



科学编程基础

1. 背景和数据类型

余恒

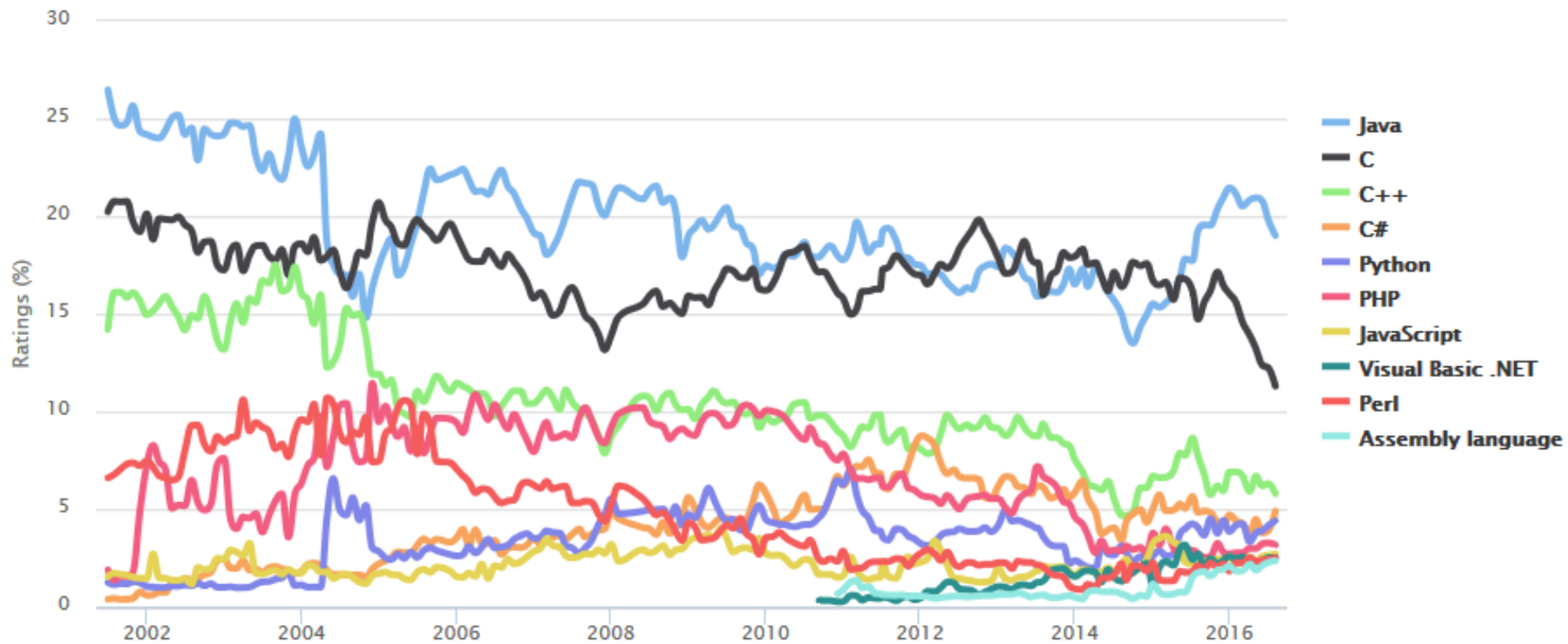
北师大天文系

编程语言排行榜

Aug 2016	Aug 2015	Programming Language	Ratings	Change
1	1	Java	19.010%	-0.26%
2	2	C	11.303%	-3.43%
3	3	C++	5.800%	-1.94%
4	4	C#	4.907%	+0.07%
5	5	Python	4.404%	+0.34%
6	7	PHP	3.173%	+0.44%
7	9	JavaScript	2.705%	+0.54%
8	8	Visual Basic .NET	2.518%	-0.19%
9	10	Perl	2.511%	+0.39%
10	12	Assembly language	2.364%	+0.60%

TIOBE Programming Community Index

Source: www.tiobe.com



课程语言排行

IDL

Programming La...

Fortran

Programming La...

MATLAB

Computer progr...

Bash

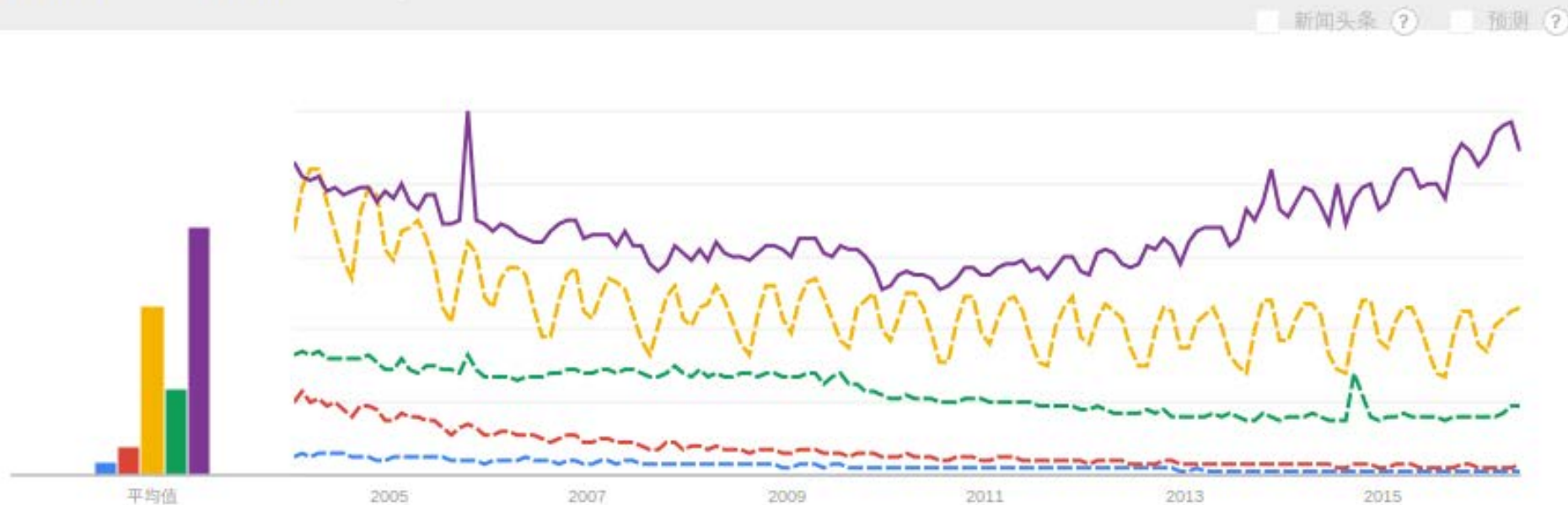
Software

Python

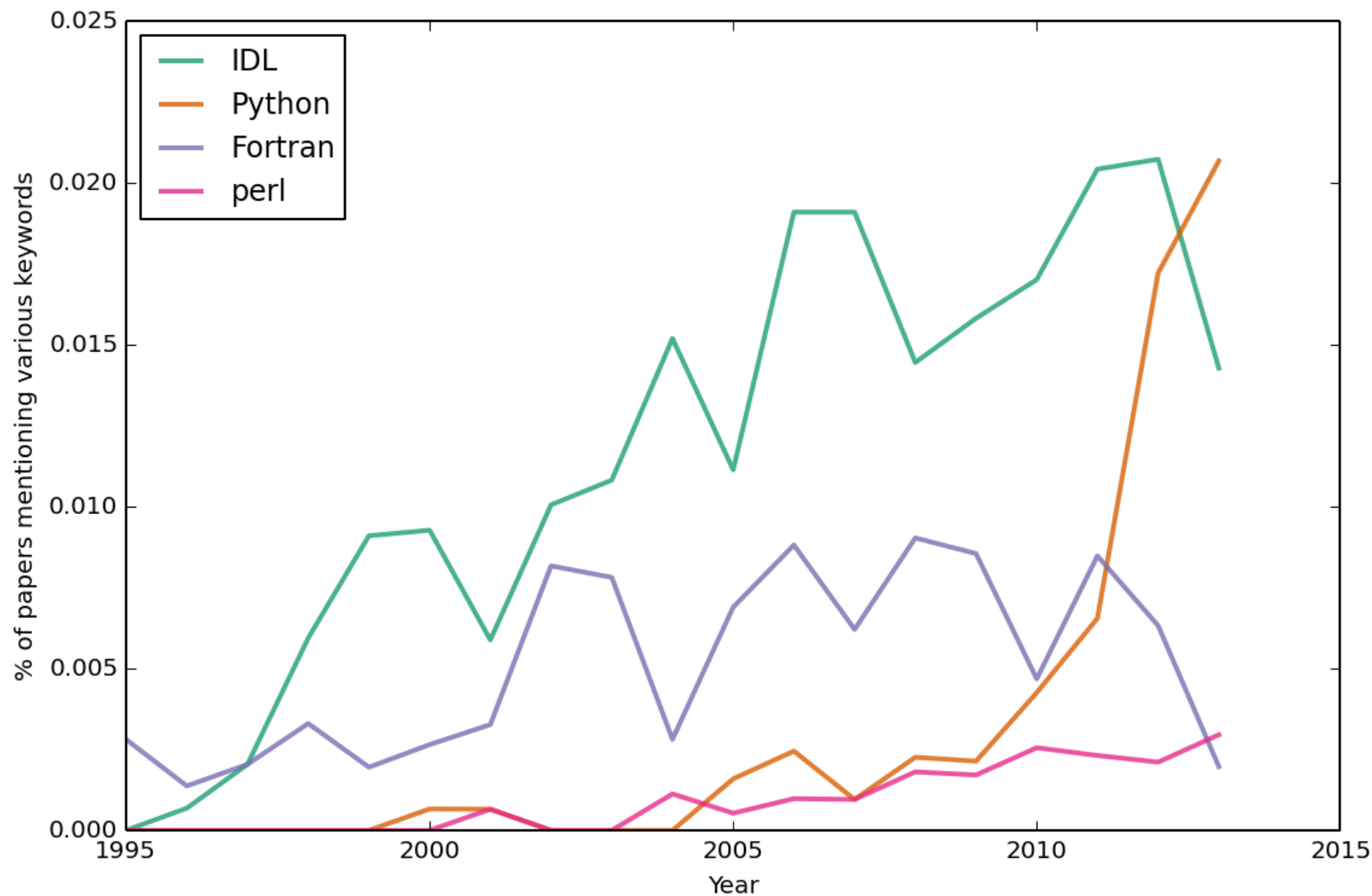
搜索字词

测试版：衡量主题的搜索热度是一项测试版功能。可快速对总体搜索热度进行准确衡量。要衡量特定查询的搜索热度，请选择“搜索字词”选项。

热度随时间变化的趋势



天文界普及情况



什么是 pythonTM

- Python是一种开源的面向对象的脚本语言。
- 1989年，CWI（阿姆斯特丹国家数学和计算机科学研究所）的研究员Guido van Rossum为开展自己的研究工作开发了一种新型的脚本编程语言。
- 他把这种新的语言命名为Python（大蟒蛇），来源于BBC当时热播的喜剧连续剧"Monty Python"。
- Python于1991年初公开发行，由于功能强大且采用开源方式发行，Python发展迅速。

为什么是 pythonTM

- 优点
 - 软件质量高，最好的可读性
 - 开发效率高，录入代码少
 - 程序可移植性，跨平台
 - 模块扩展方便
 -
- 缺点：
 - 运行速度不够快

- 教材：
 - 《Python学习手册》 第四版， 2011
- 学习材料：
 - 《Python科学计算》 张若愚， 第二版， 2012
 - 《Dive Into Python》 第二版中文版， 2010
 - 《Core Python Programming》 2nd， 2006
- 常用链接
 - <http://www.pythonchallenge.com/>
 - <http://challenge.greplin.com/>
 - <http://www.stackoverflow.com>

教学资源

- 网站
 - [Http://202.112.85.96/python](http://202.112.85.96/python)
- 系FTP（用FlashFXP连接）
 - 202.112.85.101:7712
 - 用户名: bnuaastro 密码: astronomy
 - 程序所在目录:
 - /Soft/Soft/Pub2/Science/Python/
 - 讲义目录:
 - /Incoming/astro_incoming/teacher_incoming/yuheng/



安装



- 官方安装包：仅提供核心功能，需要自行安装其它相关模块。 <http://www.python.com>
- **Canopy** (EPD, Enthought Python Distribution)：美国 Enthought 公司提供的Python套装，模块齐全，尤其增强了三维绘图功能，教育版免费。 <http://enthought.com/>
- **Anaconda**：针对数据分析和科学计算的套装，开源免费，包括2.x 和3.x 系列
- 其他版本包括： WinPython, Python(x,y)

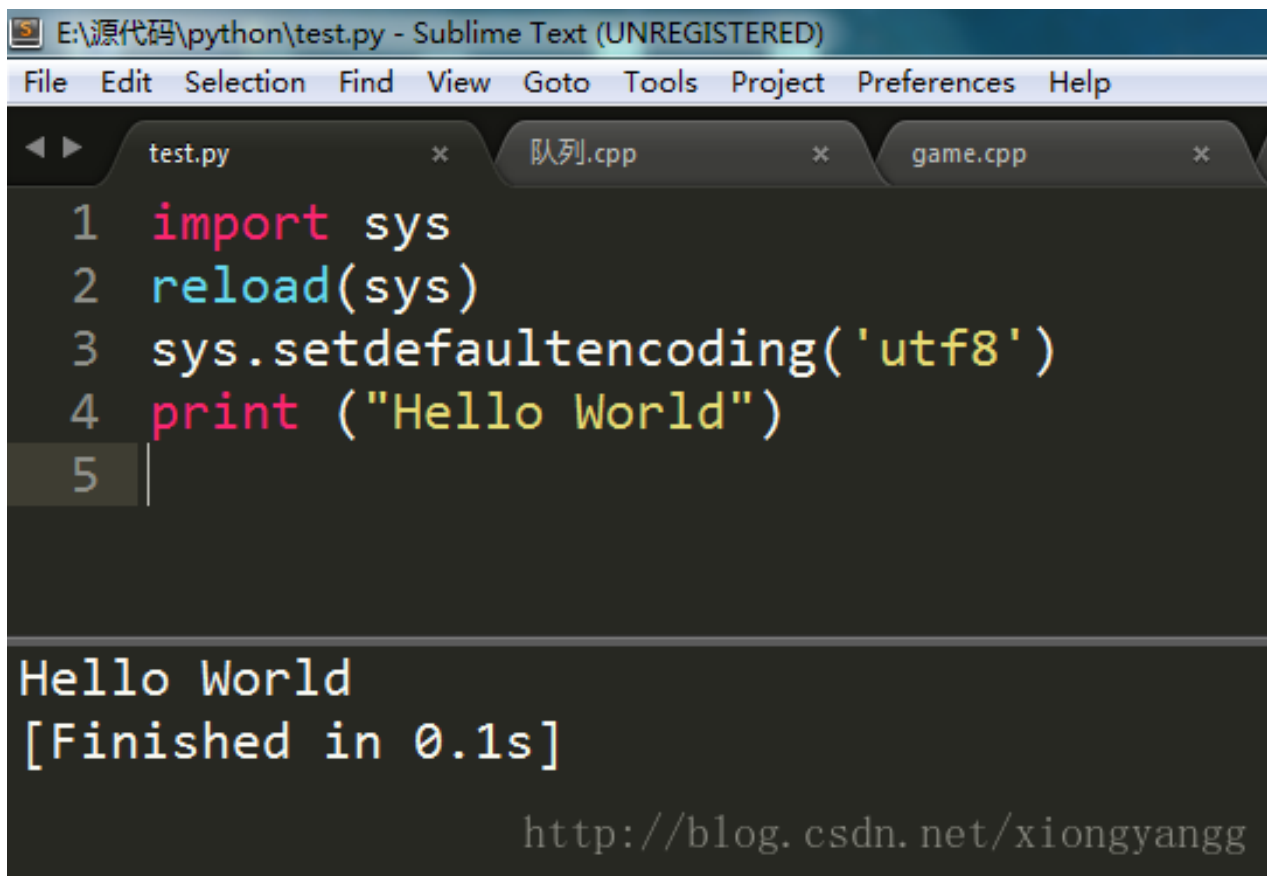
注意：目前Python有 2. x 和3. x 两个系列，语法不完全兼容，主流仍为2. x。

运行环境

- 命令行
 - python shell, 自带的输入环境
 - ipython, 增强的交互式shell输入环境
 - Jupyter Notebook, 灵活的浏览器编辑环境
- 开发编辑器(IDE)
 - IDLE: 免费编译环境
 - PyCharm : 完整的商业编译平台
 - Eclipse: 专业编译调试环境

编辑器

- 基本要求：
 - 语法高亮
 - 空白字符
 - 等宽字体
 - 代码运行
- 典型代表：
 - Emeditor
 - Sublime
 - Notepad++



The screenshot shows the Sublime Text editor interface. The title bar indicates the file path is `E:\源代码\python\test.py` and the application is `Sublime Text (UNREGISTERED)`. The menu bar includes `File`, `Edit`, `Selection`, `Find`, `View`, `Goto`, `Tools`, `Project`, `Preferences`, and `Help`. The editor has three tabs open: `test.py`, `队列.cpp`, and `game.cpp`. The `test.py` tab is active, showing the following Python code:

```
1 import sys
2 reload(sys)
3 sys.setdefaultencoding('utf8')
4 print ("Hello World")
5
```

Below the code editor, the output of the script is displayed:

```
Hello World
[Finished in 0.1s]
```

A URL is visible at the bottom right of the output area: <http://blog.csdn.net/xiongyangg>

等宽字体

- 宋体
 - 00123456789abcdefghijklmnopqrstuvwxyz
- 黑体
 - 00123456789abcdefghijklmnopqrstuvwxyz
- Arial
 - O0123456789abcdefghijklmnopqrstuvwxyz
- Consolas
 - 00123456789abcdefghijklmnopqrstuvwxyz
- Courier New
 - 00123456789abcdefghijklmnopqrstuvwxyz

范例展示

开始 -> 运行-> cmd -> python

开始 -> 程序-> Enthought -> IDLE

```
>>> print "Hello, world"
```

```
Hello, world
```

```
>>> x = 12**2
```

```
>>> x/2
```

```
72
```

```
>>> # this is a comment
```

```
>>> 2**100
```

```
1267650600228229401496703205376L
```

```
>>> exit()
```

运行方式总结

- 交互式：
 - 命令行：终端启动 `python`
 - 增强命令行：终端启动 `ipython`
- 独占式：

脚本：文本编辑器写脚本，命令行运行

```
python xxx.py
```
- 综合式： `notebook`

代码风格

- 在Python中，每行程序以换行符代表结束
- 如果一行程序太长的话，可以用"\n"符号扩展到下一行。
- 在Python中以空格缩进来区分程序功能块。
- 如果一行中有多条语句，语句间要以分号（;）分隔。
- “#”号之后的内容为注释。

```
print "Hello, world"  
x/2 # this is a comment  
2**100; 3*5
```


基本概念

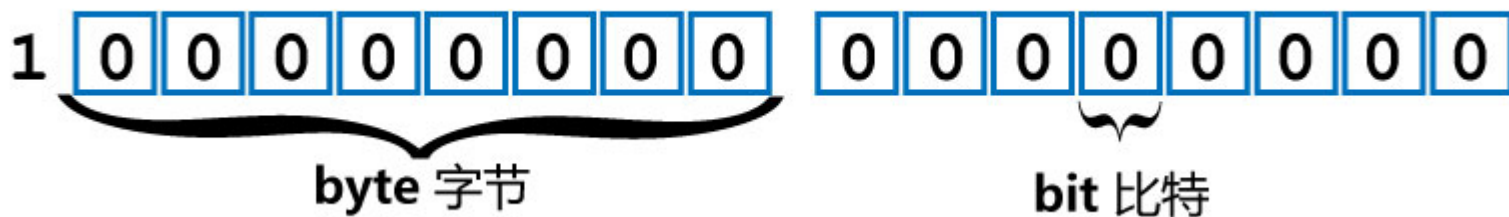
- 变量：存储数据、字符串。
 - 名称可以包括英文、数字以及下划线（_），但不能以数字开头，而且区分大小写。
 - 使用前无需声明、无需初始化，无需指定类型。
- 注意：**变量名要准确地描述该变量所代表的事物，如地球质量的变量名可以为：`mass_earth`，而`x`，`y`，`x1`，`data`之类永远是糟糕的变量名。代码阅读次数远远多于编写次数，确保所取的名字方便阅读而不是容易编写。

数据类型

- 布尔类型: `True/False`
- 数字: `1234, 3.14159, 3+4j`
- 字符串: `'hello', "I'm", """a\nb"""`
- 列表(list): `[1,[2,'three'],4]`
- 字典(dict): `{'name':"zhang","age":18}`
- 元组(Tuple): `(1,'spam',4,'K')`
- 集合(set): `set('abca'),{'a','b','c'}`

数据类型

数据类型	范围	字节数
字节型	$0 \sim 2^8$	1
整型	$-2^{15} \sim 2^{15} - 1$	2
长整型	$-2^{31} \sim 2^{31} - 1$	4
浮点型	$-10^{38} \sim 10^{38}$	4
字符串	-	



数据类型转换

- 布尔类型 (True/False) : `bool()`
- 数字:
 - 整型: `int()`, `round()`
 - 浮点型: `float()`
- 字符串: `str()`, 不可变
- 列表: `list()`, 任意数据类型组成的有序集合
- 元组: `tuple()`, 不可变的有序序列
- 集合: `set()`, 无序不重复元素集
- 字典: `dict()`, 描述映射关系的无序集合

布尔类型

函数: `bool()`; 运算符 `not, or, and`; 测试 `in, is`

```
>>> bool('')
False
>>> bool('hello')
True
>>> a = None
>>> b = 1
>>> bool(a)
False
>>> bool(b)
True
```

```
>>> not 1==3
True
>>> 1>2 or 2>1
True
>>> (1>2) + (2>1)
1
>>> (1>2) and (2>1)
False
>>> (1>2) * (2>1)
0
```

数字对象 (object)

- 通用功能
 - `3.14`, `0xFF`, `012`, `5**2`, `5%2`, `0<x<=5`
- C风格位运算
 - `1<<16`, `x & 0xff`, `x|1`, `~x`, `x^y`
- 逻辑运算
 - `a < b`, `a > b`, `a <= b`, `a >= b`
 - `a == b`, `a != b`
- 整除截断
 - `1/2` , `1./2.` , `float(1)/2`
 - 直接支持任意精度长整型和复数,
 - `2**100` -> `1267650600228229401496703205376L`
 - `1j**2` -> `(-1+0j)`

数学函数

- `abs(x)` 绝对值
- `cmp(x, y)` 比较x与y, 返回-1, 0, 1
- `pow(x, y)` x的y次方
- `complex(x, y)` 生成复数类型 `x+yj`
- `eval()` 表达式求值
例如 `>>> x = 1; eval('x+1')`
- `hex()` 将整数转换为十六进制数
- `oct()` 将整数转换为八进制数

字符串（string）

字符是计算机软件处理文字时最基本的单位，可能是字母，数字，标点符号，空格，换行符，汉字等等。字符串是0个或多个字符的序列。

- `a = "hello"`
 - `'Mrs\' single quotes'` 包含引号的字符串
 - `"""triple quotes\ncontains more"""` 可包含换行符
- `a + "world"` `"helloworld"` # 合并
- `a * 2` `"hellohello"` # 重复
- `a[0]` `"h"` # 索引
- `a[-1]` `"o"` # 倒序索引
- `a[1:4]` `"ell"` # 切片
- `len(a)` `5` # 长度
- `a = 'j' + a[1:]` 如何修改第三个字符？

字符串方法

- `.strip()` 去除首尾空白字符
- `.split()` 分割字符串（默认为空格）
- `.replace()` 替换字符
- `.find()` 查找字符
- `.count()` 字符计数
- `.upper()/lower()` 转大/小写
- `.ljust()/rjust()/zfill()` 指定宽度
- `.isalpha()/isdigit()/isalnum()`

join & split

```
>>>"+".join(['a','b','c'])
```

```
a+b+c
```

```
>>>"a+b+c".split("+")
```

```
['a','b','c']
```

```
>>>"I'm fine".split()
```

```
["I'm","fine"]
```

```
>>>"I'm fine".split(" ")
```

```
["I","m fine"]
```

isalnum

- `a = 'hello'`
- `help(a.isalnum)`
- `print a.isalnum()`
- `print "1234".isalnum()`
- `print "123abc".isalnum()`
- `print "1.23".isalnum()`
- `print "123.abc".isalnum()`