1. **基础知识**
2. 安装Python [www.python.org/download](http://www.python.org/download)
3. Windows Python2.5.x windows installer
4. Linux和Unix 大部分已经默认存在Python 在命令行 输入python 启动交互式python解释器
5. Python 不用分号 表示一行的结束 默认加上也没问题
6. 算法 告诉计算机做什么
7. 数字和表达式

默认为整数除法，参与除法的两个数中有一个数为浮点数 则运算结果为浮点数

如果希望Python 只执行普通的除法 可以加上 from \_\_future\_\_ import division

Python 提供另外一种 双斜线 实现整数除法 //

取模运算 %

幂运算 \*\* 2\*\*3 相当于 2的三次方 pow(2,3) 比取反的运算级高

**变量:**  x=3 在Python 中变量不需要 加$ 变量包括 数字 英文字母 下划线 不能以数字开头。**变量不能用关键字命名；**

表达式 可以比喻成一件事 语句 就是做一件事 print语句 和 赋值语句

**获取用户输入:**

Input (“what’s your telephone”)

raw\_input();

If 1==2 : print no;

**函数:**

像pow abs 这种标准函数称为内建函数。

**模块：**可以把模块想象成导入python的扩展功能 需要使用命令import 完成

Import math

Math.floor(1)

From math import floor 这种方法不能有同名函数

Ceil 大于或等于它的最小整数

可以使用变量 来引用函数 foo = math.floor PHP也可以

cmath 复数模块 此处跳过了

\_\_future\_\_ python 时光机 非标准python功能 可能在以后引入的功能

**新建可运行的程序：**以.py 结尾

若想直接运行python代码 可以再脚本的 头部 加上 #！/usr/bin/python2

**注释：**# PHP中为 # /\*\*/ //

**转义：** 在python中 双引号并不解释变量和单引号没有什么区别 \

**字符串：**长字符串和原始字符串，在python中 连着写两个字符串会默认拼接在一起如 “let’s go ” ‘”hello world”’ 变量拼接 类似 JS 用+ 号拼接

Str函数 它会把值转换为合理形式的字符串 reper函数 以合法的Python表达式的形势来表示值

Number = 42

Print ‘this is test’+number 这样是报错的 PHP支持这样写

Print ‘this is test’+`number` 这样是正确的 在python3.0 中已不使用 反引号 ``

print 'this number is '+repr(number) 应保持这样的代码

str ,repr,反引号 是 将python值转换为字符串的三种方法

raw\_input 是读取用户输入的原始字符串

长字符串 ‘’’ 类似 PHP的定界符 <<<# #;

原始字符串 print r’C:\\nwindows’ 请注意 r 原始字符串 最后一个字符不能使\ 除非对反斜杠进行了转义

**小结：算法，表达式，变量，语句，函数，模块程序，字符串。**

1. **列表和元组**

数据结构：通过某种方式组织在一起的数据元素的集合。

最基本的数据结构是序列，序列中每个元素被分配一个序号即元素的位置，也成为索引。第一个索引是0，第二个索引是1 以此类推。

Python 包含6中内建的序列。列表，元组，字符串，Unicode字串，buffer对象和xrange对象。

列表可以修改，元组不能修改。内建函数一般会返回元组。使用元组作为字典。

Xl = [‘fate’,27] 序列用中括号表示。序列也可以包含其他的序列。

容器的数据结构。序列和映射是两类主要的容器。

1. **通用的序列操作：加，乘，索引，分片，检查某个元素是否属于序列的成员。迭代。**
2. **索引：xl = ‘abc’;**

**Xl[0]**

**字符串就是一个由字符组成的序列。使用复数索引，Python会从右边开始**

**Xl[-1**

**字符串字面值（序列字面量亦可）能够直接使用索引，而不需要一个变量引用。**

**‘hello’[4] o**

**>>> raw\_input("what's your name ")[3]**

**what's your name Fate**

**'e'**

1. **分片：分片通过冒号隔开的两个索引来实现。**

**Tag = [‘abcdeefg’]**

**Tag[2:5] 分片的索引边界 第一个边界包含在分片内 第二个不包含在分片内**

**只要分片中最左边的索引比它右边的晚出现在序列中，结果就是一个空序列**

**优雅的捷径 tag[-3:] tag[:3]可以置空 复制一个序列 可以两边都置空[:]**

1. **更大的步长 tag[0:6:2] 步长不能为0 但是可以是复数。将从右到左提取。**

**>>> numbers = [1,2,3,4,5,6,7,8,9,10]**

**>>> numbers[0:8:-2]**

**[]**

**>>> numbers[0:8]**

**[1, 2, 3, 4, 5, 6, 7, 8]**

**>>> numbers[8:0:-2]**

**[9, 7, 5, 3]**

**>>>**

1. **相加：通过使用加运算符可以进行序列的连接操作。**

**[1,2,3] +[4,5,6] -> [1,2,3,4,5,6]**

**列表和字符串是无法连接在一起的 尽管它们都是序列 两种相同类型的序列才能进行连接操作。**

1. **乘法：用数字乘以一个序列会生成新的序列，而在新的序列中，原来的序列被重复X次**

**>>> 'python'\*5**

**'pythonpythonpythonpythonpython'**

**[42]\*10 [42,42,42,42,42,42…]**

**NONE python内建值 空列表 [] 初始化 [NONE]\*10**

1. **成员资格：检查一个值是否在序列中，可以使用in 运算符**

**>>> user = raw\_input("what's your name:"); user in users;**

**what's your name:fate**

**True**

**>>>**

**If [a,b] in [[a,b],[c,d]]: print yes**

1. **长度，最大值，最小值**

**Len([1,2,3]) max([4,5,6]) min([7,8,9])**

1. **列表 列表专门的方法**

**List 函数 把序列类型转换成列表 list(‘hello’) = [‘h’,’e’,’l’,’l’,’o’]**

**列表的基本操作**

1. **元素赋值:x = [1,1,1]**

**X[1] = 2;**

1. **删除元素**

**names = [‘fate’,’aa’,’bb’];**

**Del names[0]; del 不仅仅用于 删除列表元素 还能 用于删除其他元素。**

1. **分片赋值：**

**Name = list(‘python’);**

**Name[-3:] = list[‘php’];**

**分片赋值时，可以使用与原序列不等长的序列将分片替换**

**Name = list(‘perl’);**

**Name[1:] = list(‘ython’);**

**分片赋值时，可以在不需要替换任何原有元素的情况下插入新的元素。**

**Numbers = [1,5];**

**Numbers[1:1] = [2,3,4];**

**Numbers -> [1,2,3,4,5,]**

**分片赋值时 删除**

**numbers = [1,5];**

**numbers[1:1] = [2,3,4]**

**numbers[1:-1] = []**

**print numbers;**

**列表的方法：**

**方法是一个与某些对象紧密联系的函数。对象被放在方法名之前 并用点好连接**

1. **append();用于在列表的末尾追加新的对象。直接修改原列表 而不是简单地返回一个修改过的新列表。**

**numbers.append(6);**

1. **count() 统计某个元素在列表中出现的次数。**

**numbers.append('6,6');**

**print numbers;**

**print numbers.count(6);**

**（3） extend 方法可以在列表的末尾一次性追加一个序列的多个值。可以扩展原有的列表。修改了 被扩展的序列 原始的连接操作 会返回一个全新的列表 不会改变原有的列表。**

**numbers.extend([2,3,4]);**

**（4）index 方法用于从列表中找出某个值第一个匹配项的索引位置。**

**numbers.index(4)**

**（5）insert 方法 用于将对象插入到列表中 只能插入一个元素**

**Numbers.insert(3,’four’) 第一个参数为列表索引 第二个参数为要插入的对象**

**（6） pop 方法 移除列表中的一个元素 默认是最后一个 并且返回该元素的值**

**X = [1,2,3]**

**x.pop();**

**x.pop(0) 0 指定索引位置。**

**利用 pop 方法可以实现一种常见的数据结构 栈 LIFO last in first out**

**入栈 append 出栈 pop**

**队列 FIFO**

**( 7 ) remove 移除列表中某个值的第一个匹配项。**

**Numbers.remove(2);**

**（8） reverse 方法 将列表中的元素反向存放。 返回一个迭代器对象**

**Numbers.reverse();**

**（9）sort**

**Y = x[:]**

**y.sort();**

**x = y**

**y.sort();**

**numbers.sort(len)**

**numbers.sort(key=len)**

**numbers.sort(revers=false)**

**元组：不可改变的序列 与列表唯一的不同 是不可以修改**

**创建元组 ： 1,3,4用逗号分隔一些值 自动创建了元组**

**元组 用圆括号括起来**

**空元组：（）**

**一个值得元组：（1，）**

**只有圆括号 没有 逗号是没有用的**

**元组 可以映射中当做键使用 而列表不行**

**元组作为很多内建函数和方法的返回值使用**

**第三章 使用字符串**