[**《编程小白的第1本Python入门书》**](https://www.cnblogs.com/xiaohei4/p/10848491.html)

**第一章 为什么选择Python ？**

1.1 Python追求的是找到最好的解决方案。

1.2 Python有很丰富的库，有官方的，也有第三方开发的，你想做的功能模块很有可能已经有人写好了，你只需要调用，不需要重新发明轮子。

**第三章 变量与字符串**

3.1 初学者经常会遇到的困惑是，看书或者听课都懂，但还是不明白要怎么编程。这是因为缺乏足够多的实践......学习时，模仿示例敲一遍代码。

3.2 变量 变量会暂时性地储存你放进去的东西。

3.3 字符串

* 单引号其实和双引号完全一样
* '''三个引号被用于过于长段的文字或者是说明，只要三引号不完你就可以随意换行写下文字'''
* 字符串格式化符format()

print('{} a word she can get what she {} for.'.format('With','came'))

print('{ab} a word she can get what she {cd} for.'.format(ab = 'With', cd = 'came'))

print('{a1} a word she can get what she {a2} for.'.format(a2 = 'came', a1 = 'With'))

# 这三个语句，输出结果一样:

With a word she can get what she came for.

With a word she can get what she came for.

With a word she can get what she came for.

**第四章 函数的魔法**

4.1 内建函数（自带函数）

Python3.7.3 版本有69个内建函数，具体使用说明见[官方文档](https://docs.python.org/3/library/functions.html)：

Built-in Function

|  | **Built** | **in** | **Function** |  |
| --- | --- | --- | --- | --- |
| abs() | delattr() | hash() | memoryview() | set() |
| all() | dict() | help() | min() | setattr() |
| any() | dir() | hex() | next() | slice() |
| ascii() | divmod() | id() | object() | sorted() |
| bin() | enumerate() | input() | oct() | staticmethod() |
| bool() | eval() | int() | open() | str() |
| breakpoint() | exec() | isinstance() | ord() | sum() |
| bytearray() | filter() | issubclass() | pow() | super() |
| bytes() | float() | iter() | print() | tuple() |
| callable() | format() | len() | property() | type() |
| chr() | frozenset() | list() | range() | vars() |
| classmethod() | getattr() | locals() | repr() | zip() |
| compile() | globals() | map() | reversed() | \_\_import\_\_() |
| complex() | hasattr() | max() | round() |  |

4.2 开始创建函数

* def ：定义一个函数
* arg ：参数
* return ：返回结果
* 缩进 ：表名语句和逻辑的从属关系

def function(arg1, arg2):

return 'Sometihing'

def 函数名(参数1, 参数2):

缩进 return '结果'

求直角三角形斜长边：

def length\_bian(a,b):

c = (a\*\*2 + b\*\*2)\*\*(1/2)

return c

dd = length\_bian(3,4)

print(dd)

* 位置参数、关键词参数

de(1,2,3) # 位置参数

de(dup = 1, ddown = 2, dheight = 3) # 关键词参数

* open()函数

file = open('c://python/hello.txt','w')

file.write("HelloWorld!")

* 数学运算符

假设 a=10, b=20

| **符号** | **描述** | **实例** |
| --- | --- | --- |
| + | 相加 | a+b = 30 |
| - | 相减 | b-a = 10 |
| \* | 相乘 | a\*b = 200 |
| / | 相除 | b/a = 2 |
| % | 取余 | b%a = 0 |
| \*\* | 次方 | a\*\*b = 10^20 |
| // | 取商 | b//a = 2 |

**第五章 循环与判断**

5.1 逻辑判断True&False

1. 比较运算符

| **符号** | **描述** |
| --- | --- |
| == | 左右相等时返回True |
| != | 左右不等时返回True |
| > | 左边大于右边时返回True |
| < | 左边小于右边时返回True |
| >= | 左边大于等于右边时返回True |
| <= | 左边小于等于右边时返回True |

1. 成员运算符与身份运算符in&is

in、not in，表示归属关系的布尔运算符 is、is not，表示身份鉴别的布尔运算符

1. 布尔运算符

| **运算符** | **描述** |
| --- | --- |
| not x | 如果x是True，返回Flase |
| x and y | 如果x和y同时为True，返回True |
| x or y | 如果x和y同时为False，返回False |

5.2 条件控制if...else

# 如果...条件是成立的，就做...；反之，就做...

if condition:

do something

elif condition:

do something

else :

do something

if 成立条件（冒号）

（缩进）执行什么

elif 成立条件（冒号）

（缩进）执行什么

else（冒号）

（缩进）执行什么

5.3 循环（Loop）

* for循环

于...其中的每一个元素，做...事情

for item in iterable:

do something

for 元素 in 集合（冒号）

（缩进）执行什么

* 嵌套循环（Nested Loop）

for i in range(1,10):

for j in range(1,10):

print('{} \* {} = {}'.format(i,j,i\*n))

* while循环 只要...条件成立，就一直做...

while condition:

do something

while 成立条件（冒号）

（缩进）执行什么

如何控制while循环？

1. 在循环过程中制造某种可以使循环停下来的条件P69
2. 改变使循环成立的条件P70

**第六章 数据结构**

* 数据结构（Data Structure） 6.1 储存大量数据的容器，在Python称为内置数据结构（Build-in Data Structure）

6.2 Python有四种数据结构：列表，字典，元组，集合。

list = [val1, val2, val3, val4]

dict = {key1 : val1, key2 : val2}

tuple = (val1, val2, val3, val4)

set = {val1, val2, val3, val4}

6.3 列表（list）

1. 列表中的每一个元素都是可变的；
2. 列表中的元素是有序的；
3. 列表可容纳Python中的任何对象。

列表的增删改查

fruit = ['apple', 'orange']

fruit.insert(1,'grape') #增

fruit[0:0] = ['banana'] #增

fruit.remove('grape') #删

del fruit[0:2] #删

fruit[0] = 'Grapefruit' #改

fruit.extend('b1', 'b2') #增

6.4 字典（Dictionary）

1. 字典中数据必须以键值对的形式出现；
2. 逻辑上讲，键不能重复，值可以重复；
3. 字典中的键（key）是不可变的，而值（value）是可变的。

字典的增删改查

url\_code = {'b' : 'baidu', 's' : 'sina'}

#增

url\_code['t'] = 'tmall'

url\_code.update({'fb' : 'facebook', 'tb' : 'taobao'})

#删

del url\_code['fb']

6.5 元组（Tuple） 元组不可修改，类似稳定版的列表

6.6 集合（Set） 每一个集合中的元素是无序的、不重复的任意对象。可以通过集合去判断数据的从属关系，也可以通过集合去掉数据结构中的重复的元素。

a\_set = {1, 2, 3, 4}

a\_set.add(5) #增

a\_set.discard(5) #删

6.7 多重循环

sorted(num\_list,reverse = True) sorted函数，先将列表复制后按照长短、大小、字母顺序进行排序，但不改变列表本身。

num\_list [4,6,7,9,1,8,3,2]

print(sorted(num\_list))

整理列表时，如果同时需要两个列表怎么办？

for a,b in zip(num,str):

print(b, ' is ', a)

6.8 推导式

a = []

for i in range(1,10):

a.append(i)

b = [i for i in range(1,10)]

# 两种执行效果一样，第二种效率更高

# 看几个例子

a = [i\*\*2 for i in range(1,10)]

b = [j+1 for j in range(1,11)]

c = [n for n in range(1,10) if n % 2 == 0]

d = [letter.lower() for letter in 'ABCDEFG']

e = {i:i+1 for i in range(4)}

f = {i:j for i,j in zip(range(1,6),'abcde')}

g = {i:j.upper() for i,j in zip(range(1,6),'fghij')}

6.9 循环列表获取元素索引

letters = ['a','b','c','d','e']

for num,letter in enumerate(letters):

print(letter, ' is ', num+1)

**第八章 开始使用第三方库**

**8.1 令人惊叹的第三方库**

* 当你想搭建网站时，可以选择功能全面的Django、轻量的Flask等web框架；
* 当你想写小游戏时，可以使用PyGame框架；
* 当你想做一个爬虫时，可以使用Scrapy框架；
* 当你想做数据统计分析时，可以使用Pandas数据框架...
* 丰富的资源可以帮我们高效快捷地做到想做的事，就不需要再重新造轮子。