

# Python

Lu, Phil

<http://www.liaoxuefeng.com/>

## 1. Day One: Basics

### 1.1. 输入输出

```
name = input('please enter your name: ')
print('hello,', name)
```

### 1.2. 变量类型

```
n = 123                # 整数
f = 456.789            # 浮点
a = True and False or True # 布尔
b = None               # 空
s1 = 'Hello, world'    # 字符串
s3 = r'Hello, "Bart"'   # r 为默认不转义
s4 = r'''Hello,
Lisa!'''               # ''' 内为自动加上换行符
```

### 1.3. 编码

一般脚本前加上这两行

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
```

## 1.4. list & tuple

```
classmates = ['Michael', 'Bob', 1.23, ['asp', 'php']]    #list
数据类型
len(classmates)
classmates[0]
classmates[-1]
classmates.append('Adam')    # 加在末尾
classmates.insert(1, 'Jack')    # 加在指定位置
classmates.pop(1)    # 删除制定元素并返回，默认为最后一个

t = (1, 2)    #tuple 数据类型，一旦指定不能改变
t = ('a', 'b', ['A', 'B'])    # 指向一个 list，就不能改成指向
其他对象
t[2][0] = 'X'    # 但指向的这个 list 本身是可变的
```

## 1.5. if

```
age = 20    #nothing special
if age >= 6:
    print('teenager')
elif age >= 18:
    print('adult')
else:
    print('kid')
```

## 1.6. 循环

```
names = ['Michael', 'Bob', 'Tracy']# 第一种，for 只有
for...in... 这种写法
for name in names:
    print(name)

while n > 0:# 第二种
```

```
sum = sum + n
n = n - 2
```

## 1.7. dict & set

```
d = {'Michael': 95, 'Bob': 75, 'Tracy': 85} #dict 数据类型
d['Michael'] = 80 # 修改
'Thomas' in d # 返回 False
d.get('Thomas') # 返回 None
d.pop('Bob') # 删除元素并返回
```

和 list 比较, dict 有以下几个特点:

- 查找和插入的速度极快, 不会随着 key 的增加而变慢;
- 需要占用大量的内存, 内存浪费多
- 要保证 hash 的正确性, 作为 key 的对象就不能变
- 在 Python 中, 字符串、整数等都是不可变的, 因此, 可以放心地作为 key。而 list 是可变的, 就不能作为 key

还有一种数据类型 set, 类似于数学中的集合, 反正我没用过

## 1.8. 函数

- 一些内置函数

```
abs(-20)
int('123')
str(1.23)
a = abs # 变量 a 指向 abs 函数
a(-1)
```

- 定义函数

```
import math

def move(x, y, step, angle=0):
    nx = x + step * math.cos(angle)
```

```
ny = y - step * math.sin(angle)
return nx, ny      # 实际上是返回了一个 tuple
```

定义函数时，需要确定函数名和参数个数；  
 如果有必要，可以先对参数的数据类型做检查；  
 函数体内部可以用 return 随时返回函数结果；  
 函数执行完毕也没有 return 语句时，自动 return None。  
 函数可以同时返回多个值，但其实就是一个 tuple。

- 函数参数

参数定义的顺序必须是：必选参数、默认参数、可变参数、命名关键字参数和关键字参数。

```
d = {'Michael': 95, 'Bob': 75, 'Tracy': 85} #dict 数据类型
d['Michael'] = 80 # 修改
'Thomas' in d # 返回 False
d.get('Thomas') # 返回 None
d.pop('Bob') # 删除元素并返回

def f2(a, b, c=0, *args, d, **kw):
    print('a =', a, 'b =', b, 'c =', c, 'd =', d, 'kw =', kw)
f2(1, 2, d=99, ext=None)
a = 1 b = 2 c = 0 d = 99 kw = {'ext': None} #output
```

- \*args 是可变参数，args 接收的是一个 tuple；
- \*\*kw 是关键字参数，kw 接收的是一个 dict。