Previously I mentioned that we would be taking a layered approach to creating the theme template file. This is where we start and what we are looking to do is simplify the layout as much as possible by removing unnecessary tables and cells.

The Header Component

Open “template.htm” in DW.

The first thing to do is fix the tables.

- ✓ Replace all instances of “**undefined**” with “**0**”
- ✓ Remove all instances of **align=“left”** from between the **<table>** tags

Next we will concentrate on the header of the template and we are looking to separate this table from the rest of the HTML and move it to the top. Let’s have a look at the header code:

```
<body>
<table border="0" cellpadding="0" cellspacing="0" width="778">
  <tr>
    <td><table border="0" cellpadding="0" cellspacing="0" width="778">
      <tr>
        <td></td>
        <td><table border="0" cellpadding="0" cellspacing="0" width="462">
          <tr>
            <td></td>
          </tr>
          <tr>
            <td></td>
          </tr>
        </table></td>
      </tr>
      <tr>
        <td><table border="0" cellpadding="0" cellspacing="0" width="20">
          <tr>
            <td></td>
          </tr>
          <tr>
            <td></td>
          </tr>
        </table></td>
      </tr>
    </table></td>
  </tr>
</table></td>
</tr>
```

The HTML in **bold** is what we are looking to keep so we are going to separate this from the main table that contains the page layout; things are made a lot easier by doing this due to the fact that the header of a theme normally contains a complex layout structure and we really need to be able to manipulate this without messing up the rest of our page layout.

Cut the bold code and paste it above the first **<table>** tag in your HTML document. Now save it and view in a browser. It should look exactly the same as before, but now it is independent of the rest of the template. Put in some comment tags to show where your header begins and ends:

<!-- Header Begins --> *header table* **<!-- Header Ends -->**

Nice and neat! A modular type approach to template design means that you can easily find the code you are looking for and also aids in the reusability of your template. We will return to the header shortly, but let's first create the other layout components of the template.

The Body Component

Now to separate the template body, which is the area that will contain all of your blocks. The process is very similar to the one above so let's get cracking!

Here is the code. We are starting at the first **<table>** tag (below the header component):

```
<table border="0" cellpadding="0" cellspacing="0" width="778">
  <tr>
    <td></td>
  </tr>
  <tr>
    <td><table border="0" cellpadding="0" cellspacing="0" width="778">
      <tr>
        <td></td>
        <td></td>
      </tr>
    </table></td>
  </tr>
  <tr>
    <td></td>
  </tr>
</table>
</body>
</html>
```

Again, we will cut the code in **bold** and paste it above the original first **<table>** tag (below the header component).

Block Preparation

Now we can prepare for our blocks. Let's take a look at the body code, where our blocks will be displayed:

```
<!-- Begin Body Component -->
<table border="0" cellpadding="0" cellspacing="0" width="778">
  <tr>
    <td></td>
    <td></td>
  </tr>
</table>
<!-- End Body Component -->
```

This is where we will create our block layout. First we will work out our columns which will for a start, comprise: left, center and right:

```
<!-- Begin Body Component -->
<table border="0" cellpadding="0" cellspacing="0" width="100%">
  <tr>
    <!-- Begin Left Column -->
    <td style="width:25%;vertical-align:top">
      Left Blocks will be here
    </td>
    <!-- End Left Column -->

    <!-- Begin Center Column -->
    <td style="width:50%;vertical-align:top">
      center blocks will be here
    </td>
    <!-- End Center Column -->

    <!-- Begin Right Column -->
    <td style="width:25%;vertical-align:top">
      right blocks will be here
    </td>
    <!-- End Right Column -->

  </tr>
</table>
<!-- End Body Component -->
```

You can make this as complex as you like, but to do this you need a good understanding of tables and how they work. For now we will just concentrate on a simple 3 column layout. Notice that I have used % to define the widths of the columns. This means that they will hold their layout regardless of screen resolution.

The Footer Component

Now all we have to do is separate the footer component, which is easy because we are only removing some garbage tags from the original HTML.

```
<table border="0" cellpadding="0" cellspacing="0" width="778">
  <tr>
    <td></td>
  </tr>
  <tr>
    <td></td>
  </tr>
  <tr>
    <td></td>
  </tr>
</table>
```

Delete the code that is in **bold**. We now have a solid foundation upon which to build our theme, which is easy to read and very easy to manipulate. If you want to see the finished product, open “template.htm” include in the tutorial.

The Header Component

Now we have to set up our header in order for it to be capable of displaying CMS output.

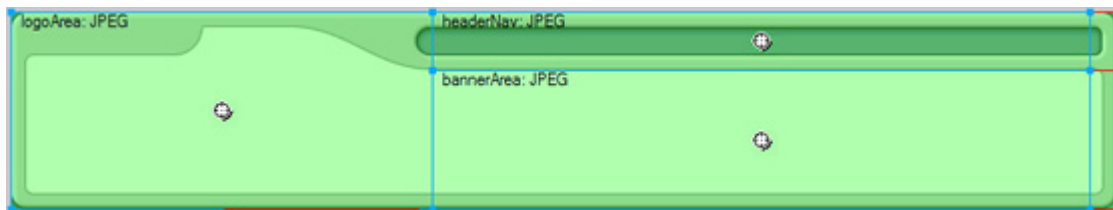


Figure 3 - Header image

As seen in the image above, our header is sliced into 3 main areas whose HTML is in **bold** below:

```
<!-- Begin Header Component-->
<table border="0" cellpadding="0" cellspacing="0" width="778">
  <tr>
    <td></td>
    <td><table border="0" cellpadding="0" cellspacing="0" width="462">
      <tr>
        <td></td>
      </tr>
      <tr>
        <td></td>
      </tr>
    </table></td>
  </tr>
  <tr>
    <td><table border="0" cellpadding="0" cellspacing="0" width="20">
      <tr>
        <td></td>
      </tr>
    </table>
  </tr>
</table>
```



```

                <tr>
                <td></td>
                </tr>
            </table></td>
        </tr>
    </table>
<!-- End Header Component-->

```

Readying the HTML

logoArea

This area will contain the site logo, specific to the theme. There is a way to set this from the CMS Administration, but I prefer to just include it as a blank for the end user to add their logo, because it is then easier for them to make their logo fit seamlessly into the theme.

The HTML for this area is in **bold** and we do not have to do anything more with it (some may want to make it a link to the home page of their website).

headerNav

This area will contain a user login bar along with some links once the user has logged in. We need to change the code a bit for it to be able to hold the data. First we will have to make it so the image “headerNav.jpg” becomes the background of the table cell **<td>**. This is easily done using CSS styles, so replace this code:

```
<td></td>
```

With this code:

```
<td style="background-image:url(images/headerNav.jpg);width:100%;height:41px;">Header Navigation
Content</td>
```

Notice the width is 100%? We will get to this later.

bannerArea

This area will contain the banner from the native CMS banner system. Replace this code:

```
<td></td>
```

With this code:

```
<td style="background-image:url(images/bannerArea.jpg);width:100%;height:98px;">
  <a href="http:web-templates.webvida.com"></a>
</td>
```

We have also added a banner here for reference. This is in the images directory of the tutorial.

We are also going to add another style attribute to the bannerArea table. This will make the banner nice and centered.

New Code:

```
<td style="background-image:url(images/bannerArea.jpg);width:100%;height:98px; padding-bottom:5px;">
  <a href="http:web-templates.webvida.com"></a>
</td>
```

The padding style attribute is handy to position elements within a table cell. You can also use padding-left, right, top or just plain padding for all sides.

Let's make it 100% width!

Probably the most challenging (yet rewarding) thing you can do in template design is to make the layout 100% width with the ability to dynamically adjust to a user's screen resolution.

If you have a look at "bannerArea.jpg" and "headerNav.jpg", you will see that the images can repeat horizontally in a seamless way. The images only need to be 1px wide to do this. Check it out for yourself! Edit the images so they are only one pixel wide and replace the original images – see no change!

All we need to do now for the header to be 100% width is add some style attributes which are depicted in **bold**:

```
<table border="0" cellpadding="0" cellspacing="0" style="width:100%">
  <tr>
    <td></td>
    <td style="width:100%"><table border="0" cellpadding="0" cellspacing="0" style="width:100%">
      <tr>
        <td style="background-image:url(images/headerNav.jpg);width:100%;height:41px;">Header
        Navigation Content</td>
      </tr>
      <tr>
        <td style="background-image:url(images/bannerArea.jpg);width:100%;height:98px;padding-
        bottom:5px;">
          <a href="http:web-templates.webvida.com"></a>
        </td>
      </tr>
    </table></td>
    <td><table border="0" cellpadding="0" cellspacing="0" width="20">
      <tr>
        <td></td>
      </tr>
      <tr>
        <td></td>
      </tr>
    </table></td>
  </tr>
</table>
```

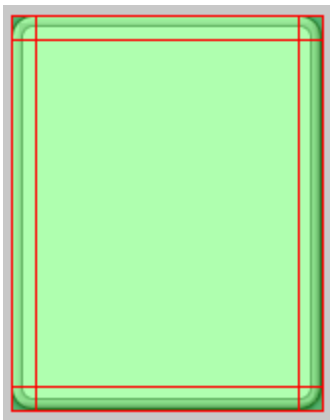
Cool, now our header is 100% width! Let's fix the banner so it will align to the right and look much better! Simply add the style attribute in bold to the line of HTML:

```
<td style="background-image:url(images/bannerArea.jpg);width:100%;height:98px;padding-bottom:5px;text-align:right;">
```

Easy! You may notice that we are using lots of styles here in place of normal HTML tag attributes. We do this because many of these HTML tag attributes are becoming obsolete and will not be available in future w3c standards. This won't happen for a while, but it is good to get into the routine of using CSS styles!

Block Components

Now we are going back to FW. Open "blockTemplate.png" and take a look at it:



We have cut this block into 9 slices and the output will result in a table like the one below:

This means our block will be able to resize horizontally and vertically.

Readying the HTML

Let's take a look at the sliced output from FW:

```
<table border="0" cellpadding="0" cellspacing="0" width="155">
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</table>
```

Nice and neat, not much to here really, but we must make it possible for this table to resize horizontally and vertically. The code that needs to be changed is in **bold** above. Below is what we have done:

```

<table border="0" cellpadding="0" cellspacing="0" width="300">
  <tr>
    <td></td>
    <td style="background-image:url(images/bTop.jpg);width:100%;height:12px;"></td>
    <td></td>
  </tr>
  <tr>
    <td style="background-image:url(images/bLeft.jpg);width:12px;height:100%;"></td>
    <td style="background-color: #ffffff">
      Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed
        diam nonumy eirmod tempor invidunt ut labore et dolore magna
        aliquyam erat, sed diam voluptua. At vero eos et accusam et
        justo duo dolores et ea rebum. Stet clita kasd gubergren, no
        sea takimata sanctus est Lorem ipsum dolor sit amet.
    </td>
    <td style="background-image:url(images/bRight.jpg);width:12px;height:100%;"></td>
  </tr>
  <tr>
    <td></td>
    <td style="background-image:url(images/bBottom.jpg);width:100%;height:12px"></td>
    <td></td>
  </tr>
</table>

```

From now on, no matter what content is present, the block will handle it without breaking its layout and we are now ready to add it to the main template. You can have all sorts of blocks and we will cover this later on.

Open up your “template.htm” and focus on the template body component:

```

<!-- Begin Body Component -->
<table border="1" cellpadding="0" cellspacing="0" width="100%">
  <tr>
    <!-- Begin Left Column -->
    <td style="width:25%;vertical-align:top">
      Left Blocks are here
    </td>
    <!-- End Left Column -->

    <!-- Begin Center Column -->
    <td style="width:50%;vertical-align:top">
      Center Blocks are here.
    </td>
    <!-- End Center Column -->

    <!-- Begin Right Column -->
    <td style="width:25%;vertical-align:top">
      Right Blocks are here
    </td>
    <!-- End Right Column -->

  </tr>
</table>
<!-- End Body Component -->

```

Bet you can guess where we will put the blocks! Take the complete block table and add it where instructed. Here is the left column example:

```

<!-- Begin Left Column -->
<td style="width:25%;vertical-align:top">

<!-- Begin Block -->
<table border="0" cellpadding="0" cellspacing="0" width="300">
  <tr>
    <td></td>
    <td style="background-image:url(images/bTop.jpg);width:100%;height:12px;"></td>
    <td></td>
  </tr>
  <tr>
    <td style="background-image:url(images/bLeft.jpg);width:12px;height:100%;"></td>
    <td>
      Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed
        diam nonumy eirmod tempor invidunt ut labore et dolore magna
        aliquyam erat, sed diam voluptua. At vero eos et accusam et
        justo duo dolores et ea rebum. Stet clita kasd gubergren, no
        sea takimata sanctus est Lorem ipsum dolor sit amet.
    </td>
    <td style="background-image:url(images/bRight.jpg);width:12px;height:100%;"></td>
  </tr>
  <tr>
    <td></td>
    <td style="background-image:url(images/bBottom.jpg);width:100%;height:12px;"></td>
    <td></td>
  </tr>
</table>
<!-- End Block -->

</td>
<!-- End Left Column -->

```

Easy! Now add to all the columns. Add two to each column.

Now we have a working layout, but I think we should have some gaps between the blocks. Again we use CSS for this. Let's also add some padding to the columns. First we will do the Left column:

```

<!-- Begin Left Column -->
<td style="width:25%;vertical-align:top;padding-top:10px;">

```

Then the Center column:

```

<!-- Begin Center Column -->
<td style="width:50%;vertical-align:top;padding:10px;padding-bottom:0px;">

```

Then the Right column:

```

<!-- Begin Right Column -->
<td style="width:25%;vertical-align:top;padding-top:10px;">

```

Now that we have some padding between columns, we also need to add padding to the blocks. We should have done this first as we would only have to change one instance of a block, but these things happen. The code added is in **bold**:

```
<!-- Begin Block -->
<table border="0" cellpadding="0" cellspacing="0" width="300">
  <tr>
    <td></td>
    <td style="background-image:url(images/bTop.jpg);width:100%;height:12px;"></td>
    <td></td>
  </tr>
  <tr>
    <td style="background-image:url(images/bLeft.jpg);width:12px;height:100%;"></td>
    <td style="background-color: #ffffff;">
      Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed
        diam nonumy eirmod tempor invidunt ut labore et dolore magna
        aliquyam erat, sed diam voluptua. At vero eos et accusam et
        justo duo dolores et ea rebum. Stet clita kasd gubergren, no
        sea takimata sanctus est Lorem ipsum dolor sit amet.
    </td>
    <td style="background-image:url(images/bRight.jpg);width:12px;height:100%;"></td>
  </tr>
  <tr>
    <td></td>
    <td style="background-image:url(images/bBottom.jpg);width:100%;height:12px;"></td>
    <td></td>
  </tr>
  <tr>
    <td colspan="3" style="height:10px;"></td>
  </tr>
</table>
<!-- End Block -->
```

We also span the 3 columns. You need to do this, otherwise it will only have the width of the first column. It doesn't matter really because you can't see it anyway.

The Footer Component

Readying the HTML

You can now use the same process that we used for the blocks to make the footer. It operates with the exact same principle and I am not going to demonstrate this as it was done before. It's safe to say that the code is available in "footerTemplate.htm" and the layout in "footerTemplate.png". Please attempt this yourself using the block methods above as a reference.

CSS Styles

If you are not familiar with styles, then please do a Google search for some tutorial. They are not very difficult to use and we will hold your hand through this stage of the tutorial.

First we will attach a style sheet to our template. To do this we must modify “template.htm”, adding the code in bold between the **<head>** tags:

```
<head>
<title>template.gif</title>
<meta http-equiv="Content-Type" content="text/html;">
<link href="style/style.css" rel="stylesheet" type="text/css">
</head>
```

Open the file “style.css” in the style directory.

Let’s start with the **<body>** tag. If you have a look in “template.htm”, you will see that the tag has a background color:

```
<body bgcolor="#7f90a0">
```

We are going to move this over to the style sheet and add another attribute which will set the page margins to 0:

```
body {
margin: 0px;
background-color: #7f90a0;
}
```

Now we will create a global class which will define the styles of all fonts that do not have an individual class. This is handy because some modules have text which does not have a class declared. Also if you want to make a theme multi-platform this will take care of the modules.

```
table {font-family: Geneva, Arial, Helvetica, sans-serif; font-size: 11px; font-weight: normal; color: #174674;}

table A:link {COLOR: #FF5500;FONT-SIZE: 11px;font-weight: bold; FONT-FAMILY: Geneva, Arial, Helvetica, sans-serif;TEXT-DECORATION: underline;}

table A:visited {COLOR: #FF5500;FONT-SIZE: 11px;font-weight: bold; FONT-FAMILY: Geneva, Arial, Helvetica, sans-serif;TEXT-DECORATION: underline;}

table A:hover {COLOR: #666666;FONT-SIZE: 11px;font-weight: bold; FONT-FAMILY: Geneva, Arial, Helvetica, sans-serif;TEXT-DECORATION: underline;}

table A:active {COLOR: #666666;FONT-SIZE: 11px;font-weight: bold; FONT-FAMILY: Geneva, Arial, Helvetica, sans-serif;TEXT-DECORATION: none;}
```

Now we will work from the top to the bottom (our tried and trusted method) and set classes for the different components and blocks.

The Header Component

```
<!-- Begin Header Component-->
<table border="0" cellpadding="0" cellspacing="0" style="width:100%">
  <tr>
    <td></td>
    <td style="width:100%"><table border="0" cellpadding="0" cellspacing="0" style="width:100%">
      <tr>
        <td style="background-image:url(images/headerNav.jpg);width:100%;height:41px;">Header
Navigation Content</td>
      </tr>
      <tr>
        <td style="background-image:url(images/bannerArea.jpg);width:100%;height:98px;padding-
bottom:5px;text-align:right;">
        <a href="http:web-templates.webvida.com"></a>
      </td>
      </tr>
    </table></td>
    <td><table border="0" cellpadding="0" cellspacing="0" width="20">
      <tr>
        <td></td>
        </tr>
        <tr>
        <td></td>
        </tr>
      </table></td>
    </tr>
  </table>
<!-- End Header Component-->
```

We are going to move all the style attributes over to the style sheet for the navigation bar. The line that we will be altering is pictured above in **bold**. Change the line of code to this:

```
<td class="headernav">Header Navigation Content</td>
```

Add these classes to your style sheet:

```
.headernav {
  background-image:url(images/headerNav.jpg);
  width:100%;
  height:41px;
  font-family: Geneva, Arial, Helvetica, sans-serif;
  font-size: 11px;
  font-weight: normal;
  color: #ffffff;
}
.headernav A:link {
  COLOR: #ffffff;
  FONT-SIZE: 11px;
  font-weight: bold;
  FONT-FAMILY: Geneva, Arial, Helvetica, sans-serif;
  TEXT-DECORATION: underline;
}
.headernav A:visited {
  COLOR: #ffffff;
  FONT-SIZE: 11px;
```



```

font-weight: bold;
FONT-FAMILY: Geneva, Arial, Helvetica, sans-serif;
TEXT-DECORATION: underline;
}
.headernav A:hover {
COLOR: #cccccc;
FONT-SIZE: 11px;
font-weight: bold;
FONT-FAMILY: Geneva, Arial, Helvetica, sans-serif;
TEXT-DECORATION: underline;
}
.headernav A:active {
COLOR: #cccccc;
FONT-SIZE: 11px;
font-weight: bold;
FONT-FAMILY: Geneva, Arial, Helvetica, sans-serif;
TEXT-DECORATION: none;
}

```

Now you can easily control the appearance of fonts and links in your header navigation bar.

Block Components

Now we are going to do the same with the blocks, but first we will give the block a title and content class. The added code is in **bold**.

```

<!-- Begin Block -->
<table border="0" cellpadding="0" cellspacing="0" width="300">
<tr>
<td></td>
<td style="background-image:url(images/bTop.jpg);width:100%;height:12px;"></td>
<td></td>
</tr>
<tr>
<td style="background-image:url(images/bLeft.jpg);width:12px;height:100%;"></td>
<td style="background-color: #ffffff;">
<div class="block-title">Block Title</div>
<div class="block-content">
Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed
    diam nonumy eirmod tempor invidunt ut labore et dolore magna
    aliquyam erat, sed diam voluptua. At vero eos et accusam et
    justo duo dolores et ea rebum. Stet clita kasd gubergren, no
    sea takimata sanctus est Lorem ipsum dolor sit amet.
</div>
</td>
<td style="background-image:url(images/bRight.jpg);width:12px;height:100%;"></td>
</tr>
<tr>
<td></td>
<td style="background-image:url(images/bBottom.jpg);width:100%;height:12px;"></td>
<td></td>
</tr>
<tr>
<td colspan="3" style="height:10px"></td>
</tr>
</table>
<!-- End Block -->

```

Now you can create corresponding classes in “style.css”. I’m not going to show them here because they take up too much space. Just have a look at “style.css”. Now you are not constrained to only one style class for all of your blocks. You could make a class for each block (i.e. left-block-title, left-block-content). Just make sure that you declare the class in your template and create one in the style sheet.

The Footer Component

Ok now create a class for the footer: are you up to it? Try to do it yourself! You can see how it works by checking the “style.css”, “template.htm” and “footerTemplate.htm”.

Summary

This tutorial has shown you how to take a graphic template such as those available on Spidean and Webvida, and elsewhere on the Internet, and create theme ready HTML. Part 2 of this tutorial shows you how to take this HTML and create HTML templates for use with AutoTheme.

Copyright

Copyright © 2004 Shawn McKenzie. All rights reserved.

No part of this document may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, recording, or otherwise, without the prior written permission of Shawn McKenzie.

The content of this document is furnished for informational use only, is subject to change without notice, and should not be construed as a commitment by the authors or copyright holder. The authors and copyright holder assume no responsibility or liability for any errors or inaccuracies that may appear in the informational content contained in this document.

All names and trademarks are property of their respective companies.