特工大危机

1.探索新知

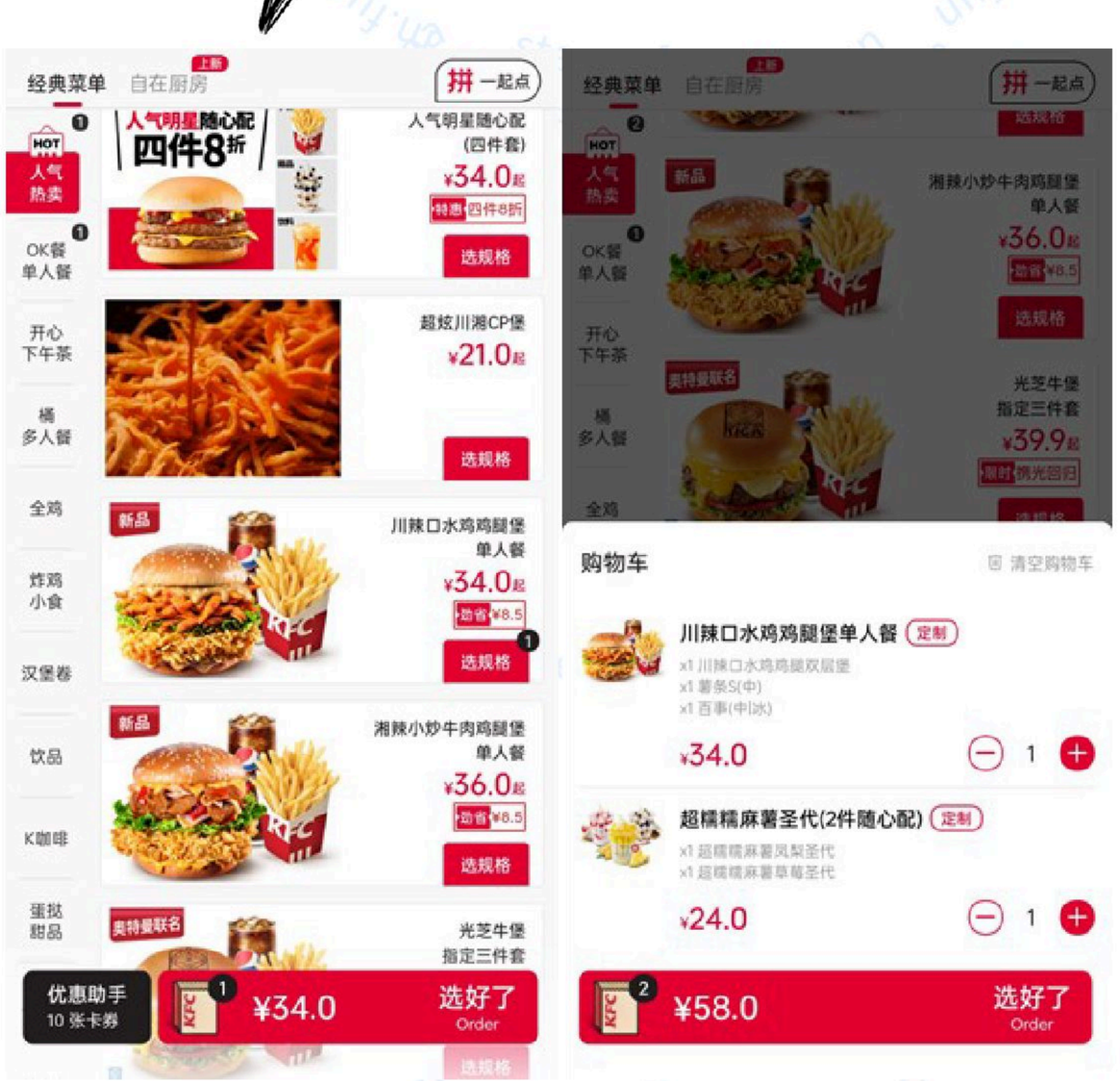
1.1 元组

列表【list】：

列表创建后，还可以随意添加、删除和修改其中的元素

元组【tuple】：

元组一旦创建，就不可改变，不能再做任何修改操作



列表就像购物车，可自由修改里面的内容

元组就好像菜单，菜品是固定的

steam.fun

1.1 元组

- **创建一个元组**

```
tupleName = (元素1, 元素2, ..., 元素n)
```

代码测试

```
fruit = ('橘子')
print( type( fruit ) )
```

控制台

```
<class 'str'>
```

```
fruit = ( '橘子', )
print( type( fruit ) )
```

控制台

```
<class 'tuple'>
```

注意： 若元组中只有一个元素，必须要在元素后面添加一个逗号

创建元组的方法合集

1. 创建空元组　　tup = ()

2. 创建元组，小括号　　tup = (1,2)

3. 创建元组，无括号　　tup = 1,2

（2020.6 电子学会二级）

下列语句，不能创建元组的是？（　　）

A. tup = ( )
B. tup = (1)
C. tup = 1,2
D. tup = (1,2)

1.1 元组

- **元组的访问**

通过索引可以获取指定元素，但不能修改指定元素

代码测试

访问 ✓

```
tupleName = ('苹果', '鸭梨', '李子', '山竹')
print(tupleName[1])   # 输出“鸭梨”
```

修改 ✗

```
tupleName = ('苹果', '鸭梨', '李子', '山竹')
tupleName[1] = '冻梨'   # 报错TypeError
```

试着分别运行两段代码，看看效果吧！

- **元组的运算**

+ 连接

测试用例：

```
tuple1 = ('苹果','鸭梨','李子','山竹')
tuple2 = ('冻梨','苹果')
tupleAll = tuple1 + tuple2
print(tupleAll)
```

输出：('苹果','鸭梨','李子','山竹','冻梨','苹果')

*** 复制**

测试用例：

```
tuple1 = ('冻梨','苹果')
tupleDouble = tuple1 * 2
print(tupleDouble)
```

输出：('冻梨','苹果','冻梨','苹果')

1.1 元组

元组的常见操作

1. len()　返回元组中元素的个数

2. tuple()　用于将列表、区间(range)等转换为元组

3. in　如果"包含"返回True，反之返回False

4. max()/min()　返回最大元素/返回最小元素

tuple()

测试用例：

```
list1 = [1,2,5,4,'独特的']
tup = tuple(list1)
print(tup)
# 输出（1,2,5,4,'独特的'）
```

in

测试用例：

```
tup = ('苹果','鸭梨','李子','山竹')
print("鸭梨" in tup)
# 输出True
```

（2020.6 电子学会二级）

下列程序的执行结果是？（　）

```
s = (1,2,3,4,5,6,7,8)
print(len(s), max(s), min(s))
```

A. 7 8 1

B. 8 8 1

C. 8 1 8

D. 7 1 8

1.2 修复特工名单

已知：

其中一份破损名单上写有“黄昏”，“黎明”，“007”，
另一份破损名单上写有“萧炎”，“唐三”，
帮忙把名单修复，合成一份吧！

1. 两份破损的名单

```
# tup_all是完整的特工名单，被撕成了tup1和tup2共两份
tup1 = ('黄昏', '黎明', '007')
tup2 = ('萧炎', '唐三')
```

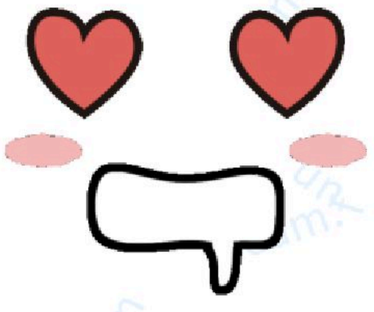
2. 修复特工名单

```
tup_all = tup1 + tup2
```

1.3 字符串

- **字符串【string】的概念**

字符串是python中最常用的数据类型。
字符串是由数字、字母、符号、空格等组成的一串字符。



理解：
把羊肉穿在一起叫做“羊肉串”，
把字符连在一起就叫做“字符串”。

1.3 字符串

- **字符串的创建**

1. 使用单引号(' ')或者双引号(" ")创建字符串

```
a = 'hello'
b = "world"
```

2. 使用三引号("' '")创建多行字符串

```
c = ''' hello
world
!!!'''
```

- **字符串的运算**

+ 连接

测试用例：

```
s = 'hello ' + 'world'
print(s)
# 输出 hello world
```

*** 复制**

测试用例：

```
s = 'hi' * 3
print(s)
# 输出 hihihi
```

- **字符串的索引**

使用索引可以访问字符串中的单个字符。
索引从0开始计数，而负数索引表示从字符串的末尾开始计数。

测试用例

```
s = '公鸡母鸡战斗机哦耶'
print(s[2])  # 输出：_____
print(s[-2])  # 输出：_____
```

想想看，会输出哪个字符呢？

1.3 字符串

- **字符串的切片**

使用切片从字符串中获取子字符串

切片操作的格式：str[start:end:step]

start：切片的起始索引（包含）。

如果未提供，默认为字符串的开头（索引为0）

end：切片的结束索引（不包含）。

如果未提供，默认为字符串的结尾（索引为字符串长度）

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

好熟悉的格式，还记得在哪里见到过类似的格式吗？

测试用例

```
s = '公鸡母鸡战斗机哦耶'
j = s[4:7]  # j = _____________
k = s[-1:]  # k = _____________
l = s[2:]   # l = _____________
m = s[4::2] # m = _____________
n = s[:4]   # n = _____________
```

1.4 字符串加密

加密规则：

1. 提取每位特工的代号，将首位放在末尾，并在最后加"ay"
2. 按照加密顺序依次输出加密后的特工代号

例如：

Jackie Chan ----> ackie ChanJay

1.4 字符串加密

1. 将密钥“ay”存储在变量 key 中，方便调用

```
key = 'ay'
```

2. 将代号挨个提取出，存储在变量name中（遍历）

```
for name in tup_all:
```

3. 每提取出一个代号存储在name中，都会将name切片处理为首字符和剩余字符两部分，分别存储在变量first和last中，如下操作：

```
first = name[0]     # name的首字符
last = name[1:]     # name的剩余字符
```

4. 根据加密规则，生成新的代号

```
new_name = last + first + key
```

5. 存储加密后的代号

由于元组不可变，所以无法向内添加元素，这里我们：
①建立一个空列表 list_all ；
②使用append 添加加密后的代号；
③再使用 tuple() 函数将列表转换为元组；
④最后输出

```
① list_all = [ ]
② list_all.append(new_name)
③ new_tup = tuple(list_all)
④ print(new_tup)
```

1.4 字符串加密

```
key = "ay"
list_all = [ ]          # ①
for name in tup_all:
    first = name[0]
    last = name[1:]
    new_name = last + first + key
    list_all.append(new_name)     # ②
new_tup = tuple(list_all)    # ③
print(new_tup)     # ④
```

完整代码

```
# 名单在争抢中被撕成两份，需要先修复特工名单
tup1 = ('黄昏','黎明','007')
tup2 = ('萧炎','唐三')
tup_all = tup1 + tup2
# 加密（字符串、切片）
key = "ay"
list_all = []
for name in tup_all:
    first = name[0]
    last = name[1:]
    new_name = last + first + key
    list_all.append(new_name)
new_tup = tuple(list_all)
print(new_tup)
```

2.强化练习

1. 下列说法正确的是？（ ）

A. 元组的元素值可以随意修改

B. 可以用del语句删除元组中的某个元素

C. tup1=(5)是一个合法的元组

D. tup1=('turtle','fish',65536)是一个合法的元组

提示：
del语句可以删除整个元组

2. 若a= (1, 2, 3)，执行下列哪个命令时会报错？（ ）

A. a[1]

B. list(a)

C. a[1] = 4

D. a * 5

3. 运行以下程序，当从键盘输入1，程序运行结果是？（ ）

A. 星期一

B. 星期+一

C. 星期二

D. 星期+二

```
str1 = '一二三四五六日'
strid = int(input('请输入1-7的数字：'))
print('星期' + str1[strid-1])
```

4. 右侧代码的运行结果是？（ ）

A. 运行会报错

B. ydti pa

C. ydtipa

D. yadhtrib yppa

```
s = 'happy birthday'
print(s[13:-15:-2])
```

2.强化练习

5. 字符串str="maixnm"包含"min"和"max"两个单词，下面哪个操作不能将"min"从中找出来？（ ）

A. str1 = str[0] + str[2] + str[4]

B. str1 = str[0:-1:2]

C. str1 = str[::2]

D. str1 = str[0:4:2]

3.术语箱

tuple	元组	first	首先，第一的
string	串	last	最后的
key	钥匙，关键		

4.课后挑战

颁奖典礼

又到了学校一年一度的颁奖典礼，P仔当选为主持人，千万不要把获奖人的名字报错哦！

编写一段代码，要求如下：

已知 tup = ("迪迦", "赛文", "佐菲", "戴拿"),

(1) 给每个获奖人名字后面加"同学"两个字；

(2) 将修改后的获奖人名字挨个加入列表；

(3) 将列表转换为元组，并输出