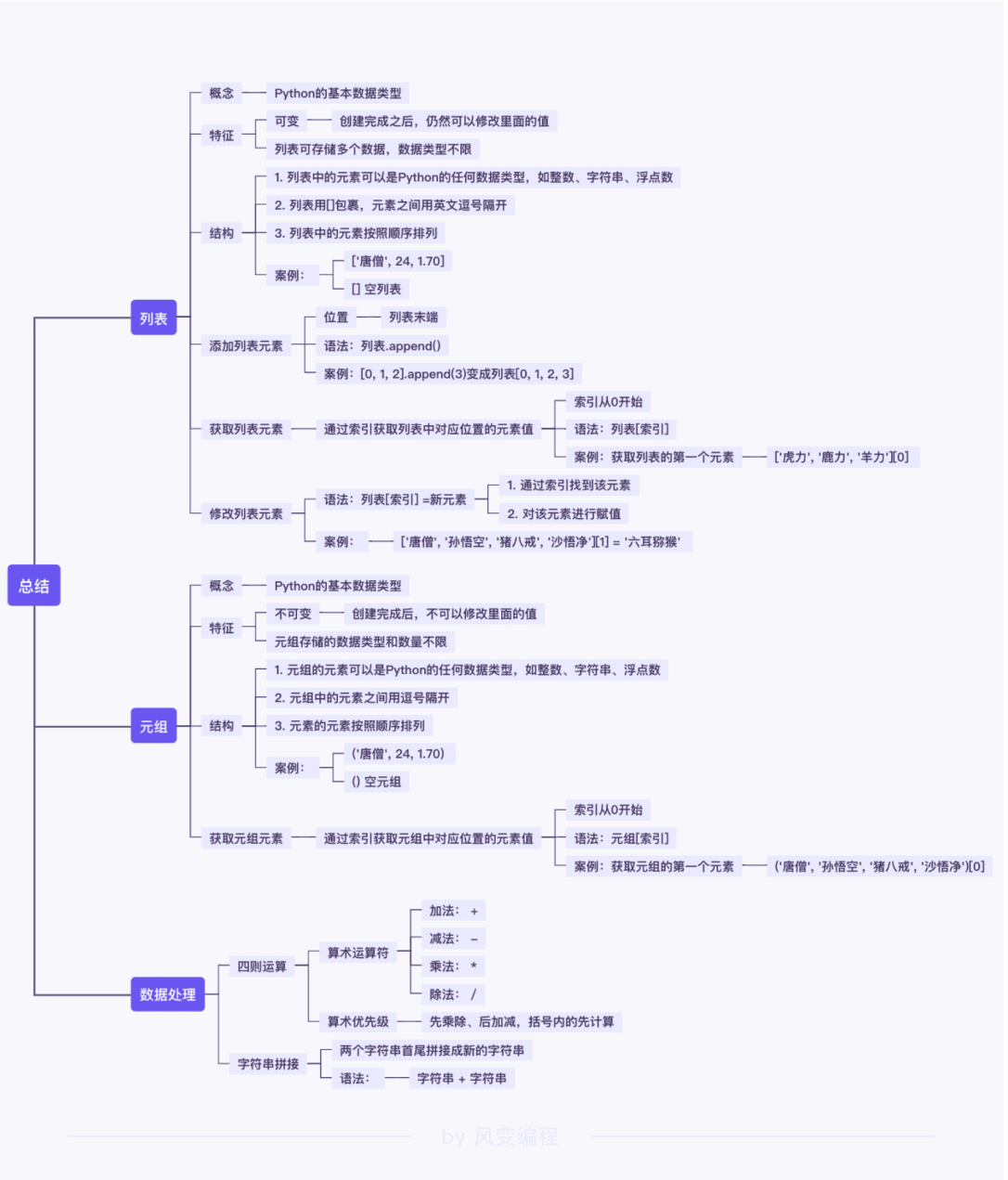


# 基础语法第1关笔记



## 一、列表与元组

与整数、浮点数、字符串一样，列表和元组也是Python的基本数据类型：



## 1.列表与元组的定义与区别

你可以将它们想象成两间“酒店”，你认识的所有数据都可以“住”进去：

```
1 # 创建一个列表
2 [1, '草莓蛋糕', 13.0, 2, '抹茶蛋糕', 14.0, 3, '提拉米苏', 15.0]
3 # 创建一个元组
5 (1, '草莓蛋糕', 13.0, 2, '抹茶蛋糕', 14.0, 3, '提拉米苏', 15.0)
```

上面一个列表，一个元组，两间“酒店”各存储了9个元素，元素分别属于三种不同的数据类型：整数（如2）、字符串（如‘抹茶蛋糕’）和浮点数（如15.0）。

列表和元组的形态非常相似，一个是中括号[]，一个是圆括号()。注意两者都是英文的括号。

它们最大的区别是：列表“可变”，元组“不可变”。

“可变”的意思是说，列表在创建完成后，我们还可以增添元素，或对里面的元素进行修改。

相反“不可变”的意思是说，元组创建完，就定型了，不可增添元素，元素也不能再做修改了。

### 列表

```
1 [1, '草莓蛋糕', 13.0, 2, '抹茶蛋糕', 14.0, 3, '提拉米苏', 15.0]
```

- ✓ 可以存储不同的数据类型
- ✓ 可以存储多个数据
- ✓ 创建之后可以修改或增加数据

### 元组

```
1 (1, '草莓蛋糕', 13.0, 2, '抹茶蛋糕', 14.0, 3, '提拉米苏', 15.0)
```

- ✓ 可以存储不同的数据类型
- ✓ 可以存储多个数据
- ✗ 创建之后可以修改或增加数据

by 风变编程

## 2.列表和元组的使用

### 2.1列表

列表的结构:

- ① 中括号[]将数据集中起来;
- ② 不同元素之间用英文逗号,隔开。

列表可以存储多个数据,当然也可以啥也不存,那就是空列表:[]。

举例:

### 列表

```
1 journey = ['唐僧', '孙悟空', '猪八戒', '沙悟净']
```

变量名

左中括号      英文逗号      右中括号

by 风变编程

列表的append()方法:

它的语法是这样的:列表.append(元素)。

```
1 journey = ['唐僧']
2 print(journey)
3 # 列表添加新元素'孙悟空'
```

```
4 journey.append('孙悟空')
5 print(journey)
6 #结果为:
8 # ['唐僧']
9 # ['唐僧', '孙悟空']
```

列表索引:



在Python中，“门牌号”被称为索引值，索引值从0开始，列表中的每个元素都有自己的索引值。

列表[0]表示列表的第一个元素，列表[1]则表示列表的第二个元素，以此类推。



通过索引值就可以访问到列表中的元素，语法是这样的：列表[索引值]。

```
1 monster = ['虎力', '鹿力', '羊力']
2 # 访问列表当中的第一个元素'虎力'，并将元素打印出来
```

```
3 print(monster[0])
5 # 结果:
6 # 虎力
```

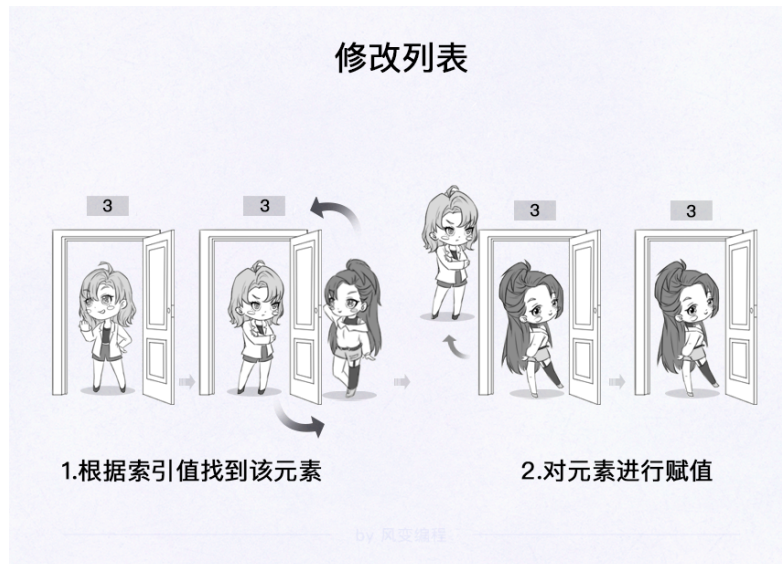
### 修改列表:

回到列表“大酒店”，“酒店”换房不过是两个步骤：

- ① 根据房间号找到房间的“主人”
- ② 更换房间的“主人”

列表修改也是：

- ① 根据索引值找到元素
- ② 替换元素



这两个步骤一条语句就可以实现，替换元素需要借助**赋值符号=**来实现，语法为**列表[索引值] = 新元素**。

```
1 journey = ['唐僧', '孙悟空', '猪八戒', '沙悟净']
2 # 将列表中的元素'孙悟空'修改成'六耳猕猴'
3 journey[1] = '六耳猕猴'
4 print(journey)
5 # 结果
6 # ['唐僧', '六耳猕猴', '猪八戒', '沙悟净']
```

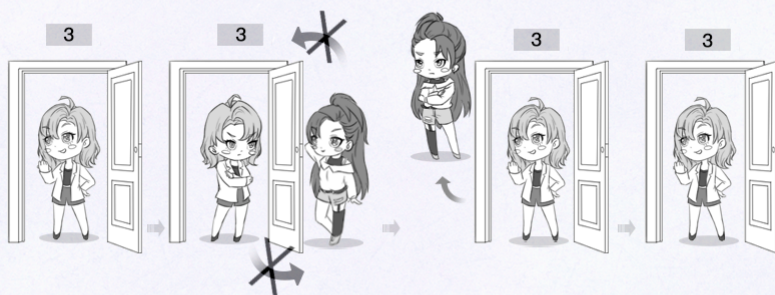
## 2.2 元组

元组和列表的特性很像。

和列表一样，元组也可以通过索引值的方式获取元素，语法是：元组[索引值]。

不同的是，不能对创建好的元组进行修改，包括修改元素，增添元素等，所以元组不能使用append()方法。

## 修改元组



元组不支持元素修改

by 风安编程

## 二、数据处理

### 3.1 合并列表

两个列表可以使用最简单的加号+进行合并：

```
1 list1 = [1,2,3]
2 list2 = [4,5,6]
3 # 拼接两个列表
4 list3 = list1 + list2
5 print(list3)
6 # 结果：
8 # [1,2,3,4,5,6]
```

### 3.2 四则运算

运算符	作用
+	加
-	减
*	乘
÷	除

by 风安编程

当好几个运算符放在一起的时候，运算就有优先级之分。

口诀：先乘除后加减，有括号先算括号。

除了列表和数字可以相加，字符串也可以。

### 3.3 字符串拼接

两个字符串相加，相当于两个字符串进行首尾拼接。

```
1 print('5'+ '2')
2 # 结果:
3 # 52
```

列表只能和列表相加，数字只能和数字相加，字符串也只能和纯字符串相加。

当多个字符串通过+符号相加，运算的结果是字符串从左到右进行首尾拼接。

```
1 name = ['祁祁', '迪哥']
2 place = ['图书馆', '餐厅', '沙滩上', '家里', '马路上', '办公室']
3 action = ['告白', '吃西瓜', '开车', '爬山', '洗脚', '唱歌']
4 # 通过字符串拼接，打印句子'祁祁在图书馆吃西瓜'
5 print(name[0] + '在' + place[0] + action[1])
6 # 通过字符串拼接，打印句子'迪哥在马路上开车'
7 print(name[1] + '在' + place[4] + action[2])
8 # 通过字符串拼接，打印句子'祁祁在沙滩上洗脚'
9 print(name[0] + '在' + place[2] + action[4])
10 # 结果:
11 # 祁祁在图书馆吃西瓜
12 # 迪哥在马路上开车
13 # 祁祁在沙滩上洗脚
```