2022-05-23

May 23, 2022 3:56 PM

- Comments are descriptions that help programmers better understand the intent and functionality of the program.
- In Python, we use the hash symbol # to write a single-line comment.
- If we do not assign strings to any variable, they act as comments.
- We use triple quotation marks for multi-line strings.
- docstrings are strings used right after the definition of a function, method, class, or module.
- Even though they are single-lined, we still use the triple quotes around these docstrings as they can be expanded easily later.
- The closing quotes are on the same line as the opening quotes.
- There's no blank line either before or after the docstring.
- They should not be descriptive, rather they must follow "Do this, return that" structure ending with a period.
- Multi-line docstrings consist of a summary line just like a one-line docstring, followed by a

blank line, followed by a more elaborate description.

- The docstrings for Python Modules should list all the available classes, functions, objects and exceptions that are imported when the module is imported.
- They should also have a one-line summary for each item.
- The docstring for a function or method should summarize its behavior and document its arguments and return values.
- It should also list all the exceptions that can be raised and other optional arguments.
- we have included a short description of what the function does, the parameter it takes in and the value it returns.
- The docstrings for classes should summarize its behavior and list the public methods and instance variables.
- The subclasses, constructors, and methods should each have their own docstrings.
- We can also use the help() function to read the docstrings associated with various objects.
- The docstrings for Python script should document the script's functions and command-line

syntax as a usable message.
- It should serve as a quick reference to all the functions and arguments.
- The docstrings for a Python package is written in the package's __init__.py file.
- It should contain all the available modules and sub-packages exported by the package.
- Anatomy of docstring
  - Descriotion of what funcition does
  - Description of arguments if any
  - Description of return value if any
  - Description of errors raised if any
  - Optional extra notes or examples of usage
- Docstring formats
  - Google Style
  - Numpydoc
  - ReStructuredText
  - EpyText
- Google style example
  -
  -

```
def function(arg_1, arg_2=42):
  """Description of what the function does.

  Args:
    arg_1 (str): Description of arg_1 that can break onto the next line
      if needed.
    arg_2 (int, optional): Write optional when an argument has a default
      value.

  Returns:
    bool: Optional description of the return value
    Extra lines are not indented.

  Raises:
    ValueError: Include any error types that the function intentionally
      raises.

  Notes:
    See https://www.datacamp.com/community/tutorials/docstrings-python
```

```
    for more info.
    """
```

- Numpydoc example
  -
  -

```
def function(arg_1, arg_2=42):
    """
    Description of what the function does.

    Parameters
    ----------
    arg_1 : expected type of arg_1
        Description of arg_1.
    arg_2 : int, optional
        Write optional when an argument has a default value.
        Default=42.

    Returns
    -------
    The type of the return value
        Can include a description of the return value.
        Replace "Returns" with "Yields" if this function is a generator.
    """
```

https://github.com/suhasid098/docstring