

查看自己电脑装的python是哪一个版本的方法：

windows系统下是cmd里输入python -v

mac里terminal直接输入python

python 2.7输出含有中文字符：print '品牌id'.decode('utf-8')

一，基本语法：

1，注释

①python中的注释以 “#” 开头，注释和代码之间至少要有两个空格

②如果注释中有中文，则需要在中文注释的前一行写上下面这句话：

```
 -*- coding:utf-8 -*-
```

③文档字符串，文档字符串是包、模块、类、函数里面的第一个语句

使用三个三重双引号组织：“""" 注释内容 """”，注释内容可以通过对象的_doc_成员
自
动提取

2，行与缩进

使用缩进来代表代码块，最好使用四个空格进行悬挂式缩进，并且同一个代码块的语句，必须含

有相同的缩进空格数

注意：在python中使用缩进时，绝对不要使用tab，也不要使用tab与空格混用

3，语句换行

python建议每行代码的长度不要超过80个字符，对于长的代码，建议进行换行，我们可以根据python会将圆括号（（））、中括号（[]）、大括号（{}）中的行隐式连接起来的这个特点，将过长的语句换行显示，并在语句外侧添加一对圆括号

注意：在（）、[]、{}中的语句需要换行，不需要再使用圆括号进行换行

二，变量与数据类型：

1，变量和赋值：变量的类型和值在赋值的那一刻被初始化，变量的赋值是通过=实现的

2，变量的类型：1）数字类型：整型、浮点型、复数类型

2）布尔类型:true和false，在进行数值运算时，true=1;

false=0;

3) 字符串类型：字符串被定义为一个字符的集合，被引号包含（单引号和双引号甚

至三引号（三个连续的单引号或者三个连续的双引号）），具有索引规则，索引

下标从0开始

4) 列表和元组类型：可以当做是数组，可以保存任意类型任意数量的数值

列表：元素使用中括号[]包含，元素的个数和元素的值是可以进行修改的

元组：元素使用小括号（）包含，元素的个数和元素的值是不可以进行修改的

5) 字典类型：字典类型是具有映射关系的数据类型，由键--值对组成，可以存储不

同类型的数据，元素使用大括号{}包含，键可以以字符串或者数值的形式来存

储，值可以是任意类型

注意：定义了一个变量，name变量的数据类型就随之确定了；

可以使用type（变量的名字）来查看变量的类型

3, 标识符和关键字：

1) 标识符：①标识符由字母、数字、下划线组成，且不能以数字开头

②标识符区分大小写

③标识符不能使用关键字

④见名知意

⑤类名使用大写字母开头，模块名使用小写加下划线的方式

命名

2) 关键字：具有特殊功能的标识符称为关键字

注意：help（）-----进入帮助系统

help> keywords-----查看所有关键字列表

help> return-----查看return关键字的说明

help>quit-----退出帮助系统

4, 简单值类型：

1), 整型(int)：十进制、八进制（以数字“0”开头）、二进制（以“0B”或者“0b”开头）、十六进

制（以“0X”或者是“0x”开头），32位机器上整型是32位， -2^{31} -- -2^{31} ，同理64位机器上整

型是64位的, -2^{63} ---- 2^{63}

`bin(20)` -----将十进制的20转换为二进制

`oct(20)` -----将十进制的20转换为八进制

`hex(20)` -----将十进制的20转换为十六进制

`a = int(x, y)` -----将任意进制的x转换为十进制的整型数据, y表示x的进制数

注意: 长整型(long)是整型数据的超集, 可以表示无限大的整数(实际上只受限于

计算机的虚拟内存的大小)长整数的数值后面带有“L”或者“l”, 长整型的数

据与整型的数据操作完全相同

2), 浮点型(float): (用于表示实数)用十进制或者是科学计数法表示

(实数E/e整数, E/e后面的整数表示指数, 指数的正负使用+和-表示, 是正数时,

+可以省略) 占用8个字节, E/e表示的基为10

3), 布尔类型: (每一个python对象天生具有布尔值)

4), 复数类型: `x=complex(a, b)` 定义一个复数 $x=a+bi$

①每一个复数由实部和虚部组成

②复数的实部和虚部都是浮点类型的数据

例: 实现复数相加:

```
a=complex(1,3)
```

```
b=complex(3,1)
```

```
c=a+b
```

注意: 数字类型之间的转换: `int(x[, base])` 将x转换为一个base进制的整数

`float(x)` 将x转换为浮点数

`complex(a, b)` 创建一个复数值为 $x = a+bi$

5, 运算符: 对数据的变换称为运算, 表示运算的符号称为运算符, 参与运算的数据称为操作数

1), 算术运算符: +, -, *, /, %, ** (第一个操作数的第二个操作数次幂), //(向下取整)

2), 赋值运算符: = 将等号右边的数据赋值给等号左边的数据

注意: python不支持自增(++)和自减(--)操作

3), 符合赋值运算符: +=, -=, *=, **=, %=, /=, //=

4)，比较运算符：用于比较俩个数，返回true或者false

==, !=, >, <, >=, <=

5)，逻辑运算符：and（布尔与），or（布尔或），not（布尔非）

6)，成员运算符：in（指定序列中找到值返回true，否则返回false，指定序列可以直接是元组或列表）

not in（指定序列中没有找到返回true，找到返回false）

6，位运算：位运算就是直接对整数在内存中的二进制位进行操作

1) 按位左移(<<)：将二进制位全部左移n位，左边高于原来二进制位数的位丢弃，右边低于原来

来二进制位数的位补0

左移n位，相当于乘以 2^n 次方，如果需要快速计算一个数乘以 2^n 可以借助左移

2) 按位右移(>>)：将二进制位全部右移n位，右边移出原二进制位的位丢弃，左边少于原来二进制

制位的位用0补齐

右移n位，相当于除以 2^n 次方，如果需要快速计算一个数除以 2^n 可以借助左移

3) 按位与(&)：参与运算的俩个数各自对应的二进制位进行“与”操作，只有对应的俩个二进制位都

是1时，结果位为1，否则为0

4) 按位或(|)：参与运算的俩个数各自对应的二进制位进行“或”操作，对应的俩个二进制位只要有

一个是1时，结果位为1，俩个都是0的时候，结果位为0

5) 按位异或(^)：参与运算的俩个数各自对应的二进制位进行“或”操作，如果一位为1，一位为0，

则结果位为1，如果俩个操作数对应位置都为1或者都为0，则结果位为0

6) 按位取反(~)：将操作数的二进制位每一位进行取反，0变1, 1变0

注意了解：

正数的反码和补码都与原码相同。

负数的反码为对该数的原码除符号位外各位取反。

负数的补码为对该数的原码除符号位外各位取反，然后在最后一位加1