



科学编程基础

4. 原生模块

余恒

北师大天文系

模块

- Python的脚本：扩展名为py的文本文件，
 - 可以单独运行，也可以导入另一个脚本中运行。
 - 当脚本被导入运行时，我们将其称为模块（`module`）。模块是Python组织代码的基本方式。
- 模块名与脚本的文件名相同，例如我们编写了一个名为Items.py的脚本，则可在另外一个脚本中用 `import Items` 语句来导入它。
- 在导入时，Python解释器会先在脚本当前目录下查找，如果没有则在 `sys.path` 包含的路径中查找。

系统内置模块

- **os** : 对操作系统服务提供了可移植的 (portable) 的接口
- **sys**: 包括跟python解析器和环境相关的变量和函数
- **decimal**: python中的float使用双精度的二进制浮点编码来表示的, 这种编码导致了小数不能被精确的表示, 例如0.1实际上内存中为0.10000000000000000001, 还有 $3*0.1 == 0.3$ 为False.
decimal就是为了解决类似的问题的, 拥有更高的精确度, 能表示更大范围的数字, 更精确地四舍五入。
- **math**: 定义了标准的数学方法, 例如 $\cos(x)$, $\sin(x)$ 等。
- **random**: 随机数生成
- **string**: 包含大量处理字符串的函数
- **io**: 实现了各种IO形式和内置的open()函数
- **datetime**: 时间日期相关函数
- **timeit**: 计时函数, 调试相关

科学计算常用模块

- [Numpy](#) – Arrays manipulation library 科学计算的必装模块，几乎所有的其他科学模块都依赖于它。
- [Scipy](#) – 扩展的科学计算模块。
- [PyGSL](#) C/C++语言中著名的科学计算函数库GNU Scientific Library (GSL)的python版
- [Matplotlib](#) – 高质量的2D作图模块，足以替代GNUPlot
- [Mayavi](#): 强大的三维作图模块，属于EPD公司套件的一部分
- [Sympy](#): 符号计算模块
- [StatLib](#): 统计学工具箱
- [Escript/Finley](#): 偏微分方程求解
- [Parallel Python](#): 并行计算模块

模块导入方式

- 使用相关函数时需要指定模块名
 - `import sys, os` # 同时导入多个模块
 - `import math as m` # 导入同时给缩写
- 函数直接使用，不加模块名
 - `from math import cos, sin` # 导入具体函数
 - `from socket import *` # 导入模块中全部函数
- Python解释器在第一次import一个py文件的时候，会尝试将其编译为字节码文件，扩展名为.pyc。后边的导入操作会直接读取.pyc文件而不是.py文件，通常速度会更快。

模块属性

- 模块所包含的对象和方法可以通过 `dir(模块名)` 查看,
- 具体的用法和解释可以通过 `help(模块名.函数名)` 查看
- 比如:

```
>>> import math
```

```
>>> dir(math)
```

```
['__doc__', '__name__', '__package__', 'acos',  
'acosh', 'asin', 'asinh', 'atan', 'atan2', 'atanh',  
'ceil', 'copysign', 'cos', 'cosh', 'degrees', 'e',  
'erf', 'erfc', 'exp', 'expm1', 'fabs', 'factorial',  
'floor', 'fmod', 'frexp', 'fsum', 'gamma', 'hypot',  
'isinf', 'isnan', 'ldexp', 'lgamma', 'log', 'log10',  
'log1p', 'modf', 'pi', 'pow', 'radians', 'sin',  
'sinh', 'sqrt', 'tan', 'tanh', 'trunc']
```

```
>>> help(math.modf)
```

函数帮助

```
>>> help(math.log)
```

```
Help on built-in function log in module math:
```

```
log(...)
```

```
    log(x[, base])
```

Return the logarithm of x to the given base.

If the base not specified, returns the natural logarithm (base e) of x.

OS模块

- `os.name`: 获取当前体系平台，Windows下返回"`nt`"，Linux下返回"`posix`"。
- `os.linesep`: 获取当前平台应用的行终止符。Windows下返回"`\r\n`"，Linux应用"`\n`"。
- `os.getcwd()`: 获取当前工作目录，即当前python脚本工作的目录路径。
- `os.listdir(path)`: 返回指定目录下的所有文件和目录名。
- `os.path.split()`: 分离路径中的目录名和文件名。
- `os.path.exists()`: 检查路径是否存在。
- `os.path.isfile()`: 判断是否为文件

测试

- `import os`
- `print os.getcwd()`
- `print os.listdir("D:\\")`
- `print os.path.exists("D:\\test")`

decimal模块

- python 中浮点数默认提供16位精度，无法精确存储十进制数。十进制数学计算模块decimal 可以精确处理十进制。

```
>>> a="%.20f" %(1/3.)
```

```
>>> a="%.20f" %(2.645)
```

```
>>> a="%.20f" %(2.675)
```

```
>>> round(2.675,2)
```

```
>>> print 0.1+0.1+0.1-0.3
```

```
5.55111512313e-17
```

```
>>> from decimal import Decimal as D
```

```
>>> D('0.1') + D('0.1') + D('0.1') - D('0.3')
```

```
Decimal("0.0")
```

decimal精度控制

```
>>> from decimal import Decimal as D
```

```
>>> from decimal import getcontext
```

```
>>> getcontext().prec
```

```
28
```

```
>>> D(1)/D(7)
```

```
Decimal('0.1428571428571428571428571429')
```

```
>>> getcontext().prec=6
```

```
>>> D(1)/D(7)
```

```
Decimal('0.142857')
```

```
>>> D(str(1.12))/D(7)
```

SYS模块

- `sys.argv` 获取命令行传入参数。
 - `sys.argv[0]` 为脚本名， `sys.argv[1]` 为第一个参数
- `sys.getdefaultencoding()`: 获取当前终端编码，一般默认为ascii。
- `sys.getfilesystemencoding()`: 获取文件系统编码，Windows下返回"`mbcs`"，mac下返回"`utf-8`"。
- `sys.path`: 命令搜索路径。

参数传递练习

- 打印命令行输入参数
- 如: `python test.py abs a 123`

```
import sys
n = len(sys.argv)
if n > 1:
    print "Your input is: ", sys.argv[1:]
else:
    print "There is no input"
```

数学模块

```
>>> import math
```

```
>>> math.pi
```

```
>>> math.e
```

```
>>> math.sin(2*math.pi/180)
```

```
>>> math.fmod(9.8,4.0)
```

- `hypot(x,y)` 从原点 $(0,0)$ 到 (x,y) 的直线距离
- `radians(x)` 角度数转弧度
- 复数运算可使用 `cmath` 模块

练习

- 编写函数计算 n 的阶乘
- 然后调用脚本根据命令行的输入计算阶乘
- 核心代码：

```
def jc(n):  
    resu = 1  
    for i in range(n+1):  
        resu = i * resu  
    return resu  
  
jc(5)
```

string模块

- `s="hello world"`
- `string.split(s)`
- `string.capitalize(s)`
- `string.capwords(s)`
- `string.upper(s)`
- `string.count(s, "o")`
- `string.find(s, "o")`
- `string.center(s, 20)`
- `string.ljust(s, 20)`
- `string.rjust(s, 20)`

random模块

```
>>> import random
>>> random.randint(0, 99)           #随机整数
>>> random.randrange(0, 101, 2)     #随机偶数
>>> random.random()                 #随机浮点数
>>> random.uniform(1, 10)           #均匀分布
>>> random.choice('?abc&%^*f')      #随机字符
>>> random.sample('abcdefghij', 3)
>>> items = [1, 2, 3, 4, 5, 6]
>>> random.shuffle(items)           #洗牌
```

datetime 模块

- `date`: 表示日期的类; `time`: 表示时间的类; `datetime`: 表示日期时间; `timedelta`: 表示时间间隔, `tzinfo`: 与时区有关的相关信息
- `import datetime as dt`
- `print 'date.max:', dt.date.max`
- `print 'date.min:', dt.date.min`
- `print 'date.today():', dt.date.today()`
- `dt.date.isoweekday(dt.date.today())`
- `dt.date.today()+dt.timedelta(7)`
- `print 'Time:', dt.time(12,5,4)`
- `dt.datetime.now()`

时间与字符串

- `from datetime import datetime as dt`
- `dt.now()`
- `dt.strftime(dt.now(), '%c')`
- `d = dt.now() - dt.strptime('2015/12/20 20:56:30', '%Y/%m/%d %H:%M:%S')`
- `(d.days, d.seconds, d.microseconds)`
- %a 星期的简写, %A 星期的全称
- %b 月份的简写, %B 月份的全称
- %I: 小时 (12小时制), %W 周数 (星期一起算)

练习：会合时间

- 由于各个行星公转的角速度不尽相同，因此相对位置时刻在变化。行星之间重复出现相同相对位置的周期就称为会合周期（synodic period）
- 已知2014年4月9日发生了火星冲日
- 求下一次火星冲是什么时候？



timeit模块

- `import timeit as ti`
- `t = ti.Timer('x=range(100)')`
- `t.timeit()`
- `t.timeit(100)`
- `ti.timeit('x=range(100)', number=100)`
- `t.repeat(3, 20000)`

作业

- 编写猜数字游戏：
- 程序想一个数（随机整数），用户在命令行猜测数字，程序返回数字大小，用户根据比较结果重新猜测，直至猜出数字。
- 作业名称： 学号_1.py
- 交作业地址（FlashFXP）：
- ftp:202.112.85.96 端口 7712
- 用户名 python 密码 bnuaastro