作为算法竞赛的选手来说， 如果不会Python就太可惜了。 即使不打算用它编写任何大规模的程序， 快速开发实用小程序的能力往往也能让你做事效率大幅度提高。

多数算法竞赛 和OJ（包括UVa 和LA)都不支持Python, 但主流的Linux操作系统都预装了Python。考虑到多数竞赛 的比赛环境都是Linux, 选手完全可以用Python编写各种小巧的实用工具来辅助自己。

如何学习Python?很简单， 直接写程序。只要写出的程序符合自己的需要， 就不用深究其中的道理一除非你很好奇。这是一门实践性很强的语言，很多时候都可以 “望文生义 ” 。更棒的是，有一个称为REPL（交互命令行）东西可以帮助你即学即用一每敲一行代码就能马上看到执行结果，无须编译。[[1]](#footnote-1)

在算法竞赛中， 用Python编写数据生成器是再合适不过的了。下面的代码生成一个长度在5 ~10之间， 恰好包含一个大写字母， 其他字符为小写字母的串。

>>> from random import\*

>>> from string import\*

>>> L = randint(4, 9)

>>> s =''.join([choice(lowercase) for i in range(L)])

>>> p = randint(O, len(s))

>>> s[:p] + choice(uppercase) + s[p:]

' bdsqVgke'

原理是这样的：首先随机选择一个4~9之间（包括4和9)的整数L 作为大写字母的个数， 然后随机生成一个长度为L 的大写字符串。这是如何做到的？首先， 我们随机生成一 个长度为L 的列表[[2]](#footnote-2)（list)， 其中每个元素都是一个大写字母， 如下所示。

>>> [choice(lowercase) for i in range(S)]

['f','r','k','r','d']

这个语法叫做列表解析Clist comprehension)， 是一种构造列表的简单方法。range(5) 生成列表[0,1,2,3,4]。这条语句用大白话说就是 “对千列表[0,1,2,3,4]中的每个数i' 调用一次 choice(lowercase)，把结果拼成一个列表” 。接下来用join函数把列表里的字符串连接起来。

S.join(L)的作用是把字符串列表L中的各个字符串拼接起来， 用字符串S分隔。[[3]](#footnote-3)最后随机出小写字符出现的位置p, 然后插入到大写字母串中。s[a:b]代表列表或者字符串的第a个元素到第b-1 个元素。a和b都可以省略，a默认为O, b默认为列表长度。换句话说，L2 = L[：］的作用是创建一个和L 一样的数组。由千Python 的所有值都是引用类型的， 因此L2=L只是把L中保存的引用拷贝到了L2。

>>> a,b=l,2

>>> a= [l,3,7]

>>> b=[l,3,7]

>>> c,d=a,a[:]

>>> id (a), id (b), id (c), id (d)

(12598008, 12517288, 12598008, 12517208)

其中 id(a)返回的是变量 a 保存的引用。 建议读者不要在代码中使用这个函数， 仅用它来更好的理解 python。刚才的实验还证实了多重赋值的合法性， 因此可以简单地用 a,b=b,a 来交换两个变量a和b。

Python 也支持 FP （虽说和C＃比较起来， Python 的 FP 实在有点逊色）。 比如，计算 30!可以采用如下写法(j) :

>>> from functools import\*

>>> reduce(lambda x,y: x\*y, range(l,31))

265252859812191058636308480000000L

类似的东西还有很多。 总之， 当你熟练掌握 Python 之后， 能大大缩短编写数据生成器、对拍器、 “猜想验证器” 等小程序的时间， 平时还能用它编写各种实用程序呢。 事实上， Python 是笔者最常用的编程语言之一，在工作中曾用 Python 做过桌面GUI程序、网站、数据处理程序、分布式任务调度器以及各类程序的插件（如 Blender) 。

Python 也有很多不足，但随着 pyinstaller、SWIG、pypy 和 execnet 等优秀项目的出现和发展， 很多问题都得到了解决或者弱化。 尽管还有一些不如人意的地方气但它的简单实用是毋庸置疑的。

1. 专业人士一般用 ipython 代替这个自带的命令行， 但初学阶段不必如此。 [↑](#footnote-ref-1)
2. 类似于 C的数组， 但长度可变， 而且每个元素的类型不必相同。 [↑](#footnote-ref-2)
3. 在上面的代码中， S是由两个单引号组成的空串， 而不是单个双引号。 [↑](#footnote-ref-3)