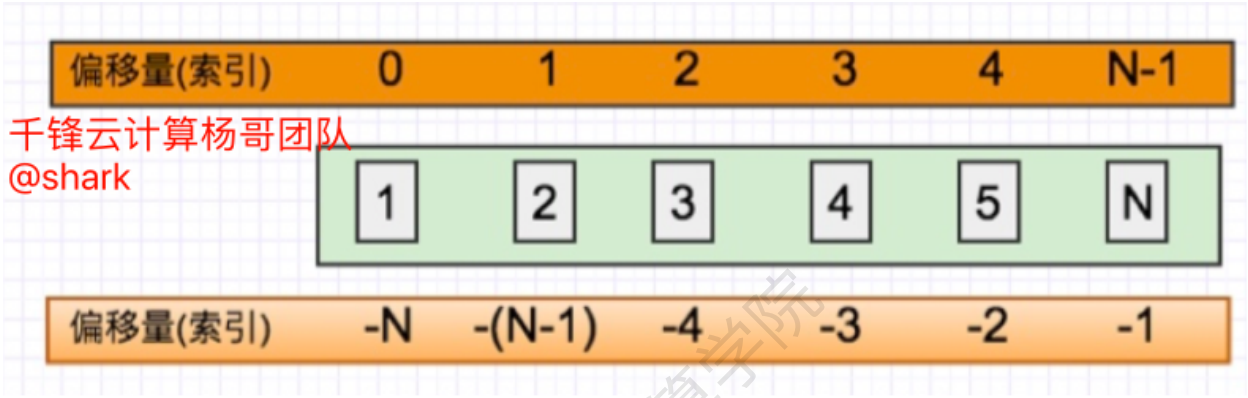
第2天两行代码下载网站数据

一、列表

1. 列表的特性介绍



> - 列表和字符串一样也是序列类型的数据
> - 列表内的元素直接用英文的逗号隔开，元素是可变的，所以列表是可变的数据类型，而字符串不是。
> - 列表的元素可以是 Python 中的任何类型的数据对象
>
> 如：字符串、列表、元组、字典、集合、函数
> - 列表中的具有相同值的元素允许出现多次
>
> [1, 2, 1, 1, 1, 1, 3, 3, 2]

2. 创建列表

```
# []  推荐，高效

li = []

l2 = ['千锋', '杨哥']


# list() 从其他类型转换
In [1]: list('hello ')
Out[1]: ['h', 'e', 'l', 'l', 'o', ' ']

# 从字符串转换 split()
In [4]: 'www.qfedu.com'.split('.')
Out[4]: ['www', 'qfedu', 'com']
```

3. 嵌套的列表

列表中可包含 python 中任何类型的元素(对象)，当然也可以包括一个或多个列表

```
li = [['one', 'two', 'three'], [1, 2, 3]]
```


4. 列表的基本操作

4.1 取值和就地修改

没有嵌套的列表取值

```
In [90]: l2 = ['insert', 'append', 'remove', 'pop',
'sort']

In [91]: l2[0]
Out[91]: 'insert'

In [92]: l2[-1]
Out[92]: 'sort'
```

4.2 嵌套的列表取值

```
In [93]: l3 = [['one', 'two', 'three'], [1, 2, 3]]

In [94]: l3[0]
Out[94]: ['one', 'two', 'three']

In [95]: ll = l3[0]

In [96]: ll[1]
Out[96]: 'two'

In [97]: l3[0][1]
Out[97]: 'two'
```


4.3 就地修改

> 可以通过索引直接修改索引对应位置的数据

```
In [113]: li
Out[113]: ['qfedu.com', 1314, '521']

In [114]: li[0] = '千锋教育'

In [115]: li
Out[115]: ['千锋教育', 1314, '521']

```

4.3 切片

同字符串的切片一样,详情参考字符串教程中的切片

几点简单示例

```
In [100]: line = 'Size: 8192 MB\n'

In [101]: line.split('\n')
Out[101]: ['Size: 8192 MB', '']

In [102]: l5 = line.split('\n')

In [103]: l5[:-1]
Out[103]: ['Size: 8192 MB']

In [104]: line.split('\n')[:-1]    # 可以直接连续操作
Out[104]: ['Size: 8192 MB']
```


4.4 必会方法

`len()`
方法是一个内置函数，可以统计序列类型的数据结构的长度。

```
In [108]: li = ['qfedu.com', 1314, '521']

In [109]: len(li)
Out[109]: 3
```

in
判断元素是否存在于列表中。

```
In [3]: li = ['qfedu.com', 1314, '521']

In [4]: '521' in li
Out[4]: True

In [5]: 1314 in li
Out[5]: True

In [6]: if 'qfedu.com' in li:
   ...:     print('ok')
   ...:
ok

In [7]:
```


append() 向列表的最后位置，添加一个元素，只接收一个参数。

```
In [7]: li.append(521)

In [8]: li
Out[8]: ['qfedu.com', 1314, '521', 521]

```

insert() 向原列表的指定位置插入一个元素，接收两个参数，第一个是索引号，第二个是要插入的元素。

```
In [9]: li.insert(0, 521)

In [10]: li
Out[10]: [521, 'qfedu.com', 1314, '521', 521]
```

`remove()` 移除列表中某个指定的元素，没有返回值，并且假如有多个相同值的元素存在，每次只会移除排在最前面的那个元素

```
In [14]: li.remove(521)

In [15]: li
Out[15]: ['qfedu.com', 1314, '521', 521, 'qf',
'yangge']
```

`pop()`

从原列表中删除一个元素，并且把这个元素返回。 接收零个或一个参数，参数是偏移量，int 类型。

```
# 删除列表中的最后一个元素
In [16]: name = li.pop()

In [17]: name
Out[17]: 'yangge'

In [18]: li
Out[18]: ['qfedu.com', 1314, '521', 521, 'qf']

 # 删除列表中第二个索引号对应的元素，并且返回这个元素，用变量名`n` 接收。
In [19]: n = li.pop(-2)

In [20]: n
Out[20]: 521

```

```
In [21]: li
Out[21]: ['qfedu.com', 1314, '521', 'qf']
```

5. 循环列表

5.1 for 循环语法

```
for 变量    in    可迭代对象:
    循环体的代码,必须缩进 4 个空格
    多行代码缩进要一致
```

> 可迭代对象 可以理解为可以被 for 循环的数据。 比如： 字符串、列表、元组、文件对象（后面讲）等。

5.2 for 循环列表中的元素

`for-list.py` 文件内容如下：

```
li = ['qfedu.com', 1314, '521', 'qf']

for item in li:
    print(item)
```

执行

```
python3 for-list.py
```

输出结果

```
qfedu.com
1314
521
qf
```

二、元组

1. 元组特性介绍

- 元组和列表一样，也是一种序列类型的数据。
- 唯一的不同是，元组是相对不可变的。

2. 高效创建元组

```
t1 = ()     # 创建 空 元素的元组
```

单一元素元组怎么搞？ 有元素的元组实际上是使用英文的逗号创建的

```
In [168]: n = (3)

In [169]: t = 3,

In [170]: type(n)

Out[170]: int

In [171]: type(t)
Out[171]: tuple
```

> 创建非空元素的元组是用`逗号`，而不是用小括号

3. 转换

`tuple()` 可以对其他序列类型的数据转换为元组。

```
In [173]: s1 = 'car'

In [174]: li = [1,2,3]

In [175]: tuple(s1)
Out[175]: ('c', 'a', 'r')

In [176]: tuple(li)
Out[176]: (1, 2, 3)

In [177]: dl = [1,2,3,['a','b']]

In [178]: tuple(dl)
Out[178]: (1, 2, 3, ['a', 'b'])
```

4. 元组的取值

> 元组也是序列类型的数据，取值和切片和列表的操作一样

```
In [33]: t1 = (1, 2, 3, ['a', 'b'])
In [34]: t1[3][0]=0
In [35]: t1
Out[35]: (1, 2, 3, [0, 'b'])
```

5. 元组的方法

> - count 统计一个元素在元组内出现的次数
> - index 返回一个元素在元组内的索引

```
In [117]: t1 = (1,'hello', ['a', 'b'])

In [118]: t1.                                  # 按 Tab 键
                count() index()          # 可以看出没有可以改变其自身的方法
```

6. 元组的相对不可变

> 元组本身是不可变的，就是元组内的元素是不可变的，一旦创建一个元组，这个元组内的元素个数和数据都是固定的了 相对不可变的意思是，元组内的元素自身是可变的数据对象，就可以通过修改这个可变元素，来间接改变元组的样子。

说下面的示例之前，先说一个内置函数 `id()`，这个函数可以返回 python 中一个对象的内存地址（id 号）

```
In [119]: id('hello')
Out[119]: 4478905344

In [120]: id(t1)
Out[120]: 4479260568

In [121]: id(t1[-1])
Out[121]: 4478661192
```

接下来就来验证元组是相对不可变的

假设我想把上个示例中的元组 `t1` 中的最后一个元素，修改为 `['a']`

```
In [122]: t1[-1]
Out[122]: ['a', 'b']

In [123]: t1[-1] = ['a']
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-123-9d326d207854> in <module>
----> 1 t1[-1] = ['a']

TypeError: 'tuple' object does not support item assignment
```

可以看到，我们不能通过元组的索引就地复制为 `['a']` 但是，元组中的最后一个元素是列表，列表中的元素是可以改变的，所以可以直接操作列表本身就行


```
In [129]: t1 = (1, 'hello', ['a', 'b'])

In [130]: t1[-1]
Out[130]: ['a', 'b']

In [131]: id(t1[-1])            # 改变前的 id
Out[131]: 4473983368

In [132]: t1[-1].pop()
Out[132]: 'b'

In [133]: t1
Out[133]: (1, 'hello', ['a'])

In [134]: id(t1[-1])           # 改变后的 id
```

```
Out[134]: 4473983368
```

7. for 循环元组

`for-tuple.py` 文件内容

```
t = ('qfedu.com', 1314, 521)

for item in t:
    print(item)
```

执行

```
python3 for-tuple.py
```

输出结果

```
qfedu.com
1314
521
```


8. 元组的优点

给我一个理由

1. 占用内存空间小
2. 元组内的值不会被意外的修改
3. 可作为字典的键
4. 函数的参数是以元组形式传递的