

Python文档资源

获取Python的帮助信息？

注释 文件中的文档

dir函数 对象中可用属性的列表

文档字符串:**doc** 附加在对象上的文件中的文档

Pydoc Help函数 对象的交互帮助

Pydoc HTML报表 浏览器中的文档模块

标准手册 一些Python标准库的官方文档

网站资源 在线教程 例子等

出版的书籍 商业参考书籍

通常来说文档在程序员之间的优先级都偏低, 在一般情况下能在代码中看到一些注释都已经是很幸运的事情了.

1. 通过#号来注释 Python会忽略#号后面的所有的文字, 可以在一行语句之后插入一些对阅读代码有益处的文字说明.
2. dir函数是抓取对象内所有可用属性列表的方式.

```
In [1]: l = [1, 2, 3]
dir(l)
```

```
Out[1]: ['__add__',
'__class__',
'__contains__',
'__delattr__',
'__delitem__',
'__delslice__',
'__doc__',
'__eq__',
'__format__',
'__ge__',
'__getattribute__',
'__getitem__',
'__getslice__',
'__gt__',
'__hash__',
'__iadd__',
'__imul__',
'__init__',
'__iter__',
'__le__',
'__len__',
'__lt__',
'__mul__',
'__ne__',
'__new__',
'__reduce__',
'__reduce_ex__',
'__repr__',
'__reversed__',
'__rmul__',
'__setattr__',
'__setitem__',
'__setslice__',
'__sizeof__',
'__str__',
'__subclasshook__',
'append',
'count',
'extend',
'index',
'insert',
'pop',
'remove',
'reverse',
'sort']
```

3. 文档字符串 通过成对的"""将多行注释放在类和函数的顶端, 这个注释会保留在 **doc**属性中以供查看.

```
In [4]: #示例代码1. 文档字符串

x = 100

def f1():
    """
    this is function1
    """
    return 'hello'

print f1.__doc__

class c1:
    """
    this is class c1
    """
    def f1(self):
        """
        this is class c1 function f1
        """
        return 'hello'

print c1.__doc__
print c1.f1.__doc__
```

this is function1

this is class c1

this is class c1 function f1

4. 或者通过help()函数查看

In [5]:

help(f1)

Help on function f1 in module __main__:

f1()

this is function1

In [6]:

help(c1)

Help on class c1 in module __main__:

class c1

| this is class c1

|

| Methods defined here:

|

| f1(self)

| this is class c1 function f1

In [7]:

help(c1.f1)

Help on method f1 in module __main__:

f1(self) unbound __main__.c1 method

this is class c1 function f1

In []: