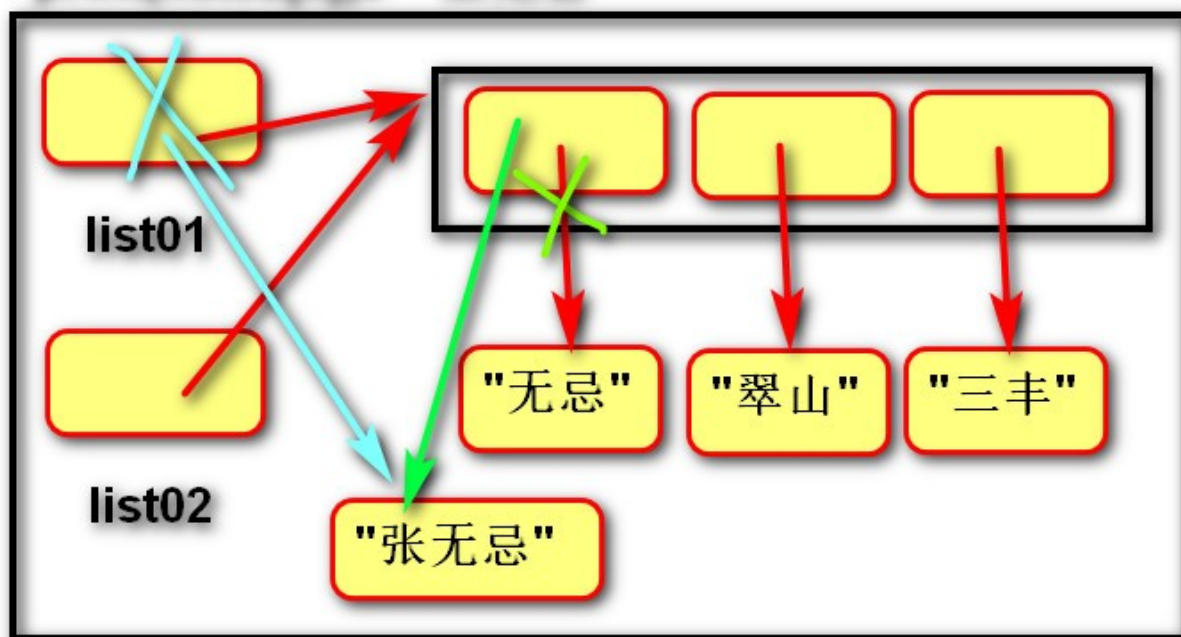


## 定义

## 基础操作

- 列表名 = list(可迭代对象)

```
print(list02[0])# "张无忌"
```

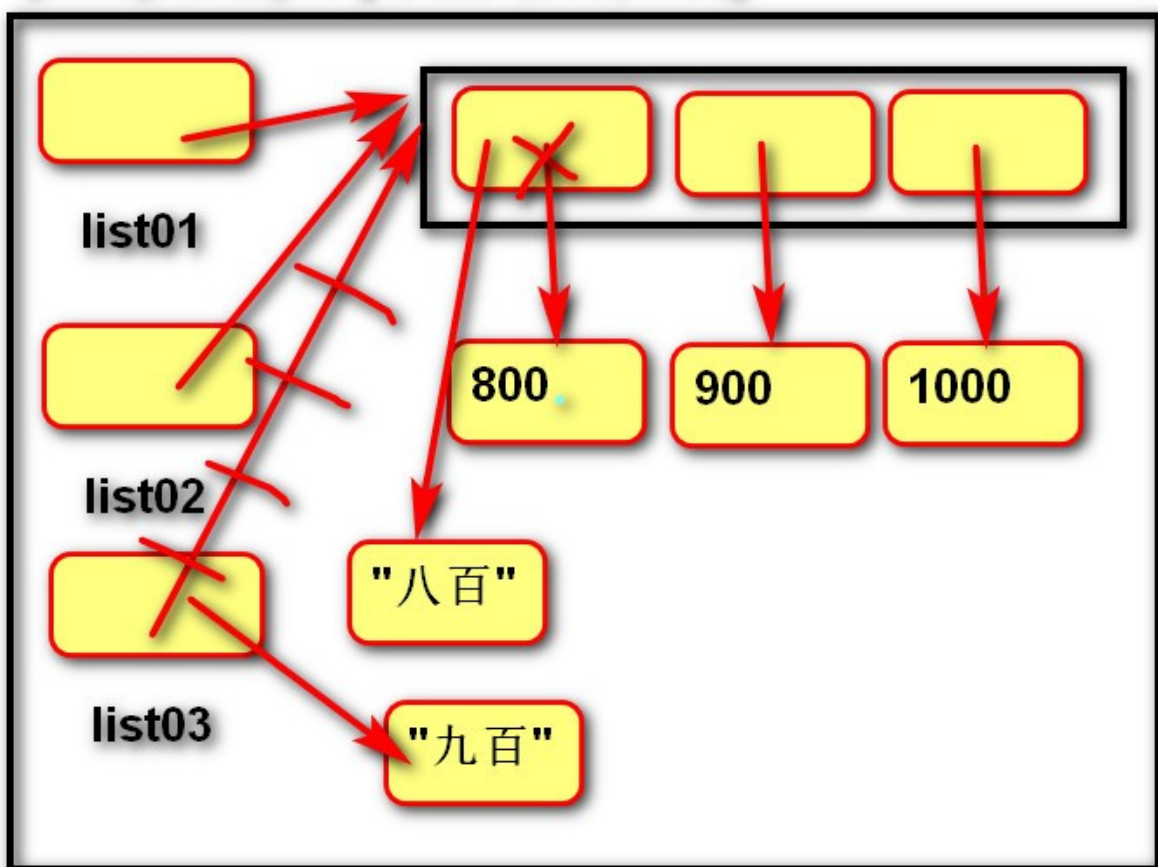


## # 修改列表第一个元素

```
# list01[0] = "张无忌"
# 修改变量list01的引用(指向)
list01 = "张无忌"
print(list02[0])
```

练习:

```
list01 = [800,900,1000]
list02 = list01
list03 = list01
list01[0] = "八百"
print(list02[0])? "八百"
list03 = "九百"
print(list02)?# ["八百",900,1000]
```



```
list01 = [800,900,1000]
list02 = list01
list03 = list01
list01[0] = "八百"
print(list02[0])# "八百"
list03 = "九百"
```

```
print(list02)# ["八百",900,1000]
```

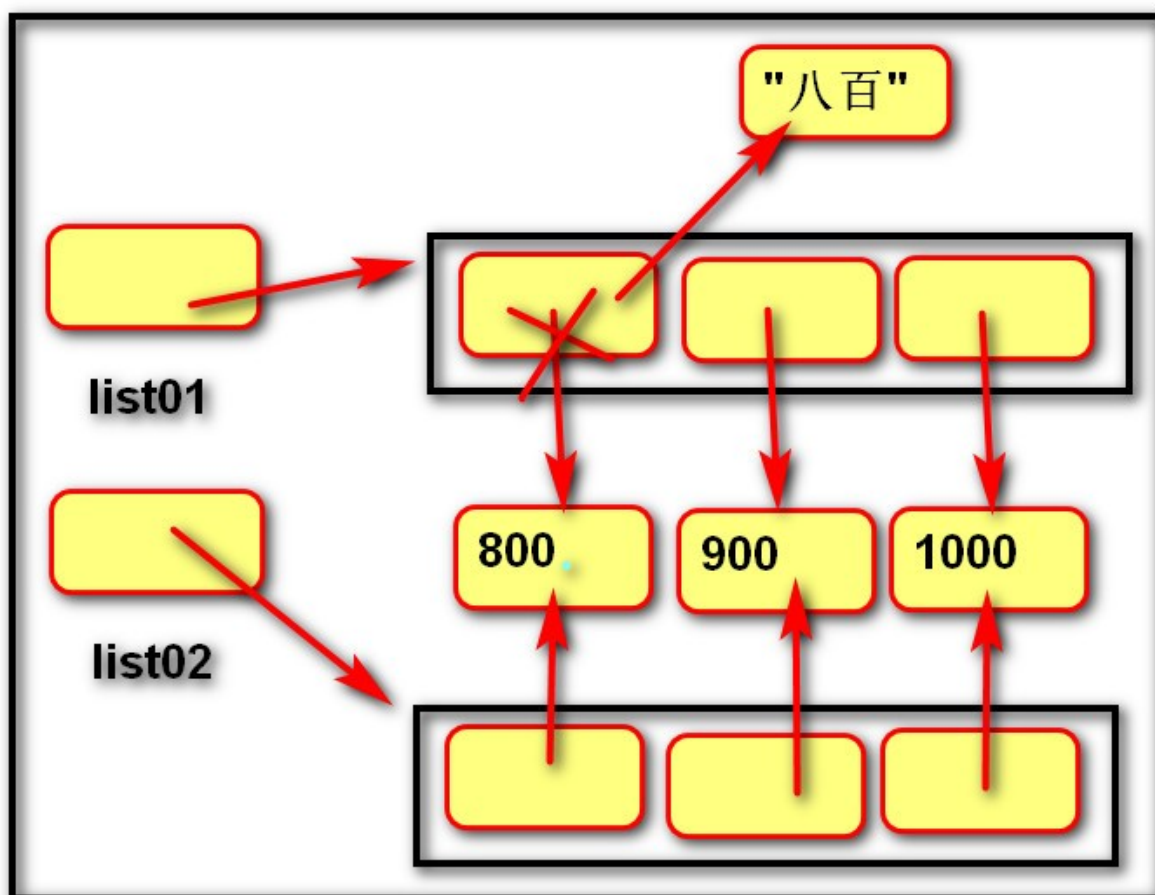
练习:

**list01 =**

**[800,900,1000]**

**list02 = list01[:]**

**list01[0] = "八百"**



```
list01 = [800,900,1000]
```

```
# 通过切片创建新列表，赋值给变量list02
```

```
list02 = list01[:]
```

```
list01[0] = "八百"
```

```
print(list02[0])#?
```

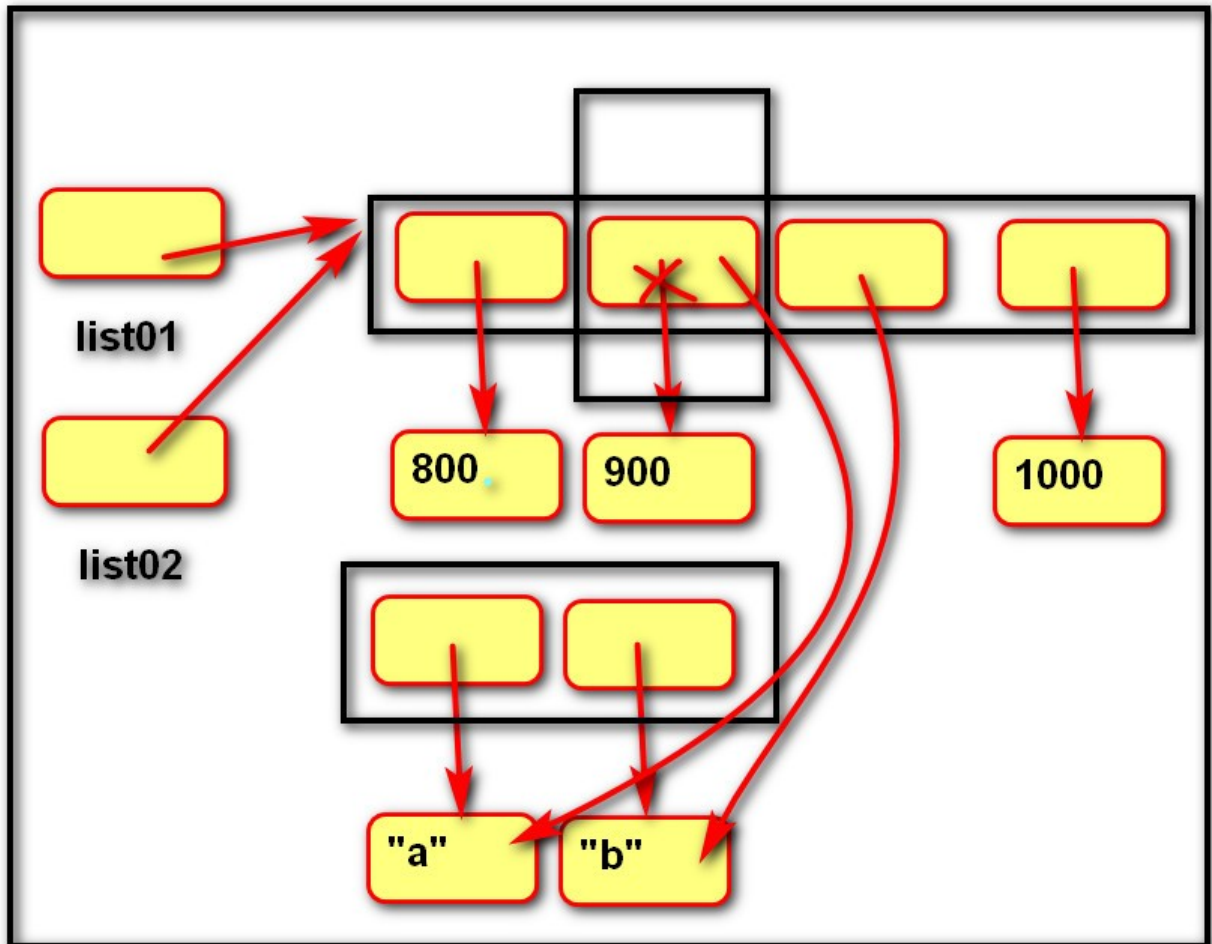
练习:

```
list01 = [800,900,1000]
```

```
list02 = list01
```

```
list01[1:2] = ["a","b"]
```

```
print(list02)
```



```
list01 = [800,900,1000]
```

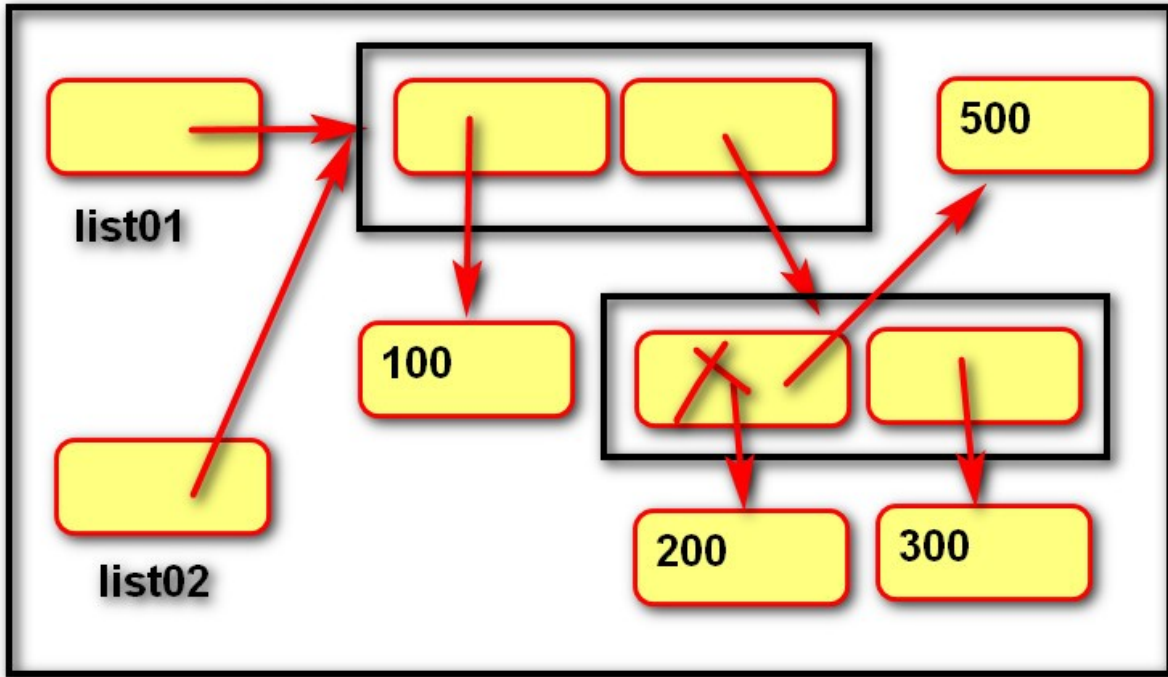
```
list02 = list01
```

```
# 定位第二个元素，然后进行修改。
```

```
list01[1:2] = ["a","b"]
```

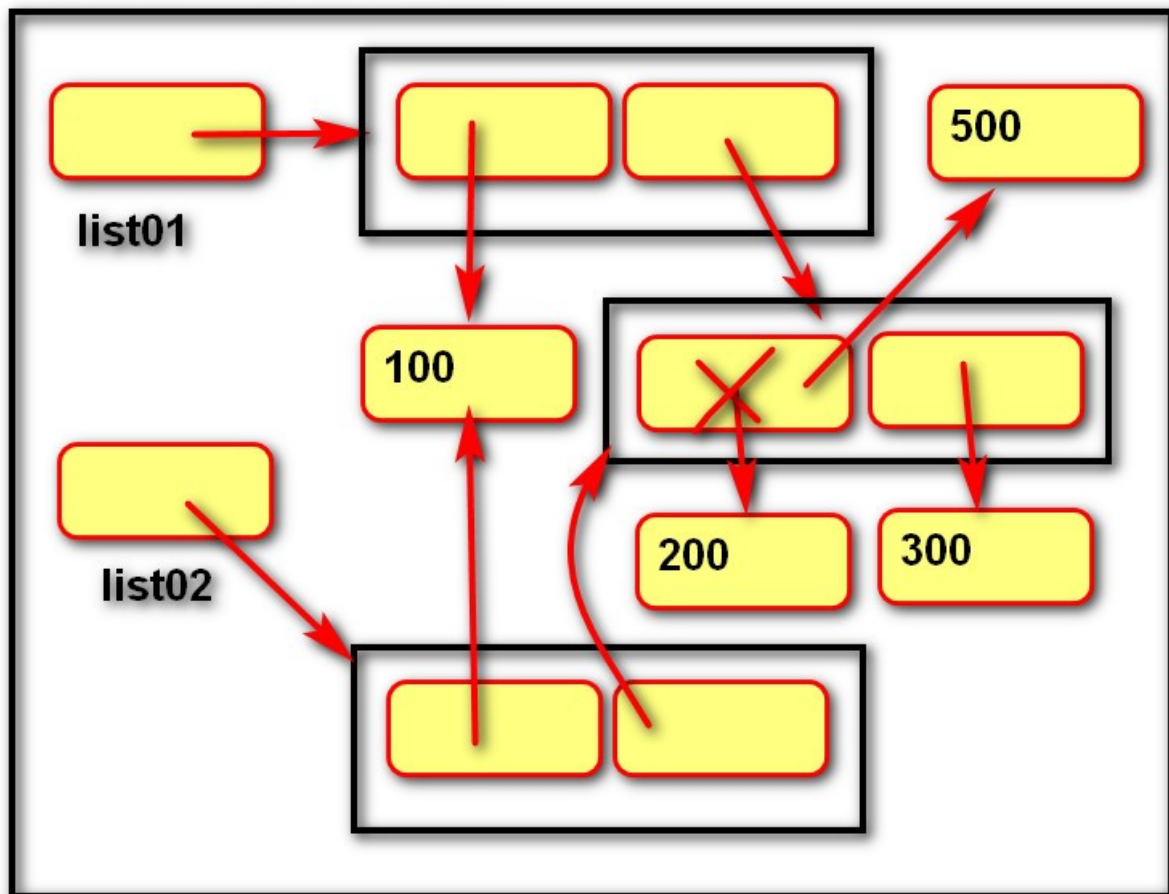
```
print(list02) # [800, 'a', 'b', 1000]
```

```
list01 = [100,[200,300]]
list02 = list01
list01[1][0] = 500
print(list02[1][0])# 500
```



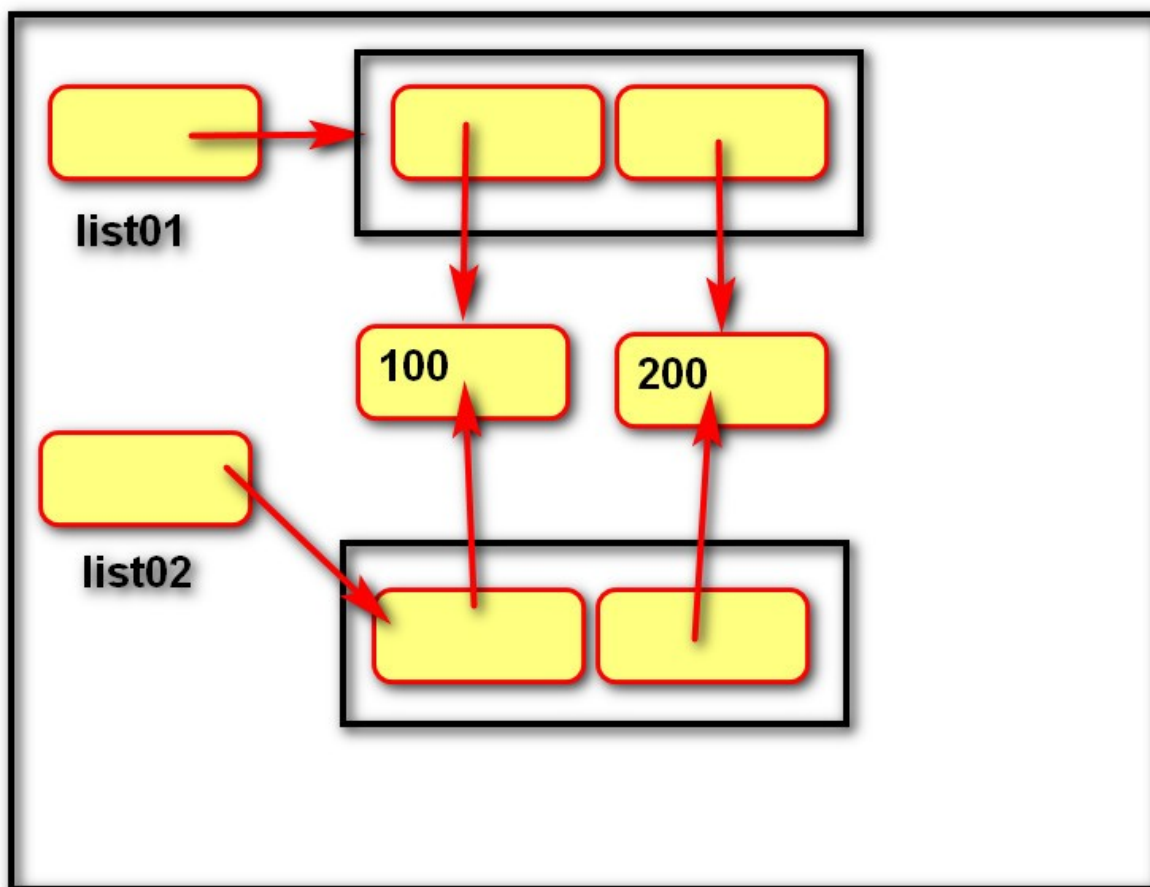
```
list01 = [100,[200,300]]
list02 = list01
list01[1][0] = 500
print(list02[1][0]) # 500
```

```
list01 = [100,[200,300]]  
list02 = list01[:]  
list01[1][0] = 500  
print(list02[1][0])# 500
```



```
list01 = [100,[200,300]]  
# list02 = list01[:]  
list02 = list01.copy()  
list01[1][0] = 500  
print(list02[1][0])# 500
```

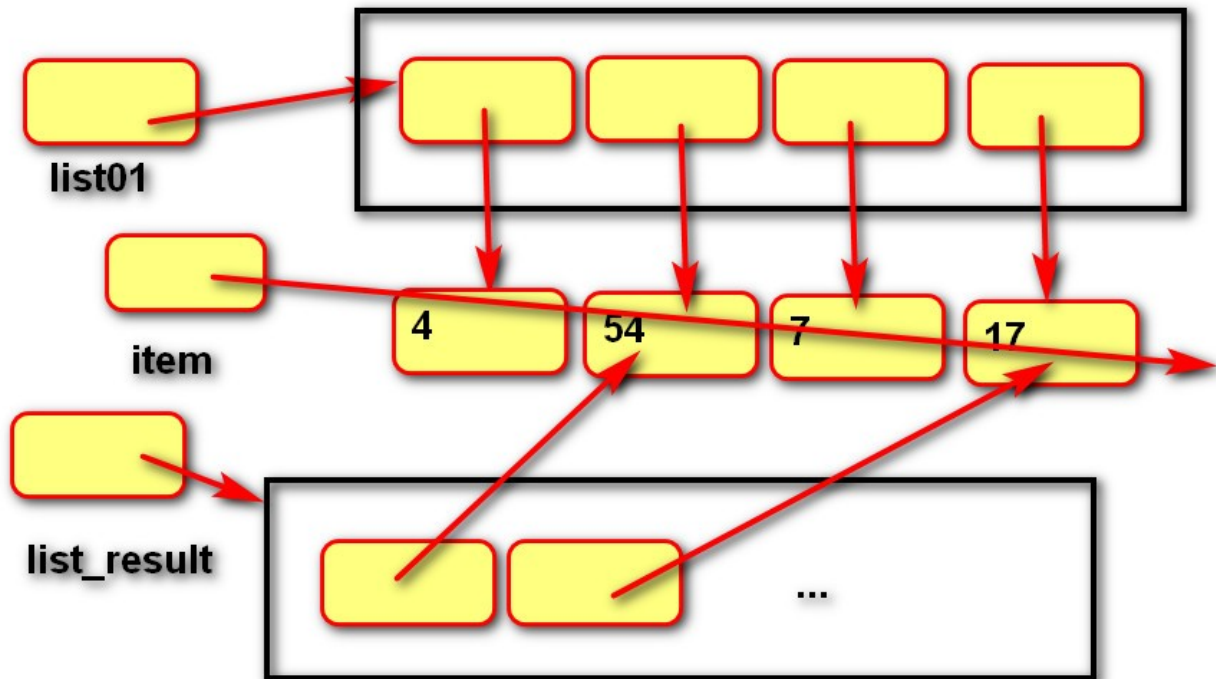
```
list01 = [100,200]
list02 = [100,200]
print(list01 == list02)#true 比较的是内容
print(list01 is list02)#false id(list01) == id(list02)
```



```
list01 = [100,200]
list02 = [100,200]
print(list01 == list02)#true 比较的是内容
print(list01 is list02)#false 比较的是地址 id(list01) == id(list02)
```



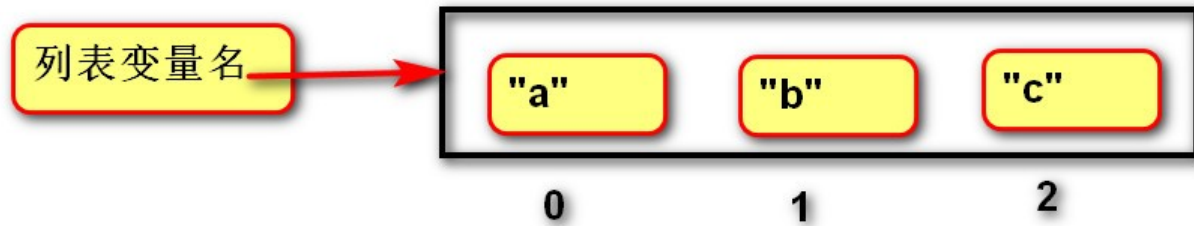
```
list01 = [4,54,5,6,67,17,8]
list_result = []
for item in list01:
    if item > 10:
        list_result.append(item)
print(list_result)
```



2. 添加元素：
  - 列表名.append(元素)
  - 列表.insert(索引, 元素)
3. 定位元素：
  - 索引、切片
4. 遍历列表：
  - 正向：
    - for 变量名 in 列表名:
    - 变量名就是元素
  - 反向：
    - for 索引名 in range(len(列表名)-1,-1,-1):
    - 列表名[索引名]就是元素



倒序获取列表中所有元素



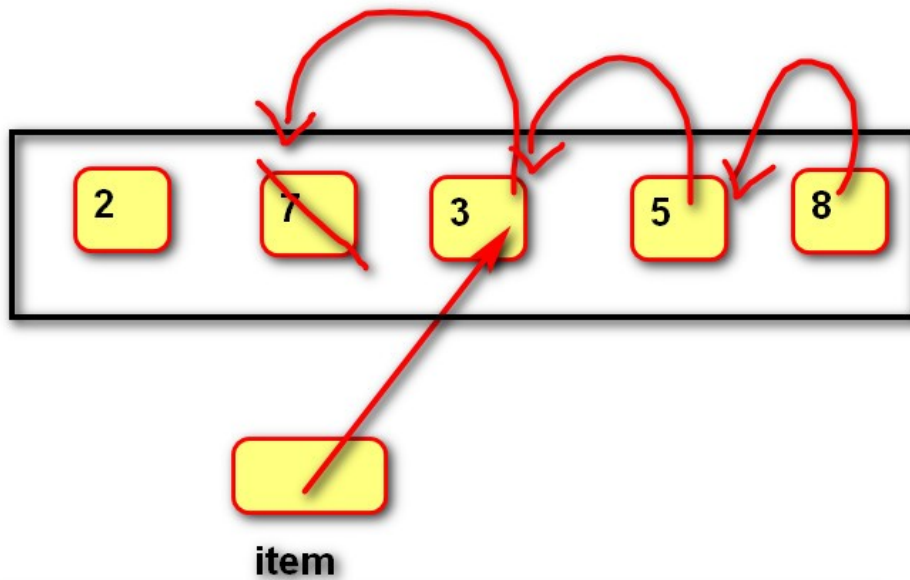
```
for i in range(len(列表变量名)-1,-1,-1):  
    列表变量名[i]
```

备注：**len(列表变量名)-1** 是最后一个索引

5. 删除元素：

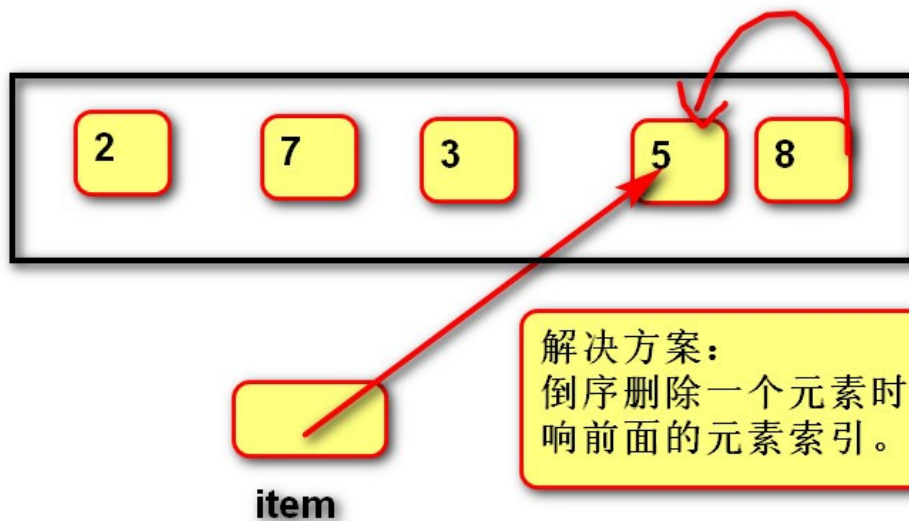
列表名.remove(元素)

del 列表名[索引或切片]



### 删除元素：

程序内部会用下一个元素覆盖上一个元素。  
如果正向判断进行删除，会跳过下一个元素。



### 解决方案：

倒序删除一个元素时，不影响前面的元素索引。

"""

列表：

适用性：存储多个数据。

练习：`exercise01.py`

`exercise02.py`

`exercise03.py`

"""

# 1. 创建列表

# 空列表

```

list01 = []
list01 = list()
# 具有默认值的列表
list02 = [1,"a",True]
# list02 = list(可迭代对象)
# str --> 列表
list02 = list("齐天大圣")
# range --> 列表
list02 = list(range(5))
# 2. 增加
# 追加
list02.append("qtx")
# 插入(索引,元素)
list02.insert(3,"lzmly")
# 3. 删除
# 根据元素删除
list02.remove(4)
# 根据索引删除
del list02[3]
# 4. 修改
list02[0] = "第一个元素"
print(list02)
# 5. 切片:定位多个元素
# 获取 (重新创建新列表)
# 前三个元素
list03 = list02[:3]
# 修改
# 修改后两个元素
list02[-2:] = ["倒数第2个元素","倒数第1个元素"]
print(list02)
# 通过切片定位2个元素,修改为3个元素
list02[2:4] = ["a","b","c"]
# 通过切片定位3个元素,修改为0个元素 [删除]
list02[2:5] = []
print(list02)
# 6. 查询
# 获取所有元素
for item in list02:
    print(item)
# 倒序获取所有
# 不建议通过切片(因为重新创建新列表,索引不建议.)
# for item in list02[::-1]:
#     print(item)
# 建议通过索引
#2 1 0

```

```
# for i in range(2,-1,-1):  
for i in range(len(list02)-1, -1, -1):  
    print(list02[i])
```

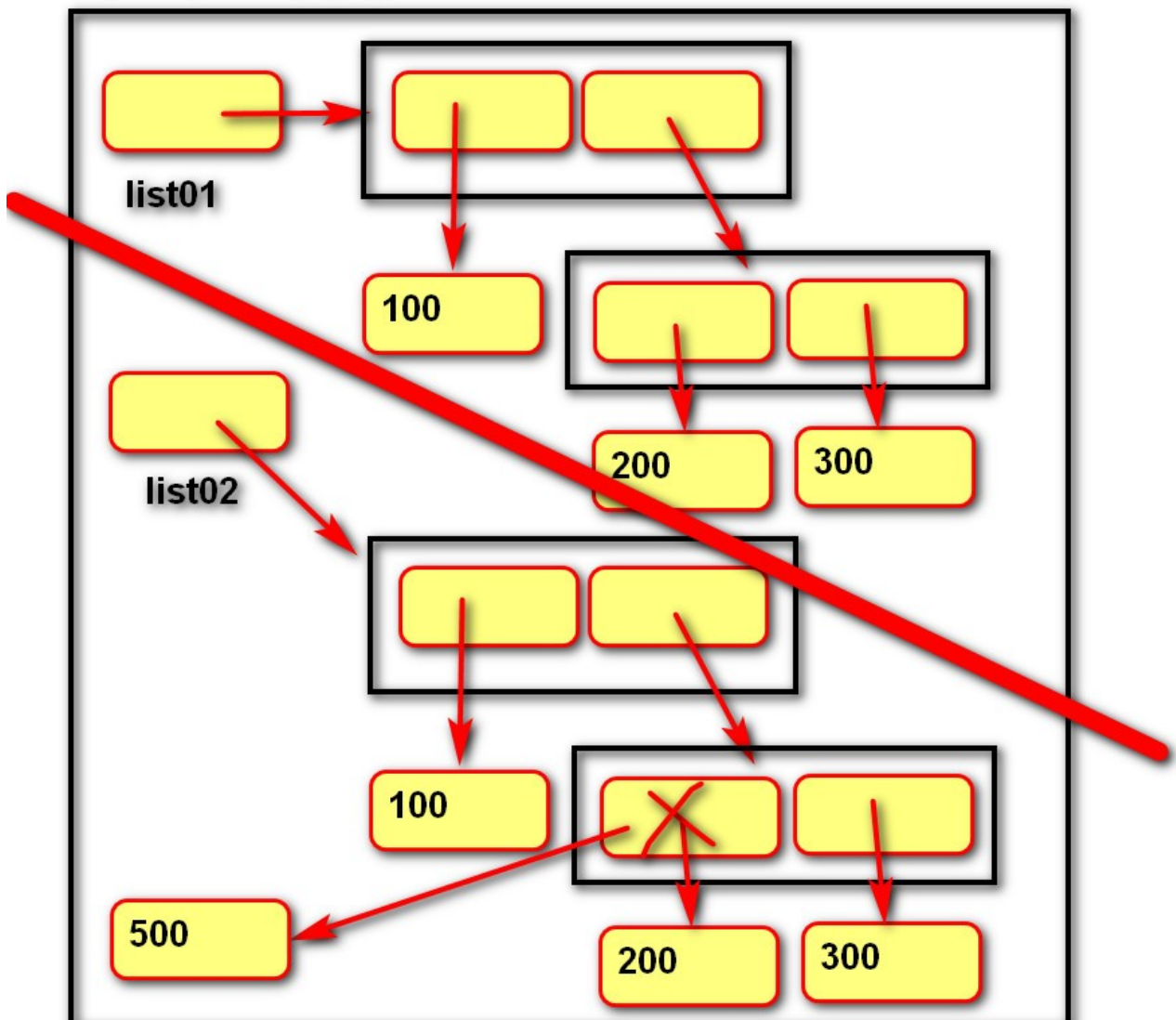
## 深拷贝和浅拷贝

浅拷贝：复制过程中,只复制一层变量,不会复制深层变量绑定的对象的复制过程。

深拷贝：复制整个依赖的变量。

```
list01 = [100,[200,300]]  
?????? # 深拷贝  
list01[1][0] = 500  
print(list02[1][0])#200
```

备注：不考虑其他优化机制



*# 准备拷贝工具*

```
import copy
list01 = [100,[200,300]]
# 深拷贝(“划清界限”:拷贝前与拷贝后的对象互不影响)
# 注意:深拷贝可能占用内存过大.
list02 = copy.deepcopy(list01)
list01[1][0] = 500
print(list02[1][0])# 200
```

## 列表 VS 字符串

1. 列表和字符串都是序列,元素之间有先后顺序关系。
2. 字符串是不可变的序列,列表是可变的序列。
3. 字符串中每个元素只能存储字符,而列表可以存储任意类型。
4. 列表和字符串都是可迭代对象。
5. 函数:

将多个字符串拼接为一个。

result = "连接符".join(列表)

*# 练习:将英文语句每个单词反转:*

```
# How are you --> you are How
# 将字符串按照空格拆分为列表
# 将列表反转
# 将列表中元素拼接为字符串
message = "How are you"
list_temp = message.split(" ")
message = " ".join(list_temp[::-1])
print(message)
```

将一个字符串拆分为多个。

列表 = “a-b-c-d”.split(“分隔符” )

"""

```
str --> list
练习:exercise08.py
```

"""

```
str01 = "张无忌_张翠山_张三丰"
list_result = str01.split("_")
print(list_result)
```

## 列表推导式

1. 定义：  
使用简易方法，将可迭代对象转换为列表。
2. 语法：  
变量 = [表达式 for 变量 in 可迭代对象]  
变量 = [表达式 for 变量 in 可迭代对象 if 条件]
3. 说明：  
如果 if 真值表达式的布尔值为 False,则可迭代对象生成的数据将被丢弃。

## 列表推导式嵌套

1. 语法：  
变量 = [表达式 for 变量 1 in 可迭代对象 1 for 变量 2 in 可迭代对象 2]
2. 传统写法：  
result = []  
for r in ["a", "b", "c"]:  
 for c in ["A", "B", "C"]:  
 result.append(r + c)
3. 推导式写法：  
result = [r + c for r in list01 for c in list02]

```
"""  
    list --> str  
    练习:exercise07.py  
"""  
list01 = ["a","b","c"]  
str01 = "+".join(list01)# a+b+c  
print(str01)  
# 案例:根据某些逻辑,拼接字符串.  
# str_result = ""  
# for item in range(10):#0 1 2 3 4 5 6 7 8 9  
#     # str_result += str(item)  
#  
#     # "" + "0" --> 产生一个新字符串对象 "0"  
#     # "0" + "1" --> 产生一个新字符串对象 "01"  
#     # "01" + "2" --> 产生一个新字符串对象 "012"  
#     # ...  
#     str_result = str_result + str(item)  
#  
# print(str_result)  
# 核心思想:通过可变对象(list)收集需要拼接的字符串,最后再转换为字符串。  
list_result = []
```

```
for item in range(10):#0 1 2 3 4 5 6 7 8 9
    list_result.append(str(item))
str_result = "".join(list_result)
print(str_result)
```