

开始阅读前的一些准备

内容概要

关于 Python 编程的第一部分总计 7 章，主要内容概括为：

1. 以布尔值为入口开始理解程序本质
2. 了解值的分类和运算方法
3. 简要了解流程控制的原理
4. 简要了解函数的基本构成
5. 相对完整地理解字符串的操作
6. 了解各种容器的基础操作
7. 简要了解文件的读写操作

阅读策略

首先，不要试图一下子就全部搞懂。这不仅很难，在最初的时候也完全没必要。

因为这部分的知识结构中，充满了“过早引用”。请在第一遍粗略完成第 1 部分中的 E1 ~ E7 之后，再去阅读《[如何从容应对“过早引用”？](#)》。

其次，这一部分，注定要反复阅读若干遍。

在开始之前，要明确这一部分的阅读目标。

这一部分的目标，不是让你读完之后就可以开始写程序；也不是让你读完之后就对编程或者 Python 编程有了完整的了解，甚至不是让你真的学会什么……这一部分的目标，只是让你“脱盲”。

不要以为脱盲是很容易的事情。你看，所有人出生的时候，都天然“文盲”；人们要上好多年的学，才能够真正“脱盲”——仔细想想吧，小学毕业的时候，所有人就真的彻底脱盲了吗？

以中文脱盲为例，学字的同时，还要学习笔划；为了学更多的字，要学拼音，要学如何使用《新华字典》……

学会了一些基础字之后，要学更多的词，而后在练习了那么多造词造句之后，依然会经常用错……你看，脱盲，和阅读能力强之间距离很长呢；不仅如此，阅读能力强和写作能力强之间的距离更长……

反复阅读这一部分的结果是：

- 你对基本概念有了一定的了解
- 你开始有能力相对轻松地阅读部分官方文档
- 你可以读懂一些简单的代码

仅此而已。

心理建设

当我们开始学习一项新技能的时候，我们的大脑会不由自主地紧张。可这只不过是多年之间在学校里不断受挫的积累效应——学校里别的地方不一定行，可有个地方特别行：给学生制造全方位、无死角、层层递进的挫败感。

可是，你要永远记住两个字：

别怕！

用四个字也行：

啥也别怕！

六个字也可以：

没什么可怕的！

我遇到最多的孱弱之语大抵是这样的：

我一个文科生.....

哈哈，从某个层面望过去，其实吧，编程既不是文科也不是理科..... 它更像是“手工课”。你越学就越清楚这个事实，它就好像是你做木工一样，学会使用一个工具，再学会使用另外一个工具，其实总共就没多少工具。然后，你更多做的是各种拼接的工作，至于能做出什么东西，最后完全靠你的想象力.....

十来岁的孩子都可以学会的东西，你怕什么？

别怕，无论说给自己，还是讲给别人，都是一样的，它可能是人生中最重要鼓励词。

关于这一部分内容中的代码

所有的代码，都可以在选中代码单元格（Code Cell）之后，按快捷键 `⇧ ⌘` 或 `⇧ ⌥` 执行，查看结果。

少量执行结果太长的代码，其输出被设置成了“Scrolled”，是可以通过触摸板或鼠标滑轮上下滑动的。

为了避免大量使用 `print()` 才能看到输出结果，在很多的代码单元格中，开头插入了以下代码：

```
from IPython.core.interactiveshell import InteractiveShell
InteractiveShell.ast_node_interactivity = "all"
```

你可以暂时忽略它们的意义和工作原理。注意：有时，你需要在执行第二次的时候，才能看到全部输出结果。

另外，有少量代码示例，为了让读者每次执行的时候看到不同的结果，使用了随机函数，为其中的变量赋值，比如：

```
import random
r = random.randrange(1, 1000)
```

同样，你可以暂时忽略它们的意义和工作原理；只需要知道因为有它们在，所以每次执行那个单元格中的代码会有不同的结果就可以了。

如果你不是直接在网站上浏览这本“书”、或者是在阅读印刷版，而是在本地自己搭建 Jupyterlab 环境使用，那么请参阅附录《[Jupyterlab 的安装与配置](#)》。

注意：尤其需要仔细看看《[Jupyterlab 的安装与配置](#)》的《关于 Jupyter lab themes》这一小节——否则，阅读体验会有很大差别。

另外，如果你使用的是 [nteract](#) 桌面版 App 浏览 [.ipynb](#) 文件，那么有些使用了 `input()` 函数的代码是无法在 [nteract](#) 中执行的。