

HTML for the SAS® Programmer

Lauren Haworth, Genentech, Inc., San Francisco

➤ ABSTRACT

With more and more output being delivered via the Internet, a little knowledge of HTML can go a long way toward improving the appearance of your output. This paper introduces some simple HTML coding techniques that are useful for SAS programmers. The paper will show how your SAS output is converted into HTML, and demonstrate HTML tricks that you can use in your SAS code to dress up your SAS HTML output.

The examples in this paper are based on SAS version 8.

➤ INTRODUCTION: HOW TO CREATE HTML OUTPUT FROM SAS

There are several ways you can move your SAS output to the web. If you are going to be doing a lot of web publishing, you may want to investigate the SAS/InterNet® product. However, if you just want to publish the occasional table, report, or printout on the web, you can do this with SAS/BASE®.

To put your output on the web, you need to convert standard SAS output into HTML files. HTML stands for HyperText Markup Language, and it is the common language understood by web browsers like Netscape and Internet Explorer.

To create HTML-formatted results, all you have to do is use the Output Delivery System and specify HTML as the output format. A simple example is shown below:

```
ODS HTML BODY='myfile.htm';  
PROC PRINT DATA=TEMP NOOBS LABEL;  
  VAR JobType Salary;  
RUN;  
ODS HTML CLOSE;
```

➤ WHAT YOUR OUTPUT LOOKS LIKE ON THE WEB

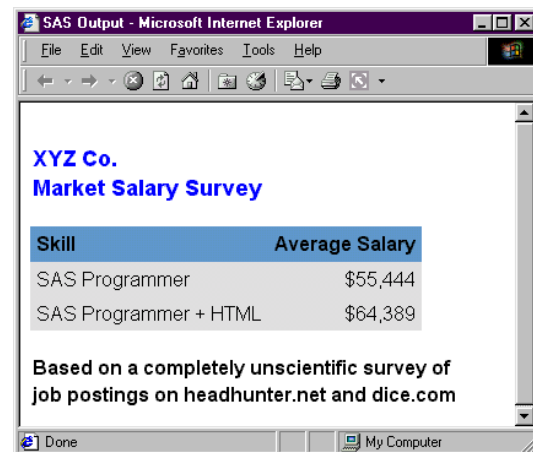
When you create an HTML file from your SAS output, the usual SAS output format of text and numbers separated by spaces is “marked up” with HTML “tags.” These tags tell Internet browsers how to display your output. They identify headers and footers, specify fonts and type

sizes, reference images, and point to other locations on the web (hyperlinks).

For example, the following PROC PRINT output ...

XYZ Co.	
Market Salary Survey	
Skill	Average Salary
SAS Programmer	\$55,444
SAS Programmer + HTML	\$64,369
Based on a completely unscientific survey of job postings on headhunter.net and dice.com	

... would look like this on the web ...



... and would contain the HTML code shown in Figure 1. The figure is considerably simplified, with a number of formatting tags removed for the sake of clarity.

FIGURE 1: HTML CODE

<HTML>

<HTML> </HTML> surrounds all code, and identifies it to the browser as HTML code.

<HEAD>

<TITLE>SAS Output</TITLE>

</HEAD>

<HEAD> </HEAD> surrounds your header. In this section, you will find general information that applies to the entire file. The <TITLE> </TITLE> tags identify the text to display in the browser's title bar when the page is viewed.

<BODY>

<BODY> </BODY> tags enclose the "guts" of your HTML output. All of the content that will be displayed on the page is contained between these two tags.

<TABLE><TR><TD>XYZ Co. </TD></TR>

<TR><TD>Market Salary Survey</TD></TR></TABLE>

<TABLE> </TABLE> tags enclose a table that is used to align the two titles for this output. The <TR> </TR> tags mark the beginning and end of each table row. The <TD> </TD> tags mark the beginning and end of each table cell in the row. Table tags are also used to align the results and the footnotes.

<P>

<P> </P> marks a break between paragraphs. It puts a large break between the lines above and below. If you don't want such a large break, use the
 tag

<TABLE>

The <THEAD> </THEAD> tags mark the beginning and end of the heading row of the table. Text in these cells is displayed in bold in the browser.

<TABLE><THEAD><TR><TD>Skill</TD><TD>Average Salary</TD></TR></THEAD>

The <TBODY> </TBODY> tags mark the beginning and end of the body rows of the table. Text in these cells is not displayed in regular weight (not bold) in the browser.

<TBODY><TR><TD>SAS Programmer</TD><TD>\$55,444</TD></TR>

<TR><TD>SAS Programmer + HTML</TD><TD>\$64,389</TD></TR>

</TBODY></TABLE>

<P>

<TABLE><TR><TD>Based on a completely unscientific survey of</TD></TR>

<TR><TD>job postings on headhunter.net and dice.com</TD></TR></TABLE>

</BODY>

</HTML>

As you review the information in Figure 1, you will see that the HTML file is composed of the basic text of the SAS output (shown in italics), surrounded by numerous HTML tags enclosed in angle brackets (shown in bold face).

If you look closer, you will also see that HTML tags come in pairs. If you look at the very first tag on the page: `<HTML>`, you will see that there is a matching `</HTML>` tag at the very bottom of the file.

Figure 1 explains the meaning of each of the tags in the file. This is just a small sampling of HTML tags; there are dozens more. However, if you can understand the tags used here, you will understand 80-90% of the tags you will find in more complicated web pages.

➤ TRICK #1: ADDING A HYPERLINK TO YOUR FOOTNOTE.

Now that you know a little about HTML tags, it's time to turn this knowledge to our advantage. We can use a couple of HTML tags in our FOOTNOTE statement to add a hyperlink to our output. This is a useful trick because it allows you to annotate your output. You can use the footnote to list the source of your data, and then attach a hyperlink so when the viewer clicks on the footnote, it automatically links them to the web site for that source.

To do this, you need to learn a new HTML tag. The tag for a hyperlink is:

```
<A HREF="xxx">text</A>
```

The "xxx" is where you put the information on where you want the hyperlink to go. This needs to be a valid URL (Universal Resource Locator) like "http://www.sas.com" or "http://www.mycompany.org/sales.htm".

The "text" is where you put the text for the hyperlink. This is the text that will be displayed on the page (with an underline to indicate that this is a hyperlink). For example, you could use "SAS Institute" or "January 2001 Sales Figures."

To see this code in action, we can add the following hyperlinks to the footnote statement from our previous examples. The original FOOTNOTE statement for the second line of the footnote was:

```
FOOTNOTE2 'job postings on  
headhunter.net and dice.com';
```

The revised FOOTNOTE statement with hyperlinks added is:

```
FOOTNOTE2 'Job postings on  
<A HREF="http://www.headhunter.net">  
headhunter.net</A> and  
<A HREF="http://www.dice.com">  
dice.com</A>';
```

When you run this code, SAS passes the HTML tags along with the rest of the text for the footnote. When the file is viewed with a browser, these tags are interpreted, and the footnote is displayed with a hyperlink. This is how the file looks when viewed with a browser:

XYZ Co.	
Market Salary Survey	
Skill	Average Salary
SAS Programmer	\$55,444
SAS Programmer + HTML	\$64,389
Based on a completely unscientific survey of job postings on headhunter.net and dice.com	

One warning: version 8.1 will sometimes fail to correctly convert the HTML tags in your footnotes. Any tags you put in titles work fine. The problem has been corrected in the 8.2 release.

➤ TRICK #2: CHANGING THE TITLES

Another HTML tag that we can use to our advantage is the header tag. This tag identifies the level of the header. Lower levels (`<H1>` or `<H2>`) create big, bold titles. Higher levels create smaller titles. SAS picks a size for your titles based on the ODS style you are using.

However, if we want a bigger title, all we have to do is add header tags to the text in our titles. The following code shows how this is done:

```
TITLE "<H1>XYZ Co.</H1>";  
TITLE "<H2>Market Salary Survey</H2>";
```

This code uses the `<H1>` `</H1>` tags on the first title for a really big title, and the `<H2>` `</H2>` tags for the second title for a slightly smaller title. The results are shown below:

XYZ Co.

Market Salary Survey

Skill	Average Salary
SAS Programmer	\$55,444
SAS Programmer + HTML	\$64,389

Based on a completely unscientific survey of job postings on [headhunter.net](#) and [dice.com](#)

The results are shown below:

XYZ Co.

Market Salary Survey

Skill	Average Salary
SAS Programmer	\$55,444
SAS Programmer + HTML	\$64,389

Based on a completely unscientific survey of job postings on [headhunter.net](#) and [dice.com](#)

➤ TRICK #3: CHANGING THE FONTS

The titles in the previous example are certainly bigger, but they're not very attractive. This is because each web browser has a default font and point size for each heading level. Unfortunately, the fonts are pretty basic.

This example will show how to specify the font and point size for each title, instead of just using the standard heading definitions.

To do this, you need to learn a new pair of HTML tags: `<FONT>` `</FONT>`. What these tags do is control the appearance of the text between the two tags. You can specify the typeface, color, and size, among other attributes.

For this example, we're going to put some tags into our titles to select two new typefaces and point size settings. The syntax is:

```
TITLE "<FONT FACE='Comic Sans MS'
      SIZE=5>XYZ Co.</FONT>";
TITLE2 "<FONT FACE='Arial'
       SIZE=3>Market Salary Survey</FONT>";
```

Notice how the two parameters for the FONT tag are used. The FACE= tag is used to specify the typeface. The SIZE= tag is used to specify the font size. A SIZE setting of 3 is the standard font size used for the rest of the text on the screen. By using a larger setting for the first title, it stands out from the rest of the text.

The technique can also be used on footnotes. In this case, it would be nice if the footnote were smaller, so it wouldn't draw so much attention. The following code changes the footnote to the typeface Arial and the size to 1, which is much smaller.

```
FOOTNOTE "<FONT FACE='Arial' SIZE=1>
          the text goes here </FONT>";
```

➤ TRICK #4: CHANGING THE FONT COLOR

In addition to controlling font size and typeface, the FONT tag can also be used to change the color of your fonts. If you don't like these bright blue titles, you can change them to a more aesthetically pleasing color.

The syntax for assigning a font color is:

`<FONT COLOR=color>`

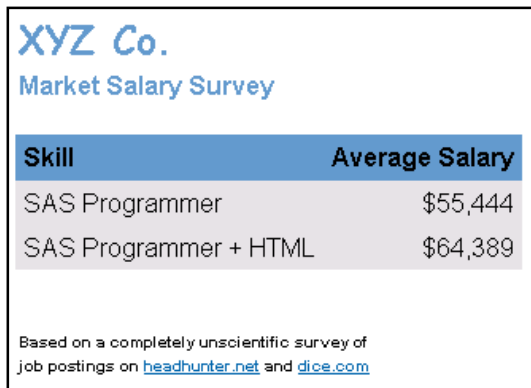
There are several ways you can define the color. First, you can use a color name. For example, to get a red title, you could assign the font to `COLOR="red"`. However, there is a limitation here. Browsers vary in terms of which color names they understand, and also in how those colors are rendered. "Red" may be bright and orangish in one browser and dull and bricklike in another browser.

To get exactly the color you want, the best way is to specify the color using RGB codes. These are hexadecimal values that indicate the amount of red, green, and blue to use to create your color. You can look up these codes in the SAS/GRAPH manual, or use a "Color Picker" on the internet (just use your favorite search engine to find one, there are dozens of free color pickers out there).

For this example, we'll use the RGB code for a lighter blue that matches the table heading to create a new color for our titles. The code for the title statement is:

```
TITLE "<FONT COLOR='#639ACE'
      FACE='Comic Sans MS' SIZE=5>
      XYZ Co.</FONT>";
TITLE2 "<FONT COLOR='#639ACE'
       FACE='Arial' SIZE=3>
       Market Salary Survey</FONT>";
```

The new output is shown below. The titles are now the same color as the table heading. (You'll have to take the author's word on this, as it's hard to see the difference in this black and white output).



XYZ Co.
Market Salary Survey

Skill	Average Salary
SAS Programmer	\$55,444
SAS Programmer + HTML	\$64,389

Based on a completely unscientific survey of job postings on [headhunter.net](#) and [dice.com](#)

➤ TRICK #5: ADDING A LOGO TO THE PAGE

One of the most powerful aspects of the HTML is its ability to display images along with the text on a page. Images are added to the page using the `<IMG>` tag.

This is a very simple tag to use. All you have to do is specify the image source by listing the file name for the image. Ideally, you should use a web-friendly image format like GIF or JPG for your images.

There are two ways you can add this logo to your output. The first is to add an `<IMG>` tag to your TITLE statement. The following code shows how to display an image in the top left corner of the page by adding an `<IMG>` tag to the first TITLE statement.

```
TITLE "<IMG SRC='xyzlogo.gif'
      HEIGHT=40>";
```

This IMG tag uses parameters to specify the file name to be used and the size to which the image will be expanded or compressed. It is important to specify a size, or you may end up with an image being displayed either too small to see or too large to fit on the screen. Picking the right size usually involves some trial and error.

The other way to add a logo is to create a new style definition. This approach allows you to add a logo at the top of every page of output, without having to edit all of the TITLE statements.

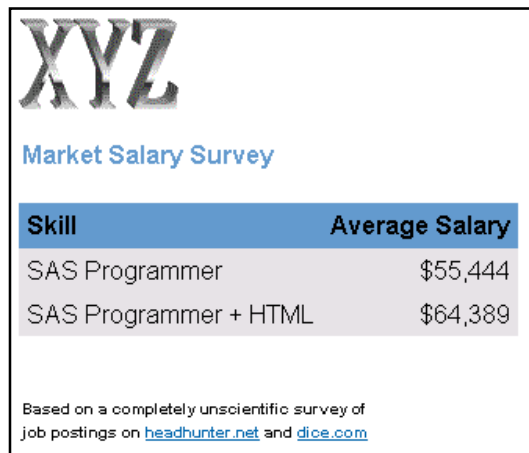
The code to create the new style definition is shown below:

```
proc template;
  define style MyHTMLStyle;
    parent=Styles.BarrettsBlue;
    replace Body from Document /
      prehtml="<IMG SRC='logo.gif'><br>";
  end;
run;
```

The PREHTML attribute is used to add the same `<IMG>` tag we used in the TITLE statement. The only difference is that putting it in the style means that it will be used at the top of every HTML page.

Now you can use this new style by adding `STYLE=MyHTMLStyle` to your ODS HTML statement.

Whether you use the TITLE approach or create a new style, your results will look like the output shown below.



XYZ
Market Salary Survey

Skill	Average Salary
SAS Programmer	\$55,444
SAS Programmer + HTML	\$64,389

Based on a completely unscientific survey of job postings on [headhunter.net](#) and [dice.com](#)

➤ TRICK #6: ADDING AN IMAGE TO THE BACKGROUND

Each of the previous examples has shown some basic black text on a white background. However, if you've spent much time surfing the Web, you probably know that it's possible to use all sorts of colors and images for the page background.

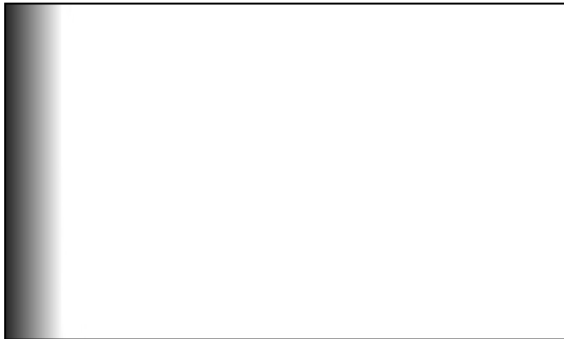
This next example will show how you can add a background image to your output.

To add a background image, we need to modify the `<BODY>` tag. We are going to add two new attributes to that tag.

First, we need to tell the browser what image to use. This is done using the following tag:

```
<BODY BACKGROUNDIMAGE=image>
```

The image is identified by file name, just as we did with the logo in the previous example. Again, web-friendly formats like GIF or JPG are best. The image we will use is shown below.



The other change we are going to make is to widen the left margin. The reason for this is that this image has a dark pattern on the left side, and we don't want our results to be hidden in the page background. If your background image does not detract from your results, you can skip this step.

The syntax for a margin change is:

<BODY LEFTMARGIN=number>

To apply these two new settings, we are going to create a new style. Just as in the previous example, we'll use PROC TEMPLATE.

```
proc template;
  define style MyHTMLStyle;
    parent=Styles.BarrettsBlue;
    replace Body from Document /
      prehtml="<IMG SRC='logo.gif'><br>"
      backgroundimage='xyzback.jpg'
      leftmargin=125;
  end;
run;
```

We're adding attributes to the <BODY> tag by adding attributes to the Body style element. Notice how the style attributes are named to match the <BODY> tag attributes. This is true of a number of the SAS style element attributes. The new results follow. If you look back to the original version of this table on the first page of this paper, you will see that this is quite an improvement.

Skill	Average Salary
SAS Programmer	\$55,444
SAS Programmer + HTML	\$64,389

Based on a completely unscientific survey of job postings on [headhunter.net](#) and [dice.com](#)

If you looked at the HTML code that was generated for this output, you would see that ODS has added two new parameters to the <BODY> tag in the file. The new body tag that controls the background is as follows.

<BODY BACKGROUND='xyzback.jpg' LEFTMARGIN=125>

Although you didn't have to specify this tag in order to use it, it's still helpful to review the code to see what tags ODS has used to build your output.

➤ TRICK #7: ADDING A NAVIGATION BAR

The previous examples have shown how to add some fairly simple HTML code to your results. This next example will show how you can add some fairly elaborate features using the tags we have already learned.

This example will add a navigation bar at the bottom of the page. This is a series of links back to other parts of your web site. It's a useful tool to help users move around your site without getting lost.

To add the navigation bar at the bottom of the page, we will use the POSTHTML attribute of the Body style element. This works just like the PREHTML attribute we used earlier.

The hyperlinks that make up the navigation bar will be organized into a single-row table. The HTML code we will use is:

```

<BR><TABLE CELLPADDING=8><TR>
<TD><A HREF='home.htm'>Home Page
</A></TD>
<TD><A HREF='about.htm'>About XYZ
</A></TD>
<TD><A HREF='contact.htm'>Contact Us
</A></TD>
<TD><A HREF='copyright.htm'>Copyright
Information</A>
</TD>
</TR></TABLE>

```

These tags create a table with one row and four columns. Each column contains a hyperlink to a different part of the web site. The CELLPADDING is set to 8 pixels to put some space between the links.

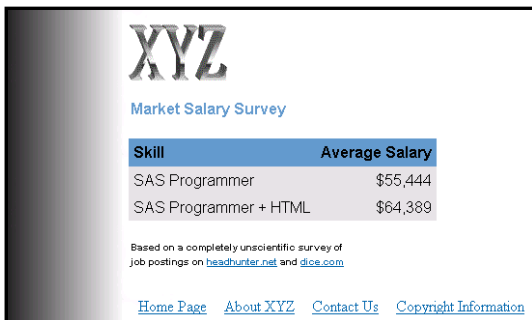
This code is added to our style as follows:

```

proc template;
  define style MyHTMLStyle;
    parent=Styles.BarrettsBlue;
    replace Body from Document /
      prehtml1="<IMG SRC='logo.gif'><br>";
      backgroundimage='xyzback.jpg'
      leftmargin=125
      posthtml1="<BR><TABLE CELLPADDING=8><TR>
        <TD><A HREF='home.htm'>Home Page
        </A></TD>
        <TD><A HREF='about.htm'>About XYZ
        </A></TD>
        <TD><A HREF='contact.htm'>Contact Us
        </A></TD>
        <TD><A HREF='copyright.htm'>
        Copyright Information
        </A></TD>
        </TR></TABLE>";
  end;
run;

```

The new results are shown below:



Skill	Average Salary
SAS Programmer	\$55,444
SAS Programmer + HTML	\$64,389

Based on a completely unscientific survey of job postings on [headhunter.net](#) and [dice.com](#)

[Home Page](#) [About XYZ](#) [Contact Us](#) [Copyright Information](#)

➤ CONCLUSIONS

These examples are just a brief introduction to the world of HTML. You can use these tricks to spice up your HTML output.

However, if you're going to be creating a lot of HTML output from SAS, I strongly recommend that you take the time to learn more about HTML. Although SAS gives you great tools to create HTML output, you will find it a lot easier to use these tools if you can understand the resulting output.

Start with reading up on ODS styles and templates. These tools will give you a lot of control over your output. You'll want to pay special attention to the documentation on the PREHTML and POSTHTML attributes.

The references section lists several good tutorials and books to help you learn more about HTML. Also, see the Appendix for a listing of common HTML tags.

Finally, the next time you're out surfing the web, take a look at the HTML code behind the pages you're viewing. You can learn a lot by viewing other people's web sites.

Netscape and Internet Explorer both have menu options to view the source code for the page on the screen. You won't recognize all of the tags you see, but the basic tags presented here will show up again and again.

➤ REFERENCES

Free downloadable HTML tutorial:

<http://www.pagetutor.com/pagetutor/index.html>

Another tutorial (aimed at kids, but it's a lot of fun)

hotwired.lycos.com/webmonkey/kids/

Good books for HTML beginners:

Castro, Elizabeth, *HTML 4 For The World Wide Web Visual Quickstart Guide*, Peachpit Press

Powell, Thomas, *HTML: the Complete Reference*, McGraw Hill

Lemay, Laura, and Denise Tyler, *Teach Yourself HTML 4 in 24 Hours*, SAMS

➤ ACKNOWLEDGEMENTS

SAS is a registered trademark of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are registered trademarks or trademarks of their respective companies.

➤ CONTACTING THE AUTHOR

Please direct any questions or feedback to the author at: info@laurenhaworth.com

APPENDIX: SELECTED HTML TAGS

➤ BASIC TAGS

<html></html> Creates an HTML document
<head></head> Sets off the title and other information that isn't displayed on the Web page itself
 <title></title> Puts the name of the document in the title bar
<body></body> Sets off the visible portion of the document
 <body bgcolor=?> Sets the background color, using name or hex value
 <body text=?> Sets the text color, using name or hex value

➤ TEXT TAGS

<pre></pre> Creates preformatted text
<h1></h1> Creates the largest headline
<h6></h6> Creates the smallest headline
**** Creates bold text
<i></i> Creates italic text
**** Emphasizes a word (with italic or bold)
**** Sets size of font, from 1 to 7
**** Sets font color, using name or hex value
**** Sets font typeface, using name (in quotes)

➤ LINKS

**** Creates a hyperlink

➤ FORMATTING

<p> Creates a new paragraph
<p align=?> Aligns a paragraph to the left, right, or center
**
** Inserts a line break
<blockquote></blockquote> Indents text from both sides
**** Creates a numbered list
 **** Precedes each list item, and adds a number
**** Creates a bulleted list
 **** Precedes each list item, and adds the bullet

➤ GRAPHICAL ELEMENTS

**** Adds an image
**** Sets display size of an image
<hr> Inserts a horizontal rule

➤ TABLES

<table></table> Creates a table
<tr></tr> Sets off each row in a table
<td></td> Sets off each cell in a row
<th></th> Sets off the table header (a normal cell with bold, centered text)

➤ TABLE ATTRIBUTES

<table border=#> Sets width of border around table cells
<table cellpadding=#> Sets amount of space between a cell's border and its contents
<tr align=?> or **<td align=?>** Sets alignment for cell(s) (left, center, or right)
<tr valign=?> or **<td valign=?>** Sets vertical alignment for cell(s) (top, middle, or bottom)