[tu\函数.png](tu/函数.png)一·函数：Int() float() input() len() type()（查询变了的类型） end=””（输出不换行）

自定义函数：def Tudent():

代码块

函数调用 : import

**关键字 global**

***全局变量声明***

***+=*** 在**列表**上不会进行相加 ，而是调用extend方法

Del 删除对象

二·**方法**增删改插：(self)

改： append(“Object”) 在末尾追加数据

Insert(Index,”新对象名”) 插入数据指定位置数据

extend(Iteration) 把其它变量数删除据追加到当前列表末尾

删除： remove(“object”) 删除指定对象

Pop(Index) 默认删除最后插入的数据，可带参数

Clear() 默认删除所有元素

Delit iteration 默认将变量从内存中释放，可带参数 iteration[index]

Count(“object”) 统计表中某一个元素出现的个数

Sort() 默认升序排序

Sort(reverse=True) 降序排序 reverse ：反转

Reverse() 反转排序 (倒过来排列)

Update() 把其它变量数删除据追加到当前列表末尾，如果其他键值对出现同样的就会覆盖原有的键值对 (用于字典)

Isspace() 判断字符中是否有空格，有则返回True (用于字符串)

**189-字符串-05-判断数字的三个方法**：

Isdecimal() 只可判断阿拉伯数字 isdigit() 可判断转义字符数组字 isnumeric() 可判断中文数字

**190-字符串-06-字符串的查找和替换**

Startswith() 判断指定字符开始 返回(True OR False) endswith() 判断指定字符结束 find() 查找是否与指定字符,么有返回 -1 replase() 替换原有字符(old) “不修改原字符，而是返回新字符”

Left right

**文本右**·左·剧中 对齐 rjust() ljust() center()

**文本*右***·左·左右**移除空白字符** rstrip() lstrip() strip()

**范例：**

strips **= [" 静 夜思 "]  
for** lstrips **in** strips**:** print**("|%s|" %**lstrips.lstrip**())** # 左移除  
 print**("|%s|" %**lstrips.rstrip**())** # 右移除  
 print**("|%s|" %**lstrips.strip**())** # 左右同时移除

**字符串的拆分和拼接 split() 去除所有空白字符 join() 排列字符**

**范例**

# 网上获取内容  
#1. 去除所有空白字符  
#2. 用“ ” 分割排列  
pore\_str **= "静夜思\n\n李白\t床前明月光\n疑似地上霜\t举头望明月\n\n低头思故乡"**print**("1:"**,pore\_str**)**# 1.  
pore\_split **=** pore\_str.split**()**print**("2："**,pore\_split**)**# 2.  
pore\_join **= " "**.join**(**pore\_split**)**print**("3:"**,pore\_join**)**

**4.3字符串切片**

# 字符串[开始索引:结束索引:步长]  
sum\_str **= "0123456789"**# 1.截取从 2 ~ 5 位置 的字符串  
print**(**sum\_str**[**2**:**6**])**# 2.截取从 2 ~ 末尾 的字符串  
print**(**sum\_str**[**2**:])**# 3.截取从 开始 ~5 位置的字符串  
print**(**sum\_str**[:**6**])**# 4.截取完整的字符串  
print**(**sum\_str**[:])**# 5.从开始位置，每隔一个字符截取字符串  
print**(**sum\_str**[::**2**])** # 步长 从第二个开始  
# 6.从索引 1 开始，每隔一个取一个  
print**(**sum\_str**[**1**::**2**])**# 7.截取从 2~末尾 -1的字符串  
print**(**sum\_str**[**2**:-**1**:])**# 8.截取字符串末尾两个字符  
print**(**sum\_str**[**8**:])**print**(**sum\_str**[-**2**::])**# 9.字符串的逆序（面试题）  
print**(**sum\_str**[::-**1**])**print**(**sum\_str**[-**1**::-**1**]) # 去一个就向左移动一个**

**5．X公共函数**

**5.2（切片）**

# [1:3] 对有序的集合进行切片，列表和元组都是有序的集合

list\_str = "012345"

# 对列表的索引1到2 不包括尾索引元素 输出结果为：第一个元素1

print(list\_str[1:2])

**5.3** [**算数运算符**](python_keywords/5.3算术运算符.png)

**5.4** [**完整for语法循环**](python_keywords/5.4完整的for循环语法.png)

**3.3**[**多值参数**](python_keywords/3.3多只参数.png)

**3.4** [**元组 和 字典的拆包**](python_keywords/3.4元组和字典的拆包.png)

**04 函数的递归**

[递归①](python_keywords/04函数递归_1.png)

[递归②](python_keywords/04函数递归_2.png)

**二． 面向对象**

**（一、封装）**

1. [初始化方法](python_keywords/面向对象_初始化方法01.png) ，\_\_INIT\_\_方法
2. [对象消失时调用](python_keywords/05.__del__方法.png)，\_\_del\_\_方法
3. [返回一个值](python_keywords/06__str__方法.png) ，\_\_str\_\_方法
4. [身份运算符](python_keywords/02_身份运算符.png)，is (i == j) “IS”运算符 是对比在内存中的地址、==只是比较内容
5. [伪私有属性和私有方法](python_keywords/02伪私有属性和私有方法（科普）.png)（科普）

**（二、继承）**

1. [继承的概念](python_keywords/面向对象/1.1_继承的概念.png)
   1. 专业术语： Dog 类是 Animal类的**子类**，animal类是dog类的**父类**，dog类从Animal类**继承**
   2. Dog类是 Animal类的**派生类**，Animal类是dog类的**基类**，dog类从Animal**类派生**

1.3——访问父类的私有方法和私有属性（[间接访问](file:///D:\Java%20for%20developers\python\New_python_大纲\python_keywords\面向对象\1.3_父类的私有属性和私有方法.png)）

2.1------ 新式类和旧式类 ([经典类](python_keywords/面向对象/2.2_新式类和旧式类(经典类).png))

**（三、多态）**

1. 多态的概念([polymorphic](python_keywords/面向对象/三、多态/3.1_多态.png))
2. **类的结构**

**术语——**[**实例**](python_keywords/面向对象/01、类的结构/1.1_术语——实例.png)

**01 、类是一个**[**特殊的对象**](python_keywords/面向对象/01、类的结构/1.2_类是个特殊的对象.png)

01、**类属性的**[**获取机制**](python_keywords/面向对象/三、多态/2.2属性的获取机制.png) （科普） 所谓的获取机制就是： 一个属性count在外界使用对象调用时如果没有在对象中找到，就会向上找类中是否有count属性

03.类方法和静态方法（[类方法](python_keywords/面向对象/三、多态/03.类方法和静态方法.png)）（[静态法法](python_keywords/面向对象/三、多态/3.2_静态方法.png)） 在不需要访问类属性和对象属性时，可以定义静态方法

**三、单例设计模式**

1、 [单例](python_keywords/面向对象/四、单例/3.4单例.png)

①\_\_new\_\_方法（[new](python_keywords/面向对象/四、单例/3.5__NeW__方法（只分配一个对象）.png)） [案例](python_keywords/面向对象/四、单例/3.5.1单例_案例.png)

**四、异常**

1. 捕获异常([Try、except](python_keywords/面向对象/四、单例/异常_02.png))
2. 错误类型捕获（[2.2](python_keywords/面向对象/四、单例/错误类型异常_03.png)）
3. 异常完整捕获语法（[2.3](python_keywords/面向对象/四、单例/03_异常捕获完整语法.png)）
4. 异常的传递（[03](python_keywords/面向对象/四、单例/04_异常的传递.png)）
5. 抛出 raise 异常（[04](python_keywords/面向对象/四、单例/05_抛出%20raise%20异常.png)）

**五·模块导入**

1. 基本import 导入([import](python_keywords/面向对象/五·模块/hz_01_import导入.png))
2. Form......import导入([from..import](python_keywords/面向对象/五·模块/hz_02_from.........import导入.png))
3. 模块搜索顺序[[扩展](python_keywords/面向对象/五·模块/hz_03_模块搜索顺序.png)]
4. A
5. A
6. A
7. \_\_name\_\_属性（[注意](python_keywords/面向对象/五·模块/hz_04___name__属性.png)）

**六、包(package)**

1、package([\_\_init\_\_](python_keywords/面向对象/五·模块/包/hz_01_package.png))

**七、文件的基础操作**

**1、操作文件的**[**套路**](python_keywords/面向对象/五·模块/包/hz_07_文件的基础操作.png)**（open、**[**read**](python_keywords/面向对象/五·模块/包/hz_08_read方法_读取文件.png)**、write、close）**

**2、**[**文件指针**](python_keywords/面向对象/五·模块/包/hz_09_文件指针.png)**(并非C语言中的指针;指向第几条记录)**

**3、打开**[**文件方式**](python_keywords/面向对象/五·模块/包/hz_10_打开文件方式.png)**（R、W、A、R+、W+、A+）；如 file = open(“文件名”,”方式”, conding = “utf-8”)**

**4、读取文件的另一种方式** [**readline**](python_keywords/面向对象/五·模块/包/hz_12_readline.png) **方法(只读取一行类容)**

**5、如何在python 2.0中使用 中文(**[**coding**](python_keywords/面向对象/五·模块/包/hz_11_如在python%202.0中使用中文.png)**) coding 译码的意思**

**6、复制文件（用 read） ,大型类容文件（readline）**

**7、文件/目录的常见管理(**[**os**](python_keywords/面向对象/五·模块/包/hz_14_目录、文件管理(os).png)**包)**

**☆八、eval函数(**[**eval**](python_keywords/面向对象/五·模块/包/hz_15_eval函数.png) **函数，传入一个表达式 就会求出值 并返回计算结果)**

**1、 eval 函数 不可滥用(**[**eval（）**](python_keywords/面向对象/五·模块/包/hz_15_eval函数_合理使用.png)**)**