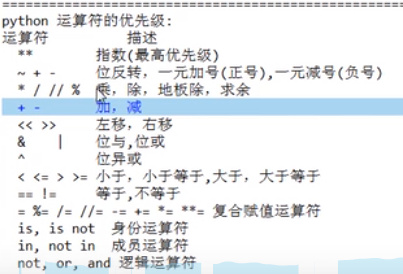
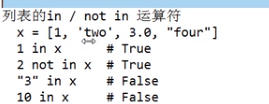
【注】windows下运行py的方法，进入某个磁盘=》直接输入盘符：即可；进入某个目录cd+路径；dir类似ls；cd不加路径类似pwd；

函数名一般下划线断开，类名一般各单词首字母大写！

0、运算符





【py是动态语言】

1、行注释：#

块注释：'''、"""

2、windows/linux下查看py版本，python --version或者python3 --version

3、进入交互式模式python或者python3，退出exit(）

4、input(“提示字符”)和print()函数

5、逻辑运算符：A and B、A or B、not A；

逻辑位运算符：&、|、~、^

整数：1，-1，0

十六进制：0x121231

数字可以以\_分割

浮点：1.2e-3，12e3，3.0=3.，0.4=.4

bool类型：False、True

空值：None（不表示任何类型）

复数complex类型：a+bj

字符串类型：str

6、'I\'m \"OK\"!'中/表示直接输出后面字符，结果为：I'm "OK"!

7、//取整，%取余，/正常除，\*乘，\*\*幂运算，=赋值，==等于

8、0b、0o、0x进制（2、8、16）

9、a=100; b=a; a=1; 此时打印出a，b的值都改变为了1（浅拷贝）

del 变量1，变量2（释放变量，不影响关联变量的值）

a = None（也不影响）

10、类似类型转换：（泛型计算）

int(a)，float(a)，complex(a)，str(a)，bool(a)

type(a)查看变量类型，type(fun())：查看函数fun()返回值类型，无返回值则为None

11、函数：绝对值abs(a)，四舍五入round(a)，幂运算pow(a,b)，help(abs)查看帮助文档

bin(a)转换为二进制

12、整体赋值：a，b = 10，100

13、分号表示分割语句=换行写语句，\表示续行符

14、类似三目运算符：

表达式1 if 真值表达式 else 表达式2（根据真值表达式返回表达式 1或者2）

15、结构：

while 表达式：（else可省略）

执行1

else：

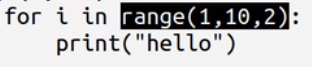
执行2

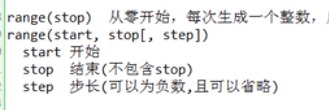
for 变量列表 in 可迭代对象：（else可省略）

执行1

else：

执行2

 break；continue跳出循环

[]内表示可省略

if 表达式1：

执行1

elsif 表达式2：

执行2

else：

执行3

16、字符串（str）：

表示：’’’…’’’ , “””….”””可以有换行，里面的换行转换为\n

‘…’ , “…”不可以有换行

b = ‘ ” ’ 与b=’ \” ’一样，都输出 ”

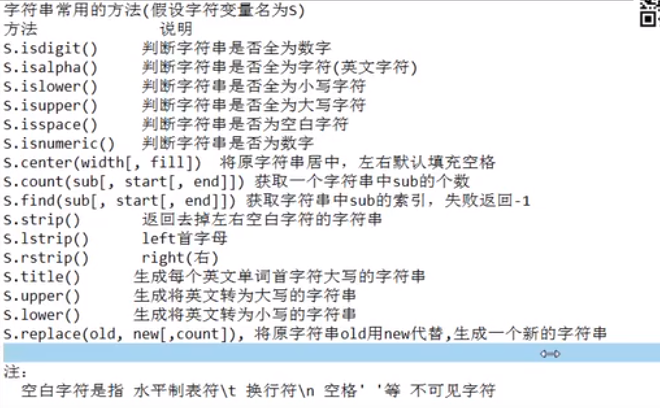
编码：有3种方式utf8（linux，mac），gbk（windows），unicode（作为媒介，是ascii 的扩展）

.py文本的第一行写上#coding=utf-8，或者在 a = u‘ghiuh0ioh’（字符串前加u）

字符串运算: a=“…”+”…”，拼接两个字符串；a\*2：复制2份（\*=）；><=等：比较ascii直 到不等为止

字符串访问：a[0]，a[1]…，切片操作：访问某一段：a[2:3:1]起始,终止,步长，步长为负到 着打，最后一个元素为索引元素为-1，倒着往前-2…，a[:]所有元素的切片

函数：len(),max(),min(),ord()获取编码,chr()获取编码对应的字符



格式控制（py2）：

a = 250

b = 100

c = -100

print('hello Bob! you are %d,%d,%d' %(a,b,c))

format函数：（py3）：{}符号标注，：表示操作

print('{}在{}'.format('郭某人', '写代码')—->不指定位置顺序匹配

print('{1}在{0}'.format('写代码','郭某人'))--0,1表示format中参数放的位置

print('{name}在{option}'.format(name="谢某人",option="写代码"))--关键字锁定

----------------------

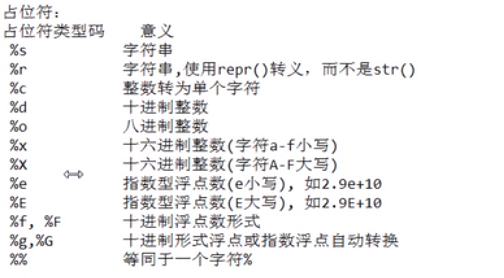
print(‘{:^30}’.format(‘guojian’))--^居中,<左对齐，>右对齐,30为显示出来的字符串长度

print('{:.2f}'.format(3.14159))  
print('{:x}'.format(20))—->20以十六进制显示

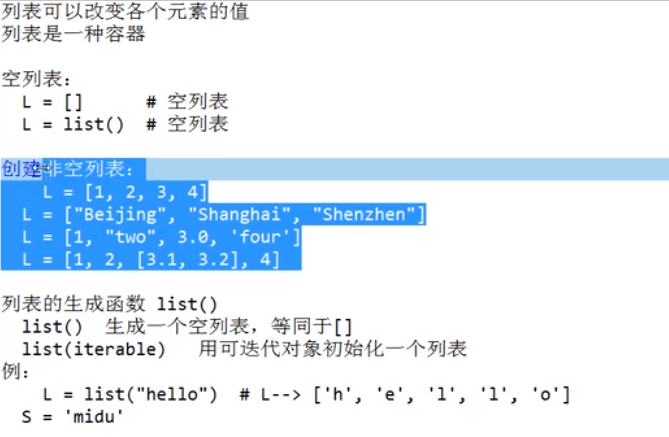
print('{:,}'.format(100000000))---千分位分割100，000，000

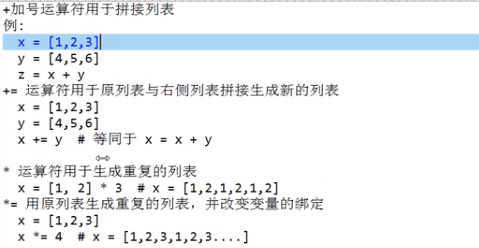
a = 100

print('nishi{}'.format(a))



17.1、列表







list与str不同在于，list可以改变其某个元素值 ，如：list[1] = 2（此方式只能修改，不能添加）；添加新元素list. append(x)

切片操作与str一样。

直接引用列表名，表示引用该列表的所有元素！！！

17.2、元组（touple）：与list相似，不同在于，元组不能修改，用（…）；元组只有一个元素时：（1，）要加逗号

创建空元组： tup1 = ()

t[0]，下标进行元素访问。（元素不允许修改/删除）

17.3、集合 （无序的不重复元素序列，若有重复元素，会自动去除多的一个）

创建空集合： a = set() ，a = {元素}（创建空集合不能用a={}，因为a={}是用来创建空字典）

添加元素x：a.add(x) 添加列表，元组，字典：a.update(x)

移除元素：a.remove(x)，a.discard(x)

清除集合：a.clear()

一般只能判断某个元素是否存在集合里，而不能直接访问集合元素（无序）

17.4、字典（键值对）

键必须是唯一的，但值则不必；

键必须不可变，所以可以用数字，字符串或元组充当，而用列表就不行

stuff = {'x': 5, 'y': 10, 2：3}

stuff['x'] ##访问5（只能通过冒号前的键进行访问）

stuff[2] = 3 ##新增/修改都可以 2：3

stuff['city'] = "San Francisco" ##新增'city': "San Francisco"

del stuff[2] 删除索引为2的键值对

del stuff[‘x’] 删除'x': 5

对照表：<https://www.runoob.com/python3/python3-string.html>

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | str | 列表（list） | 元组（touple） | 集合（set） | 字典（dict） |
| 切片 | Ok | Ok | Ok | Not | Not |
| 是否可以改变元素 | 不可修改/删除 (del a[0])/新增某个元素 | Ok | 不可修改/删除某个元素(del a[0]) | 不可修改/删除某个元素(del a[0])  可以：见17.3 | Ok |
| 新建 | A = “” | A = list()  或者  A = [] | A = () | A = set() | A = {} |
| 访问 | 下标 | 下标 | 下标 | 不可访问某个元素（无序） | 键 |

注意：list()，set()可以在括号里放入可迭代对象从而创建非空的列表与集合

18、from sys import argv

script, first, second, third = argv

命令行输入: python 1.py 参数1 参数2 参数3

（传参数给脚本）

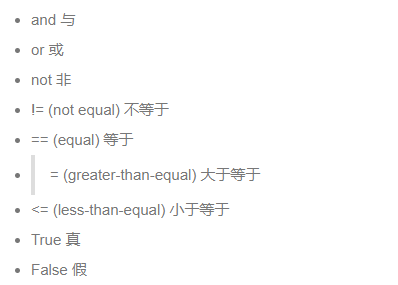
19、函数定义：

def 函数名(参数列表):

函数体

可以带return。且可以多个返回值，如，return a,b,c

20、常用运算符



21、（37课）关键字：

| **KEYWORD** | **DESCRIPTION** | **EXAMPLE** |
| --- | --- | --- |
| assert | 声明 | assert False, "Error!" |
| exec | 将字符串作为Python代码执行 | exec 'print "hello"' |
| lambda | 创建一个无名函数 | s = lambda y: y \*\* y; s(3) |
| raise | 代码出错时，抛出一个异常 | raise ValueError("No") |
| Import/With…as | 一个变量的别名 | with X as Y: pass |
| yield | 暂停， 返回给调用者 | def X(): yield Y; X().next() |

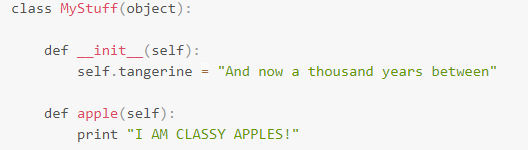
23、模块调用（Module）

在main2.Py文件按中导入自己写的main1.py文件：import main1

在main2.py中使用mian1.py的函数及变量：main1.函数名()、main1.变量名



24、类对象



构造函数：\_\_init\_\_(self)，创建新对象时，自动调用

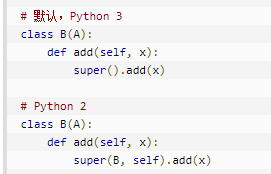
self为参数，此处为一个空对象（self非关键字，但是熟定的）

object是所有类的基类（py3的一个特点：所有类均继承于基类object，故可以不写object）

定义对象：A = MyStuff(“1111”)，第一个参数，永远是空对象，熟定使用self。如果self后面还有其他参数，则1111会依次传给后续参数。

调用类的方法及成员变量：A.apple()、A.tangerine

25、



①继承

隐式继承：继承后，直接继承父类的方法（函数）

重写继承：继承后，在子类中改写父类的方法（函数）

super：super().要调用父类的函数名() -----py3 （作用：其他类里调用父类的函数）

or super(本类名,self).要调用父类的函数名()-------py2

super类详解：<https://www.runoob.com/w3cnote/python-super-detail-intro.html>

class FooParent(object):

def \_\_init\_\_(self):

pass

class FooChild(FooParent):

def \_\_init\_\_(self):

super(FooChild,self).\_\_init\_\_()

# super(FooChild,self) 首先找到 FooChild 的父类（就是类 FooParent），然 后把类 FooChild 的对象转换为类 FooParent 的对象

②包含：（类不继承而调用类）

class Other(object):

def implicit(self)：

pass

class Child(object):

def \_\_init\_\_(self):

self.other = Other()

def implicit2(self):

self.other.implicit()-----------在其他类直接调用其他类的方法

注意：在类中以\_\_开头的函数或变量为私有，外部不可访问！！！！

26、异常处理：（异常可以自定义，异常应该是典型的继承自Exception类）

①

try:

<语句> #运行的代码

except 异常类型：

<语句> #如果在try部份引发了该异常

except 异常类型，参数:

<语句> #如果引发了该异常，获得异常的附加数据信息

else: #如果没有异常发生

<语句>

简单结构：try：（无异常类型，捕获所有异常）

……

except：

…..

else:

……

<https://www.runoob.com/python/python-exceptions.html>

②

try:

<语句> #运行别的代码

finally: #不论异常与否都会执行

<语句>

异常参数信息：

try:

正常的操作

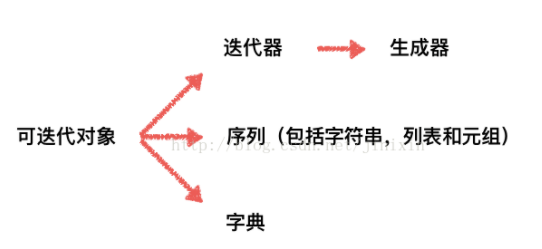
......................

except ExceptionType, Argument: #异常信息存储于Argument里

print Argument

raise：（用于触发异常）

raise Networkerror("Bad hostname")



27、全局变量global

①定义在函数外就是全局。要在一个子函数中使用，则使用global进行声明

②在其他.py文件中使用

单独写一个gl.py的文件，里面放全局。在要引用的文件中import gl，需要引用格式：gl.变量名，即可。且此时外部操作是可以改变该全局值的。