python documentation style Cheat Sheet

by shakhand via cheatography.com/33748/cs/10529/

Part1

```
# -- coding: utf-8 --
"""Example Google style docstrings.
This module demonstrates documentation as specified by the `Google Python
Style Guide`_. Docstrings may extend over multiple lines. Sections are created
with a section header and a colon followed by a block of indented text.
Example:
    Examples can be given using either the `Example or Examples`
    sections. Sections support any reStructuredText formatting, including
    literal blocks::
        $ python example_google.py
Section breaks are created by resuming unindented text. Section breaks
are also implicitly created anytime a new section starts.
Attributes:
    module_level_variable1 (int): Module level variables may be documented in
        either the `Attributes` section of the module docstring, or in an
        inline docstring immediately following the variable.
        Either form is acceptable, but the two should not be mixed. Choose
        one convention to document module level variables and be consistent
        with it.
Todo:
    * For module TODOs
    * You have to also use `sphinx.ext.todo` extension
.. _Google Python Style Guide:
   http://google.github.io/styleguide/pyguide.html
"""
```

Part2

```
module_level_variable1 = 12345
module_level_variable2 = 98765
"""int: Module level variable documented inline.
The docstring may span multiple lines. The type may optionally be specified
on the first line, separated by a colon.
"""
def function_with_types_in_docstring(param1, param2):
    """Example function with types documented in the docstring.
    PEP 484_ type annotations are supported. If attribute, parameter, and
    return types are annotated according to PEP 484_, they do not need to be
    included in the docstring:
    Args:
        param1 (int): The first parameter.
        param2 (str): The second parameter.
    Returns:
        bool: The return value. True for success, False otherwise.
    .. _PEP 484:
        https://www.python.org/dev/peps/pep-0484/
    """
def function_with_pep484_type_annotations(param1: int, param2: str) -> bool:
    """Example function with PEP 484 type annotations.
    Args:
        param1: The first parameter.
        param2: The second parameter.
    Returns:
        The return value. True for success, False otherwise.
    """
```