Programming Guide

Written by David Delene: 19 October 2013

Revised by David Delene: 17 November 2015, 14 December 2015, 9 July 2016, 12 August 2016, 23 August 2017

Title

File names should not include any special characters and please note that a "space" is a special character.

Header Comments

All source code should include header comment section at the start of each file that includes at a minimum: Name, Purpose, Syntax, Parameters, Execution Example, Modification History (with name, date, and summary of changes) and Copyright. Do not use tabs to indent code unless you provide the tab marks in a comment line. Code will not look correct unless others viewing the code using the same tab marks; hence, it is better to use spaces. A "Modification History" section should always be included. For the first version of the code include the date and author.

Syntax Output

All programs (scripts) are written so that execution information is provided via required and optional arguments given on the command line. Programs should never ask for input during execution since this type of coding make it more difficult to automate and combine scripts. Furthermore, configuration files should be avoided and simpler scripts used that only require a limited number of parameters.

Each program should provide the execution syntax when the program is executed with a -h (and/or –help) option and when an incorrect number of required parameters is given. The first, and sometimes only line, should start with "Syntax:" and provides the execution syntax. On the Syntax Line optional parameters are given within []. The Syntax Line may be followed by and "For Example:" line where an example of execution is provided. Further explanation of parameters definitions may be given on additional lines if a normal user would not understand things from the Syntax Line. Parameters need to be defined and not what it enables or provides (State what the parameter is, not how it would be used).

Coding Style

- **Lines of Code**
  1. Indent code that is within loops.
  2. Lines should be a maximum of 80 characters, however, words should not be split between lines.
  3. To keep lines from becoming too long, it is usual best to put comments above lines of code instead of at the end of lines of code.
  4. When a command spans two lines, indent second line so the lines has same indention as first line.
  5. Remove "commented out" lines of code from final program.
  6. Remove testing lines of code like printing out variables.
  7. Use a single blank line to separate sections of code, two blank lines can be use to separate major sections of code.
  8. Do not include additional blank lines of code at end of program.
- **Variables**

1. Define all parameters at start of program, not within.
2. Provide comment that give the unit of variables.
3. Use variables/constants when possible, particularly in middle of code, instead of numerical values.
4. Use descriptive variable names.
5. Use relative paths or environment variables for directory information. Do not use absolute paths.
6. While some languages allow it (for example IDL), don’t use different combinations of upper and lower cases for the same variables at different places in your program. Make it very difficult to search and find all occurrences of the variable.

- **Equations**
    1. Provide description/reference in comment line for all equations.
    2. Provide comment that describe unit of calculated parameter.
    3. Use same format for all equations, best to have space before and after '=' sign.
- **Loops**
    1. Some languages, like python, have defined structure (4 spaces) for indenting code lines.
    2. Use the same indentation for all loops and if-then statements.
    3. Use array, lists, and dictionaries within loops as to have code with few lines and easy to understand.
- **Comments**
    1. Provide comments that describe the design of the code.
    2. Provide reference for all equations used.
    3. Define units of variables.
    4. Use a space between comment symbol and start of sentence.
    5. Write comments that are complete sentences. Do not use all capitals in comments lines.
    6. Don't leave blank lines at end of code.
- **Library Usage**
    1. Import all libraries at start of program.
    2. Use library name or standard convection for libraries.

Coding Execution

- Provide sufficient information (using print lines) that the user knows what is going on while the code executes.
- File names should not have special characters and a “space” is a special character.
- Provide any data files required to conduct the example execution of the code.

Plot Creation

- Include axis labels that have unis.
- Use tick spaces and start/end intervals that are easy to understand, including the minor tick intervals.
- Use fonts that are large enough for insertion of the plot into a document or presentation.
- Do not use titles that just repeat what is on the x and y axis.