

# Python 简介

长郡中学 罗雨屏

roosephu@gmail.com

2013 年 1 月 26 日

# Python 简介

简介

变量类型

基本语法

常用函数

常用模块

简单实例

FAQ

其它应用

Acknowledge

- 1989 年圣诞节，由 Guido van Rossum 开发，后转交社区
- 面向对象，且支持部分函数式编程
- 运行时检查变量类型
- 变量不需要提前定义，第一次赋值相当于定义
- 语法简洁，开发效率高，运行效率较低
- Python 格言：在 Python 解释器里输入 `import this`

# *int/float/complex*

简介

变量类型

基本语法

常用函数

常用模块

简单实例

FAQ

其它应用

Acknowledge

- `int` 为高精
- `float` 的精度相当于 C 的 `double`
- `**` 为幂运算
- 注意在 Py3k 中，若 `a`, `b` 均为 `int`，则 `a / b` 返回一个 `float`，`a // b` 返回一个 `int` 向下取整。
- `int` 的取模运算为 `%`

# list/tuple

- 变长数组，每个元素的类型可以不相同
- slice
  - $x[a:b]$  :  $x$  中下标在  $[a, b)$  内的元素所组成的数组
    - 下标从 0 开始
  - $x[a:b:c]$  :  $x[a], x[a+c], \dots, x[a+c * \lfloor \frac{b-a-1}{c} \rfloor]$  所组成的数组
  - 省略  $a$  表示从头开始，省略  $b$  表示到尾结束
  - 负下标表示从后往前数
- list comprehension : 从 Haskell 借鉴而来
  - $[x * x \text{ for } x \text{ in range}(n)] : [0, 1, 4, \dots, (n-1)^2]$
- `in` : `5 in [1, 3, 5]` 为 `True` ,  $O(n)$
- tuple : 只读的 list (好没存在感的 tuple)

# *list/tuple cont'd*

- `len(x)` : `x` 的长度
- `min(x)/max(x)` : `x` 中最小/最大的元素
- `s.sort()/s.reverse()` : 将 `s` 排序/翻转
  - `sorted(s)/reversed(s)` 返回一个新的 `list`
- `s.index(x)` : `s` 中 `x` 第一次出现的位置
- `s.count(x)` : `s` 中 `x` 出现的次数
- `s + t` : 合并两个 `list` , `s` 和 `t` 均为 `list`
- `s.append(x)` : 在 `s` 的最后添加一个 `x`
- `s.insert(i, x)` : 在 `i` 处插入 `x` , 使得 `s[i] = x`
- `s.remove(x)` : 删除 `s` 中第一个出现的 `x`
- `s.pop()` : 返回 `s[0]` , 并把 `s[0]` 从 `s` 中删除

# set/dict

简介

变量类型

基本语法

常用函数

常用模块

简单实例

FAQ

其它应用

Acknowledge

- 集合/字典
- 支持 `in`
- 基本操作均为  $O(1)$
- 插入
  - `set`: `S.add(x)`
  - `dict`: `D[x] = y`
- 删除
  - `set`: `S.remove(x)`
  - `dict`: `D[x] = None`

# string

- 支持 + 作为字符串连接
- `str(123)` 可以把 123 转为 '123'
- `str(['a', 'c'])` 得到 'ac'
- `'Py' in 'Python'` 为 True
- `'%d+%d+%d' % (1, 2, 3)` 为 '1+2+3'
- 单引号与双引号是没区别的，方便 + 个人喜好
- `s.find(t)`: s 中 t 第一次出现的位置。如果未找到返回 -1。（参数均为 string，下同）
- `x.replace(s, t)`: 把 x 中所有出现的 s 替换成 t。

# 基本规则

简介

变量类型

基本语法

常用函数

常用模块

简单实例

FAQ

其它应用

Acknowledge

- Python 最大的特点是用缩进表达块的概念
  - 建议 4 格空格缩进
- 多个变量一起赋值
  - 交换两个变量  $x, y = y, x$
  - 不要问我为啥  $f[x], x = x, f[x]$  会导致错误的结果



# if

## Example (code)

```
if x == 1 or (x == 2 and k == 3) :  
    print('orz', 'WJMZBMR')  
elif x == 2 :  
    print('orz', 'sevenkplus')  
else :  
    print('orz', 'oimaster')
```

- 注意逻辑与和逻辑或用的是 `and/or`
- 注意冒号

# 循环

简介

变量类型

基本语法

常用函数

常用模块

简单实例

FAQ

其它应用

Acknowledge

## Example (code)

```
for i in range(1, 100) :  
    print('orz', 'LYD&SHY')  
while True :  
    print('orz', 'SHY&LYD')
```

# with

- 一个临时变量
- 要求这个类型支持 `close()`

## Example (code)

```
with open('output.txt', 'w') as fout :  
    print('orz', 'XLk', file = fout)
```

# function

简介

变量类型

基本语法

常用函数

常用模块

简单实例

FAQ

其它应用

Acknowledge

## Example (code)

```
def add_1(a, b) :  
    print('orz', 'fotile')  
    return a + b  
  
add_2 = lambda x, y : x + y  
print(add_1(3, 2), add_2(3, 5))
```

- 函数可以返回一个 tuple/list

# import

- 导入模块
- `import math`
  - 调用 `math.sin(math.pi)`
- `import math as M`
  - 调用 `M.sin(M.pi)`
- `from math import *`
  - 调用 `sin(pi)`
  - 当然这里的 `*` 可以换成 `sin, pi`
- 鉴于语法过于简单，这里就不能继续 orz 各位大神了  
……T\_T

# 类型构造函数

[简介](#)[变量类型](#)[基本语法](#)[常用函数](#)[常用模块](#)[简单实例](#)[FAQ](#)[其它应用](#)[Acknowledge](#)

- `int/str/float/dict/set/tuple/list`
- 支持 `int/str/float` 之间的转换以及  
`dict/set/tuple/list` 之间的转换

# *range*

简介

变量类型

基本语法

常用函数

常用模块

简单实例

FAQ

其它应用

Acknowledge

- $\text{range}(n) : [0, 1, \dots, n - 1]$
- $\text{range}(a, b) : [a, a + 1, \dots, b - 1]$
- $\text{range}(a, b, c) :$   
 $[a, a + c, a + c * 2, \dots, a + c * \lfloor \frac{b - a - 1}{c} \rfloor]$
- 以上参数均为整数
- 在 Py3k 中, `range` 返回的是一个 `generator`, 如果转成 `list`, 可以利用 `list()`, 例如 `list(range(n))`

# functional

[简介](#)[变量类型](#)[基本语法](#)[常用函数](#)[常用模块](#)[简单实例](#)[FAQ](#)[其它应用](#)[Acknowledge](#)

- 一些函数式编程风格的函数
- `sum/map/filter/enumerate`
- 写过 Haskell 的人应该知道这些函数是干啥的吧。
- `sum/enumerate` 比较常用。
- `enumerate(x)`: 返回一个 `[(idx, val)]`, `idx` 表示序号, `val` 表示值
- `map(f, x)`: `[f(i) for i in x]`
- `filter(f, x)`: `[i for i in x if f(x)]`



# file

简介

变量类型

基本语法

常用函数

常用模块

简单实例

FAQ

其它应用

Acknowledge

- 文件处理
- `open(filename, mode)`: mode 同 C 中 `freopen` 的格式, 返回一个 `file`
- `f.close()`: 关闭文件
- `f.write(S)`: 写入一个字符串 `S`

# *print/input*

- 基本的输入输出语句
- Py2 和 Py3k 的 `print/input` 是不同的
- 注意用 ‘;’ 隔开的两个值中间会有一个空格
- Py2 中, `print` 为一关键字
  - `print >> fout, x, y`
- Py3k 中, `print` 为一函数
  - `print(x, '_', y, file = fout)`
- Py2 中, `raw_input()` 返回 `string`, `input()` 等同于 `eval(raw_input())`
- Py3 中, `input()` 相当于 Py2 中的 `raw_input()`

# *eval*

- 运行一段代码
- 表达式求值？不怕不怕啦……
  - `eval('1+3+6') = 10`

# *pow*

简介

变量类型

基本语法

常用函数

常用模块

简单实例

FAQ

其它应用

Acknowledge

- 快速幂
- $\text{pow}(b, e) : b^e$
- $\text{pow}(b, e, m) : b^e \bmod m$

# math

简介

变量类型

基本语法

常用函数

常用模块

简单实例

FAQ

其它应用

Acknowledge

- 大名鼎鼎的 `math` 库
- 相当于 `math.h(C)/cmath(C++)/math(Pascal)`
- 自带两个常数  $\pi, e$
- `cmath` 为复数的 `math` 库
  - 复数的表示  $1 + 2j$  ,  $j$  表示虚数单位  $i$

# *fractions*

简介

变量类型

基本语法

常用函数

常用模块

简单实例

FAQ

其它应用

Acknowledge

- 分数类型
  - 构造函数为 `Fraction(a, b)` 表示  $\frac{a}{b}$
- 已重载各种运算符
- 附赠 `gcd` 函数一份
  - 保证任意时刻分子分母互质
- 用 `numerator` 来访问分子, `denominator` 来访问分母

# *subprocess*

简介

变量类型

基本语法

常用函数

常用模块

简单实例

FAQ

其它应用

Acknowledge

- 产生新的进程。
- 一般导入 `call` 足以。
  - `call('./program')`
  - 返回程序的返回值

# random

简介

变量类型

基本语法

常用函数

常用模块

简单实例

FAQ

其它应用

Acknowledge

- `randint(a, b)` : 产生  $[a, b]$  内的一个随机整数
- `randrange`
  - 参数同 `range` 的传参规则
  - 返回这个 `range` 内的任意一个整数
- `random()` : 产生  $[0, 1)$  内的一个随机实数
- `uniform(a, b)` : 产生  $[a, b]$  内的一个随机实数
- `shuffle(x)` : 随机打乱 `x` (`x` 是一个 `list`, 下同)
- `choice(x)` : 返回 `x` 内任意一个元素
- `sample(x, n)` : 返回 `x` 内随机 `n` 个元素



# shutil

简介

变量类型

基本语法

常用函数

常用模块

简单实例

FAQ

其它应用

Acknowledge

- `copy(src, dst)`: 把 `src` 拷贝到 `dst`
  - 不能移动文件夹
- `move(src, dst)`: 把 `src` 移动到 `dst`
  - `src` 允许为文件夹。递归移动

# 更多帮助

简介

变量类型

基本语法

常用函数

常用模块

简单实例

FAQ

其它应用

Acknowledge

- Python 的帮助文档非常详细
- 在交互式终端里面输入 `help()` 后进入帮助界面
  - 再输入模块名或函数名即可查看对应的帮助
  - `<Ctrl-D>` 退出
- Python 的官方帮助文档 <http://www.python.org/doc/>
  - 不同版本有不同的文档
  - Language Reference 介绍基础语法
  - Library Reference 介绍库的应用

# 对拍程序

## example

```
from random import *
from subprocess import call

for cnt in range(100000) :
    with open("treasures.in", "w") as fout :
        n = 20
        print(n, file = fout)
        for i in range(n) :
            print(randint(-10, 10), randint(-10, 10), randint(1, 30), file = fout)

if call("./treasures") or call("./treasures_bf") or \
    call(["diff", "treasures.out", "treasures.ans"]) :
    break
print(cnt, "Accepted")
```

- 当然你还可以把暴力顺便也用 Python 写了

# 数据生成器

## example

```
from random import *
from subprocess import call
from shutil import move

info = [
    4, 5, 5,
    1000, 1000, 1000, 1000,
    int(1e18), int(1e18), int(1e18),
]

for cnt, n in enumerate(info) :
    with open("calc.in", "w") as fout :
        d = randint(6, 9)
        p = randint(0, d - 1)
        n = randint(n * 9 // 10, n)
        print(randint(5, 9), n, d, p, 1000000007, file = fout)

    call("./calc")
    move("calc.in" , "calc" + str(cnt) + ".in" )
    move("calc.out", "calc" + str(cnt) + ".out")
```

# FAQ

简介

变量类型

基本语法

常用函数

常用模块

简单实例

FAQ

其它应用

Acknowledge

- Question

- 肿么没反应了? Python 是不是死掉了?

- Answer

- Python 太慢了
- Python 太慢了
- Python 太慢了
- Python 太慢了
- 想象一下在 SPOJ 的 Pyrimid 上运行 Python ……

# FAQ

简介

变量类型

基本语法

常用函数

常用模块

简单实例

FAQ

其它应用

Acknowledge

- Question
  - 我要写类！我要写继承、重载、派生、多继承！
- Answer
  - 类的关键字为 `class`
  - 然后请到 Language Reference 里面查
  - 本弱只能帮到这了

# 库

- Python 有很多外部库
  - 图像处理: PIL
  - GUI 编程: Tkinter/wxPython/PyQt/PyGTK
  - 网页设计: Django/Pylons/web.py
  - 游戏设计: Pygame/PyOpenGL
  - 数值处理: NumPy/SciPy
  - Python 首页<sup>1</sup> 有介绍
  - 有兴趣的同学可以继续寻找
- Google 的 Python 学习课堂
  - <https://developers.google.com/edu/python/>

---

<sup>1</sup><http://www.python.org/>

# Acknowledge

## *L<sup>A</sup>T<sub>E</sub>X/beamer*

beamer theme 为 Fedora Laughlin, under LPPLv1.3c<sup>a</sup>。图片 under GPLv2<sup>b</sup>。

---

<sup>a</sup><http://www.latex-project.org/lppl/lppl-1-3c.html>

<sup>b</sup><http://www.gnu.org/licenses/gpl-2.0.html>

## *Emacs/Org-Mode*

本 pdf 由 Org-Mode<sup>a</sup> 辅助生成。

---

<sup>a</sup><http://orgmode.org/>



# Thanks

- 感谢大家倾听

简介

变量类型

基本语法

常用函数

常用模块

简单实例

FAQ

其它应用

Acknowledge