

水木书荟

理解为主，应用为王
配套96集免费视频教程，让您轻松、快速用Python
Python3入门必备，小甲鱼手把手教您Python开发，
从真实案例中领略Python的真正魅力

实用·好玩·参与

零基础
入门学习
Python



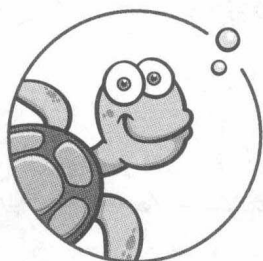
© 小甲鱼 编著

清华大学出版社



 水木书荟

零基础
入门学习
Python



© 小甲鱼 编著

清华大学出版社
北京

内 容 简 介

本书适合学习 Python3 的入门读者,也适用对编程一无所知,但渴望用编程改变世界的朋友们!本书提倡理解为主,应用为王。因此,只要有可能,小甲鱼(作者)都会通过生动的实例来让大家理解概念。

虽然这是一本入门书籍,但本书的“野心”可并不止于“初级水平”的教学。本书前半部分是基础的语法特性讲解,后半部分围绕着 Python3 在爬虫、Tkinter 和游戏开发等实例上的应用。

编程知识深似海,小甲鱼没办法仅通过一本书将所有的知识都灌输给你,但能够做到的是培养你对编程的兴趣,提高你编写代码的水平,以及锻炼你的自学能力。最后,本书贯彻的核心理念是:实用、好玩,还有参与。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

零基础入门学习 Python/小甲鱼编著. —北京:清华大学出版社,2016(2016.12 重印)

(水木书荟)

ISBN 978-7-302-43820-5

I. ①零… II. ①小… III. ①软件工具—程序设计 IV. ①TP311.56

中国版本图书馆 CIP 数据核字(2016)第 101697 号

责任编辑:刘 星 李 晔

封面设计:刘 键

责任校对:时翠兰

责任印制:李红英

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦 A 座 邮 编:100084

社总机:010-62770175 邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

课 件 下 载: <http://www.tup.com.cn>, 010-62795954

印 装 者:清华大学印刷厂

经 销:全国新华书店

开 本:185mm×260mm 印 张:23 字 数:584 千字

版 次:2016 年 11 月第 1 版 印 次:2016 年 12 月第 3 次印刷

印 数:4501~9500

定 价:49.50 元

Life is short. You need Python.

——Bruce Eckel

上边这句话是 Python 社区的名言，翻译过来就是“人生苦短，我用 Python”。

我和 Python 结缘于一次服务器的调试，从此便一发不可收拾。我从来没有遇到一门编程语言可以如此干净、简洁，如果你有处女座情节，你一定会爱上这门语言。使用 Python，可以说是很难写出丑陋的代码。我从来没想过一门编程语言可以如此简单，它太适合零基础的朋友踏入编程的大门了，如果我有一个八岁的孩子，我一定会毫不犹豫地使用 Python 引导他学习编程，因为面对它，永远不缺乏乐趣。

Python 虽然简单，其设计却十分严谨。尽管 Python 可能没有 C 或 C++ 这类编译型语言运行速度那么快，但是 C 和 C++ 需要你无时无刻地关注数据类型、内存溢出、边界检查等问题。而 Python，它就像一个贴心的仆人，私底下为你都一一处理好，从来不用你操心这些，这让你可以将全部心思放在程序的设计逻辑之上。

有人说，完成相同的一个任务，使用汇编语言需要 1000 行代码，使用 C 语言需要 500 行，使用 Java 只需要 100 行，而使用 Python，可能只要 20 行就可以了。这就是 Python，使用它来编程，你可以节约大量编写代码的时间。

既然 Python 如此简单，会不会学了之后没什么实际作用呢？事实上你并不担心这个问题，因为 Python 可以说是一门“万金油”语言，在 Web 应用开发、系统网络运维、科学与数字计算、3D 游戏开发、图形界面开发、网络编程中都有它的身影。目前越来越多的 IT 企业，在招聘栏中都有“精通 Python 语言优先考虑”的字样。另外，就连 Google 都在大规模使用 Python。

好了，我知道过多的溢美之词反而会使大家反感，所以我必须就此打住，剩下的就留给大家自己体验吧。

接下来简单地介绍一下这本书。一年前，出版社的编辑老师无意间看到了我的一个同名的教学视频，建议我以类似的风格撰写一本书。当时我是受宠若惊的，也很兴奋。刚开始写作就遇到了不小的困难——如何将视频中口语化的描述转变为文字。当然，我希望尽可能地保留原有的幽默和风趣——毕竟学习是要快乐的。这确实需要花不少时间去修改，但我觉得这是值得的。

本书不假设你拥有任何一方面的编程基础，所以本书不但适合有一定编程基础，想学习 Python3 的读者，也适合此前对编程一无所知，但渴望用编程改变世界的朋友！本书提倡理解为主，应用为王。因此，只要有可能，我都会通过生动的实例来让大家理解概念。虽然这是一本入门书籍，但本书的“野心”可并不止于“初级水平”的教学。本书前半部分是基础的语法特性讲解，后半部分围绕着 Python3 在爬虫、Tkinter 和游戏开发等实例上的应用。编程知识深似海，没办法仅通过一本书将所有的知识都灌输给你，但我能够做到的是培养你对编程的兴

趣,提高你编写代码的水平,以及锻炼你的自学能力。最后,本书贯彻的核心理念是:实用、好玩,还有参与。

本书对应的系列视频教程,可以在 <http://blog.fishc.com/category/python> 下载得到,也可扫描以下二维码关注微信号进行观看。



编者

2016年7月

第 1 章 就这么愉快地开始吧	1
1.1 获得 Python	1
1.2 从 IDLE 启动 Python	2
1.3 失败的尝试	3
1.4 尝试点儿新的东西	3
1.5 为什么会这样	4
第 2 章 用 Python 设计第一个游戏	5
2.1 第一个小游戏	5
2.2 缩进	6
2.3 BIF	6
第 3 章 成为高手前必须知道的一些基础知识	8
3.1 变量	8
3.2 字符串	9
3.3 原始字符串	10
3.4 长字符串	11
3.5 改进我们的小游戏	12
3.6 条件分支	12
3.7 while 循环	13
3.8 引入外援	14
3.9 闲聊数据类型	15
3.9.1 整型	15
3.9.2 浮点型	16
3.9.3 布尔类型	16
3.9.4 类型转换	16
3.9.5 获得关于类型的信息	17
3.10 常用操作符	18
3.10.1 算术操作符	18
3.10.2 优先级问题	19
3.10.3 比较操作符	19
3.10.4 逻辑操作符	20

第 4 章	了不起的分支和循环	21
4.1	分支和循环	21
4.2	课堂小练习	23
4.3	结果分析	24
4.4	Python 可以有效避免“悬挂 else”	24
4.5	条件表达式(三元操作符)	24
4.6	断言	25
4.7	while 循环语句	25
4.8	for 循环语句	26
4.9	range()	26
4.10	break 语句	27
4.11	continue 语句	27
第 5 章	列表、元组和字符串	29
5.1	列表：一个“打了激素”的数组	29
5.1.1	创建列表	29
5.1.2	向列表添加元素	29
5.1.3	从列表中获取元素	31
5.1.4	从列表删除元素	31
5.1.5	列表分片	32
5.1.6	列表分片的进阶玩法	33
5.1.7	一些常用操作符	33
5.1.8	列表的小伙伴们	35
5.1.9	关于分片“拷贝”概念的补充	36
5.2	元组：戴上了枷锁的列表	37
5.2.1	创建和访问一个元组	37
5.2.2	更新和删除元组	39
5.3	字符串	39
5.3.1	各种内置方法	40
5.3.2	格式化	43
5.4	序列	46
第 6 章	函数	50
6.1	Python 的乐高积木	50
6.1.1	创建和调用函数	50
6.1.2	函数的参数	51
6.1.3	函数的返回值	52
6.2	灵活即强大	52
6.2.1	形参和实参	52

6.2.2	函数文档	52
6.2.3	关键字参数	53
6.2.4	默认参数	53
6.2.5	收集参数	54
6.3	我的地盘听我的	55
6.3.1	函数和过程	55
6.3.2	再谈谈返回值	56
6.3.3	函数变量的作用域	56
6.4	内嵌函数和闭包	58
6.4.1	global 关键字	58
6.4.2	内嵌函数	59
6.4.3	闭包(closure)	60
6.5	lambda 表达式	62
6.6	递归	64
6.6.1	递归是“神马”	64
6.6.2	写一个求阶乘的函数	66
6.6.3	这帮小兔崽子	68
6.6.4	汉诺塔	70
第 7 章	字典和集合	72
7.1	字典：当索引不好用时	72
7.1.1	创建和访问字典	72
7.1.2	各种内置方法	74
7.2	集合：在我的世界里，你就是唯一	77
7.2.1	创建集合	78
7.2.2	访问集合	79
7.2.3	不可变集合	79
第 8 章	永久存储	80
8.1	文件：因为懂你，所以永恒	80
8.1.1	打开文件	80
8.1.2	文件对象的方法	81
8.1.3	文件的关闭	81
8.1.4	文件的读取和定位	82
8.1.5	文件的写入	83
8.1.6	一个任务	83
8.2	文件系统：介绍一个高大上的东西	85
8.3	pickle：腌制一缸美味的泡菜	91

第 9 章 异常处理	93
9.1 你不可能总是对的	93
9.2 try-except 语句	95
9.2.1 针对不同异常设置多个 except	96
9.2.2 对多个异常统一处理	97
9.2.3 捕获所有异常	97
9.3 try-finally 语句	97
9.4 raise 语句	98
9.5 丰富的 else 语句	98
9.6 简洁的 with 语句	99
第 10 章 图形用户界面入门	101
10.1 导入 EasyGui	102
10.2 使用 EasyGui	102
10.3 修改默认设置	104
第 11 章 类和对象	105
11.1 给大家介绍对象	105
11.2 对象=属性+方法	105
11.3 面向对象编程	106
11.3.1 self 是什么	107
11.3.2 你听说过 Python 的魔法方法吗	107
11.3.3 公有和私有	108
11.4 继承	109
11.4.1 调用未绑定的父类方法	111
11.4.2 使用 super 函数	112
11.5 多重继承	112
11.6 组合	113
11.7 类、类对象和实例对象	114
11.8 到底什么是绑定	115
11.9 一些相关的 BIF	116
第 12 章 魔法方法	120
12.1 构造和析构	120
12.1.1 __init__(self[, ...])	120
12.1.2 __new__(cls[, ...])	121
12.1.3 __del__(self)	122

12.2	算术运算	122
12.2.1	算术操作符	123
12.2.2	反运算	125
12.2.3	增量赋值运算	126
12.2.4	一元操作符	126
12.3	简单定制	126
12.4	属性访问	131
12.5	描述符(property 的原理)	135
12.6	定制序列	137
12.7	迭代器	139
12.8	生成器(乱入)	142
第 13 章	模块	145
13.1	模块就是程序	145
13.2	命名空间	146
13.3	导入模块	146
13.4	<code>__name__ = '__main__'</code>	147
13.5	搜索路径	149
13.6	包	150
13.7	像个极客一样去思考	150
第 14 章	论一只爬虫的自我修养	157
14.1	入门	157
14.2	实战	159
14.2.1	下载一只猫	159
14.2.2	翻译文本	161
14.3	隐藏	166
14.3.1	修改 User-Agent	166
14.3.2	延迟提交数据	168
14.3.3	使用代理	169
14.4	Beautiful Soup	171
14.5	正则表达式	174
14.5.1	re 模块	175
14.5.2	通配符	175
14.5.3	反斜杠	175
14.5.4	字符类	176
14.5.5	重复匹配	177
14.5.6	特殊符号及用法	178

14.5.7	元字符	180
14.5.8	贪婪和非贪婪	182
14.5.9	反斜杠+普通字母=特殊含义	183
14.5.10	编译正则表达式	184
14.5.11	编译标志	184
14.5.12	实用的方法	185
14.6	异常处理	190
14.6.1	URLError	190
14.6.2	HTTPError	191
14.6.3	处理异常	193
14.7	安装 Scrapy	194
14.8	Scrapy 爬虫之初窥门径	196
14.8.1	Scrapy 框架	196
14.8.2	创建一个 Scrapy 项目	198
14.8.3	定义 Item 容器	198
14.8.4	编写爬虫	199
14.8.5	爬	199
14.8.6	取	201
14.8.7	在 Shell 中尝试 Selector 选择器	201
14.8.8	使用 XPath	203
14.8.9	提取数据	203
14.8.10	使用 item	206
14.8.11	存储内容	207
第 15 章	GUI 的最终选择: Tkinter	208
15.1	Tkinter 之初体验	208
15.2	Label 组件	210
15.3	Button 组件	212
15.4	Checkbutton 组件	213
15.5	Radiobutton 组件	214
15.6	LabelFrame 组件	215
15.7	Entry 组件	216
15.8	Listbox 组件	221
15.9	Scrollbar 组件	223
15.10	Scale 组件	224
15.11	Text 组件	225
15.11.1	Indexes 用法	227
15.11.2	Marks 用法	229

15.11.3	Tags 用法	231
15.12	Canvas 组件	237
15.13	Menu 组件	242
15.14	Menubutton 组件	245
15.15	OptionMenu 组件	246
15.16	Message 组件	247
15.17	Spinbox 组件	248
15.18	PanedWindow 组件	248
15.19	Toplevel 组件	250
15.20	事件绑定	252
15.21	事件序列	254
15.21.1	type	254
15.21.2	modifier	255
15.22	Event 对象	256
15.23	布局管理器	258
15.23.1	pack	258
15.23.2	grid	259
15.23.3	place	261
15.24	标准对话框	262
15.24.1	messagebox(消息对话框)	262
15.24.2	filedialog(文件对话框)	265
15.24.3	colorchooser(颜色选择对话框)	266
第 16 章	Pygame: 游戏开发	268
16.1	安装 Pygame	268
16.2	初步尝试	269
16.3	解惑	271
16.3.1	什么是 Surface 对象	271
16.3.2	将一个图像绘制到另一个图像上是怎么一回事	272
16.3.3	移动图像是怎么一回事	273
16.3.4	如何控制游戏的速度	273
16.3.5	Pygame 的效率高不高	273
16.3.6	我应该从哪里获得帮助	274
16.4	事件	274
16.5	提高游戏的颜值	277
16.5.1	显示模式	277
16.5.2	全屏才是王道	278
16.5.3	使窗口尺寸可变	279

16.5.4	图像的变换	279
16.5.5	裁剪图像	281
16.5.6	转换图片	287
16.5.7	透明度分析	287
16.6	绘制基本图形	291
16.6.1	绘制矩形	292
16.6.2	绘制多边形	293
16.6.3	绘制圆形	293
16.6.4	绘制椭圆形	294
16.6.5	绘制弧线	294
16.6.6	绘制线段	295
16.7	动画精灵	296
16.7.1	创建精灵	298
16.7.2	移动精灵	299
16.8	碰撞检测	301
16.8.1	尝试自己写碰撞检测函数	301
16.8.2	sprite 模块提供的碰撞检测函数	303
16.8.3	实现完美碰撞检测	304
16.9	播放声音和音效	305
16.10	响应鼠标	308
16.10.1	设置鼠标的位置	308
16.10.2	自定义鼠标光标	309
16.10.3	让小球响应光标的移动频率	311
16.11	响应键盘	313
16.12	结束游戏	314
16.12.1	发生碰撞后获得随机速度	314
16.12.2	减少“抖动”现象的发生	315
16.12.3	游戏胜利	317
16.12.4	更好地结束游戏	318
16.13	经典飞机大战	319
16.13.1	游戏设定	319
16.13.2	主模块	320
16.13.3	我方飞机	322
16.13.4	响应键盘	322
16.13.5	飞行效果	323
16.13.6	敌方飞机	324
16.13.7	提升敌机速度	325
16.13.8	碰撞检测	326

16.13.9	完美碰撞检测	329
16.13.10	一个 BUG	330
16.13.11	发射子弹	331
16.13.12	设置敌机“血槽”	333
16.13.13	中弹效果	335
16.13.14	绘制得分	336
16.13.15	暂停游戏	337
16.13.16	控制难度	338
16.13.17	全屏炸弹	339
16.13.18	发放补给包	340
16.13.19	超级子弹	343
16.13.20	三次机会	344
16.13.21	结束画面	347
参考文献		350

第 1 章

就这么愉快地开始吧

1.1 获得 Python

我观察到这么一个现象：很多初学的朋友都会在学习论坛上问什么语言才是最好的？他们的目的很明确，就是要找一门“最好”的编程语言，然后持之以恒地学习下去。没错，这种“执子之手与子偕老”的专一精神是我们现实社会所推崇的。但在编程的世界里，我们并不提倡这样。我们更提倡“存在即合理”，当前热门的编程语言都有其存在的道理，它们都有各自擅长的领域和适用性。因此我们没办法去衡量哪一门语言才是最好的。

Python 的语法是非常精简的，对于一位完美主义者来说，Python 将是他爱不释手的伙伴。Python 社区的目标就是构造完美的 Python 语言！本书将使用 Python3 来进行讲解，而 Python3 不完全兼容 Python2 的语法，这样做无疑会让大多数程序员心生怨愤且喋喋不休，因为他们用 Python2 写的大量代码经过层层调试已经趋近完美，并已部署到服务器或应用上了。Python3 对 Python2 的语法不兼容，意味着他们的这些应用需要进行转换和重新调试……但是，Python 社区仍旧坚持推出全新的 Python3。只有勇敢地割掉与时代发展不相符的瑕疵部分，才能缔造出真正的完美体验！

工欲善其事，必先利其器。我们要成为“大牛”，要用 Python 去拯救世界，要做的第一件事就是要下载一个 Python 的安装程序并成功地将它安装到你的计算机上。

安装 Python 非常容易，你可以在它的官网找到最新的版本并下载（注：本书所需要的程序、例子均附带在本书配套资源中），地址是 <http://www.python.org>。

如图 1-1 所示，进入 Python 官网后找到 Download 字样，下载最新版本的 Python 即可。

如果是其他操作系统（例如，Mac OS X），在页面下方可以找到对应的下载地址，如图 1-2 所示。

此处演示的是本书截稿前的最新版本 Python 3.4.3（32 位）（注：这里建议大家安装 32 位版，因为本书第 16 章安装 Pygame 时需要 32 位版本的 Python），一般大家下载最新版本即可。安装 Python3 非常简单，打开下载好的安装包，按照默认选项安装即可。

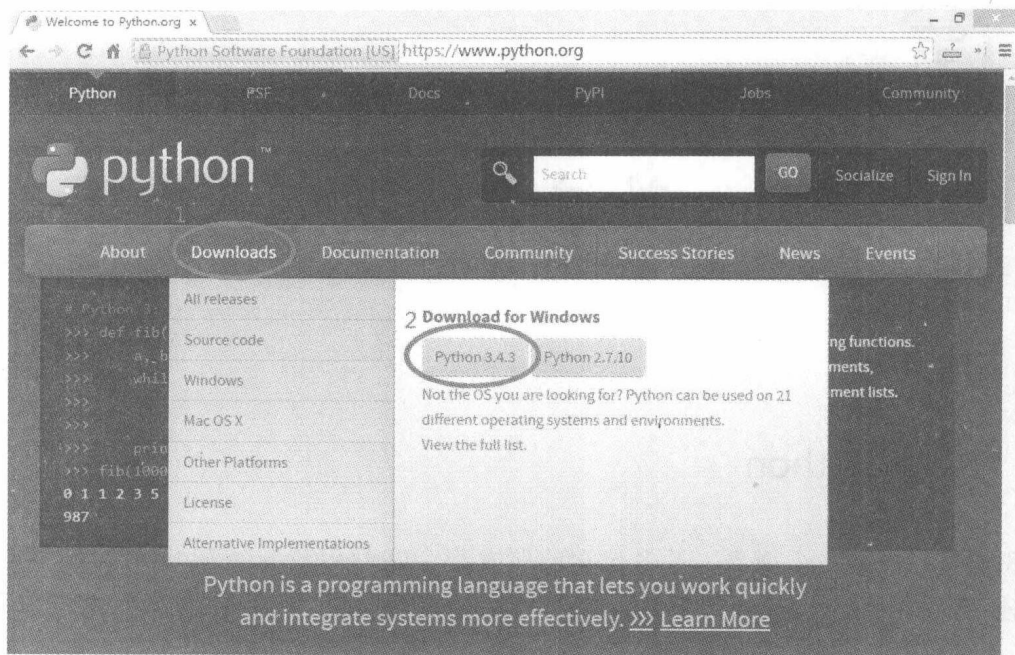


图 1-1 下载 Python3

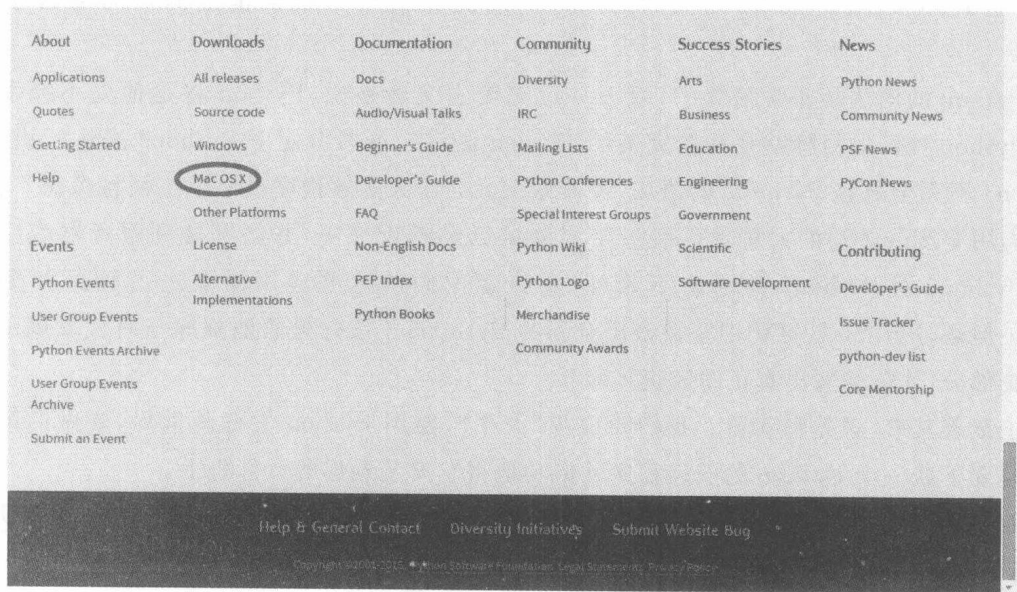


图 1-2 下载 Python3

1.2 从 IDLE 启动 Python

IDLE 是一个 Python Shell, shell 的意思就是“外壳”,从基本上说,就是一个通过输入文本与程序交互的途径。像 Windows 的 cmd 窗口,像 Linux 那个黑乎乎的命令窗口,它们都是 shell,利用它们,就可以给操作系统下达命令。同样,可以利用 IDLE 这个 shell 与 Python 进

行互动。

>>>这个提示符含义是：Python 已经准备好了，在等着输入 Python 指令呢。如图 1-3 所示，可以看到 Python 已经按照我们的要求去做了，在屏幕上打印（注：这里打印的意思是“打印”到屏幕上）I love fishc.com 这个充满浓浓爱意的字符串，这说明什么？没错，这说明我们是“爱鱼 C”的，也说明了我们跟 Python 的第一次亲密接触是有感觉的，她完全能够理解我的想法。

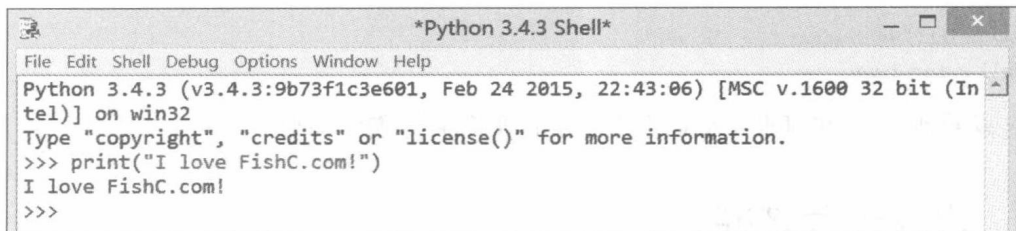


图 1-3 在 Python 的 IDLE 中输入命令

1.3 失败的尝试

像下面这样输入，Python 就会“笨笨地”出错：

```
>>> print "I love fishc.com" # 这是 Python2.x 的语法
SyntaxError: Missing parentheses in call to 'print'
>>> printf("I love fishc.com"); # 这是 C 语言的语法
Traceback (most recent call last):
  File "<pyshell#1>", line 1, in <module>
    printf("I love fishc.com");
NameError: name 'printf' is not defined
```

其实 Python3 哪里是“笨”，她只是小气，所以显得蠢萌蠢萌的。我们仿佛听到她在说：为什么此时此刻你跟我在一起还想着前任？为什么你跟我在一起还想着其他女人，小 C 她哪点儿比我好，她还要加分号呢，我可不用！

大家看到上边的代码中井号（#）后边加了段中文，井号起到的作用是注释，也就是说，井号后边的内容是给人们看的，并不会被当作代码运行。

1.4 尝试点儿新的东西

尝试点儿新的东西，在 IDLE 中输入 `print(5+3)` 或者直接输入 `5+3`：

```
>>> print(5+3)
8
>>> 5+3
8
```

看起来 Python 还会做加法！这并不奇怪，因为计算机最开始的时候就是用来计算的，任何编程语言都具备计算能力，那接下来看看 Python 在计算方面有何神奇。