目录

[运行Python程序的方法 1](#_Toc406145152)

[输入和输出 1](#_Toc406145153)

[输出 1](#_Toc406145154)

[输入 1](#_Toc406145155)

[Python变量在内存中的变化 1](#_Toc406145156)

[常量 2](#_Toc406145157)

[list[ ]的用法 2](#_Toc406145158)

# 运行Python程序的方法

* 新建一个XXX.py文件，并在里面输入代码。
* 进入命令控制台(cmd)，切换到XXX.py文件所在的目录下.
* 输入：python XXX.py

# 输入和输出

## 输出

* 单个字符串输出：print ‘XXXX’
* 多个字符串输出：print ‘XXX’,’xxx’,’XXX’(每次遇到“,”号都会输出一个空格)

## 输入

* raw\_input( )函数可以让用户输入：

name = raw\_input( )

name = raw\_input(‘please enter your name: ‘ )

# Python变量在内存中的变化

a = 'ABC'

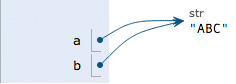
b = a

a = 'XYZ'

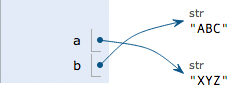
print b

py-var-code-1

执行b = a，解释器创建了变量b，并把b指向a指向的字符串'ABC'：



执行a = 'XYZ'，解释器创建了字符串'XYZ'，并把a的指向改为'XYZ'，但b并没有更改：



所以，最后打印变量b的结果自然是'ABC'了。

# 常量

* 在Python中，通常用全部大写的变量名表示常量

PI=3.14444444

但事实上PI仍然是一个变量，Python根本没有任何机制保证PI不会被改变，所以，用全部大写的变量名表示常量只是一个习惯上的用法，如果你一定要改变变量PI的值，也没人能拦住你。

# list[ ]的用法

* 列表是有序的集合，可以随时添加和删除其中的元素
* list中的元素的数据类型也可以不同
* list中的元素也可以是另一个list
* 生成一个列表：

classmate = [‘a’,’b’,’c’]

* 获取列表元素的个数：

len(classmates)

* 访问列表元素靠索引：

classmates[0]

* 访问列表最后一个元素：

classmates[-1]

* 按倒序访问列表元素：

classmates[-2] 访问倒数第2个元素

* 往list中追加元素到末尾

classmates.append(‘xxx’)

* 往list中追加元素到指定的位置

classmates.insert(1,’XXXX’) classmates[1]=’XXXX’

* 删除list末尾的元素：

classmates.pop()

* 删除指定位置的元素：

classmates.pop(i) //会返回该位置上的值。

# tuple( )的用法

* tuple一旦初始化就不能修改
* 他没有append(),insert()这样的方法
* tuple的一个陷阱：

当你定义一个tuple时，在定义的时候，tuple的元素就必须被确定下来。

* t=(1,2), t=()表示定义了一个空的元组。

但当你定义t=(1)是，会产生歧义，因为他还可以表示数学公式中的小括号。所以为了消除歧义，采用以下方法：

* t=(1,)
* 有一点很重要：

虽然tuple