

---

# **pydocstring Documentation**

***Release 0.2.1***

**Alisdair Robertson**

**Nov 30, 2019**



---

## Contents:

---

<b>1</b>	<b>CLI</b>	<b>3</b>
<b>2</b>	<b>API</b>	<b>5</b>
<b>3</b>	<b>Indices and tables</b>	<b>9</b>
	<b>Python Module Index</b>	<b>11</b>
	<b>Index</b>	<b>13</b>



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>	Google Docstring Formatter
<code>pydocstring.formatters.numpy</code>	Numpy Docstring Formatter
<code>pydocstring.formatters.reST</code>	reST Docstring Formatter
<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.formatters.google

Google Docstring Formatter

`pydocstring.formatters.google.class_docstring` (*parso\_class*)  
Format a google docstring for a class

Only documents attributes, `__init__` method args can be documented on the `__init__` method

**Parameters** `parso_class` (*Class*) – The class tree node

**Returns** The formatted docstring

**Return type** str

`pydocstring.formatters.google.function_docstring` (*parso\_function*)  
Format a google docstring for a function

**Parameters** `parso_function` (*Function*) – The function tree node

**Returns** The formatted docstring

**Return type** str

`pydocstring.formatters.google.module_docstring` (*parso\_module*)  
Format a google docstring for a module

Only documents attributes, `__init__` method args can be documented on the `__init__` method

**Parameters** `parso_module` (*Module*) – The module tree node

**Returns** The formatted docstring

**Return type** str

## 2.4 pydocstring.formatters.numpy

Numpy Docstring Formatter

`pydocstring.formatters.numpy.class_docstring(parso_class)`

Format a numpy docstring for a class

Only documents attributes, `__init__` method args can be documented on the `__init__` method

**Parameters** `parso_class` (*Class*) – The class tree node

**Returns** The formatted docstring

**Return type** `str`

`pydocstring.formatters.numpy.function_docstring(parso_function)`

Format a numpy docstring for a function

**Parameters** `parso_function` (*Function*) – The function tree node

**Returns** The formatted docstring

**Return type** `str`

`pydocstring.formatters.numpy.module_docstring(parso_module)`

Format a numpy docstring for a module

Only documents attributes, `__init__` method args can be documented on the `__init__` method

**Parameters** `parso_module` (*Module*) – The module tree node

**Returns** The formatted docstring

**Return type** `str`

## 2.5 pydocstring.formatters.reST

reST Docstring Formatter

`pydocstring.formatters.reST.class_docstring(parso_class)`

Format a reST docstring for a class

Only documents attributes, `__init__` method args can be documented on the `__init__` method

**Parameters** `parso_class` (*Class*) – The class tree node

**Returns** The formatted docstring

**Return type** `str`

`pydocstring.formatters.reST.function_docstring(parso_function)`

Format a reST docstring for a function

**Parameters** `parso_function` (*Function*) – The function tree node

**Returns** The formatted docstring

**Return type** `str`

`pydocstring.formatters.reST.module_docstring(parso_module)`

Format a reST docstring for a module

Only documents attributes, `__init__` method args can be documented on the `__init__` method

**Parameters** `parso_module` (*Module*) – The module tree node

**Returns** The formatted docstring

**Return type** `str`

## 2.6 pydocstring.cli

pydocstring CLI provides endpoints 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 endpoint
```

## CHAPTER 3

---

### Indices and tables

---

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



### p

- `pydocstring`, 5
- `pydocstring.cli`, 3
- `pydocstring.exc`, 6
- `pydocstring.formatters.google`, 6
- `pydocstring.formatters.numpy`, 6
- `pydocstring.formatters.reST`, 7





## C

`class_docstring()` (in module `pydocstring.formatters.google`), 6  
`class_docstring()` (in module `pydocstring.formatters.numpy`), 6  
`class_docstring()` (in module `pydocstring.formatters.reST`), 7

## F

`FailedToGenerateDocstringError`, 6  
`function_docstring()` (in module `pydocstring.formatters.google`), 6  
`function_docstring()` (in module `pydocstring.formatters.numpy`), 7  
`function_docstring()` (in module `pydocstring.formatters.reST`), 7

## G

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

## I

`InvalidFormatterError`, 6

## M

`main()` (in module `pydocstring.cli`), 8  
`module_docstring()` (in module `pydocstring.formatters.google`), 6  
`module_docstring()` (in module `pydocstring.formatters.numpy`), 7  
`module_docstring()` (in module `pydocstring.formatters.reST`), 7

## P

`pydocstring(module)`, 5  
`pydocstring.cli(module)`, 3, 8  
`pydocstring.exc(module)`, 6  
`pydocstring.formatters.google(module)`, 6  
`pydocstring.formatters.numpy(module)`, 6  
`pydocstring.formatters.reST(module)`, 7