Python 快速入门

iGuo

Python 中文社区

本版本：2017 年 2 月 15 日

目录

目录

个人简介

- Python 中文社区专栏作者
- 主修经济学，辅修统计学，编程爱好者
- 爱好科研，喜欢读论文和写代码，带领着自己的课题组 China's Prices Project(CPP) 在科研和编程的道路上前进
- 主要兴趣方向：数据分析、科学计算
- 主要擅长领域：网页爬虫、数据清洗、统计分析、可视化
- 正在开发的领域：海量数据的储存、清洗、挖掘与分析

我与代码的那些故事

- 学习代码的原因：导师的要求
- 学习渠道：学校课程 - 自学

系列简介

- 目标人群：
  - 零基础或者基础比较薄弱的小白
  - 希望系统梳理知识体系的初学者
- 任务目标：
  - 系统梳理 Python 基础知识，为后续 PyLive 做铺垫
  - 向初学者介绍编程的学习方法
  - 探索合适的 PyLive 组织方式

目录

为什么学习编程?

- 明确目标
  - Python 语言的岗位?
  - 工作中的辅助工具?
  - 科研和学术用途?
  - 自我娱乐?
  - ...
- 根据目标确定学习策略
  - 刚需：系统学习
  - 非刚需：面向需求学习

如何学习编程?

- 系统学习：基本语法 - 课后练习 - 期中期末／大作业 - 实际项目
- 面向需求学习：理解需求 - 查找解决方案 - 学习对应语法 - 模仿、踩坑 - 跳坑 - 实现需求

如何学习编程?

- 需求是驱动水平提高的根本动力 - 学会创造需求
- 学习资源来源从何而来? 知乎相关问题的回答
- 遇到 bug 怎么办? 合理使用搜索引擎
- 延伸阅读: 给自学编程的初学者的小建议

目录

为什么选择 Python?

- 易学易用，开发效率高
- 强大的胶水
- 延伸阅读：Python 为何能坐稳 AI 时代头牌语言

2.x vs 3.x?

- 为什么 3.x 无法完全取代 2.x？没有真正解决 2.x 的问题。
- 2.x 和 3.x 互相迁移很困难？不是。
- 延伸阅读：
  Python2 和 Python3 哪个更适合初学者学习来爬虫呢?

如何选择 Python 版本?

- 2.x：选择 2.6 以上版本
- 3.x：选择 3.4 以上版本
- 本教程使用 2.7.13 和 3.6.0

Python 开发环境

- Python 解释器 +IDLE
- PyCharm：主流 IDE
- Anaconda：科学计算专用
- Jupyter：浏览器版 iPython
- Sublime Text 3：可拓展性极强的文本编辑器
- 本系列：**Python 解释器** / iPython / **Jupyter Notebook**

Python 学习资源

- 书籍：《Python 核心编程》(2.x)、《笨办法学 Python》等
- 网站：Crossin 编程教室、廖雪峰 Python 教程等
- 延伸阅读：Python 优质资料合集

目录

什么是程序?

- 程序：输入 $\rightarrow$ 输出
- 程序 $=$ 数据结构 $+$ 算法
  - 数据结构：储存、组织数据
  - 算法：具体步骤
- 程序：实现需求，解决问题

Python 解释器 (2.7.13)

```
>>> for i in range(10):
...     print i,
...
0 1 2 3 4 5 6 7 8 9
```

- 主提示符：>>>
- 次提示符：...
- 注释：#

输入 (2.7.13)

- **print**

```
1 >>> print 'Hello, world'
2 Hello, world
```

输出 (2.7.13)

- **input**
- **raw_input**

```
1 >>> input('please␣input␣your␣favorate␣number:'
      )
2 please input your favorate number:7
3 7
4 >>> raw_input('please␣input␣your␣favorate␣
      number:')
5 please input your favorate number:7
6 '7'
```

语句与表达式 (2.7.13)

- 语句： **print** 'Hello, world'
- 表达式： **abs**(-4)

帮助 (2.7.13)

```
>>> help(input)
Help on built-in function input in module
    __builtin__:

input(...)

    input([prompt]) -> value

    Equivalent to eval(raw_input(prompt)).
```

帮助 (2.7.13)

```
>>> help(print)
  File "<stdin>", line 1
    help(print)
               ^
SyntaxError: invalid syntax
```

操作符 (2.7.13)

- 算术操作符：+,−,*,/,//,%,**

```
1 >>> 1/2,1//2,1.0/2,1.0//2
2 (0, 0, 0.5, 0.0)
3 >>> 4**2
4 16
5 >>> 7%3
6 1
```

- 比较操作符：<,<=,>,>=,==,!=
- 逻辑操作符：**and**, **or**, **not**

变量与赋值 (2.7.13)

```
1 >>> counter = 0
2 >>> # 给变量 counter 赋值 0
```

目录

基本数据类型：数字 (2.7.13)

```
1 >>> 1
2 1
3 >>> 1.1
4 1.1
5 >>> True
6 True
```

基本数据类型：序列 (2.7.13)

```
1 >>> [1,2]
2 [1, 2]
3 >>> (1,3)
4 (1, 3)
5 >>> 'hello'
6 'hello'
```

基本数据类型：集合和字典 (2.7.13)

```
1 >>> {1,2,3}
2 set([1, 2, 3])
3 >>> {"1":1,"2":2}
4 {'1': 1, '2': 2}
```

基本控制语句：if(2.7.13)

```
1 >>> a = 1
2 >>> if a<2:
3 ...     print('hello')
4 ...
5 hello
```

基本控制语句：for(2.7.13)

```
1 >>> for i in range(10):
2 ...     print i,
3 ...
4 0 1 2 3 4 5 6 7 8 9
```

基本控制语句：while(2.7.13)

```
1 >>> a = 1
2 >>> while a<4:
3 ...     print a,
4 ...     a=a+1
5 ...
6 1 2 3
```

目录

写入文件 (2.7.13)

```
1 >>> import os
2 >>> os.getcwd()  # get current working
      dictionary
3 >>> f = open('test.txt','w')
4 >>> f.write('Hello,world')
5 >>> f.close()
```

读取文件 (2.7.13)

```
1 >>> import os
2 >>> os.getcwd()
3 >>> f = open('test.txt','r')
4 >>> f.read()
5 >>> f.close()
```

try-except 语句 (2.7.13)

```
>>> try:
...     open('test.txt')
... except IOError:
...     print 'No such file'
...
No such file
```

定义函数 (2.7.13)

```
>>> def test():
...     print('hello,world')
...
>>> test()
hello,world
```

类：定义类 (2.7.13)

```
>>> class Test(object):
...     def __init__(self,name):
...         self.name = name
...     def print_name(self):
...         print "My name is",self.name
...
>>>
```

```
1 >>> a = Test('iGuo')
2 >>> a.name
3 'iGuo'
4 >>> a.print_name()
5 My name is iGuo
```

导入和使用模块 (2.7.13)

```
>>> import string
>>> dir(string)
['Formatter', 'Template', '_TemplateMetaclass', '__builtins__', '__doc__', '__file__', '__name__', '__package__', '_float', '_idmap', '_idmapL', '_int', '_long', '_multimap', '_re', 'ascii_letters', 'ascii_lowercase', 'ascii_uppercase', 'atof', 'atof_error', 'atoi', 'atoi_error', 'atol', 'atol_error', 'capitalize', 'capwords', 'center', 'count', 'digits', 'expandtabs', 'find', 'hexdigits', 'index', 'index_error', 'join', 'joinfields', 'letters', 'ljust', 'lower', 'lowercase',
```

导入和使用模块 (2.7.13)

```
 9   'lstrip', 'maketrans', 'octdigits', '
        printable', 'punctuation', 'replace', 'rfi
10  nd', 'rindex', 'rjust', 'rsplit', 'rstrip', '
        split', 'splitfields', 'strip', 'sw
11  apcase', 'translate', 'upper', 'uppercase', '
        whitespace', 'zfill']
12  >>> string.uppercase
13  'ABCDEFGHIJKLMNOPQRSTUVWXYZ'
14  >>> string.lowercase
15  'abcdefghijklmnopqrstuvwxyz'
16  >>> string.whitespace
17  '\t\n\x0b\x0c\r '
18  >>> string.upper
19  <function upper at 0x0000000002BB8278>
```

导入和使用模块 (2.7.13)

```
20 >>> string.upper('agesrag')
21 'AGESRAG'
```

下载第三方模块 (2.7.13)

- 在命令行输入:
  - pip2 install <package>
  - easy_insall <package>
- 常见问题:
  - pip 不能使用
  - C 扩展库不能安装

目录