Python不需要将源代码编译成二进制码，你只需要直接从源代码运行该程序。在程序内部，Python会将源代码转换为称为字节码的中间形式，尔后再转换成你的电脑所使用的语言，并运行它。

Python同时支持面向过程编程和面向对象编程。

***可扩展性***

如果希望代码的某一部分快速的运行或者希望算法的某些部分不被公开，你可以在C或C++语言中编写这些程序，然后再将其运用到你的Python程序中。

***可嵌入性***

你可以在你的C或C++程序中嵌入Python，从而向你的程序用户提供脚本功能。

***注释***

使用#进行单行注释

***字面常量***

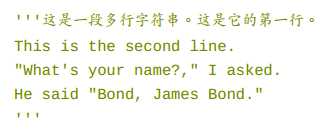
诸如，5、1.23这样的数字，或者是如“这是一串文本”或“This is a string”这样的文本。用这样的称呼是因为它们是字面上的------你用的就是它字面意义上的值或者内容。

***单引号、双引号***

被单引号包括的字符串和被双引号包含的字符串是一样的。

三引号

你可以通过使用三个引号------“””或’’’来指定多行字符串。也可以在三引号之间自由的使用单引号和多引号，比如，



***字符串是不可变的***

一旦创建了一个字符串，就不可能再改变它。

***格式化方法***

使用format()方法来格式化字符串

age = 20  
name = "Paul"  
print("{0} is {1} years old." . format(name, age))

输出

Hello, World!

Paul is 20 years old.

这里要注意，“{}”中的数字是可选项，去掉之后仍然会得到相同的输出结果。

还有更详细的使用方法



print()方法会默认的在最后使用“\n”来进行换行，可以使用end指定其应该以空白结尾。

print(‘a’, end=’’)

print(‘b’, end=’’)

输出结果为ab

或者通过end指定以空格结尾，

print(‘a’, end=’ ’)

print(‘b’, end=’ ’)

print(‘c’, end=’ ’)

输出结果为a b c

***转义序列***

使用反斜杠“\”进行转义。在一个字符串中，一个放置在末尾的反斜杠表示字符串将在下一行继续，例如，

“This is the first sentence.\

This is the second sentence.”

输出结果如下，

“This is the first sentence. This is the second sentence.”

***原始字符串***

如果你需要指定一些未经过特殊处理的字符串，比如转义序列，那么你需要在字符串前增加r或者R来指定一个原始(Raw)字符串，

***对象***

Python将程序中的任何内容称为对象(object)

input()函数来获取用户的输入。

int将一个数字字符转换成一个整数，比如将输入的数字转换成整数，int(input(“input a num:”))

***流程控制***

if—elif—else

每一个控制语句后面都有一个冒号，

rightNum = 20  
inputNum = int(input("Please input a number:"))  
if rightNum == inputNum:  
 print("Yes")  
else:  
 print("No")  
print("Exit")

while语句

rightNum = 20  
condition = True  
while condition:  
 inputNum = int(input("Please input a number:"))  
 if inputNum == rightNum:  
 print("Yes")  
 condition = False  
 elif inputNum > rightNum:  
 print("you should input a less number")  
 else:  
 print("you should input a bigger number")  
else:  
 print("Exit")

for循环

for i in range(1, 5):  
 print(i)  
else:  
 print("The for loop is over")

range()函数的值为1，2，3，4，不包括第二个参数，如果加上第三个参数，就表示步进为第三个参数，比如range(1,5,2)，会输出1，3。

range()每次只会生成一个数字，如果要获取全部数字，可以使用list()函数，例如list(range(5))，会输出[0,1,2,3,4]

Python中的中文，Python文件默认是ASCII编码方式，如果要在文件中的任何地方包括注释中输入中文，要在文件开头加上#coding=utf-8或者#coding=gbk。

***Python中函数的定义与调用***

#coding=utf-8  
#函数的定义与调用  
def say\_hello():  
 print("Hello!")  
say\_hello()

***函数参数***

def print\_max(a, b):  
 if a >= b:  
 print("前者大")  
 else:  
 print("后者大")  
  
print\_max(1, 2)

## global语句

Python中的默认参数值与PHP中的相同

## 关键字参数

给指定的关键字参数赋值，不用关心参数顺序。

def func(a, b=5, c=10):  
 print("a = ", a, "b = ", b, "c = ", c)  
  
  
func(1)  
func(25, c=20)  
func(c=50, a=100)

## 可变参数

def total(a=5, \*numbers, \*\*phonebook):  
 print('a', a)  
  
 for i in numbers:  
 print('i', i)  
  
 for key, value in phonebook.items():  
 print(key, value)  
  
total(50, 1, 2, 3, name=11, age=22)

第一个星号表示元组(Tuple)，第二个星号表示字典(Dictionary)

## return

每一个函数都在其末尾隐含了一句return None，除非你写了自己的return语句。

Python中的pass语句用于指示一个没有内容的语句块。

def some\_function():  
 pass

## DocStrings

文档字符串

# 模块

导入一个模块是一件代价高昂的事情，因此Python引入一些技巧使其能够更快速的完成。其中一种方式便是创建按字节码编译的文件，这一文件以.pyc为其扩展名。

### from..import语句

如果希望直接使用导入的变量和函数，不必每次都使用sys.等，可以使用该语句来实现。

from math import sqrt  
  
print("Square root of 16 is", int(sqrt(16)))

### 模块的 \_\_name\_\_

每个模块都有一个名称，而模块中的语句可以找到它们所处的模块的名称。这对于确定模块是独立运行的还是被导入进来运行的这一特定目的来说有用。

if \_\_name\_\_ == "\_\_main\_\_":  
 print("This program is being run by itself")  
else:  
 print("I am being imported from another module")

### 编写你自己的模块

每一个python程序同时也是一个模块，你只需要保证它以.py为扩展名即可。

例如，编写mymodule.py，导入使用其中的函数或变量

mymodule.py

def say\_hi():  
 print("hello, world!")  
  
\_\_VERSION\_\_ = 1.0

使用上述模块，使用import或者from…import

from math import sqrt  
#import mymodule  
from mymodule import say\_hi, \_\_VERSION\_\_  
  
print("Square root of 16 is", int(sqrt(16)))  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 print("This program is being run by itself")  
else:  
 print("I am being imported from another module")  
  
say\_hi()  
print(\_\_VERSION\_\_)

或者使用from mymodule import \*，该方式会导入公共名称，但不会导入\_\_VERSION\_\_名称。

### dir函数

能够返回由对象所定义的名称列表。如果这一对象是一个模块，则该列表会包括函数内所定义的函数、类与变量。

该函数接收参数。如果参数是模块名称，函数将返回这一指定模块的名称列表。如果没有参数，函数将返回当前模块的名称列表。

## 包

包是指一个包含模块与一个特殊的\_\_init\_\_.py文件的文件夹，后者向Python表明这一文件夹是特别的，因为其包含了Python模块。

# 数据结构

## 列表

python中有四种内置的数据结构------列表(List)、元组(Tuple)、字典(Dictionary)和集合(Set)。

列表应该用中括号括起来。

shoplist = ['mango','orange','apple','banana']  
print(len(shoplist))  
  
for item in shoplist:  
 print(item + " ")  
  
del shoplist[0]  
  
for item in shoplist:  
 print(item + "--")

shoplist.append('pear')

shoplist.sort()

## 元组

元组和列表很相似，但是元组是不可变的。

元组的定义是使用一对圆括号，在括号中使用逗号进行分割，当然括号可有可无，但是建议总是使用。

zoo = ("cat", "dog", "tiger")  
newzoo = "lion","penguin", zoo  
  
print(newzoo[2])  
print(newzoo[2][2])

结果

('cat', 'dog', 'tiger')

tiger

注意：如果元组中没有元素，要写成zoo=(),如果只有一个元素要写成zoo=(element,)最后要加上“,”，这样才能告诉python该变量是一个元组而不是一个被括号包裹的对象。

## 字典

字典保存的是键值对，并且键值必须是唯一的。

注意：你只能使用不可变的对象(比如string)来作为字典的键。

定义形式：d={key1:value1, key2:value2}

删除一对键值对：del d[‘key1’]，添加一个键值对：d[‘key3’]=’value3’

## 序列

shoplist = ['apple','mango','carrot','banana']  
name = 'swaroop'  
  
print('Item 1 to 3 is ', shoplist[1:3])  
print('Item 2 to end', shoplist[2:])  
print('Item -1 ', shoplist[-1])

输出结果

('Item 1 to 3 is ', ['mango', 'carrot'])

('Item 2 to end', ['carrot', 'banana'])

('Item -1 ', 'banana')

可以在切片操作中提供第三个参数，表示步长，默认是1，shoplist[::2]

## 集合

集合Set是无序的。可以通过in来判断元素是否在集合中。

bri = set(['brazil','russia','india'])  
print('india' in bri)

输出true

bri = set(['brazil','russia','india'])  
bricopy = bri.copy()  
bricopy.add('china')  
print(bricopy.issuperset(bri))  
  
print('india' in bri)

还有remove方法

直接将一个列表付给另一个列表，只是浅拷贝，要想深拷贝，可以使用listcopy=list[:]来进行。

# 面向对象编程

class Person:  
 def say\_hi(self):  
 print("Hello")

定义一个类。

\_\_init\_\_方法，会在类的对象被实例化时立即运行。

class Person:  
 def \_\_init\_\_(self, name):  
 self.name = name

## 类变量与对象变量