



链滴

python 正式学习第一个月

作者: [yuanhenglizhen](#)

原文链接: <https://ld246.com/article/1554036624476>

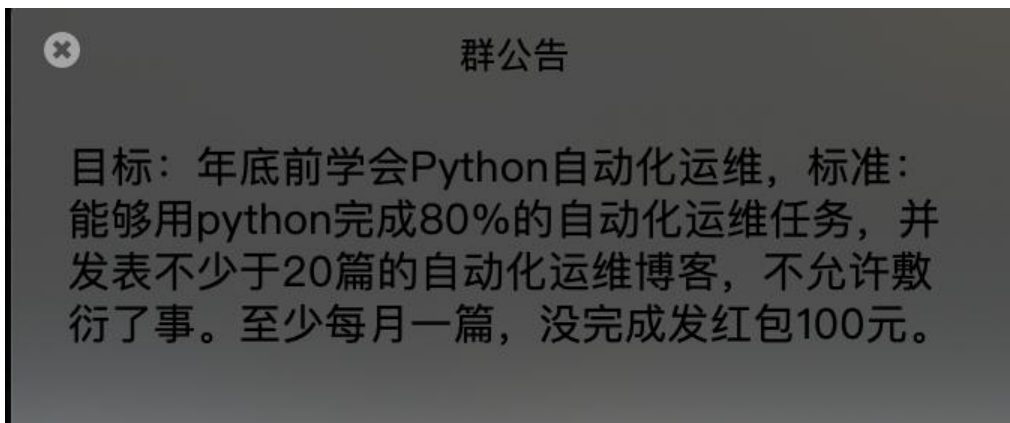
来源网站: 链滴

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

1. 起源

一次群里闲聊，加入了由阳光发起的python自动运维学习的群。起初，阳光死活不让我加入群，因为他知道我有一点编程的基础，觉的不适合一起学习。我硬是死皮赖脸的，让他把我拉到群里里面。

加群的第一天，阳光和我说了规则，具体规则如下：



我很慌，万一落下一次，我需要发一个4个人的300元红包，作为一次惩罚。因为钱少了，不会心疼。多了，反而起不到作用。300元还是属于能接受的范围。

就这样我加入了这样的一个学习团队中去，目前群李只有四个，我、阳光、老李头、还有个记不得名大佬。

2. 学习总结

我学习的方式是按着《python编程-让运维工作自动化》这边书，阅读，做习题的方式来的

2.1 书目大纲

第一部分 Python编程基础

第1章 Python基础

第2章 控制流

第3章 函数

第4章 列表

第5章 字典和结构化数据

第6章 字符串操作

第7章 模式匹配与正则表达式

第8章 读写文件

第9章 组织文件

第10章 调试

第11章 从Web抓取信息

第12章 处理Excel电子表格

第14章 处理CSV文件和JSON数据

第15章 保持时间、计划任务和启动程序

第16章 发送电子邮件和短信

第17章 操作图像

第18章 用GUI自动化控制键盘和鼠标

2.2学习过程

1. Python 中的变量不需要声明.每个变量在使用前都***==必须赋值==***, 变量赋值以后该变量才被创建。这一点和js语法有点区别,注意区分

在 Python 中, 变量就是变量, 它没有类型, 我们所说的"类型"是变量所指的内存中对象的类型。

6种数据类型

**> **Number (数字)

**> **String (字符串)

**> **List (列表)

**> **Tuple (元组)

**> **Sets (集合)

**> **Dictionary (字典)

Python3 支持 int、float、bool、complex (复数) 。常见就是==int float bool==

可以通过使用del语句删除单个或多个对象,用逗号分隔

元组 (tuple) 与列表类似, 不同之处在于元组的元素不能修改。元组写在小括号(())里, 元素之间用号隔开。

列表和元组类似javascript中的array

字典是一种映射类型, 字典用"{}"标识, 它是一个无序的键(key): 值(value)对集合。

注意:

>>1、字典是一种映射类型, 它的元素是键值对。

>2、字典的关键字必须为不可变类型, 且不能重复。

>3、创建空字典使用 {}。

表达式是由 值和操作符组成的结合,所以的表达式都求值为一个值。

表达式需要有重新赋值才能改变变量的值

```
str(99)*3
```

```
>'999999'
```

#学习编程的一定会学的语句``

```
from lufeipython.Chapter1.user_input import name
```

```
print("hello Python! ")
```

2+2是表达式也称为归约

2**2即2的2次方，也是表达式运算优先级最高的

%取余数

有趣的是python的除法

三个除法的概念 传统除法/精确除法/地板除法

1、传统除法：整数相除结果是整数，浮点数相除结果是浮点数

2、精确除法：无论操作数是整数还是浮点数，得到的商总是真实结果，总是得到浮点数

3、地板除：无论操作数是整数还是浮点数，得到的商不保留浮点数的尾数

在python2.6版本中4/2是等于2的 只有3版本的时候结果是2.0

_### 数据类型_1.整型 (int)

2.浮点型 (float)

3.字符串

用于字符串的时候表示复制 只能用于字符串和数字，不能用strstr

比如"li"*5输出 "lililili"

变量名区分大小写

整数可以与浮点型的值相等

round 函数四舍五入取值

_### 布尔型_值只有True/False

小写的会报错

比较操作符 比较两个值，求值为一个布尔值

布尔操作符

and 与 or 或 not 非 需要注意的话 这边和javascript以及其他部分语言有区别 || && !

_### 操作流的元素_操作流的元素开始部分通常为"条件"，接下来是一个代码块

```
passwd: str = input()
```

```
if name == "mufengs":  
    print("hello mufengs")
```

```
if passwd == "233":
    print("access granted")
elif passwd == "234":
    print("good")
else:
    print("wrong passwd")
```

while 语句_### break 语句_###continue语句_像break一样, continue语句在while语句内部行

_###for循环和round()函数_代码执行有限次数的时候选用 当然也有办法用while实现

类真类假的值可以让代码的可读性增强

等价于for的while循环方式

```
print('my name is')
o = 0
while o < 5:
    print('Jimmy Five Times(' + str(o) + ')')
    o = o + 1
```

range()的开始/结束/步长

如 range(0,10,2)

_### from import 语句_如from random import *

用sys.exit()提前结束程序

_## 第二章习题_等于操作符和赋值操作符的区别是什么

==是等于操作符, 它比较两个值, 求值为一个布尔值, 而=是赋值操作符将值保存在变量中

Round函数返回一个数值,该数值是按照指定的小数位数进行四舍五入运算的结果

abs() 函数返回数字的绝对值。

_## 第三章函数_def() 参数是变元也就是其他语言中的变量

_### 3.3 None值_在python中有一个值为None, 他表示没有值

```
a = print("1111")
```

```
print(a)
```

上面的例子可以明确的看出来 print是不可以参与值的赋值运算的

```
print("23", end="...")
print("456")
```

23...456 #修改默认换行的

```
print('cats', 'dogs', 'mice', sep="...")
```

cats...dogs...mice sep表示修改默认的分割符号, 默认是空格

_### 3.5 局部变量不能在全局作用域下使用_如下例子所示

```
def spam():
    eggs = 31337
spam()
print(eggs)

# Traceback (most recent call last):
#   File "<input>", line 1, in <module>
# NameError: name 'eggs' is not defined
```

_### 3.6 global语句_如果想在函数中修改全局变量中的存储的值，就必须对该变量使用global语句

_### 3.7 异常处理_try和except语句

如下面的try

_## 第三章小姐_函数是将代码逻辑分组的主要方式

_## 第四章列表_列表是一个值，它包含多个字构成的序列

在一个切片中，第一个整数是切片开始处的下标，第二个整数是切片结束处的下标

列表多重赋值

变量的数量和列表的长度必须严格相等，否则出现ValueError

```
cat = ['name', 'age', 'weight']
name, age, weight = cat
```

类似列表 字符串和元组

元组 tuple (1,2,3)

列表 list [1, 2, 3]_###copy()和deepcopy()_copy()可以用来复制列表和字典这样的可变值，而不是复制引用

如果复制列表中包含列表，就需要使用copy.deepcopy()这个函数来代替

_## 第五章 字典和结构化数据_字典类似与列表，但是没有下标

可以参考js的对象，区别是调用的时候一个用. 一个用[]

键值对输入的顺序不重要和列表的区别，列表的顺序是重要的

_## 第六章 字符串操作_多行注释可以使用 """ """

1.程序设计和数据结构

2.处理命令行参数

_## 第七章 模式匹配与正则表达式_强口令检测

2.3 学习心得体会

其实学习编程，以前什么都不知道反而好学一点，因为你会发现，在学习得过程中会不断得对比，哎

xxx语言中是这么用，python中怎么不一样，有容易记混淆得可能性。打个比方，如js中得或与非，用|| && ! 这些表示的，而在python中你需要用 or and not来表示。你可以不在句子的结尾写；但你的缩进一定要正确。这一个月，恶补了一些python 的基础知识，比如元组，列表，字典等等。也过一些简单的python程序，必须用python实现根据isbn号查书目信息，scrapy爬取豆瓣分类书目信等。今天是这个月的最后一天，继续加油！

3. 下一阶段学习计划

准备基于jumpserver二次开发，加入nginx配置管理，数据库管理，svn管理（虽然实现了，但是过简单只是基于文件修改的）等功能

github地址：<https://github.com/mufengcoding/PythonOps>

有想入群聊一起学习的，可以加我微信Do8080