Conditional Statements

Traffic Lighting

cellstyle expr-1 as style-element{style-attributes},
    expr-2 as style-element{style-attributes},
    ...
    expr-n as style-element{style-attributes};

The CELLSTYLE-AS statement is valid at
the top level of both PROC ODSLIST and
PROC ODSTEXT as well as within list and
item blocks. Each expression is a WHERE
expression that is evaluated for each item
or paragraph. If the result of the expression
is true, the given style element and style
attributes are applied. Once a matching
expression is found, execution stops.

Translating Values

translate expr-1a into expr-1b,
    expr-2a into expr-2b,
    ...
    expr-na into expr-nb;

The TRANSLATE-INTO statement is valid
at the top level of both PROC ODSLIST
and PROC ODSTEXT as well as within list
and item blocks. Each expression is a
WHERE expression. The (a) expressions
are evaluated for each item/paragraph. If
the result of the (a) expression is true, the
result of the (b) expression is used in place
of the item/paragraph value in the output.
Once a matching expression is found,
execution stops.

Example

/* static text */
proc odstext;
  p 'Class Information' /
    style=systemtitle;
  p 'The following is height, weight, and age information from the class data set.';
  run;

/* data dependent text / list */
proc odstext data=sashelp.class(obs=2);
  p 'Student name is ' || name || ' /' /
    style={fontweight=bold};
  list;
    item 'Height - ' || put(height,2.);
    item 'Weight - ' || put(weight,3.);
    item 'Height - ' || put(height,2.);
  end;
  run;

/* item block with nested list */
proc odslist data=sashelp.class(obs=2);
  item / style={liststyletype=none};
  p name || ''/age'' /
    style={fontweight=bold};
  list;
    cellstyle age > 13 as
      {fontstyle=italic};
    translate age = 13 into
      'Thirteen';
    item age / format=2.;
  end;
  end;
  run;

For complete information, refer to the Base SAS® documentation at
support.sas.com/base
PROC ODSLIST

**Basic Usage**

    proc odslist data=dataset;
     ... statements ... run;

**Procedure Options**

- `contents=“string”` specifies the string to be displayed in the table of contents for the output object.
- `data=“data-set”` specifies the data set to use for variable references in items and paragraphs. Data= is not needed if there are no variable references.
- `name=template-name` specifies the name of the template to store. This option enables you to store list templates for reuse with the DATA step.
- `pagebreak=yes | no | on | off` specifies whether or not the procedure should generate page breaks. The default is “no”.
- `store=template-store` specifies the template store to save the template in if name= is also used.
- `print` specifies that the output object should be printed. This is for use when name= is specified on static lists (i.e., lists with no variable references or data set).

**Item Block Statements**

- `list < /option(s) >;` specifies the content of the nested list. The list block statements are the same as for PROC ODSLIST (see above).

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>format=</td>
<td>SAS data format</td>
</tr>
<tr>
<td>style=</td>
<td>Style override</td>
</tr>
<tr>
<td>value=</td>
<td>Numeric value of bullet</td>
</tr>
</tbody>
</table>

**Common ODSLIST Statements**

- `item expression < /option(s) >;` specifies the content of the item.
- `item / < / options(s) >;` specifies a block of content for an item. This is used for using multiple paragraphs and/or creating nested lists.
- `end;` specifies a block of content for an item.

**PROC ODSTEXT**

**Basic Usage**

    proc odstext data=dataset;
     ... statements ... run;

**Procedure Options**

- `contents=“string”` specifies the string to be displayed in the table of contents for the output object.
- `data=“data-set”` specifies the data set to use for variable references in items and paragraphs. Data= is not needed if there are no variable references.
- `name=template-name` specifies the name of the template to store. This option enables you to store list templates for reuse with the DATA step.
- `pagebreak=yes | no | on | off` specifies whether or not the procedure should generate page breaks. The default is “no”.
- `store=template-store` specifies the template store to save the template in if name= is also used.
- `print` specifies that the output object should be printed. This is for use when name= is specified on static lists (i.e., lists with no variable references or data set).

**List Block Statements**

- `item expression < /option(s) >;` specifies the content of the item.
- `item / < / options(s) >;` specifies a block of content for an item. This is used for using multiple paragraphs and/or creating nested lists.
- `end;` specifies a block of content for an item.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>format=</td>
<td>SAS data format</td>
</tr>
<tr>
<td>style=</td>
<td>Style override</td>
</tr>
<tr>
<td>value=</td>
<td>Numeric value of bullet</td>
</tr>
</tbody>
</table>

**Common ODSTEXT Statements**

- `list < /option(s) >;` specifies the content of the nested list. The statements are the same as for PROC ODSLIST.