**Python基础语法和数据类型**

# 【第一节课笔记】

# 一、Python基础语法

## 1.命名规则

**只能包含：数字、字母、下划线，不能以数字开头，不能使用关键字。**

## 2.顶格编写，同级的代码对其写，不同级的需要进行缩进(如：判断、循环、函数)

## 3.注释

**三种注释方式：**

1. **# 号，单行注释**

# 这里是注释内容

1. **三引号，多行注释**

'''  
 这里是注释内容  
 '''

1. **快捷键：ctrl + / 第一次：进行注释，再按一次：解开注释**

## 4.基本语法

### 4.1 输出语句：print()

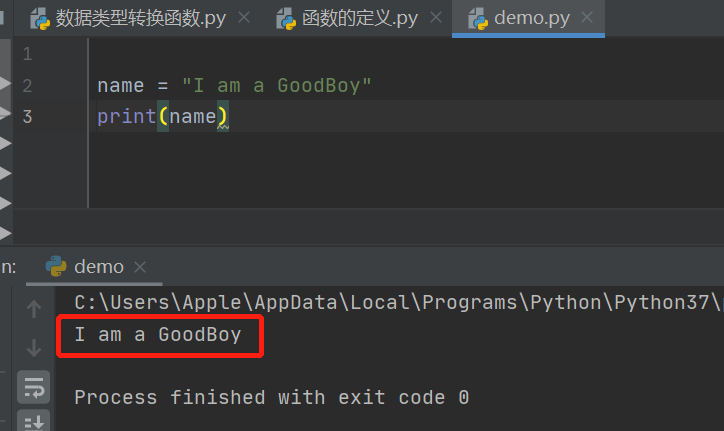
print("我是柠檬班最亮的仔")

### 4.2 判断数据类型函数：type()

print(type("你是柠檬班最美的妞"))

### 4.3 变量：存储数据的容器

name = "U are a GoodBoy"  
print(name)



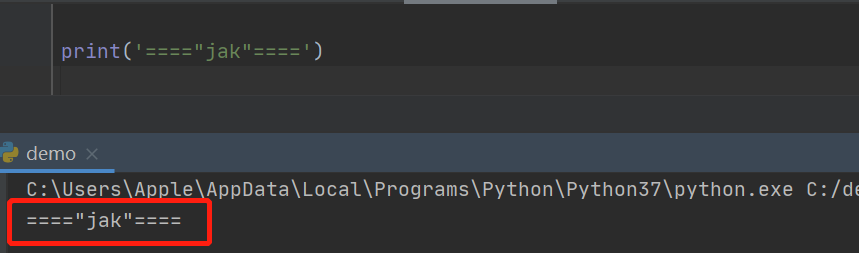
### 4.4 字符串的三种定义方式

*'''*

*字符串的三种定义方式：  
 1.使用单引号  
 2.使用双引号  
 4.使用三引号  
'''*name = '我是被单引号引起来的字符串'  
name2 = "我是被双引号引起来的字符串"  
name3 = '''我是被三引号引起来的字符串'''

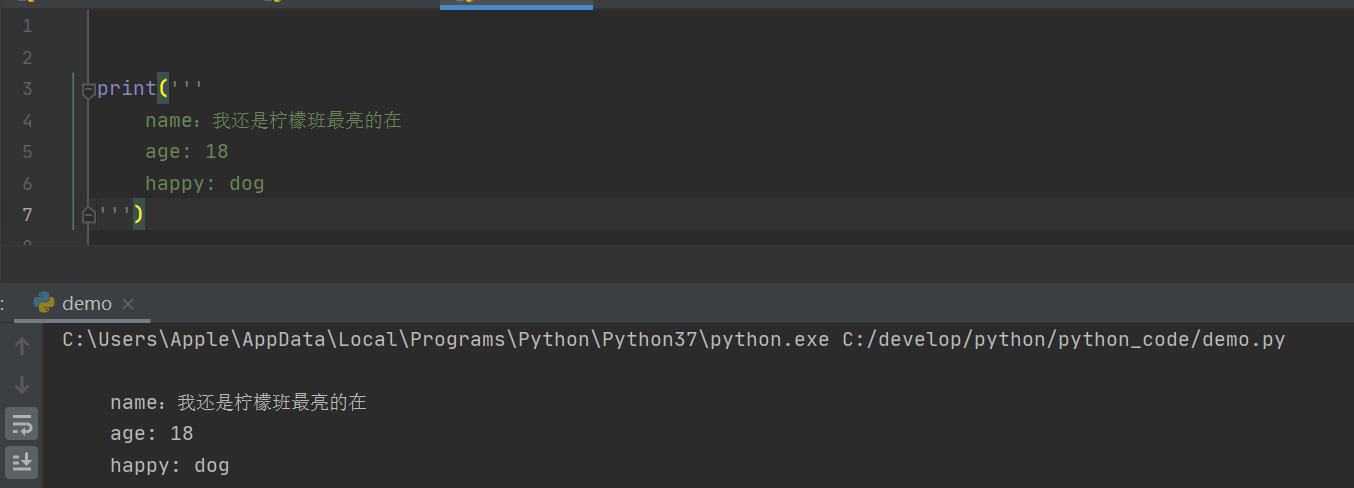
**4.4.1 双引号的使用，保持引号输出：**

print('===="jak"====')



**4.4.2 三引号的使用，保持原样输出：**

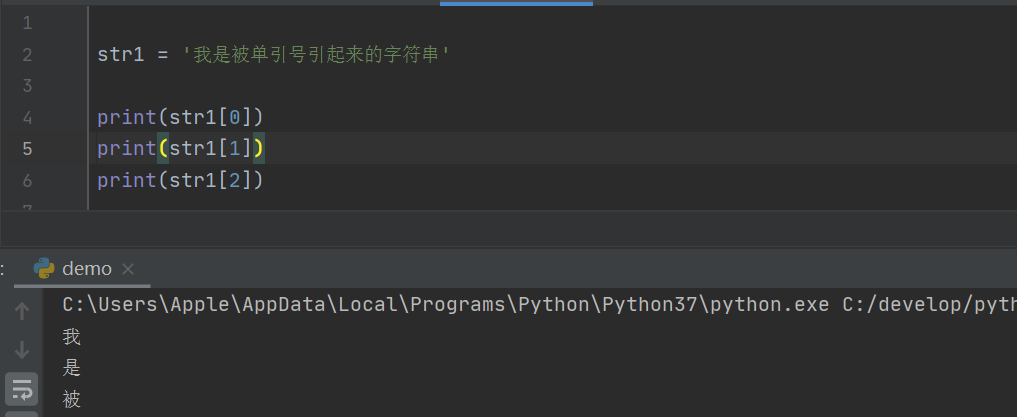
print('''  
 name：我还是柠檬班最亮的在  
 age: 18  
 happy: dog  
''')



### 4.5 字符串的常用操作

#### 4.5.1切片---取值 ---位置 = 索引 从0开始，依次往右排序

str1 = '我是被单引号引起来的字符串'  
  
print(str1[0])  
print(str1[1])  
print(str1[2])



#### 4.5.2切片---取多个值

(1)切片---- [索引头:索引尾:步长]

str1 = '我是被单引号引起来的字符串'  
print(str1[0:5:1]) # 下标从0开始，取到第五位，步长为1，每次取一个值

结果：我是被单引

说明：中括号[]左闭右开，所以只能取到下标为5之前的值。

str1 = '我是被单引号引起来的字符串'  
print(str1[0:14:2]) # 下标从0开始，取到第14位，步长为1，每两步取一个值

结果：我被引引来字串

1. 切片的默认值

索引头：默认为：0 索引尾：最后的值 步长：默认为：1

1. 取最后一个元素 [-1]

str1 = '我是被单引号引起来的字符串'  
print(str1[-1])

结果：串

1. 逆序取值：步长值为：-1

str1 = '我是被单引号引起来的字符串'  
print(str1[::-1])

结果：串符字的来起引号引单被是我

#### 4.5.3 len() 计算长度函数

str1 = '我是被单引号引起来的字符串'  
print(f'字符串str1的长度为：{len(str1)}')

结果：字符串str1的长度为：13

4.5.4 len()在切片中的应用

# 取的字符串内容使用len()，这样不管字符串长度有多长，从起始位开始，后面的内容都被取出来。

str1 = '我是被单引号引起来的字符串'  
print(str1[6:len(str1):1])

结果：引起来的字符串

#### 4.5.6 find() 字符串查找函数，根据字符串查出字符串的下标，元素不存在，返回-1，不报错。

str1 = '我是被单引号引起来的字符串'

print(str1.find("单"))

结果：3

#### 4.5.7 index() 字符串查找函数，根据字符串查出字符串的下标，元素不存在，直接报错，程序中断。

str1 = '我是被单引号引起来的字符串'  
print(str1.index("双"))

结果：

Traceback (most recent call last):

File "C:/develop/python/python\_code/demo.py", line 4, in <module>

print(str1.index("双"))

ValueError: substring not found

4.5.8 replace()函数，字符串替换函数

str1 = '我是被单引号引起来的字符串'  
print(str1.replace("我是","你是")) # 注意：替换与被替换的字符串中间用逗号隔开

结果：

你是被单引号引起来的字符串

# 【第一节课作业】

1. 下面哪些不能作为标识符？
2. find
3. \_num
4. 7Val
5. add
6. def

# 【第二节课笔记】

## 5.格式化输出

5.1第一种，格式化输出：字符串里传入变量值的时候，.format() ---函数

'''  
 格式化输出：.format() 函数  
'''  
name\_1 = "张恨水"  
age\_1 = 18  
gender\_1 = "男"  
print('''基本信息:  
 name:{}   
 age:{}   
 gender:{}'''.format(name\_1,age\_1,gender\_1))

结果：

基本信息:

name:张恨水

age:18

gender:男

5.2 第二种，% ----占位符

%s -----字符串

%d ----数字

%f ----浮点型

name\_1 = "张恨水"  
age\_1 = 18  
gender\_1 = "男"  
print('''基本信息:  
 name:%s   
 age:%s   
 gender:%s'''%(name\_1,age\_1,gender\_1))

结果：

基本信息:

name:张恨水

age:18

gender:男

## Python运算符

### 算术运算符：加(+) 减(-) 乘(\*) 除(/) 取余(%) 平方(\*\*) (\* \*\*)

'''  
Pyhon的算术运算符  
算术运算符：加(+) 减(-) 乘(\*) 除(/) 取余(%) 平方(\*\*)   
'''  
print(f'加法：{1+2}')  
print(f'减法：{1-2}')  
print(f'乘法：{1\*2}')  
print(f'除法：{1/2}') # 结果是字符串类型  
print(f'除法，取商：{2%5}')

结果：

基本信息:

name:张恨水

age:18

gender:男

加法：3

减法：-1

乘法：2

除法：0.5

除法，取商：2

### 赋值运算符： = 左边的赋值给右边的变量 += -=

name\_02 = "张索尼"  
age = 18  
age += 1  
age -= 2  
print(name\_02,age)

结果：张索尼 17

### 比较运算符： > >= < <= == != （python2的不等于：<>） 的结果是：True、False

print(10<20)  
print(10>20)  
print(10<=10)  
print(10>=10)  
print(10==10)

节结果：

True

False

True

True

True

### 逻辑运算符 and or not

a = 10  
b = 50  
print(a<b and a>b )  
print(a<b or a>b )

结果：

False

True

### 6.5 成员运算符 in not in

a\_1 = "hello word!"  
b\_1 = "you are cool"  
print("he" in a\_1)  
print("you" not in b\_1)

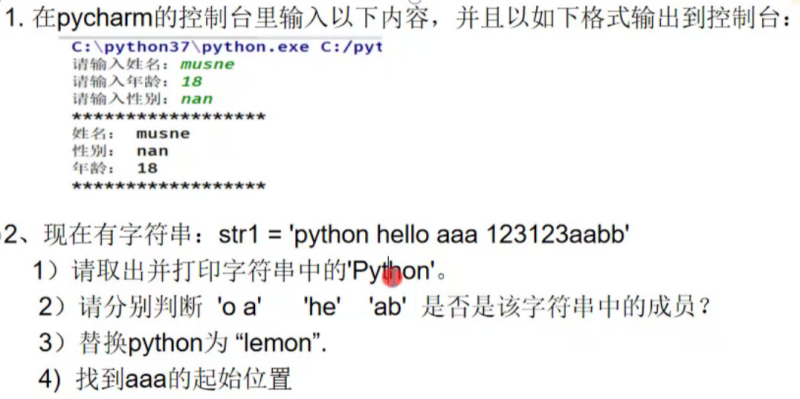
结果：

True

False

【

# 【第二节课作业】



**题目1.**

name = input('请输入你的名字：')  
age = input('请输入你的年龄：')  
sex = input('请输入你的性别：')

print('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*')  
print('姓名：'+name+'\n'+'年龄：'+age+'\n'+'性别：'+sex)  
print('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*')

结果：

请输入你的名字：柠檬

请输入你的年龄：18

请输入你的性别：女

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

姓名：柠檬

年龄：18

性别：女

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**题目2：**

**1）请取出并打印字符串中的 ‘Python’**

# 1) 请取出并打印字符串中的'Python'。  
str1 = 'python hello aaa 123123aabb'  
print(f'原字符串: {str1}')  
print(f'取出的字符串： {(str1[0:7]).capitalize()}')

结果：

原字符串: python hello aaa 123123aabb

取出的字符串： Python

**2）请分别判断 'o a' 'he' 'ab' 是否是该字符串中的成员?**

# 2) 请分别判断 'o a''he’'ab’是否是str1字符串中的成员?

str1 = 'python hello aaa 123123aabb'  
print('a o' in str1)  
print('he' in str1)  
print('ab' in str1)

结果：

False

True

True

1. **替换”python” 为“lemon”**

# 3)替换python为“lemon"  
str1 = 'python hello aaa 123123aabb'  
print(str1.replace("python","lemon"))

结果：lemon hello aaa 123123aabb

1. **找到aaa的起始位置**

# 4) 找到aaa的起始位置

str1 = 'python hello aaa 123123aabb'  
print(str1.find('aaa'))  
print(str1.index('aaa'))

结果：13

# 【第三节课笔记】

## 列表 list[ ] 内置函数：list()

### 7.1列表的定义及特点

列表里元素可以是任意的字符类型：int、float、bool、str、list，tuple，dict元素之间用英文逗号隔开，列表元素可以重复、改变。

list1 = [] # 空列表

list\_2 = [22,"liumazi",{"name":"zhaosi"},True ,(1,2),[1,2,3,4,5]]

结果：

[]

[22, 'liumazi', {'name': 'zhaosi'}, True, (1, 2), [1, 2, 3, 4, 5]]

### 7.2 增加列表

list\_2 = [1,"liumazi",{"name":"zhaosi"},True ,(1,2),[1,2,3,4,5]]

# 增加元素  
list\_2.append("海王") # 添加元素到列表---追加到最后  
print(list\_2)  
list\_2.extend([{"name":"mixi"},(66,99)]) # 添加多个元素到列表  
print(list\_2)  
list\_2.insert(2,"zhaosi") # 添加元素到指定位置  
print(list\_2)  
list\_2.append(["Andy","Rose"]) # 保持原样列表格式添加到列表  
print(list\_2)

结果：

[1, 'liumazi', {'name': 'zhaosi'}, True, (1, 2), [1, 2, 3, 4, 5], '海王']

[1, 'liumazi', {'name': 'zhaosi'}, True, (1, 2), [1, 2, 3, 4, 5], '海王', {'name': 'mixi'}, (66, 99)]

[1, 'liumazi', 'zhaosi', {'name': 'zhaosi'}, True, (1, 2), [1, 2, 3, 4, 5], '海王', {'name': 'mixi'}, (66, 99)]

[1, 'liumazi', 'zhaosi', {'name': 'zhaosi'}, True, (1, 2), [1, 2, 3, 4, 5], '海王', {'name': 'mixi'}, (66, 99), ['Andy', 'Rose']]

### 7.3 删除列表元素

#### 7.3.1 pop()函数，无参数默认删除最后一个元素

# 删除元素  
list\_7 = [1,"liumazi",{"name":"zhaosi"},True ,(1,2),[1,2,3,4,5]]  
list\_7.pop() # 默认删除最后一个元素

print(list\_7)

list\_7.pop(2) # 也可以指定索引进行删除  
print(list\_7)

结果：

[1, 'liumazi', {'name': 'zhaosi'}, True, (1, 2)]

[1'，liumazi', True, (1, 2)]

#### 7.3.2 remove() 函数，指定元素本身进行删除

# remove()函数  
list\_8 = [1,"liumazi",{"name":"zhaosi"},True ,(1,2),[1,2,3,4,5]]  
list\_8.remove("liumazi")  
print(list\_8)

结果：

[1, {'name': 'zhaosi'}, True, (1, 2), [1, 2, 3, 4, 5]]

### 7.4 修改列表元素

# 修改列表元素  
list\_9 = [1,"liumazi",{"name":"zhaosi"},True ,(1,2),[1,2,3,4,5]]  
list\_9[0] ="憨憨"  
list\_9[3] = False  
print(list\_9)

结果：

[1, {'name': 'zhaosi'}, True, (1, 2), [1, 2, 3, 4, 5]]

['憨憨', 'liumazi', {'name': 'zhaosi'}, False, (1, 2), [1, 2, 3, 4, 5]]

### 7.5 统计列表的元素长度、个数

print(len(list\_9))

结果：6

## 元组 tuple

### 元组的定义，元组是不可变的

tuple\_1 = (1,2,3,4,5)  
print(tuple\_1)

结果：

(1, 2, 3, 4, 5)

### 元组转换成列表

tuple\_2 = (1,2,3,4,5)  
print(f'元组转换成列表前：{tuple\_2}')  
print(f'元组转换列表后：{list(tuple\_2)}')  
print(f'列表转换元组：{tuple(tuple\_2)}')

结果：

元组转换成列表前：(1, 2, 3, 4, 5)

元组转换列表后：[1, 2, 3, 4, 5]

列表转换元组：(1, 2, 3, 4, 5)

## 9.字典 dict

### 9.1 定义

### 字典没有索引，可以通过key、get()方法取值，key是唯一的，不可重复的和不可变的，所以key不能使用列表和字典本身来定义，最常用的是字符串来定义。

#### 9.1.1 元素为一对键值对：{key:value},key---键值 value---值

#### 9.1.2 字典使用场景：描述一个物体的属性---name: Bobo，age:18, city:深圳

dict\_1 = {"name":"Bobo","age":18,"city":"深圳"}  
print(dict\_1)

结果：

{'name': 'Bobo', 'age': 18, 'city': '深圳'}

### 9.2 字典的取值

dict\_1 = {"name":"Bobo","age":18,"city":"深圳"}  
print(dict\_1)  
print(dict\_1["name"]) # 直接通过key去取value  
print(dict\_1.get("age")) # 通过get方法去取value

结果：

Bobo

18

### 9.3 字典的增加及修改，如果增加的元素原来就有，那么就会替换原来。

**单个增加：**

dict\_2 = {"name":"Bobo","age":18,"city":"深圳"}  
dict\_2["身高"] = 160  
dict\_2["体重"] = 60

print(dict\_2)

dict\_2["age"] = 25  
print(dict\_2)

结果：

{'name': 'Bobo', 'age': 18, 'city': '深圳', '身高': 160, '体重': 60}

{'name': 'Bobo', 'age': 25, 'city': '深圳', '身高': 160, '体重': 60}

**多个增加：**

dict\_2 = {"name":"Bobo","age":18,"city":"深圳"}  
dict\_2.update({"gender":"女","Hobby":"学习"})  
print(dict\_2)

结果：

{'name': 'Bobo', 'age': 18, 'city': '深圳', 'gender': '女', 'Hobby': '学习'}

### 9.4 字典的删除，因为字典是无序的，所以字典的 pop()方法必须传 key

dict\_3 = {"name":"Bobo","age":18,"city":"深圳"}  
dict\_3.pop("name")  
print(dict\_3)

结果：

{'age': 18, 'city': '深圳'}

## 10.集合set{}

### 10.1 集合的定义，集合不可重复的，无序的

set\_1 = {1,2,3,4,5}  
print(set\_1)

结果：

{1, 2, 3, 4, 5}

### 10.2 集合的使用场景

#### 10.2.1 对list去重

list\_8 = [22,22,33,33,"aa","aa"]  
print(set(list\_8))

结果：

{33, 'aa', 22}

## 流程控制

### if语句

if 判断条件:

执行的语句

else if:

执行的语句

else:

执行的语句

money = int(input("请输入你的存款金额:"))  
if money >= 500:  
 print("学习python")  
elif money >= 100:  
 print("学习java")  
else:  
 print("放弃吧")

结果：

请输入你的存款金额:300

学习java

请输入你的存款金额:100

学习java

请输入你的存款金额:50

放弃吧

### for循环

list\_001 = ["aaa","bbb","ccc"]  
for i in list\_001:  
 print(i)

print("\*\*" \* 20)

结果：

aaa

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

bbb

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

ccc

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

#### break 跳出循环，后面的代码都不会再执行

list\_001 = ["aaa","bbb","ccc"]  
for i in list\_001:  
 if i == "bbb":

Break

print(i)

print("\*\*" \* 20)

结果：

aaa

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

#### continue 跳出本次循环，后面的代码还会执行

list\_001 = ["aaa","bbb","ccc"]  
for i in list\_001:  
 if i == "bbb":  
 continue   
 print(i)  
 print("\*\*" \* 20)

结果：

aaa

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

ccc

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

## range () 函数，生成一个整数序列

**参数：start stop step**

'''  
range() 函数，经常与for循环使用  
函数的参数：start stop step , 取头不取尾，stop必须要写。  
'''  
for num in range(1,10,2):  
 print(num)

结果：

1

3

5

7

9

# 【第三节课作业】

# 1.使用for循环遍历列表、字典、元组  
  
print('=======================使用for循环遍历列表元素=======================')  
list\_011 = ["柠檬","苹果","香蕉","葡萄"]  
for i in list\_011:  
 print(i)

# 输出结果：

柠檬

苹果

香蕉

葡萄  
print('=======================使用for循环遍历列表元素=======================')  
  
  
print('=======================使用for循环遍历字典值========================')  
dict\_011 = dict(name="李四",age=22,city= "深圳")  
for items in dict\_011.values():  
 print(items)

# 输出结果：

李四

22

深圳  
print('=======================使用for循环遍历字典值========================')  
  
  
print('=======================使用for循环遍历元组==========================')  
girl\_tuple = ("貂蝉", "狐狸精","范金链","翠花","小班")  
for everyOne in girl\_tuple:  
 print(everyOne)

# 输出结果：

貂蝉

狐狸精

范金链

翠花

小班  
print('=======================使用for循环遍历元组==========================')

# 第1题: a=[1,2,'6','summer']请用成员运算符判断i是否在这个列表里面  
a=[1,2,'6','summer']  
print(f'判断的结果是：{"i" in a}')

# 输出结果：False  
# 输出结果：

判断的结果是：False  
  
  
# 第2题: dict\_1={"class\_id":45,'num':20}请判断班级上课人数是否大于5  
# : num表示课堂人数。如果大于5，输出人数。  
dict\_1={"class\_id":45,'num':20}  
if dict\_1["num"] > 5:  
 print(f'班级人数是：{dict\_1["num"]}')  
# 输出结果：

班级人数是：20  
  
# 第3题、list3=['菠萝','球球'，'萌萌','瓜瓜'，'兔兔,'不停']  
'''  
列表当中的每一个值包含:姓名、性别、年龄、城市。以字典的形式表达。并且把字典都存在新的list中，最后打印最终的列表。  
方法1:手动扩充--字典--存放在列表里面;{}  
方法2:自动--dict() --不强制list.appen()  
'''  
# 方式一：  
print('===========方法1:手动扩充--字典--存放在列表里面;{}===============')  
list3=['菠萝','球球','萌萌','瓜瓜','兔兔','不停']  
list3\_01 = []  
list3\_01.append({"姓名":"菠萝","性别":"男","年龄":18,"城市":"北京"})  
list3\_01.append({"姓名":"球球","性别":"男","年龄":19,"城市":"深圳"})  
list3\_01.append({"姓名":"萌萌","性别":"男","年龄":20,"城市":"深圳"})  
list3\_01.append({"姓名":"瓜瓜","性别":"男","年龄":21,"城市":"上海"})  
list3\_01.append({"姓名":"兔兔","性别":"男","年龄":22,"城市":"广州"})  
list3\_01.append({"姓名":"不停","性别":"男","年龄":23,"城市":"北京"})  
print(list3\_01)  
print('===========方法1:手动扩充--字典--存放在列表里面;{}===============')  
# 输出的结果：

[{'姓名': '菠萝', '性别': '男', '年龄': 18, '城市': '北京'}, {'姓名': '球球', '性别': '男', '年龄': 19, '城市': '深圳'}, {'姓名': '萌萌', '性别': '男', '年龄': 20, '城市': '深圳'}, {'姓名': '瓜瓜', '性别': '男', '年龄': 21, '城市': '上海'}, {'姓名': '兔兔', '性别': '男', '年龄': 22, '城市': '广州'}, {'姓名': '不停', '性别': '男', '年龄': 23, '城市': '北京'}]

# 方式二：

neme\_list = ['波波','蛋蛋','兰兰','花花','兔兔','洋洋']  
gender = [0,1,1,0,0,1] # 0：男 1：女  
age\_list = [22,18,23,19,20,18]  
city\_list = ['上海','宁波','深圳','广州','北京','天津']  
  
user\_info\_list = []  
for i in range(len(neme\_list)):  
 user\_info\_dict = dict(姓名 = neme\_list[i],性别 =gender[i] ,年龄 = age\_list[i],城市 = city\_list[i])  
 user\_info\_list.append(user\_info\_dict)  
print(user\_info\_list)

print('===========方法2:自动--dict()--不强制list.appen()=============')

# 输出的结果：

[{'姓名': '波波', '性别': 0, '年龄': 22, '城市': '上海'}, {'姓名': '蛋蛋', '性别': 1, '年龄': 18, '城市': '宁波'}, {'姓名': '兰兰', '性别': 1, '年龄': 23, '城市': '深圳'}, {'姓名': '花花', '性别': 0, '年龄': 19, '城市': '广州'}, {'姓名': '兔兔', '性别': 0, '年龄': 20, '城市': '北京'}, {'姓名': '洋洋', '性别': 1, '年龄': 18, '城市': '天津'}]