

第二章：序列类型的操作方法代码笔记

列表

增

```
# 追加
# 列表名.append(元素)          将制定元素追加到列表的末尾
list1 = ["M762猛男枪", "7.62mm子弹", "平底锅", 1024, "1024"]
list1.append("手榴弹")        # 由于列表是可变数据类型，因此这个追加是直接在源数据进行修改的
print(list1)

# 插入
# 列表名.insert(下标, 元素)      将指定数据插入到指定位置
list2 = ["M762猛男枪", "7.62mm子弹", "平底锅"]
list2.insert(0, 1024)         # 将 1024 插入到list2下标为 0 的位置
print(list2)

# 追加序列
# 列表名.extend(序列元素)        将指定序列中的元素提取出来，逐个追加到列表末尾
list3 = ["M762猛男枪", "7.62mm子弹", "平底锅"]
list3.extend(("大花袄", "吉利服", "沙滩裙"))
print(list3)
```

删

```
# 下标删除
# 列表名 .pop(下标)            根据下标删除元素，并返回被删除元素。
# 列表名.pop()                不填下标默认删除最后一个元素。
li = ["M762猛男枪", "7.62mm子弹", "平底锅"]
x = li.pop(2)                 # 在删除完 下标2对应的数据后，会将删的数据进行返回
print("被删的数据:", x, "  被删后的列表: ", li)    # 打印

x = li.pop()                  # 没有填下标，在删除完 默认下标-1对应的数据后，会将删的数据进行返回
print("被删的数据:", x, "  被删后的列表: ", li)    # 打印

# 删除指定元素
# 列表名.remove(元素)          删除列表中的指定元素
li = ["M762猛男枪", "7.62mm子弹", "平底锅", 1024]
li.remove(1024)
print("被删后的列表: ", li)

# 清空列表
li = ["M762猛男枪", "7.62mm子弹", "平底锅", 1024]
li.clear()
print("被清空后的列表", li)

# 删除一段数据
li = ["M762猛男枪", "7.62mm子弹", "平底锅", 1024]
```

```
#           0           1           2           3
del li[1:3:1]      # 先找个 li[1:3:1] 切片的这段数据，然后将这段数据进行删除
print(li)
```

改

```
# 修改（先查后改）
li = ["M762猛男枪", "7.62mm子弹", "平底锅", 1024]
#                                     3/-1
# 希望将 1024 变成 手榴弹
li[3] = "手榴弹"
# 先找到要替换的数据的位置，然后传一个新的值，将旧值进行顶替
print(li)
```

查

```
# 查下标
# 列表名.index(元素)           查询指定元素的下标
# 如果查询到下标就返回      -> index只查询一次，查到之后就不在进行
# index 翻译成中文是索引的意思
li = ["M762猛男枪", 1024, "7.62mm子弹", "平底锅", 1024]
#           0           1           2           3           4
x = li.index(1024)           # 查到会返回查询结果
print(x)

# 查数量
# 列表名.count(元素)           统计指定元素在列表中出现的次数，统计好的数据会返回
li = ["M762猛男枪", 1024, "7.62mm子弹", 1024, "平底锅", 1024]
x = li.count(1024)
print(x)
```

元组

查

```
# 查
# 查下标
# 列表名.index(元素)           查询指定元素的下标
# 如果查询到下标就返回      -> index只查询一次，查到之后就不在进行
# index 翻译成中文是索引的意思
li = ("M762猛男枪", 1024, "7.62mm子弹", "平底锅", 1024)
#           0           1           2           3           4
x = li.index(1024)           # 查到会返回查询结果
print(x)

# 查数量
# 列表名.count(元素)           统计指定元素在列表中出现的次数，统计好的数据会返回
li = ("M762猛男枪", 1024, "7.62mm子弹", 1024, "平底锅", 1024)
x = li.count(1024)
print(x)
```

字符串

增

```
# 增加    +    拼接符号
str1 = '窗前明月光'
str2 = '疑是地上霜'
str3 = str1+", "+str2+"。"
print(str3)
```

删

```
# 删除
str1 = "床前明月光,疑是地上霜。"
# 字符串名.replace(旧值,新值)    以新值替换旧值
str2 = str1.replace('疑是地上霜','低头思故乡')
print("str1----->",str1)    # 不可变数据类型,源数据无法改变
print("str2----->",str2)    # 修改后新生成的数据

# 实现删除效果: 将指定的字符串,替换成空字符串。
str3 = str1.replace('疑是','')
print("str3---->",str3)
```

改

```
# 字符串名.upper()    转大写
str1 = "hello python !"
str2 = str1.upper()
print("str1----->",str1)    # 字符串是不可变数据类型,源数据无法改变
print("str2----->",str2)    # 修改后新生成的数据

# 字符串名.lower()    转小写
str1 = "HELLO PYTHON !"
str2 = str1.lower()
print("str1----->",str1)    # 字符串是不可变数据类型,源数据无法改变
print("str2----->",str2)    # 修改后新生成的数据

#字符串切割
zuoBiao = "东149,西北11,东北132,西南168"
# 字符串名.split(切割规则,次数)    切割规则: 根据什么进行切割    切割后的数据是一个列表包裹字符串的数据
shuju = zuoBiao.split(',',1)    # 字符串是不可变数据类型,源数据无法改变
print("切割后的数据:",shuju)
print("切割前的数据:",zuoBiao)
str1 = "今天天气好好哦,天哪天哪"
str2 = str1.split('天')    # 以天字为分隔符号,将字符串数据进行分开,并且依次保存到列表中
print("切割后的数据:",str2)
print("切割前的数据:",str1)

# 字符串名.replace(旧值,新值)    以新值替换旧值
str2 = str1.replace('疑是地上霜','低头思故乡')
print("str1----->",str1)    # 不可变数据类型,源数据无法改变
print("str2----->",str2)    # 修改后新生成的数据
```

查

```
# 查下标
# 列表名.index(元素)          查询指定元素的下标
# 如果查询到下标就返回      -> index只查询一次，查到之后就不在进行
# index 翻译成中文是索引的意思
li = "M762猛男枪,你好,1024,你好,7.62mm子弹,你好,平底锅"      # str
#           0           1           2           3           4
x = li.index('你好')      # 查到会返回查询结果
print(x)      # 输出8 ,是找到的 字符串1024第一个字符的下标

# 查数量
# 列表名.count(元素)          统计指定元素在列表中出现的次数,统计好的数据会返回
li = "M762猛男枪,1024,你好,7.62mm子弹,平底锅,你好,1024,1024"      # str
x = li.count('1024')
print(x)

# find  查询是否包含指定字符串元素，若包含，返回序列中第一个元素的下标，没有则返回-1
str1 = "床前明月光,疑是地上霜。"
x = str1.find("今天")
print(x)

str1 = "hello python !"
x = str1.islower()
print(x)      # True

y = str1.isupper()
print(y)      # False
```

课后作业

- 1、将课堂代码练习一遍，并写好注释
- 2、完成作业文档中的题目

刷题网站

http://www.pythontip.com/coding/code_oj