

SAS[®] and HTML - HTML Publishing Using SAS

Caroline Bahler, Biopop, Radford, VA

Sally Muller, UNC, Chapel Hill, NC

David Doolittle, Biopop, Radford, VA

Arturo Barrios, UNC, Chapel Hill, NC

Abstract

The current version of HTML allows the developer to have multiple frames on one web page, clickable table cells and graphics (such as maps), and different types of selection boxes. The question is how can a SAS[®] user expand their HTML universe and utilize these portions of HTML functionality? The answer is to learn some HTML and utilize the data _null_ step to publish! The objective of this paper is to discuss some of the advanced functionality within HTML and discuss how to integrate the SAS Web Publishing macros (and other SAS program goodies) with these advanced techniques.

Introduction

Hypertext markup language (HTML) is an evolving language that uses tags to signal internet browsers of different types on different hardware environments to present a document identically over all of those environments¹. A tag is a command that will be interpreted by the browser to display a portion of a document in a specific manner.

HTML Elements

The following HTML elements are specifically used to format information presented within an HTML document. This is not a definitive list and does not define all of the attributes available for each of the HTML elements discussed. The objective of this section is to introduce SAS users to some useful HTML elements and indicate how they can be used. At the end of the paper is a fairly comprehensive reference list of both books and Internet sites on HTML.




A word of warning, the elements described below require at least version 2 or greater of Netscape[®] or Microsoft Internet Explorer[®] (IE). In some cases, a given version of Netscape will have the element but IE will not. In any HTML document before using browser version specific elements it is a good practice to determine under what circumstances each element can be used and provide alternatives if necessary.

Lists

List tags can be divided into two groups: single line item lists and lists that allow a multi-line description of terms, ie. “regular” and “glossary” lists. Tags used to mark list are described below and all require a starting and finishing tag except the LI and DT tags where only the

starting tag is required. A “glossary” list consist of a term and a multi line description. “Regular” lists can be ordered (each item is assigned a number or letter in ascending order) or unordered (each item is assigned a bullet).

The elements used to create lists are detailed below (example 1):

<i>Element</i>	<i>Tags</i>	<i>Description</i>
Glossary List 	<DL>.. </DL>	Defines a list containing terms and their descriptions (see references for attributes).
Term	<DT>	List item text.
Term Desc.	<DD>.. </DD>	Description of term may be text or graphic images.
Ordered List 	.. 	Defines a list of items that are marked by an ascending number or letter (see references for attributes).
Unordered List 	 	Defines a list of items marked by a bullet (see references for attributes).
List Item		Defines a member of an ordered or unordered list and can be either text or graphic images.

A list is defined by first specifying <DL>, or and then using the appropriate tags to define the list items.

For example :

Glossary List	Ordered List
<DL>	
<DT> text or graphic	 text or graphic
<DD> text or graphic </DD> etc.	 text or graphic etc.
</DL>	

List elements can be used singly or nested within each other. For example, an unordered list can be nested within an ordered or glossary list or vice versa. In addition, lists can be nested within other HTML elements (table for example). This leads to great versatility in formatting a document.

Tables

The Table HTML element is a very flexible formatting tool. Tables allow the document designer to divide the document into cells that can contain graphic images, anchors, numeric data, paragraphs etc.. In other words, all information contained within a document can be placed in specific areas of the document by a table. This can be extremely useful for lining up information or

segregating information into specific areas of a document.

The elements used to create tables are detailed below (example 1):

<i>Element</i>	<i>Tag</i>	<i>Description</i>
Table	<TABLE> .. </TABLE>	Specifies that the following information will be in table format. There are several format attributes available see references.
Table Row	<TR>.. </TR>	Specifies a row within a table (see references for attributes).
Table Header	<TH> ... </TH>	Labels a column or row by bolding text automatically (see references for attributes).
Table Data	<TD> .. </TD>	Contains the information (text or graphic) for one cell in a table (see references for attributes).

The number of rows within a table is established by then number of <TR></TR> combinations present within the <TABLE></TABLE> elements. Information within a row can consist of both “table header” and “table data” elements. The number of columns within a table is established by the number of <TD> </TD> groupings within each row.

Graphic Images

Graphics images within a HTML document can be in either GIF or JPEG graphic formats. GIF and JPEG files can be easily created within SAS (see example 1 and web references 30, 31, and 32). The tag used to specify a graphic reference is . Some of the attributes associated with IMG are : HEIGHT, WIDTH, USEMAP, ALIGN and ISMAP (a complete list and other information can be found within the references listed at the end of the paper). Height and width specify the size of the graphic in pixels. Align specifies alignment of the graphic with its neighboring text within a document (top, middle or bottom). USEMAP and ISMAP are used to divide a graphic image into sections and link other HTML documents or areas within a HTML document to a given section when clicked on by the mouse, i.e. create an “active” image. USEMAP differs from ISMAP in that the information needed to section and link the graphic image is within the HTML document. ISMAP indicates that the information required resides within a separate file.

Defining USEMAP and ISMAP coordinates:

1. Create graphic image.
2. Insert image into document using the IMG element.
3. Determine size of image using the WIDTH and HEIGHT attributes if necessary.

4. Add an anchor element (any name will do at this stage) and ISMAP to the IMG element. This will allow you to test whether your browser supports ISMAP. If your browser does support ISMAP then when the mouse pointer is passed over the graphic image numbers will appear on the bottom bar of the browser. These numbers are the coordinates you will need to define “clickable” areas of the graphic image (fig.1). The number of groups of x,y coordinates required depends upon the shape of the graphic image section.
5. Add the following information to your HTML document (example 1):
 - To IMG element - USEMAP = “#name of map section”
 - <MAP name = ”name of map section”>
<AREA SHAPE = RECT (only rectangle supported at present by USEMAP)
COORDINATES = “lx,ly,rx,ry”
HREF = “linked url”>
where : lx,ly = upper left hand coordinates
and rx,ry = lower right hand coordinates
 - </MAP>
 - case is important in both the USEMAP = and NAME = statements! You must use the same case for both or the browser will not recognize the map.
6. If using ISMAP (make sure your site has the appropriate set-up by checking with your web master) then change the HREF = “MAPFILE” to the appropriate path information for the mapping file. This file will contain the same information as the MAP element in the following order: shape url coordinates. More shapes are available than in USEMAP - rectangle, circle and polygon (see SAS Technical Support Web examples page for a good example of creating a US map image that is active).
7. Test both the USEMAP and ISMAP functionality

Creating graphic image for internet display in SAS:

1. Create graphic image using SAS/GRAPH® version 6.12. (see example 2 and web references 30-32).
2. Export graphic image using the IMGGIF, GIF, or IMGJPEG drivers provided with SAS/GRAPH version 6.12 (see example 2 and web references 30-32).

Note: If you are not using a full size image then create your own version of one of the above graphics drivers and set the size parameters to meet your size requirements (see example 2). This is a necessary step especially if you are using ISMAP.SAS program available from SAS Tech Support Web Page which creates the ISMAP file and the graphics image in one program. Otherwise your coordinates will be off.

Example 1. HTML Document

```

<HTML>
<BODY>
<CENTER> <IMG SRC="BARS.GIF" height=150
width=400 border=0 usemap="#map1">❶
<MAP NAME="map1">
<AREA SHAPE=rect href="diploid.html"
coords="56,24,202,137">
<AREA SHAPE=rect href="tetra.html"
coords="209,24,362,137">
</MAP>
</CENTER>
<H3> <CENTER> List Examples <CENTER> </H3>
<TABLE BORDER>❷
<TR>
<TH> Glossary List </TH>
<TH> Ordered List </TH>
<TH> Unordered List </TH>
</TR>
<TR> <TD><font size=2 color=green>
<DL>❸
<DT>BITSY
<DD>LEMON YELLOW SELF </DD>
<DT>CHESTER CYCLONE
<DD>LEMON YELLOW SELF, GREEN THROAT </DD>
<DT>OPEN HEARTH
<DD>RED & COOPER BITONE, RUBY HALO </DD>
<DT>ORIENTAL RUBY
<DD>CARMINE RED SELF </DD>
<DT>PRAIRIE BLUE EYES
<DD>LAVENDAR, NEAR BLUE EYEZONE </DD>
<DT>PRAIRIE MOONLIGHT
<DD>CREAMY YELLOW SELF, GREEN THROAT </DD>
<DT>STELLA D'ORO
<DD>GOLD SELF </DD>
</DL>
</TD>
<TD VALIGN="TOP"> <OL> <font size=2 color=blue>
<LI> Diploid
<OL>❹
<LI> BITSY
<LI> CHESTER CYCLONE
<LI> OPEN HEARTH
<LI> ORIENTAL RUBY
<LI> PRAIRIE BLUE EYES
<LI> PRAIRIE MOONLIGHT
<LI> STELLA D'ORO
</OL>
<LI> Tetraploid
<OL>
<LI> AMBER BALLERINA
<LI> BELOVED BALLERINA
<LI> CHICAGO RUBY
<LI> CHICAGO SUNRISE
<LI> HUDSON VALLEY
<LI> OLIVE BAILEY LANGDON
</OL>
</TD>
<TD VALIGN="TOP">
<UL>❺ <font size=2 color=red>
<LI> BITSY
<LI> CHESTER CYCLONE
<LI> OPEN HEARTH
<LI> ORIENTAL RUBY
<LI> PRAIRIE BLUE EYES
<LI> PRAIRIE MOONLIGHT
<LI> STELLA D'ORO
<LI> AMBER BALLERINA
<LI> BELOVED BALLERINA
<LI> CHICAGO RUBY
<LI> CHICAGO SUNRISE
<LI> HUDSON VALLEY
<LI> OLIVE BAILEY LANGDON
</UL>
</TD></TR>
</TABLE>
</BODY>
</HTML>

```

Example 1. SAS program that creates HTML document.

```

proc sort data=sasuser.daylilys out=daylily;
  by ploid;
run;

filename html 'c:\sugi\daylily.html';
data _null_;
  set daylily end=eof;
  file html ps=60 ls=132 n=ps;
  if _n_ = 1 then
    put @01 '<HTML>'
      /@01 '<BODY>'
      /@01 '<CENTER>'
      /@01 '<IMG SRC="BARS.GIF" height=150
width=400 border=0 usemap="#map1">'
      /@01 '<MAP NAME="map1">'
      /@01 '<AREA SHAPE=rect href="diploid.html"
coords="56,24,202,137">'
      /@01 '<AREA SHAPE=rect href="tetra.html"
coords="209,24,362,137">'
      /@01 '</MAP>'
      /@01 '<h3> Daylilies </h3>'
      /@01 '</CENTER>'
      /@01 '<TABLE BORDER>'
      /@01 '<TR>'
      /@01 '<TH> Glossary List </TH>'
      /@01 '<TH> Ordered List </TH>'
      /@01 '<TH> Unordered List </TH>'
      /@01 '</TR>'
      /@01 '<TR> <TD>'
      /@01 '<DL>'
      /@01 '<DT>' cultivar $20.
      /@01 '<DD>' color $36. +1 '</DD>';

  if _n_ ge 2 then
    put @01 '<DT>' cultivar $20.
      /@01 '<DD>' color $36. +1 '</DD>';

  if eof then put @01 '</DL> </TD>';
run;

data _null_;
  set daylily end=eof;
  by ploid;
  file html ps=60 ls=132 n=ps mod;

  if ploid='T' then ploidy='Tetraploid';
  if ploid='D' then ploidy='Diploid';
  if _n_ = 1 then put @01 '<TD VALIGN="TOP"><OL>';

  if first.ploid then
    put @01 '<LI>' +1 ploidy $20.
      /@01 '<OL>';
  put @01 '<LI>' +1 cultivar $20.;
  if last.ploid then put @01 '</OL>';

  if eof then put @01 '</TD>';
run;

data _null_;
  set daylily end=eof;
  by ploid;
  file html ps=60 ls=132 n=ps mod;

  if ploid='T' then ploidy='Tetraploid';
  if ploid='D' then ploidy='Diploid';

  if _n_ = 1 then put @01 '<TD VALIGN="TOP"><UL>';
  put @01 '<LI>' +1 cultivar $20.;

  if eof then put @01 '</UL>'
    /@01 '</TD></TR>'
    /@01 '</TABLE>'
    /@01 '</BODY>'
    /@01 '</HTML>';
run;

```

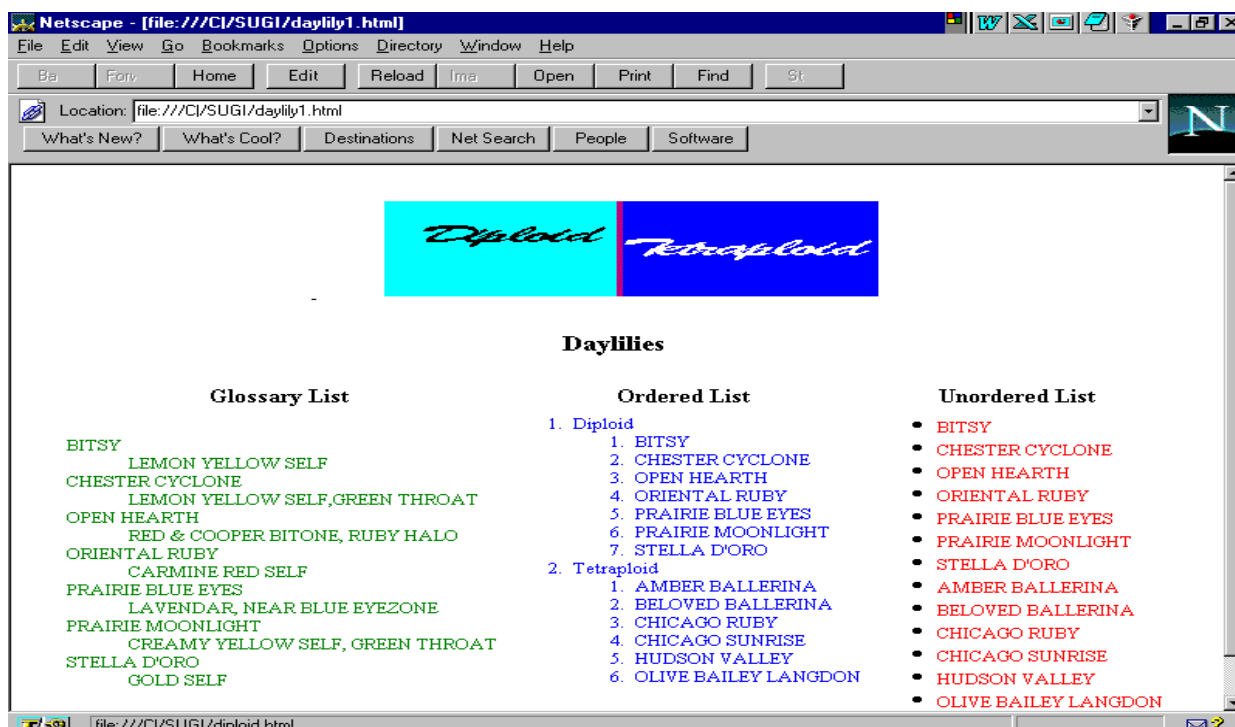


Fig 1. Web page created by HTML document in Example 1.

Example 2 - Creating your own graphics driver and image.

```
libname gdevice0 'c:\sugi';

proc gdevice c=gdevice0.devices nofs;
  copy imggif from=sashelp.devices
  newname=imggif;
run;

/* at this point you need to make appropriate
changes */

proc gdevice c=gdevice0.devices;
run;

proc sort data=sasuser.daylilys out=daylily;
  by ploidy;
run;

goptions dev=devices.imggif
gaccess="sasgastd>%unquote(c:\sugi\bars)";

proc gchart data=daylily;
  pattern1 v=solid c=blue;
  pattern2 v=solid c=red;
  axis1 label=none value=none;
  vbar ploidy / type=freq subgroup=ploidy nolegend
width=40 maxis=axis1 raxis=axis1;
run;
quit;
```

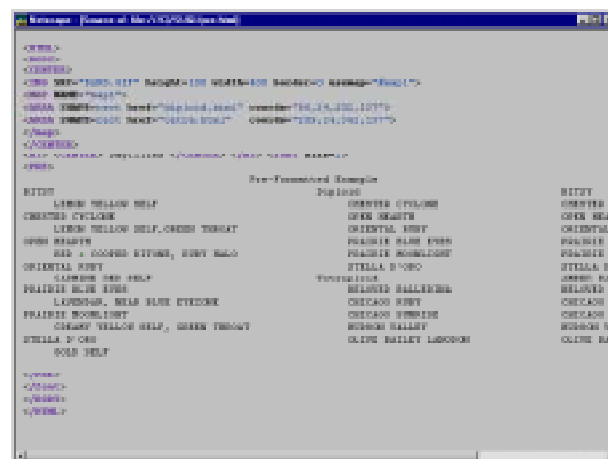
See SAS/GRAPH® - PROC GDEVICE for information on making modifications to and utilization of a personalized graphics driver³.

Pre-formatted Text

<PRE> element is used to mark a block of text to be displayed "as is". The main advantage of this element is that text that ordinarily would be recognized as HTML

tags or other control characters by the browser can be displayed. Another use of the <PRE> element is to format text on the HTML document page. Therefore, text set-up in columns is displayed without using a "table".

The SAS HTML Output Format Macro utilizes the <PRE> element to publish SAS Output as a browser document (see web reference 26). Example 3 shows how the <PRE> element can be used to create columns.



Example 3. HTML source.

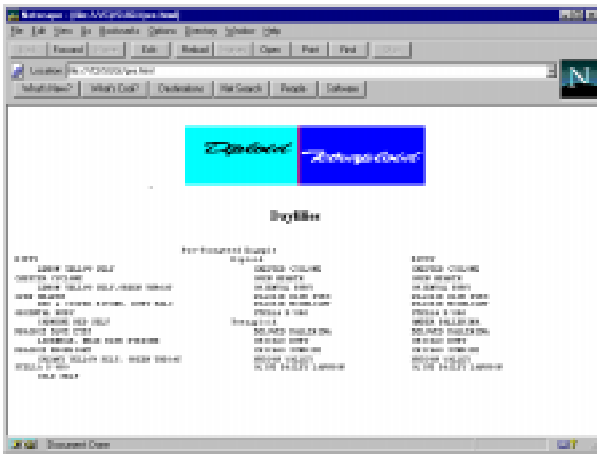


Fig 2. Web page create by HTML document.

Forms

HTML “form” elements can be used in conjunction with CGI (common gateway interface) and PERL scripts to allow user interaction with a web page¹. A good example of a form is the search engine interface available on the web (Web Crawler or Lycos). Typically, HTML “forms” can be used to enter information into a database or select options to create a query that can then be processed. An advantage of using a “form” is that through the PERL script SAS programs can be modified and run.

HTML elements used to create “forms” are listed below:

| <i>Element</i> | <i>Tag</i> | <i>Description</i> |
|----------------|-----------------------------|---|
| Form | <FORM>...
</FORM> | Used to designate the starting and ending of a form and where to send the information . |
| Input | <INPUT> | Creates a text-boxes, check boxes and radio buttons within a form. |
| Selection Box | <SELECT>..
</SELECT> | Creates a multi-line selection box or “pop-up box” . |
| | <OPTIONS>..
</OPTIONS> | Item within selection box. |
| Comment box | <TEXTAREA>..
</TEXTAREA> | Creates a text boxes that allows the user to input multiple lines of text. |

An extremely good reference for the use of “forms”, CGI and Perl scripts is Larry Hoyle’s paper from SUGI 23². In this paper Larry has an example of a Perl script, HTML document and SAS program interaction.

Frames

Frames allow the document developer to divide a web page into two or more individual areas that can be referenced individually and contain separate HTML documents within each section. For example, one portion of a document could contain a table of contents for your web application and the other portion the actual HTML document associated with a particular item within the table of contents. The major advantage to using

frames is the ability to keep the information within one frame section static while the contents of other sections can change. Therefore, to utilize frames a single web page will contain at least three separate: HTML documents; one HTML document containing the frame definition information and multiple HTML documents containing the contents for each frame defined.

HTML elements used to create frames are listed below:

<i>Element</i>	<i>Tag</i>	<i>Description</i>
Frame Definition	<FRAMESET> ... </FRAMESET>	Defines the number frames, whether the page is divided into rows or columns and their size.
Frame content	<FRAME>	Defines content of an individual frame section.
Frames n/a	<NOFRAME>... </NOFRAME>	Defines formatting when frames are not supported by browser.

The “driving” HTML document will contain the following information:

```
<HTML>
<FRAMESET COLUMNS=”20%,80%”>
  <FRAME NAME=”FRAME1” SRC = ‘cultivar.html’>
  <FRAME NAME=”FRAME2” SRC = ‘diploid.html’>
</FRAMESET>
</HTML>
```

where columns = size of the 2 frames created and
url = document to place in the frame.

To populate a specific frame section from an anchor in another frame section, the TARGET attribute within the anchor <A> tag must be specified. This attribute also can be used in other HTML elements, see Netscape FRAME Tutorial site for more information. In this case the CULTIVAR.HTML document points to the larger frame so that all linked documents will be displayed in FRAME2. Note: The Javascript guide is a great example of frame usage (see Web reference 20).

The <NOFRAME> tag is needed since frames are not supported by all browsers, the <NOFRAME> function is to tell the browser what to do if frames are not supported.

Dynamic HTML

“Dynamic” HTML is an evolving term that changes as the “technology” of the browser changes. Originally, a dynamic HTML page would display the current date or post the number of hits at a given web site. What encompasses “dynamic” HTML is function of the person defining it. Currently, “dynamic” HTML can be the combination of HTML and Javascript language to allow documents to change as the user interacts with the document. In other words, “dynamic” HTML in this instance refers to the information displayed by a

document changing as the user interacts with the document. Another typical use of “dynamic” HTML is data validation of form data, on the client side instead of the server. Javascript is embedded within the HTML document and allows a document developer to greatly extend the functionality of a document. For example, the current date can be displayed in a document by creating a short Javascript AND then accessing it within the HTML portion of a document. The reference for Javascript language is Netscape’s Javascript manual. Three caveats when using Javascript -

1. Case is critical unlike HTML which is NOT case sensitive and
2. Internet Explorer does not support all Javascript functionality.
3. Not all browsers versions support Javascript.

The Java script functionality that will be discussed are the event handlers: `onClick`; `onmouseover`; `onmouseout` and `onsubmit`. An event handler is a Javascript function that does not necessarily require a Javascript block of code to support it, for example `onClick`.

<i>Event Handler</i>	<i>Usage</i>
<code>onClick</code>	used to define what action takes place when a user clicks on a form element or link.
<code>onmouseover</code>	used to define what action takes place when a user moves the mouse pointer over a link.
<code>onmouseout</code>	used to define what action takes place when the mouse pointer is moved off of a link.
<code>onsubmit</code>	used in conjunction with <code>onClick</code> to define what action takes place when a user submits a form.

The following example contains both a Javascript and all the event handlers. The Java script is used to determine whether the required text had been entered. Fig. 3 illustrates the web page created.

```
<html>
<body>
<script>
var check=false
function checkCultivar (){
  if (document.form1.cultivar.value.length > 0) {
    return true}
  else {
    alert("Enter a Daylily Cultivar")
    return false}
}
</script>

<h2> Use of Java script Event Handlers </h2>
<br>
<p> <h4> Event handlers allow the document
developer to put some "smarts"
into the form section of a document. For
example:</h4> </p>
```

```
<form name="form1"
      onSubmit="alert('Continuing')">
<p> <b> Enter Daylily Cultivar </b>
<input type="text" name="cultivar" size=40></p>
<hr>
<h4> Example of onClick & onSubmit </h4>
<input type="button"
      name="cont" value="Continue"
      onClick="checkCultivar()" >
<input type="button" name="help" value="Help"
      onClick="msgWindow=
        window.open('help.html','window2',
          'resizable=no,width=300,height=400')">
</form>
<hr>
<h4> Example of onMouseOver - Status Bar
Information </h4>

<a href="event.html"
  onClick="msgWindow=
    window.open('help.html','window2',
      'resizable=no,width=300,height=400')"
  onMouseOver="window.status='Daylily Cultivar
List'; return true"
  onMouseOut="window.status='';return true">
<h4>HELP</h4>
</a>
</body>
</html>
```

The use of `onmouseover` in the above example is to put a message in the status line at the bottom of the window, i.e. this example is similar to the “mouse tip” in SAS/AF® Frame.

SAS, SAS/GRAPH, SAS/AF, and SAS/EIS are registered trademarks or trademarks of SAS Institute, Inc. in the USA and other countries. Netscape is a registered trademarks or trademarks of Netscape, Inc. in the USA and other countries. Microsoft Internet Explorer is a registered trademarks or trademarks of Microsoft, Inc. in the USA and other countries. ® indicates USA registration.

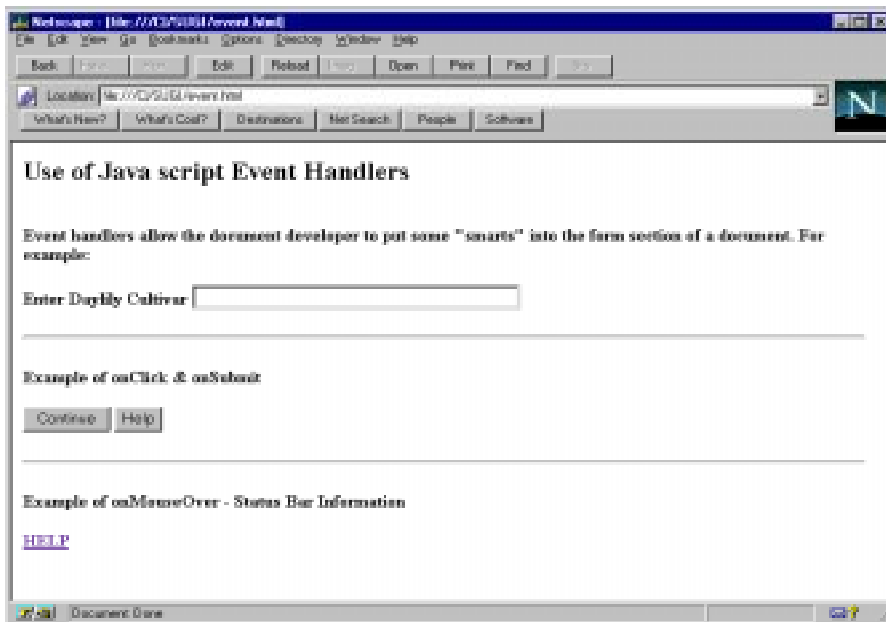


Fig. 3. Example of a Dynamic web page.

Resources and References

1. Graham, Ian S. (1996), HTML Sourcebook. A Complete Guide to HTML 3.0, Second Edition. New York, New York, John Wiley and Sons, Inc.
2. Hoyle, Larry (1997), Interfacing the SAS System with the World Wide Web. Proceedings of the Twenty-second SAS Users Group International Conference, 22:723-732.
3. SAS Institute Inc. (1990), SAS/GRAPH User's Guide, Version 6, Fourth Edition. Cary, NC, SAS Institute, Inc.



Warning screen that pops up if no information is entered in text box.



Help screen.

Web References

Below is a listing of web sites with information on HTML, Java Script Language and other general information about publishing on the web. All of these references have some interesting and useful material and point to other web sites of interest.

HTML

1. *HTML Reference Guide*
<http://developer.netscape.com/library/documentation/htmlguid/index.htm>
2. *Advanced HTML*
<http://www.geocities.com/SiliconValley/Park/7476/advancedhtml.htm>
3. *Index Dot HTML: THE Advanced HTML Reference*
<http://hepax6.rl.ac.uk/delphi/adye/blooberry/html/index.html>
4. *Wilbur - HTML 3.2*
<http://www.flirble.org/alister/docs/html3/index.html>
5. *Alphabetic sorted HTML Tag Listing*
<http://hepax6.rl.ac.uk/delphi/adye/blooberry/html/alphabet.htm>
6. *How do they do that with HTML?*
<http://www.nashville.net/~carl/htmlguide/index.html>
7. *THE Netscape Frames Tutorial*
<http://www.newbie.net/frames/3ed/>
8. *The HTML Goodies Home Page - Thanks for stopping by...*
<http://www.htmlgoodies.com/>
9. *Quad's Ultimate HTML Site - Your one stop Web Resource, period.*
<http://www.quadzilla.com/>
10. *Welcome to The HTML Station*
<http://www.december.com/html/>
11. *Yale C/AIM Web Style Guide*
<http://info.med.yale.edu/caim/manual/>
12. *The Ten Commandments of HTML FAQ*
<http://www.visdesigns.com/design/commandments.html>
13. *Willcam's Comprehensive HTML Cross Reference - "The Tag List"*
<http://www.willcam.com/cmat/html/tags.html>
14. *Dynamic HTML*
<http://www.stars.com/Authoring/HTML/Dynamic.html>

General Reference

15. *Netscape Gold Rush Tool Chest*
http://home.netscape.com/assist/net_sites/starter/samples/index.html
16. *Webreference.com (sm) - The Webmaster's Reference Library*
<http://www.webreference.com/>
17. *Web Review -- The Legion of DOM: Exploring the Document Object Model*
<http://webreview.com/>
18. *The Web Developer's Virtual Library*
<http://www.stars.com/>
19. *World Wide Web Tools and Guides for Authors*
<http://manuel.brad.ac.uk/WWW/authtool.html>

Java Script

20. *JavaScript Guide - Netscape's Javascript Language Manual*
<http://home.netscape.com/eng/mozilla/3.0/handbook/javascript/index.html>
21. *Cut-N-Paste JavaScript*
<http://www.infohiway.com/javascript/indexf.htm>
22. *The WDWL: Introduction to Authoring JavaScript*
<http://www.stars.com/Authoring/JavaScript/Intro/>
23. *FAQs, Technical Resources - NetscapeWorld*
<http://www.netscapeworld.com/netscapeworld/common/nw.jumps.html#javascript>

SAS

24. SAS Home Page
<http://www.sas.com>
25. SAS Research and Development Web Tools (Going Places with SAS Tools)
<http://www.sas.com/rnd/web/sitemap.html>
26. Output Formatter
<http://www.sas.com/rnd/web/format/out/index.html>
27. Data Set Formatter
<http://www.sas.com/rnd/web/format/ds/index.html>
28. Tabulate Formatter
<http://www.sas.com/rnd/web/format/tab/index.html>
29. Creating Graphics for the Web
<http://www.sas.com/rnd/web/publish.html#graph>
30. GIF
<http://www.sas.com/rnd/web/driver/GIF/GIF.html>
31. Animated GIF
<http://www.sas.com/rnd/web/driver/GIFANIM/GifAnim.html>
32. IMGJPEG
<http://www.sas.com/rnd/web/driver/JPEG/Jpeg.html>