
pydocstring Documentation

Release 0.2.1

Alisdair Robertson

Apr 06, 2022

Contents:

1	CLI	3
2	API	5
3	Indices and tables	7
	Python Module Index	9
	Index	11

Using pydocstring is really simple, just check out the CLI and API.

CHAPTER 1

CLI

pydocstring CLI provides entrypoints for CLI commands

Currently only the `pydocstring` command is supported

To use `pydocstring` from the command line you call `pydocstring` with the source, and optionally the position of the cursor within that source (defaults to the end).

Note that `pydocstring` doesn't insert docstrings in place (yet), as it's designed for editor integration. It prints out the generated docstring for the scope the given cursor position is in.

You may also want to provide the `-f` flag with the formatter you want to use.

```
usage: pydocstring [-h] [-f {google,numpy,reST}] [--version] source [position]

positional arguments:
source                  Source code to process, or the path to a file
position               Position of the cursor in the document, defaults to
                        the end. Row, then column

optional arguments:
-h, --help              show this help message and exit
-f {google,numpy,reST}, --formatter {google,numpy,reST}
                        docstring formatter to use
--version               show program's version number and exit
```


If you're looking to use this in an editor you're probably looking for `pydocstring.generate_docstring()`

<code>pydocstring</code>	Top level API - all you need to integrate with an editor, just call <code>pydocstring.generate_docstring()</code> with the source code and the position of your cursor.
<code>pydocstring.exc</code>	pydocstring exceptions
<code>pydocstring.formatters.google</code>	
<code>pydocstring.formatters.numpy</code>	
<code>pydocstring.formatters.reST</code>	
<code>pydocstring.cli</code>	pydocstring CLI provides entrypoints for CLI commands

2.1 pydocstring

Top level API - all you need to integrate with an editor, just call `pydocstring.generate_docstring()` with the source code and the position of your cursor.

```
pydocstring.generate_docstring(source, position=(1, 0), formatter='google', autocomplete=False)
```

Generate a docstring

Parameters

- **source** (*str*) – the text of the source
- **position** (*tuple*) – the position of the cursor in the source, row, column. Rows start at 1 Columns start at 0
- **formatter** (*str*) – the format of the docstring choose from google, numpy, reST.
- **autocomplete** (*bool*) – Whether or not to remove three characters from before the position prior to parsing the code. This is to remove the “""" before a docstring default:

False

Raises `exc.InvalidFormatter` – If the value provided to *formatter* is not a supported formatter name

Returns docstring, excluding quotation marks, or None, if one could not be generated

Return type str or None

2.2 pydocstring.exc

pydocstring exceptions

exception `pydocstring.exc.FailedToGenerateDocstringError`
Could not generate a docstring for the given code and position

exception `pydocstring.exc.InvalidFormatterError`
Provided name of formatter is not one supported

2.3 pydocstring.cli

pydocstring CLI provides entrypoints for CLI commands

Currently only the `pydocstring` command is supported

To use pydocstring from the command line you call `pydocstring` with the source, and optionally the position of the cursor within that source (defaults to the end).

Note that pydocstring doesn't insert docstrings in place (yet), as it's designed for editor integration. It prints out the generated docstring for the scope the given cursor position is in.

You may also want to provide the `-f` flag with the formatter you want to use.

```
usage: pydocstring [-h] [-f {google,numpy,reST}] [--version] source [position]

positional arguments:
  source                Source code to process, or the path to a file
  position              Position of the cursor in the document, defaults to
                        the end. Row, then column

optional arguments:
  -h, --help            show this help message and exit
  -f {google,numpy,reST}, --formatter {google,numpy,reST}
                        docstring formatter to use
  --version             show program's version number and exit
```

`pydocstring.cli.main()`
CLI entrypoint

CHAPTER 3

Indices and tables

- `genindex`
- `modindex`
- `search`

p

- `pydocstring`, 5
- `pydocstring.cli`, 3
- `pydocstring.exc`, 6

F

`FailedToGenerateDocstringError`, 6

G

`generate_docstring()` (*in module pydocstring*), 5

I

`InvalidFormatterError`, 6

M

`main()` (*in module pydocstring.cli*), 6

P

`pydocstring` (*module*), 5

`pydocstring.cli` (*module*), 3, 6

`pydocstring.exc` (*module*), 6