Python 学习笔记

1. 以#开头的语句是注释，注释是给人看的，可以是任意内容，解释器会忽略掉注释。其他每一行都是一个语句，当语句以冒号“:”结尾时，缩进的语句视为代码块。
2. 字符串是以''或""括起来的任意文本。"I'm OK"。'I\'m \"OK\"!' Python还允许用r''表示''内部的字符串默认不转义。

>>> print r'\\\t\\'

[\\\t\\](file:///\\\t\\)

1. 用'''...'''的格式表示多行内容：

>>> print '''line1  
... line2  
... line3'''  
line1  
line2  
line3

1. 可以把任意数据类型赋值给变量，同一个变量可以反复赋值，而且可以是不同类型的变量。这种变量本身类型不固定的语言称之为动态语言，与之对应的是静态语言。静态语言在定义变量时必须指定变量类型，如果赋值的时候类型不匹配，就会报错。

字符编码

1. Unicode把所有语言都统一到一套编码里，这样就不会再有乱码问题了。Unicode标准也在不断发展，但最常用的是用两个字节表示一个字符（如果要用到非常偏僻的字符，就需要4个字节）。UTF-8编码把Unicode编码转化为“可变长编码”。UTF-8编码把一个Unicode字符根据不同的数字大小编码成1-6个字节，常用的英文字母被编码成1个字节，汉字通常是3个字节，只有很生僻的字符才会被编码成4-6个字节。在计算机内存中，统一使用Unicode编码，当需要保存到硬盘或者需要传输的时候，就转换为UTF-8编码。
2. Python提供了ord()和chr()函数，可以把字母和对应的数字相互转换：  
   >>> ord('A')  
   65  
   >>> chr(65)  
   'A'
3. 以Unicode表示的字符串用u'...'表示，比如：

>>> print u'中文'

中文

>>> u'中'

u'\u4e2d'

写u'中'和u'\u4e2d'是一样的，\u后面是十六进制的Unicode码

1. 把u'xxx'转换为UTF-8编码的'xxx'用encode('utf-8')方法：

>>> u'ABC'.encode('utf-8')

'ABC'

>>> u'中文'.encode('utf-8')

'\xe4\xb8\xad\xe6\x96\x87'

1. len()函数可以返回字符串的长度：>>> len(u'ABC') 3
2. 把UTF-8编码表示的字符串'xxx'转换为Unicode字符串u'xxx'用decode('utf-8')方法：>>> 'abc'.decode('utf-8') u'abc'
3. 源代码中包含中文的时候，通常在文件开头写上这两行：

#!/usr/bin/env python

# -\*- coding: utf-8 -\*-

第一行注释是为了告诉Linux/OS X系统，这是一个Python可执行程序，Windows系统会忽略这个注释；第二行注释是为了告诉Python解释器，按照UTF-8编码读取源代码，否则，你在源代码中写的中文输出可能会有乱码。