

# 派发特工任务





## 1. 探索新知

### 1.1

### 特工任务列表

#### ● 列表【list】

- 列表是Python中一种常用的数据类型，用于存储一组值
- 列表可以包含任意类型的元素，如数字、字符串、布尔值等
- 列表以方括号 [ ] 表示，元素之间使用逗号分隔

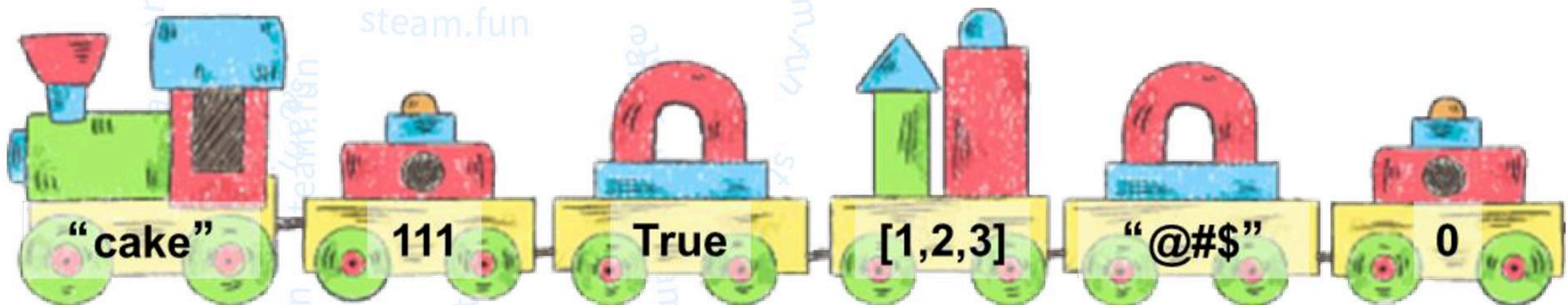
#### ● 创建一个列表

```
lis = [元素1, 元素2, 元素3, 元素4, ...]
```

我们拿名为“lis”的火车来类比



"lis"



```
lis = [ "cake", 111, True, [1,2,3], "@#\$", 0 ]
```

# 1.1

## 特工任务列表



接着，我们试着创建一个名为 name 的列表来存储特工的代号



黄昏



黎明



007



萧炎



唐三

```
name = [ "黄昏", "黎明", "007", "萧炎", "唐三" ]
```

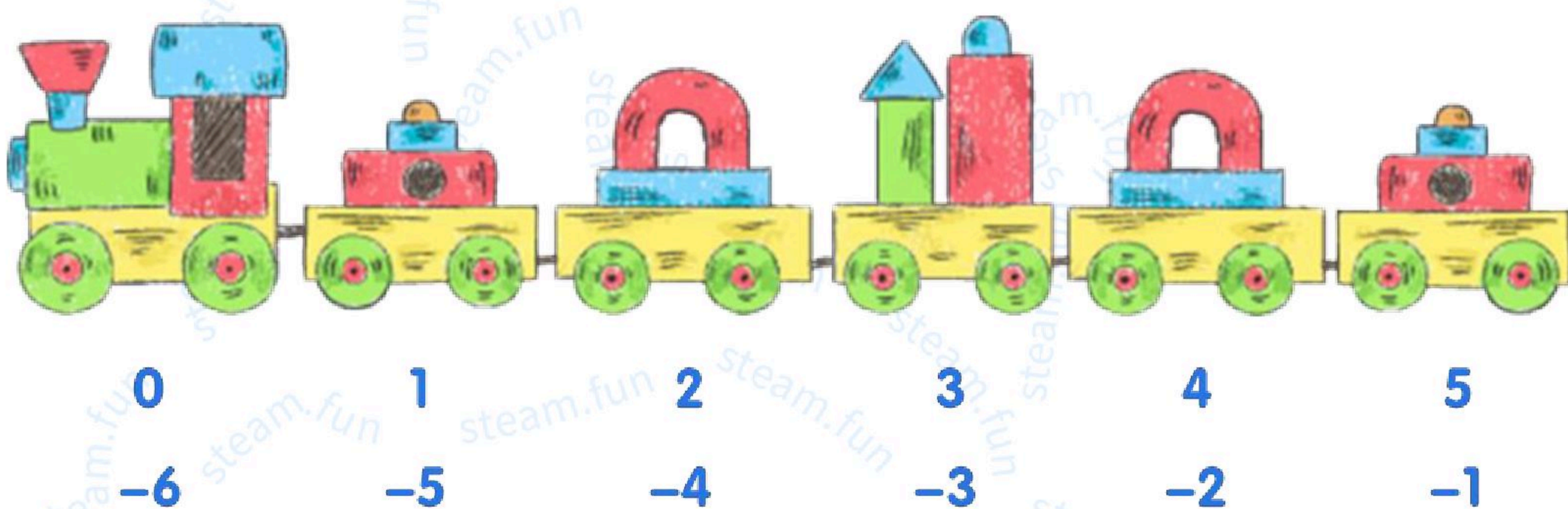
大家自己试试，创建一个名为 task 的列表，记得把各位特工的任务按顺序存储进去

- '潜入敌国获取情报'
- '采摘生长在天山的雪莲花'
- '给隔壁出差李阿姨家的喵喵加粮'
- '营救被扣押的人质'
- '破解电脑系统获取目标数据'

继续拿火车来类比，

从车头数，车厢代号从左往右为0、1、2、...；

从后往前倒着数，车厢代号从右往左为-1、-2、-3、...



### ● 列表的索引

- 列表中的每个元素都有一个索引，用于唯一标识该元素在列表中的位置
- 从左往右从 0 开始计数，从右往左从 -1 开始计数

### ● 访问列表元素

列表名[索引]

如: lis[0]、lis[1]、lis[-1]

"007" 的  
索引为2

```
name = [ "黄昏", "黎明", "007", "萧炎", "唐三" ]
        0       1       2       3       4
```



"007" 的索引为 2，我们推理得知007的任务密钥为 2，  
接下来我们试着查询对应的特工任务

- 查询007的任务

我们使用密钥（索引）定位，007对应的特工任务是 `task[2]`，将其存到一个新的变量`task_007`中

```
task_007 = task[2]
```

- 输出007的任务

```
print("007的特工任务：" + task_007)
print() # 空行（可以隔开上下文，非必要）
```

一名神秘的长官，不知从哪里拿到了特工名单，  
首先清点了名单上的特工数量，  
接着挨个查询每位特工的任务详情...

#### 控制台

```
共有特工5名
黄昏的特工任务：潜入敌国获取情报
黎明的特工任务：采摘生长在天山的雪莲花
007的特工任务：给隔壁出差李阿姨家的喵喵加粮
萧炎的特工任务：营救被扣押的人质
唐三的特工任务：破解电脑系统获取目标数据
```

# 1.3

## 神秘长官

length: 长度

### • 列表的长度

➤ 要获取列表的长度，可以使用内置函数 `len()`

```
# 获取特工列表长度  
print('共有特工'+ str(len(name)) + '名')
```

做一做

```
fruits = ["苹果", "桃子", "香蕉"]  
sports = ["足球", "篮球"]  
  
>>> len(fruits)  
_____  
  
>>> len(sports)  
_____
```

### • 列表的遍历

➤ 【遍历列表】意味着 **逐个** 访问列表中的元素。



**思考:**  
如何得到下面的输出效果?

在Python中，我们可以使用循环来遍历列表中的元素。常见的遍历方法有两种：使用for循环和使用while循环

```
控制台  
name task  
共有特工5名  
黄昏的特工任务: 潜入敌国获取情报  
黎明的特工任务: 采摘生长在天山的雪莲花  
007的特工任务: 给隔壁出差李阿姨家的喵喵加粮  
萧炎的特工任务: 营救被扣押的人质  
唐三的特工任务: 破解电脑系统获取目标数据
```

```
print ( name[0] + "的特工任务：" + task[0] )
print ( name[1] + "的特工任务：" + task[1] )
print ( name[2] + "的特工任务：" + task[2] )
print ( name[3] + "的特工任务：" + task[3] )
print ( name[4] + "的特工任务：" + task[4] )
```

**观察：**

观察代码可以发现规律，只有索引在发生变化，从0递增到4，我们可以将索引设为  $i$ ，使用for循环优化代码

我们需要遍历的是从0-4共5个值，每个特工对应一个值，有几个特工就遍历几个值

```
for i in range( len(name) ):
    print(name[i] + "的特工任务：" + task[i])
```

range( ? )      整数序列

len(name)      特工的数量

print()    # 空行（可以隔开上下文，非必要）

特工不是永远都在单打独斗，就像这次，“萧炎”和“唐三”特工就需要合作完成任务，唐三负责破解电脑系统获取目标数据，接着萧炎会根据唐三获得的信息营救被扣押的人质...



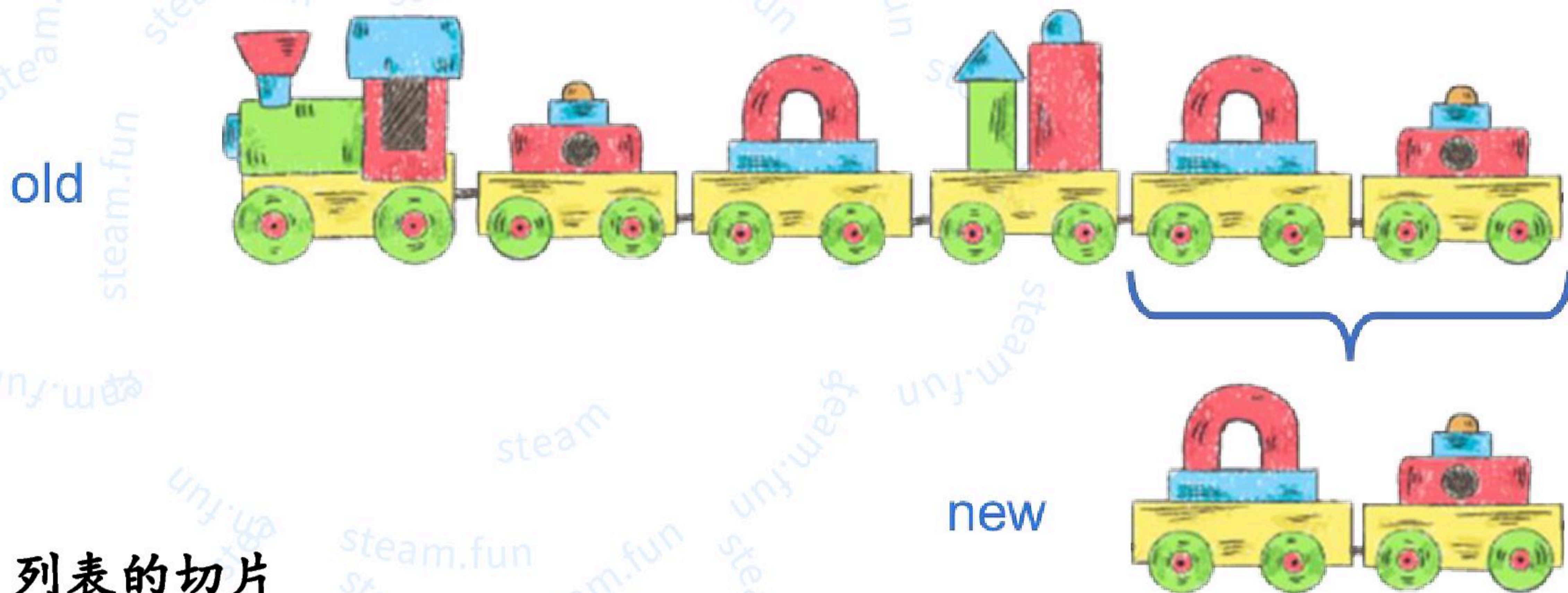
萧炎



唐三

继续拿火车来类比，

切片的过程可以理解成，从旧的火车中取出几节车厢，重新组成一列新的火车



### ● 列表的切片

- 列表切片是一种从列表中**获取子列表**的方法，它可以根据指定的索引范围来提取需要的元素
- 切片使用的是**冒号 (:)** 进行分隔

### ● 列表切片各参数

```
new_list = lis[start: end: step]
```

**start**: 切片的起始索引（包含）。

如果未提供，默认为列表的开头（索引为0）

**end**: 切片的结束索引（不包含）。

如果未提供，默认为列表的结尾（索引为列表长度）

**step**: 切片的步长（可选）。

指定切片中相邻元素的间隔。如果未提供，默认为1

## 1.4

### 派发合作任务

举例理解：

```
fruit_list = ["苹果", "香蕉", "橙子", "葡萄", "草莓", "樱桃"]  
              0       1       2       3       4       5
```

# 1. 使用切片获取列表中的部分元素

```
slice1 = fruit_list[1:4]  
print(slice1)
```

# 输出：  
-----

# 2. 使用切片获取列表的前三个元素

```
slice2 = fruit_list[:3]  
print(slice2)
```

# 输出：  
-----

# 3. 使用切片获取列表的后两个元素

```
slice3 = fruit_list[4:]  
print(slice3)
```

# 输出：  
-----

# 4. 使用切片获取列表的每隔一个元素

```
slice4 = fruit_list[::2]  
print(slice4)
```

# 输出：  
-----

# 5. 使用切片逆序反转列表

```
slice5 = fruit_list[::-1]  
print(slice5)
```

# 输出：  
-----



接下来我们试着从 `task[]` 中获取一个子列表，  
里面存储“萧炎”和“唐三”的合作任务

```
# “萧炎”和“唐三”特工合作完成任务（列表的切片）  
task_together = task[3:]  
print(task_together)
```

## 完整代码

```
name = ['黄昏', '黎明', '007', '萧炎', '唐三']
task = ['潜入敌国获取情报',
        '采摘生长在天山的雪莲花',
        '给隔壁出差李阿姨家的喵喵加粮',
        '营救被扣押的人质',
        '破解电脑系统获取目标数据']

# 007通过密钥2获取相应的任务信息(访问列表元素)
task_007 = task[2]
print("007的特工任务：" + task_007)
print()

# 获取特工列表长度
print('共有特工' + str(len(name)) + '名')
# 长官要看到每位特工的任务详情
for i in range(len(name)):
    print(name[i] + "的特工任务：" + task[i])
print()

# “萧炎”和“唐三”特工合作完成任务(列表的切片)
task_together = task[3:]
print(task_together)
```



## 2. 强化练习

1. 运行以下程序，输出的结果是？（ ）

A. 1

B. 2

C. 3

D. 4

```
list5 = ['1', '2', '4', '6', '9']  
print( list5[2] )
```

2. 在Python语言中，表达式 `[1,2]*2` 的值是？（ ）

A. [2,4]

B. 6

C. [1,2,1,2]

D. [1,2,2]

 提示：

列表的乘法是将列表中元素重复N次

列表的加法是将列表中元素合成一个列表

3. 运行以下程序，输出的结果是？（ ）

A. 1

B. 4

C. 5

D. 8

```
list3=['11', '4', 5, 1, 4]  
print( len(list3) )
```

4. 列表 `ls=['H', 'a', 'p', 'p', 'y', '!]`，下列说法错误的是？（ ）

A. `ls[:6]`的值是 `['H', 'a', 'p', 'p', 'y', '!]`

B. `ls[:-6]`的值是 `[]`

C. `ls[6]`的值是 `'!'`

D. `ls[2:-2]`的值是 `['p', 'p']`



## 2. 强化练习

5. `ls = ['富强', '民主', '文明', '和谐', '自由', '平等', '公正', '法治']`, 下列切片结果为 ['和谐', '平等'] 的选项是? ( )

A. `ls[-5:-2:-2]`

B. `ls[-5:-2:2]`

C. `ls[4:-3:1]`

D. `ls[4:-2:2]`



### 3. 术语箱

list 列表

task 任务

length 长度

together 一起

name 名字

### 4. 课后挑战

水果舞会

编写一段代码，要求如下：

已知 `fruit = ["苹果", "香蕉", "橙子", "草莓", "葡萄", "樱桃"]`

(1) 提取出索引 1-3 位置（包含1和3）的水果放入列表 `fruit1`

(2) 提取出所有偶数索引位置的水果放入列表 `fruit2`

(3) 输出 `fruit1`、`fruit2`

