

Good Programming Practice – Coding Style Conventions



Overview

In order to be efficient and streamline the sharing of program code between programmers, with regulatory agencies, and with external partners or vendors, it is vital for code structure to follow standard conventions. We propose that these conventions be divided into those which should be considered as mandatory, and those which are merely suggestions to be followed as applicable.

Mandatory Conventions

The following summarizes standard programming conventions which should be followed in the creation of SAS programs.

- Use lowercase

It is easier to read

- Precede each data step and procedure with a comment
- Separate data steps and procedures with at least one blank line
- Use 'data=dataset' option in procedure statements so that the dataset being used is explicitly stated to ensure that the statement will work if it is moved to another location
- End data steps and procedures with run or quit to provide a boundary and allow for independent execution
- Split data steps into logical parts
- Put each statement on a separate line
- Left justify global statements and data and procedure statements and their corresponding run and quit statements
- Indent statements belonging to a level by at 2 to 5 columns, i.e. every nesting level should be visibly indented from the previous level
- Do not use tabs for indentation because they will display differently depending on the platform and text editor being used
- For do loops place the end statement in the same position as the do statement so that they can be easily matched
- Limit line length to the screen width so that the whole line can be viewed without scrolling

This should also prevent lines wrapping when the code is printed out, thus making it easier to read. No more than 80 characters wide is a good guideline for this.

- Use self-explanatory names for variables and datasets
- Keep naming conventions consistent across data sets and programs
- Insert parentheses in meaningful places in order to clarify the sequence in which mathematical or logical operations are performed
- When converting character variables to numeric or vice versa, use the put and input functions to explicitly convert the variable to ensure that it is done in the way intended and to avoid errors, warnings, and notes in the program log

Suggested Conventions

The following summarizes additional suggested programming conventions for SAS programs.

- Perform only one task per module or macro
- Use logical groupings to separate code into blocks
- Double space between sections
- Group similar statements together
- Define new variables with the attrib statement in order to ensure that the variable properties such as length, format, and label are correct instead of allowing them to be implicitly determined by the circumstances in which they are initialized in the code

Examples of Well Structured SAS Code

```
** Derive gender specific result **;
data two;
  set one;
  attrib result length=8.  label='Calculated Result'
         gender length=$6. label='Subject Gender'
         ;
  if sex='M' then do;
    gender='Male';
    result=(x/ym)*100;
  end;
  else if sex='F' then do;
    gender='Female';
    result=(x/yf)*100;
  end;
run;

** Print listing of all female subjects **;
proc print data=two(where=(sex='F'));
  variables patno result;
run;

** Create a dummy dataset with each subject and visit **;
data one;
  do patno=1 to 40; * cycle thru patients;
    do visit=1 to 3; * cycle thru visits;
      output;
    end; * cycle thru visits;
  end; * cycle thru patients;
run;
```

Text is available under the [Creative Commons Attribution-ShareAlike License](https://creativecommons.org/licenses/by-sa/4.0/) ; additional terms may apply. See Terms of use for details.