

# python注释风格

---

## VsCode插件

---

[autoDocstring - Python Docstring Generator - Visual Studio Marketplace](#)

vscode中的autoDocstring插件提供了这几种风格

docblockr、pep257、google、google-notypes、sphinx、sphinx-notpyes、numpy、numpy-notypes、one-line-sphinx

安装完成后进入设置搜索Auto Docstring

Docstring Format选项选择注释风格

## PyCharm

---

PyCharm中提供了这几种风格

Epytext、reStructuredText、Numpy、Google

设置位置

File -> Settings -> Tools -> Python Integrated Tools -> Docstrings -> Docstring format

PyCharm默认不开启参数类型注释，开启类型的设置位置

File -> Settings -> Editor -> General -> Smart Keys -> Insert type placeholders in the documentation comment stub

其中，autoDocstring的sphinx 和 pycharm的reStructuredText有些像

## 注释的自动生成

---

直接在类、方法内的第一行敲三个双引号Enter即可

## 注释风格的选择

---

考虑到算法开发常用的IDE：VsCode和PyCharm，建议使用numpy、sphinx/reStructuredText、google这三种中的一种

## 常见的注释风格

---

## docblockr-autoDocstring

```
1 class PyClass(object):
2     """_summary_
3
4     Arguments:
5         object {_type_} -- _description_
6     """
7
8     def __init__(self) -> None:
9         """_summary_
10        """
11        pass
12
13    def PyMethod(self, parm1, parm2=True) -> str:
14        """_summary_
15
16        Arguments:
17            parm1 {_type_} -- _description_
18
19        Keyword Arguments:
20            parm2 {bool} -- _description_ (default: {True})
21
22        Returns:
23            str -- _description_
24        """
25        pass
```

## pep257-autoDocstring

```
1 class PyClass(object):
2     """_summary_
3
4     Arguments:
5         object -- _description_
6     """
7
8     def __init__(self) -> None:
9         """_summary_
10        """
11        pass
12
13    def PyMethod(self, parm1, parm2=True) -> str:
14        """_summary_
```

```

15
16     Arguments:
17         parm1 -- _description_
18
19     Keyword Arguments:
20         parm2 -- _description_ (default: {True})
21
22     Returns:
23         _description_
24     """
25     pass

```

## google-autoDocstring

```

1  class PyClass(object):
2      """_summary_
3
4      Args:
5          object (_type_): _description_
6      """
7
8      def __init__(self) -> None:
9          """_summary_
10         """
11         pass
12
13     def PyMethod(self, parm1, parm2=True) -> str:
14         """_summary_
15
16         Args:
17             parm1 (_type_): _description_
18             parm2 (bool, optional): _description_. Defaults to
19             True.
20
21         Returns:
22             str: _description_
23         """
24         pass

```

## google-notypes-autoDocstring

```

1  class PyClass(object):
2      """_summary_
3

```

```

4      Args:
5          object: _description_
6      """
7
8      def __init__(self) -> None:
9          """_summary_
10         """
11         pass
12
13      def PyMethod(self, parm1, parm2=True) -> str:
14          """_summary_
15
16          Args:
17              parm1: _description_
18              parm2: _description_. Defaults to True.
19
20          Returns:
21              _description_
22          """
23          pass

```

## sphinx-autoDocstring

```

1  class PyClass(object):
2      """_summary_
3
4      :param object: _description_
5      :type object: _type_
6      """
7
8      def __init__(self) -> None:
9          """_summary_
10         """
11         pass
12
13      def PyMethod(self, parm1, parm2=True) -> str:
14          """_summary_
15
16          :param parm1: _description_
17          :type parm1: _type_
18          :param parm2: _description_, defaults to True
19          :type parm2: bool, optional
20          :return: _description_
21          :rtype: str

```

```

22     """
23     pass

```

## sphinx-notpyes-autoDocstring

```

1  class PyClass(object):
2      """_summary_
3
4      :param object: _description_
5      """
6
7      def __init__(self) -> None:
8          """_summary_
9          """
10         pass
11
12     def PyMethod(self, parm1, parm2=True) -> str:
13         """_summary_
14
15         :param parm1: _description_
16         :param parm2: _description_, defaults to True
17         :return: _description_
18         """
19         pass

```

## numpy-autoDocstring

```

1  class PyClass(object):
2      """_summary_
3
4      Parameters
5      -----
6      object : _type_
7              _description_
8      """
9
10     def __init__(self) -> None:
11         """_summary_
12         """
13         pass
14
15     def PyMethod(self, parm1, parm2=True) -> str:
16         """_summary_
17

```

```

18         Parameters
19         -----
20         parm1 : _type_
21             _description_
22         parm2 : bool, optional
23             _description_, by default True
24
25         Returns
26         -----
27         str
28             _description_
29         """
30         pass

```

## numpy-notypes-autoDocstring

```

1  class PyClass(object):
2      """_summary_
3
4      Parameters
5      -----
6      object
7          _description_
8      """
9
10     def __init__(self) -> None:
11         """_summary_
12         """
13         pass
14
15     def PyMethod(self, parm1, parm2=True) -> str:
16         """_summary_
17
18         Parameters
19         -----
20         parm1
21             _description_
22         parm2, optional
23             _description_, by default True
24
25         Returns
26         -----
27         _description_
28         """

```

## one-line-sphinx-autoDocstring

```

1  class PyClass(object):
2      """_summary_
3
4      :param _type_ object: _description_
5      """
6
7      def __init__(self) -> None:
8          """_summary_
9          """
10         pass
11
12     def PyMethod(self, parm1, parm2=True) -> str:
13         """_summary_
14
15         :param _type_ parm1: _description_
16         :param bool parm2: _description_, defaults to True
17         :return str: _description_
18         """
19         pass

```

## epytext-PyCharm

```

1  class PyClass(object):
2      """
3
4      """
5
6      def __init__(self) -> None:
7          """
8
9          """
10         pass
11
12     def PyMethod(self, parm1, parm2=True) -> str:
13         """
14
15         @param parm1:
16         @type parm1:
17         @param parm2:
18         @type parm2:

```

```
19         @return:
20         @rtype:
21         """
22         pass
```

## reStructuredText-PyCharm

```
1  class PyClass(object):
2      """
3
4      """
5
6      def __init__(self) -> None:
7          """
8
9          """
10         pass
11
12     def PyMethod(self, parm1, parm2=True) -> str:
13         """
14
15         :param parm1:
16         :type parm1:
17         :param parm2:
18         :type parm2:
19         :return:
20         :rtype:
21         """
22         pass
```

## numpy-PyCharm

```
1  class PyClass(object):
2      """
3
4      """
5
6      def __init__(self) -> None:
7          """
8
9          """
10         pass
11
12     def PyMethod(self, parm1, parm2=True) -> str:
```



```

13         """
14
15         Parameters
16         -----
17         parm1 :
18         parm2 :
19
20         Returns
21         -----
22
23         """
24         pass

```

## google- PyCharm

```

1  class PyClass(object):
2      """
3
4      """
5
6      def __init__(self) -> None:
7          """
8
9          """
10         pass
11
12     def PyMethod(self, parm1, parm2=True) -> str:
13         """
14
15         Args:
16             parm1 ():
17             parm2 ():
18
19         Returns:
20
21         """
22         pass

```