Python Docstring Conventions

Friday, February 06, 2015

9:41 AM

Wing fixed their PEP287 rendering in Source Assistant with version 5.1, which has unlocked several reST formatting options to us in our docstrings. With those options available, our docstring conventions have been updated to provide better readability in code, Wing's Source Assistant and Sphinx.

Here is a sample docstring as viewed in code in Wing:

Machine generated alternative text:
sone_function( first_arg, second_arg, third_arg ):
ciption
J*:.Arguments:. ,
hd E I line descriptions in the event of text that is lon than
:‘ ‘first_arg’ : ‘arg_type’ Description of the first argument.
second ar “: arg_type’ Description of the second argument with flTI tiple
a single line should be and it needs word wrapping. j is
text will have line breaks removed from it when dispE d in
Wing and Sphinx.
a list as part of its description.
• _arg: arg_type’ Description of the third argument, which -. ires
* List Item One
* List Item Two
Keyword Arguments:tt
None
Returns:tt
return_vaF return_typ& Description of what is being returJ
(‘*ExamP1es:
**
n___
some_function( first_arg, second_arg, third_arg )
*Author:** .
Author Name, author.name@dsvolition.com, mm/dd/Š —
_______________________________ri_________________________________________________

And here is how it will look in the Source Assistant within Wing:

Machine generated alternative text:
J Defining: some_function
some_tunction(first_arg, second_arg, third_arg)
V
Descriptionoftheclassormethod. rA
Arguments:
j Iirst_arg: ai-g_type Description oftheflrstargument.
.J second_arg:arg_type Description of the second argument with
multiple line descriptions in the event of textthat is
longer than a single line should be and it needs word
- wrapping. This text will have line breaks removed from
itwhen displayed in Wing and Sphinx.
third_arg: ai-g_type Description of the third argument, which
requires a list as part of its description.
I .
. ListltemTwo
Keyword Arguments:
None
Returns:
return_val:return_type Description of what is being returned.
Examples:
some_function( first_arq, aecond_arg, third_arg )
Author
Author Name, author.nameedsvolition.com. mmiddiyyyy

Note the easily discernable section headers, argument lists and definitions.

Follow these basic rules:

* **Every line** is indented with **only tabs** (NO SPACES), which will indent the sections properly. Mixed tabs/spaces causes PEP formatting issues.
* Argument names are surrounded by colon and double-back-ticks.
  + Double-back-ticks make the name become an inline literal, which will display it in a fixed-width font in our docs and in a different color from the rest of the line or the section titles in Wing.
  + Colons make it a field list, which will align the descriptions of each section together (each section like Arguments, Keyword Arguments, etc. gets its own alignment).
  + There ***must*** be some white-space between the last colon and the beginning of the description. It is advised that this white-space is tabs and that all descriptions are tabbed to the same tab-stop for readability right in the code.
  + Multiple lines can be utilized to make the description more readable in the code, simply indent each subsequent line past the indentation of the field name. For readability, it is recommended the same tab-stop as the beginning of the description. Any line breaks added here will be removed when the doc string is converted so that the description displays word-wrapped appropriately in its intended display.
  + Lists can be included here by inserting a blank line between the description and the first item on the list, then adding a list indicator (usually asterisk). Can be useful for describing multiple entries in a collection
* If an argument takes a required type, that type should be listed as the first word of the argument description and surrounded in single-back-ticks. This emphasizes the type in italics and once Wing implements proper hyperlinking in their doc strings, this will also provide a link to that type (useful for home-brewed class types).
* A blank line between the section title and the section content. This provides extra spacing between the titles and content in Source Assist and unifies the look of all section titles, since the Examples section *requires* a blank line in order to display the example code properly.
* If a section has no values, list only the word "None" (with no text-decoration) so that the section visually fades away.

If you’re interested in other features that reST provides for nice-looking docstrings, read up on [PEP-287](https://www.python.org/dev/peps/pep-0287/) and take a look at a [reStructuredText Primer](http://sphinx-doc.org/rest.html).

**FOR YOUR CONVENIENCE:**

Docstring snippets can be inserted into your Python code very easily by using auto-complete in Wing. Typing a capital V will start the auto-complete and automatically highlight the code snippet that is appropriate for the code-level that your cursor is currently at (module, class or method). This means you can type V and then press Tab and instantly have a doc-string that will take only a matter of seconds to fill out, but save everyone who looks at your code (including your future self) hours of head-scratching from trying to decipher what in the world you were doing.