Python

## Python的发展史：

### Python之父的发展史：Guido van Rossum

Guido van Rossum：名副其实的荷兰人,龟叔

1982年从阿姆斯特丹大学取得了数学和计算机硕士学位。(matlab)

20世纪80年代中期，Python之父Guido van Rossum还在CWI(数学和理论计算机科学领域的研究中心，位于阿姆斯特丹)，那个时候他Guido构思了一门致力于解决问题的编程语言。他觉得现有的编程语言对非计算机专业的人十分不友好。

C，C++（面向过程，面向对象【世间万物皆对象，对象指挥对象来做事情】），

1989年12月份，在媳妇回娘家休假的期间，为了打发无聊 的圣诞节假期，Guido开始写Python的第一个版本。经过四周的研究终于研究出了python的第一个版本。（值得一提的是Python这个名字的由来，Python有蟒蛇的意思，但Guido起这个名字完全和蟒蛇没有关系。当Guido在实现Python的时候，他还阅读了Monty Python's Flying Circus《巨蟒剧团之飞翔的马戏团》的剧本，这是来自一部来自20世纪70年代的BBC喜剧。Guido认为他需要一个简短、独特且略显神秘的名字，因此他决定将该语言称为Python。

10 java = 1 python

）





|  |  |
| --- | --- |
| 版本 | 特性 |
| 1994年1月发布Python1.0版本。 | 主要新功能是lambda, map, filter和reduce |
| 2000年10月，Python2.0发布 | 主要新功能是内存管理和循环检测垃圾收集器以及对Unicode的支持 |
| 2008年的12月份，Python3.0发布 | Python3.x不向后兼容Python2.x，这意味着Python3.x可能无法运行Python2.x的代码。Python3代表着Python语言的未来。 |
| 官网 | [www.python.org](http://www.python.org)(社区)  com:国际域名，www.baidu.com  cn: 中国  .net 网络服务商  .gov.cn  org组织 |

## Python的应用领域

### 1.Web开发（facebook,twitter,知乎）

尽管今天PHP依然是Web开发的流行语言，但Python上升势头更劲。随着Python的Web开发框架逐渐成熟，比如耳熟能详的Django和flask你可以快速地开发功能强大的Web应用。小编我首推Django，强烈建议所以有志于从事Python Web开发的人掌握这门框架。无论是建大型网站，开发OA或Web API，Django都可以轻松胜任。

### 2.网络爬虫（python专业）

多亏有了Python，一个小白用几行代码就可以写个爬虫爬段子爬福利图了。然而爬虫的真正作用是从网络上获取有用的数据或信息，可以节省大量人工时间。能够编写网络爬虫的编程语言有不少，但Python绝对是其中的主流之一。Python自带的urllib库，第三方的requests库和Scrappy框架让开发爬虫变得非常容易。

### 3.计算与数据分析（python专业）

随着NumPy，SciPy，Matplotlib等众多程序库的开发和完善，Python越来越适合于做科学计算和数据分析了。它不仅支持各种数学运算，还可以绘制高质量的2D和3D图像。和科学计算领域最流行的商业软件Matlab相比，Python比Matlab所采用的脚本语言的应用范围更广泛，可以处理更多类型的文件和数据。

### 4.人工智能（AI：python）

当前最热工资最高的IT工作就是从事人工智能领域的工程师了。Python在人工智能大范畴领域内的机器学习、神经网络、深度学习等方面都是主流的编程语言，得到广泛的支持和应用。最流行的神经网络框架如Facebook的PyTorch和Google的TensorFlow都采用了Python语言。你不学Python,你会用那些框架吗?

### 5.自动化运维（python专业）

这几乎是Python应用的自留地，作为运维工程师首选的编程语言。在很多操作系统里，Python是标准的系统组件。大多数Linux发行版和MacOSX都集成了Python，可以在终端下直接运行Python。Python标准库包含了多个调用操作系统功能的库。通过pywin32这个第三方软件包，Python能够访问Windows的COM服务及其它WindowsAPI。使用IronPython，Python程序能够直接调用.NetFramework。一般说来，Python编写的系统管理脚本在可读性、性能、代码重用度、扩展性几方面都优于普通的shell脚本。

### 6.云计算

Python的最强大之处在于模块化和灵活性，而构建云计算的平台的IasS服务的OpenStack就是采用Python的，云计算的其他服务也都是在IasS服务之上的。

### 7.网络编程

Python提供了丰富的模块支持sockets编程，能方便快速地开发分布式应用程序。很多大规模软件开发计划例如Zope，Mnet, BitTorrent和Google都在广泛地使用它。

### 8.游戏开发（LOL,炉石传..c++，c,python）

很多游戏使用C++编写图形显示等高性能模块，而使用Python或者Lua编写游戏的逻辑、服务器。相较于Python，Lua的功能更简单、体积更小，然而Python则支持更多的特性和数据类型。Python的PyGame库也可用于直接开发一些简单游戏。

以上就是Python八大主要应用领域，你都知道吗？总之，Python是一门新手友好、功能强大、高效灵活的编程语言，学会之后无论是想进入数据分析、人工智能、网站开发这些领域，还是希望掌握第一门编程语言，都可以用Python来开启未来无限可能！

## Python的就业方向

**发展前景一：Linux运维**

**发展前景二：Python Web网站工程师**

**发展前景三：Python自动化测试**

**发展前景四：数据分析**

**发展前景五：人工智能**

Python单元测试框架开发。（python基础和高级阶段应用）贾老师

Python功能自动化脚本开发。乔老师

Python接口自动化脚本开发。贾老师

语言之间的区别：

解释型语言：python

编译型语言：java 🡪 编程机器可以识别的语言🡪执行语言

## Python的编程环境的安装

1. 安装python开发工具包：python.3.8.exe
2. 安装IDEA:pycharm编程环境

安装注意：

1. 不要有中文：极有可能编程乱
2. 不要有空格：防止解析路径只解析到空格处停止。C:\xxx yy\zzz c:\xxx
3. 安装的办公软件不要放在C盘。D：E：F
   1. 向系统里注册软件信息：软件路径
   2. 解压软件
   3. 专业卸载：控制面板（我的电脑 🡪右键 🡪属性），第三方卸载软件

平常：

1. 装一个翻译工具：金山词霸、有道词典、灵格斯
2. Everything：全文搜索
3. 问题解决：百度、必应（微软）、google(很抱歉有长城防火墙阻隔：服务器迁移到香港)，翻墙工具（vpn：蓝灯，佛跳墙）

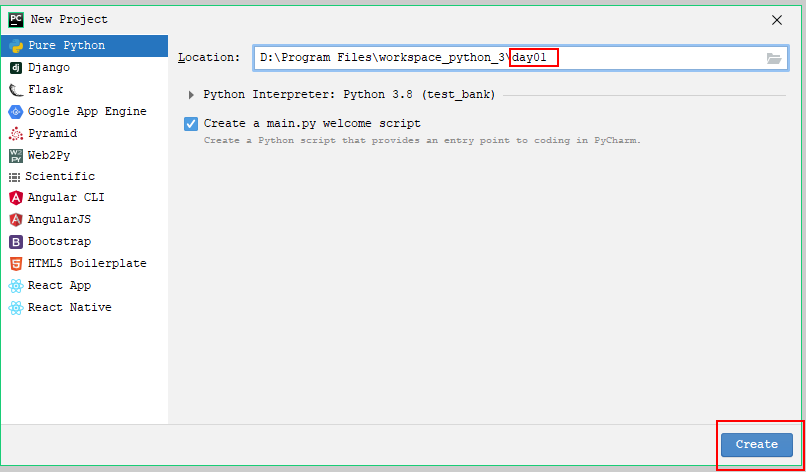
## Python入门而谈：

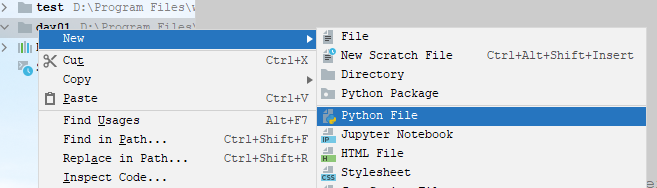
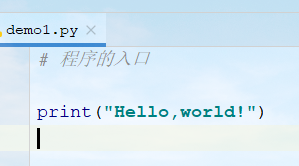
1. 我们学python2.x or 3.x呢？
   1. 2.x版本是过去式，3.x是未来的一种语言，一种去向。
   2. 2.x不会在有大的变化，3.x的库的更新，只会在3.x上更新。
      1. 因为：要清理和整理2.x。所以3.x和2.x版本就不会进行兼容。就是说2.x版本开发的程序3.x的解释器不能解释。
   3. 3.x能支持中文编程。----字符编码的问题

## Python的安装：

* 1. <http://www.python.org/downloads/>
  2. 默认安装路径：c:\python
  3. 然后配置环境变量：
     1. 直接在path变量里新增
     2. 或者间接的新建python\_home变量来存储python安装路径。
  4. Linux、mac系统有自带的python环境。自带的是2.6，请更新至2.7版本。

1. 第一个helloworld程序
   * + 创建项目



* + - 
    - 

注意事项：

* 1. 在python的交互器里输入(python)
  2. >>>print(“hello world”)
  3. 然后将该命令保存到一个.py格式的文件里。
  4. 用python hello.py执行。
  5. 第一个helloworld程序与其他程序语言的对比。

## 计算机的基本操作知识

快捷键和组合键：

Ctrl :

Ctrl + c : 复制

Ctrl +V :粘贴

Ctrl + a ：全选

Ctrl + x : 剪切

Ctrl + z : 撤销

Ctrl + o :打开

Ctrl + s:保存【ctrl + s不离手】

Shift :换挡键

半角输入法：

全角输入法：

Windows：

Window + e : 打开我的电脑

Window + r : 打开运行方式

Windows + l:锁屏

Window 常用指令：

mspaint：画板

calc：计算器

services.msc :打开服务

cmd : dick opreration system 命令行版的

mstsc : 远程连接

shutdown –s -t 10800

shutdown -a :取消关机

control:控制面板

iexplore： IE浏览器

winword: 打开word文档

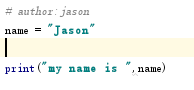
notepad 记事本

chrome : 打开谷歌浏览器

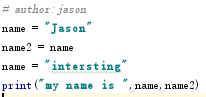
firefox : 开发火狐浏览器

## 变量、字符编码

* 1. 何为变量：存储数据的变化的量叫变量



* 1. 但是要注意一点:



Name与name2在内存里都指向了Jason,当name2指向了intersting的时候，name却没有改变指向。

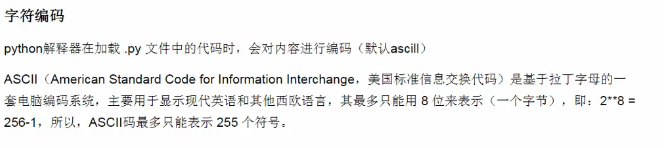
所以打印的结果是：Jason,intersting

1. 变量的命名规则：

\_name

* 1. 多个单词组成用下划线连接
  2. 常量：python没有常量的概念
     1. 常量单词全部大写

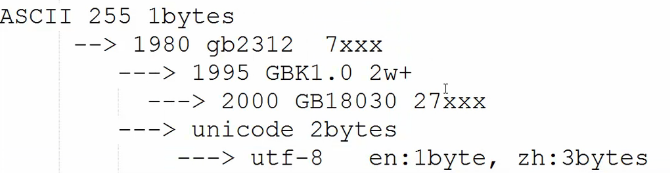
1. 字符编码与二进制
   1. 字符编码





1. 比如日本的Shift\_JIS编码：

image5.png



所以在python2里用中文的话，就在脚本里加一行：

# -\*- coding:utf-8 -\*-即可

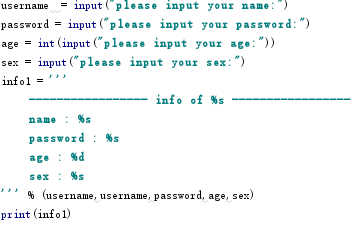
就是告诉解释器用utf-8编码来执行源代码。

1. 注释：
   1. 单行注释：#
   2. 多行注释：’’’ 注释内容 ’’’ ，三个单引号引起来

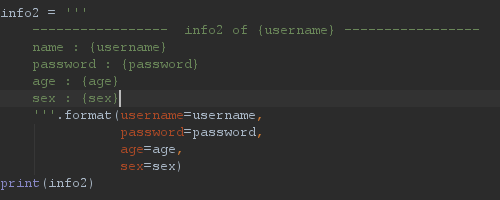
## 用户交互程序：

### 三种打印格式化方式：

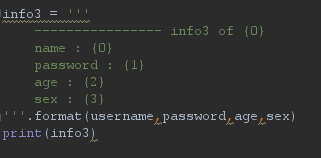
* 1. 从键盘输入：input



* 1. 第二种方式：



* 1. 第三种方式：



提醒：最好用最后两种，前面的那种%s用的不多。

## 猜数字小游戏

数字：让系统随机产生一个，范围需要自己指定0~2000

没有次数限制。

给一个系统提示。大了！小了！恭喜你，猜中了，本次猜的数字xxx。

Python技术选型：

1. 使用python的随机模板产生一个随机数
   1. Import random
   2. Random.random() \* 2000
2. 循环技术。
   1. 次数型循环： for
   2. 当型循环: while

|  |
| --- |
| i = 0  while i < 100:  print(i)  i = i + 1 |

1. 判断，终止技术。
   1. if…else

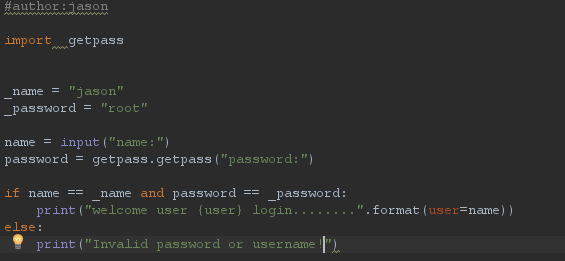
|  |
| --- |
| *'''age = 10 if age >= 18:  print("张家玮，你来啦！！！") else:  print("小屁孩，滚出去！！！") '''* |

break

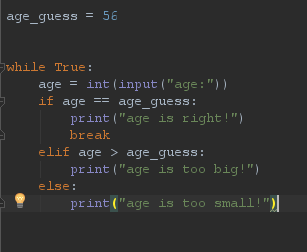
|  |
| --- |
| *# author：jason* import random  *# 1.让系统产生一个随机数* num = int(random.random() \* 2000) count = 0 *# 猜的计数器* while True:  guess = int(input(**"请输入您要猜的数字："**))  count = count + 1 *# 计数器加1* if num > guess:  print(**"小了！"**)  elif num < guess:  print(**"大了！"**)  else:  print(**"恭喜你，猜中了，您的猜的数字为："**,num,**"您猜了"**,count,**"次了！"**)  break |

## If else 流程判断

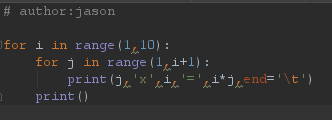
### 一个简单的if…else



### 一个简答的while循环：



## For



作业：

1. 一：将学的知识写到博客。
2. 二：写一个登录接口
   1. 输入用户名密码
   2. 认证成功后显示欢迎信息
   3. 输错三次后锁定

# 模块初识

## 模块：python的强大之处在于他有非常丰富和强大的标准库和第三方库（需要安装才能使用的像django框架），以后会更深入的讲到各种常用的库，ok,现在我们先来学两个简单的：

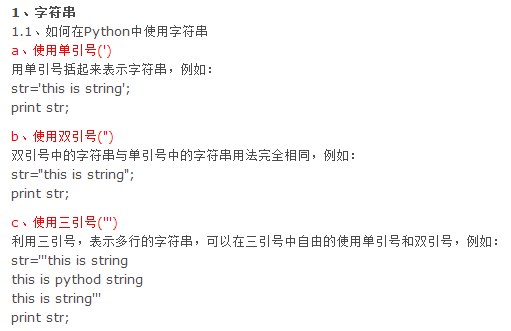
### sys库

### sys模块的作用：

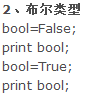
* sys是system的缩写，用来获取操作系统和编译器的一些配置，设置及操作。

# Python数据类型

## 目录1、字符串



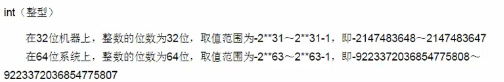
2、布尔类型



3、整数

image15.png

Int （整型）：



Long（长整型）：



严重注意：python3.x版本已将long类型和int类型整合了，当int类型表示不过来了会自动提升为long类型。

4、浮点数

5、数字

6、列表

7、元组

8、字典

9、日期

Float（浮点型）：

## 数据运算：

### 基本运算：

**+ ，-，\* ,/**

**%:取模**

**x\*\*y次幂：x的y次幂**

**//取整除，：取商的整数部分。9//2，整数部分未4.0**

### 运算符

==判断是否等于。比较对象是否相等

！=不等于，比较两个对象是否不相等

<>不等于，比较两个对象是否不相等

>大于，

<小于

>=大于等于

<=小于等于

### 赋值运算

=：简单的赋值运算

+=：加法赋值运算。C+=a 与 c = c +a同效

-=：减法赋值运算。C-=a 与c= c – a同效

\*=：乘法赋值运算。C\*=a 与 c = c\* a同效

/=：除法赋值运算。c/=a 与 c = c/a同效

\*\*=：幂赋值运算。C\*\*=a 与 c = c \*\*a 同效

//=：取整商赋值运算：c//=a 与 c = c//a同效

### 逻辑运算

And 就相当于&&

If(a > 15 and b<20) if(a>15 && b<20)

Or 就相当于||

If(a>12 or a<52) 与其他语言的 if(a>12 || a<52)

Not 就相当于 ！

If( not(a>12 || a< 52)) 与其他语言的 If( ！(a>12 || a< 52))

### 成员运算

In:判断某个值在某个字符序列是否存在

Not in:

### 身份运算

Is :判断两个标识符是引用自一个对象（可以用id(a) 来查看对象的身份标识号）

Not is :

### 位运算符

&：与运算，3 & 5:0011 & 0101 = 0001 = 1 (0b0011 & 0b0101)

| ：或运算，3 | 5 ：0011 | 0101 = 0111 = 7 (0b0011 | 0b0101)

^ :异或运算, 3 ^ 5 : 0011 ^ 0101 = 0110 = 6 (0b0011 ^ 0b0101),相同为0，不同为1

~：取反运算： ~5 ： ~ 0101 = -6

>>:右移运算符： 5 >> 1 : 0101 >> 1 =0010 =2

<<:左移运算符: 5<< 1 : 0101 << 1 = 1010 = 10

### Bytes数据类型

三元运算符：

a,b,c = 5,4,8d = a if a > c else c print(d)

a,b,c = 8,96,52d = a if a>b else (b + 1 if a +5 > b else b) print(d)

二进制：

0b0010110

八进制

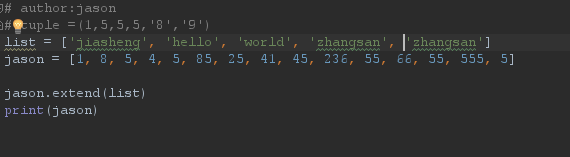
0o01076(是零o开头，不是0开头了)

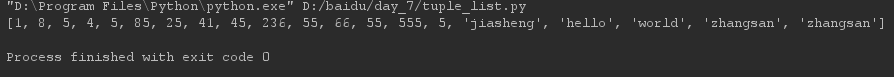
十六进制

0xd0dfd1

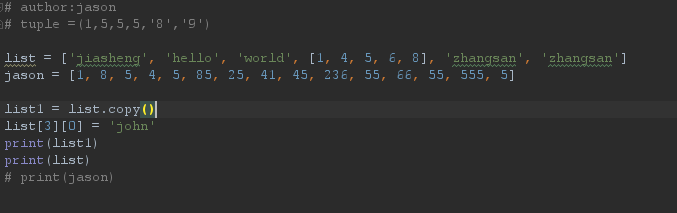
### 元组和列表

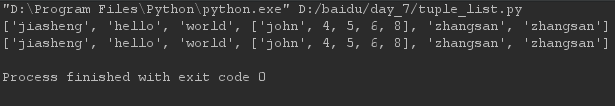
* 元组与列表：
  + 就相当于一个数组。
  + Name\_tuple=(1,”hello”,6355,’intersting’,True)
  + Name\_list = [1,”hello”,6355,’intersting’,True]
* 取值:
  + Name\_tuple[0]
  + Name\_list[0]
* 修改值
  + Name\_tuple是无法修改元组的值
  + Name\_list是可以修改该列表内的值。Name\_list[2] = 63;
* 添加值
  + Name\_tuple：元组不能添加值，初始为多少那么元组里就是多少，后期不能进行添加值。
  + Name\_list:列表可以添加值，需要调用append方法，name\_list.append()，添加的位置是在列表的结尾。
  + Name\_list:列表可以插入值，需要调用insert方法，insert(index,value);需要提供要插入的角标值和实际值
  + 删除值：
    - 对于元组无法删除值
    - 列表的删除值：
      * 删除某个值：name\_list.remove(“张三”);
        + 注意：如果该列表里存在相同的元素，remove方法只删除其中一个并不是删除全部的。
      * 删除某个角标的值：
        + Name\_list.pop();若不提供角标值则默认删除的是该列表的最后一个值。
        + Name\_list.pop(int index); 删除指定角标对应的值。
      * 删除全部列表
        + Name\_list.clear();将列表的元素全部清除。
        + Del name\_list;
  + 列表元素的查找：
    - name\_list.index(‘zhangsan’) 查找元素的角标：
  + 列表元素的统计
    - Name\_\_list.count(“zhangsan”);统计zhangsan在列表中出现的次数。Count()
  + 排序
    - Sort（）方法，默认从小到大排序
    - 注意：排序方法只能排序列表中元素为相同类型的列表。
  + 列表的扩展
    - 如果想让其他列表中的元素想添加到此列表中，用扩展即可
    - Extends
    - List1.extend(list)
    - 比如：



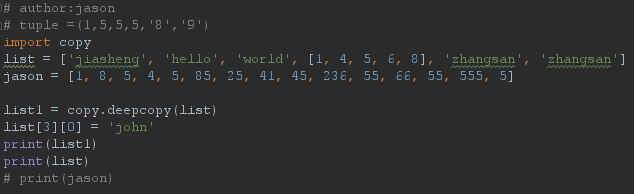


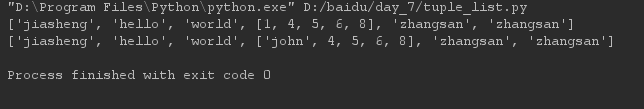
* + 列表的复制：
    - 浅copy;对于列表中没有子列表的列表可以用copy。
    - 但是如果有对于列表中有子列表的列表就不能用copy来复制了。





* + - 所以我们用深度copy。
      * Import copy
      * 用copy的deepcopy方法
      * List1 = list.deepcopy(list)
      * Print(list1)





所以呢对于浅copy我们基本上不用。

* + - 列表的切片
      * List = [4,8,7,5,6,9,2,1]
      * List[1:5];表示：从角标为1的元素开始打印一直打印到角标为5-1的元素。
      * 若切片的前面的起始角标不写默认从0开始。
        + List[:5]与list[0:5]是一样的效果。
      * 负数的情况。
        + List[-5:-1],从列表的后面开始往前数分别为-1，-2…….

所以打印结果为：5,6,9,2

* + - * 如果在打印切片的时候中间隔一个打印一个呢？
        + 那么就在切片的后面再加一个切片：
        + List[0:-2:2]：表示从0角标开始到倒数第三个数结束。中间隔一个数打印一下。

#步长切片 for i in range(0,len(list),2): print(list[i])

浅copy的适用场景：用于创建多人的共同的共享数据

water = ['cup', ['升', 60]]person = water[:] person1 = water[:] person[0] = 'kitty' person1[0] = '史努比'person[1][1] -= 3person1[1][1] -= 6 print(person) print(person1)

就好比，两个人有各自的水杯，但是他俩都用的是同一个饮水机。

那么浅copy就是将杯子的型号copy了一份，但是饮水机还是哪个饮水机。

总结：

* 所以对于元组，是一种只读的列表序列，只有count和index方法。
* 尽管有好几种表现方式，但这是列表和元组唯一的“技术差异”。比如：列表有一个 append() 的方法来添加更多的元素，而元组却没有这个方法：
* 元组并不需要一个 append() 方法，因为元组不能修改。

一个练习：购物车

# author:jason shop = [ ['phone', 6000], ['MAC phone', 5600], ['iphone 8x', 10000], ['watch', 1200], ['lenovo pc', 5200] ] while True: salary = input("input your salary please:") if salary.isdigit(): salary = int(salary)break else: print("please input salary again!") print("-------------------welcome to my shopping store!-----------------\n") my\_cart = [] while True: for index, item in enumerate(shop): print(index, item) choice = input(">:") if choice.isdigit(): choice = int(choice) if choice < len(shop): if salary >= shop[choice][1]: my\_cart.append(shop[choice]) salary -= shop[choice][1] print("\033[32;20;1m Added %s you current salary is %s \033[0m" % (shop[choice], salary)) else: print('\033[41;20;1m sorry you salary is not enough! please charge and shopping!s\033[0m') else: print("\033[42;20;1m Current shopping product is not exists!please input again!\033[0m") elif choice == 'q': break else: print("you input is not illegal ,please input again!") print('-----------------bye !----------------------------------!')print('your cart has next product:') for i in my\_cart: print(i) print("your balance is ", salary)

## 字符串常用操作：

1. 首字母大写：capitalize()
   1. Name = ‘jason’
   2. Name.capitalize()
   3. Print(name) 结果为Jason
2. 统计出现的次数
   1. Name.count(‘a’) 单独统计某个字符出现的次数
   2. print(\_\_author\_\_.count("ja")) #统计某个字符出现的次数,也可以统计出现的字符串
3. 字符串的自动居中并填充
   1. Center(length,char),将字符串居中，两边少的空白部分用字符来填充。
   2. name = ‘jason’
   3. name.center(20,’a’).

image24.png

结果：

1. 判断是否以某个字符串结尾的。
   1. Endswith();
   2. Name.endswith(“so”).判断该字符串是否是以so结尾的。

## 字典操作

### 字典是一种key-value的数据类型，使用就像我门上面用字典。

比如：info = {

'student001': 'jason',

'student002': 'john',

'student003': 'alice',

'student004': 'albert'

}

其中的学生号与学生名称是一一对应的。

### 字典的特性：

1. Dict是无序性的。
2. Key必须保持唯一性，也就是在字典里不存在两个相同的键。也就保证了唯一的键值对。

### 字典的输出

* Print(info[‘student001’]),就把其对应的值打印出来了。

### 字典的修改：

Info[‘student001’] = ‘hello’

### 字典的删除

* Del info
* Del info[‘key’] 删除字典中某一个值
* Info.clear(),清除字典里的所有键值对，
* Infor.pop(‘key’),删除某个一个键值对

### 字典的添加

* 直接在字典里写:info[‘key’] = value;

### 字典的值的存在判断

* 用in 关键字来判断是否在dict里是否存在。
* ‘student001’ in info;
* 该句意思是判断student001在info字典里是否存在。（只判断键是否在字典里存在）
  + 在python2.7版本中通常是用has\_key来写的。Info.has\_key(‘student001’);

### 三级菜单

|  |
| --- |
| # author:jason  data = {  "北京":{  "昌平":{  "天通苑":['海底捞','呷哺'],  '龙泽':['永辉超市']  },  "海淀":{  '公主坟':['军事博物馆','中华世纪园'],  '科普场馆':['中国科技馆','北京天文馆'],  '高校':['北京大学','清华大学']  },  "朝阳":{  '龙城':['鸟化石国家地质公园','朝阳南北塔'],  '双塔':['朝阳凌河公园','朝阳凤凰山']  }  }  }  ########打印城市的函数  def print\_place(choice):  for i in choice:  print('\t\t',i)  ########城市  for con in data:  print(con)  choice1 = input("请选择城市：choice1:>>>")  flag = False  while not flag :  if choice1 in data:  if choice1 == 'q':  break  print\_place(data[choice1])  choice2 = input("choice2>>>")  if choice2== 'q':  break  if choice2 in data[choice1]:  print\_place(data[choice1][choice2])  choice3 = input("choice3>>>")  if choice3 == 'q':  break  if choice3 in data[choice1][choice2]:  print\_place(data[choice1][choice2][choice3]) |

## 集合的使用

### 1、将列表转换为集合

List=[1,4,489,6,8,2,9,63,5,5,6,68555,522]

列表可以转换为集合

s=set(list) //【可以将列表转换为集合】

print(s,type(s))

### 2、直接申明集合

S=set（[5,8,85,235,52,565,423,3544,5]）

或者

S=set（{5,8,85,235,52,565,423,3544,5}）

或者

S=set（(5,8,85,235,52,565,423,3544,5)）

### 集合的一个特点：

1. 无序性：集合里的值显示都是无序排序的。
2. 唯一性：集合里的值都是唯一的，不存在重复的值。

### 集合的一些操作：

1. 两个集合取交集：
   1. s1 = set({2,6,45,2,5})
   2. s2 = set({6,9,8,2,3})
   3. print(s1.intersection(s2))
   4. 结果为：{2,6}
2. 两个集合取并集：其实就是将两个集合合并然后去除重复的数据

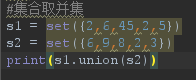


image26.png

1. 两个集合取差集

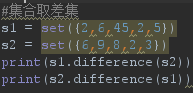
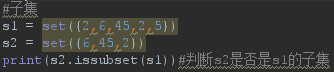


image28.png

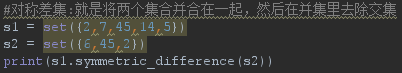
* 1. 结果：

1. 判断某个集合是否是某个集合的子集

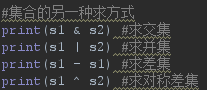


* 1. 结果为True

1. 对称子集



* 1. {7, 5, 6, 14}



### 基本操作

1. 集合元素的添加

s1 = set([5,6,9,22,10,5,20,65,5,45])s1.add(9456) print(s1)

* 1. 结果为：{65, 5, 6, 9, 10, 45, 9456, 20, 22}

1. 集合元素的移除
   1. s1.remove(22)就是将元素22从集合中移除。
2. 向集合中添加其他集合元素,添加多项
   1. s1.update([56,32,75,14,95,63]);
3. 求长度：
   1. Len(s1);求s1集合的长度。
4. 测试
   1. X in s1
   2. 判断x是否是s1集合中的一个元素
5. discard(project obj1)
   1. 如果要移除的元素是该集合中的一部分，则删除，如果不是该集合中的元素则什么也不做。
   2. S=set([25,6,5,89,2145,55])
   3. S.discard(6),若6是s集合的一个成员则删除，若不是则什么都不做。

## 文件操作

读取文件的基本操作：

1. 先打开文件得到文件的一个引用。
2. 通过引用对文件进行操作
3. 关闭文件（关闭流）

image32.png