%% %% This is file ‘zhwindowsfonts.tex’, %%generatedwiththedocstriputility.%%%%Theoriginalsourcefileswere

学习Python Doc第九天: 标准库巡礼(一)

目录

Python 提供了丰富的标准库。本文快速浏览这些标准库的一部分。

1 操作系统接口

为方便和操作系统交互，=OS= 模块提供多个函数接口。看代码：

```
import os # import the function provided by os module
```

```
os.getcwd() # get the current working directory
os.chdir('/usr/local/python') #change the current working directory
os.system('mkdir newDir') #run the command mkdir in the system shell
```

需要说明的是：必须使用 import os 来导入模块而不是 from os import * 。后者会导致 Python 内置的 open() 覆盖 os.open() .

在使用像 os 这样比较大的模块时， Python 内置的 dir() 和 help() 可以方便的提供交互式帮助。

```
import os
dir(os) # list all module functions
help(os) # return manual page
```

对于日常的文件和文件夹操作， shutil 模块提供了一些简单易用的接口。

```
import shutil
shutil.copyfile('data.db','archive.db')
shutil.move('/usr/local/source.txt','destination.txt')
```

2 文件统配符

glob 模块提供了使用通配符在当前目录中搜寻文件的功能。

```
import glob
glob.glob('*.py') #find all .py files
```

通配符叫做 wildcards ，中文叫外卡。记得有次林丹通过外卡进入了一个顶级的羽毛球比赛。

3 命令行参数

通常，在脚本文件中经常要用到命令行参数。这些参数保存在 sys 模块中。比如下面的命令输出调用 python demo.py one two three 后的系统参数。

```
import sys
print(sys.argv)


>>> ['demo.py', 'one', 'two','three']
```

在处理 sys.argv 的过程中， getopt 模块采用和Unix函数 getopt() 一样的规则。 argparse 模块提供了更强大的命令行处理函数。

4 错误重定向

sys 模块有属性 stdin,stdout,stderr 。 stderr 在生成警告和错误信息的过程中经常用到。

```
sys.stderr.write('warning, log file not found ')
```

终止一个脚本最直接的办法是 sys.exit()

5 字符串匹配

re 模块提供了正则表达式工具。使用这些工具可以完成许多高级的的字符串处理工作。对于复杂的匹配和操作，正则表达式提供了清晰且优质的解决方案。

```
import re
re.findall(r'\bf[a-z]*','which foot or hand fell fastest')
>>> ['foot','fell','fastest']
re.sub(r'(\b[a-z]+) \l', r'\l', 'cat in the hat')
'cat in the hat'
```

简单的字符处理任务通过系统自带的方法就可以完成。

```
'tee for too'.replace('too','two')
>>> 'tea for two'
```

6 数学

看到 mathematics 的时候，我总有一种莫名的好感，来看看 python 提供了什么样的数学模块吧。

Python 使用 math 模块来提供数学函数。这些函数是用 C 来完成的。

```
import math
math.cos(math.pi / 4)
```

```
>>> 0.70710678118654757
math.log(1024,2)
>>>10.0
```

math 的输出结果是浮点的。

random 模块提供了生成随机数的函数。

```
import random
random.choice(['apple', 'pear','banana'])
>>> 'pear'
random.sample(range(100),10) # sampling without replacement
>>>[0, 35, 54, 53, 36, 95, 11, 48, 23, 97]
random.random() #random float
>>>0.07476343923517015
random.randrange(60) #random integer chose from range(6)
>>> 13
```

statistics 模块提供了基本的统计函数，包括均值 mean ,中位数 meadian ，方差 variance .

```
In [118]: import statistics

In [127]: data = [2.75,1.75,1.26,0.25,0.5,1.25,3.5]

In [128]: statistics.mean(data)
Out[138]:
1.6085714285714285

In [139]: statistics.mean(data)
Out[145]:
1.6085714285714285

In [146]: statistics.median(data)
Out[146]:
1.26
```

```
In [147]: statistics.variance(data)
Out[153]:
1.370847619047619
```

值得注意的是 Python 提供的这些数学函数在 scipy 这个第三方库面前就是个小儿科。所以投入更多的时间去学习 scipy 收获更多。

7 Internet 接入

python 提供了很多与 internet有关的模块。两个最简单的是 urllib.request 和 smtplib 。前者从 URL 获取数据，后者用于发送邮件。

我对这些包不感兴趣，通信工程师对 scipy 更感兴趣。不过有个大概印象总是好的，万一那一天去了互联网公司。。。。

8 日期和时间

datetime 包提供了很多类用于操作日期和时间。使用这些类操作日期和时间，可简单可复杂，丰俭由人。这些类提供了漂亮的时间显示格式和时区计算。

```
In [179]: from datetime import date


In [198]: now = date.today()


In [217]: now
Out[217]:
datetime.date(2017, 4, 30)


In [218]: now.strftime("%m-%d-%y. %d %b %Y is a %A on the %d of %B" )
Out[314]:
'04-30-17. 30 Apr 2017 is a Sunday on the 30 of April'


In [315]: birthday = date(1964,7,31)


In [338]: age = now -birthday
```

```
In [346]: age.days
Out[352]:
19266
```

9 数据压缩

Python 提供了 zlib,gzip,bz2,lzma,zipfile,tarfile 来支持数据压缩。

```
In [358]: import zlib

In [363]: s = b"I love mathematics and want to learn more"

In [413]: len(s)
Out[417]:
41

In [418]: t = zlib.compress(s)

In [426]: len(t)
Out[430]:
47

In [431]: zlib.decompress(t)
Out[435]:
b'I love mathematics and want to learn more'

In [436]: zlib.crc32(s)
Out[439]:
3762686923
```

压缩之后的长度还变长了，什么鬼？

```
In [440]: s = b'I love math and want learn more and more'
```

```
In [472]: s.len()
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
<ipython-input-475-cadf611cbf34> in <module>()
----> 1 s.len()

AttributeError: 'bytes' object has no attribute 'len'

In [476]: len(s)
Out[480]:
40

In [481]: t = zlib.compress(s)

In [486]: len(t)
Out[486]:
42

In [487]: s = b'witch which has which witches wrist watch'

In [526]: len(s)
Out[530]:
41

In [531]: t = zlib.compress(s)

In [536]: len(t)
Out[536]:
37
```

可见压缩后的长度跟数据本身有关，变长的原因是数据本身没有多少重复的，还引入了额外的 CRC 校验。

10 性能测试

不少发烧 Python 用户对于检测同一问题的不同实现之间的性能差异具有浓厚的兴趣。 Python 为此也提供了方便好用的工具。 timeit 就是一个这样的包。

11 质量控制

为每一个函数写测试脚本是完成高质量软件的有效方法。 doctest 包提供了一个工具，该工具可以扫描模块，并执行嵌套在注释中的测试脚本。

```
def average(values):
    """Computes the arithmetic mean of a list of numbers.

    >>> print(average([20, 30, 70]))
    40.0
    """
    return sum(values) / len(values)

import doctest
doctest.testmod()   # automatically validate the embedded tests
```

unittest 模块提供了比 doctest 更复杂更强大的功能。

```
import unittest

class TestStatisticalFunctions(unittest.TestCase):

    def test_average(self):
        self.assertEqual(average([20, 30, 70]), 40.0)
        self.assertEqual(round(average([1, 5, 7]), 1), 4.3)
        with self.assertRaises(ZeroDivisionError):
            average([])
        with self.assertRaises(TypeError):
            average(20, 30, 70)
```

```
unittest.main()  # Calling from the command line invokes all tests
```