

# 列表推导式

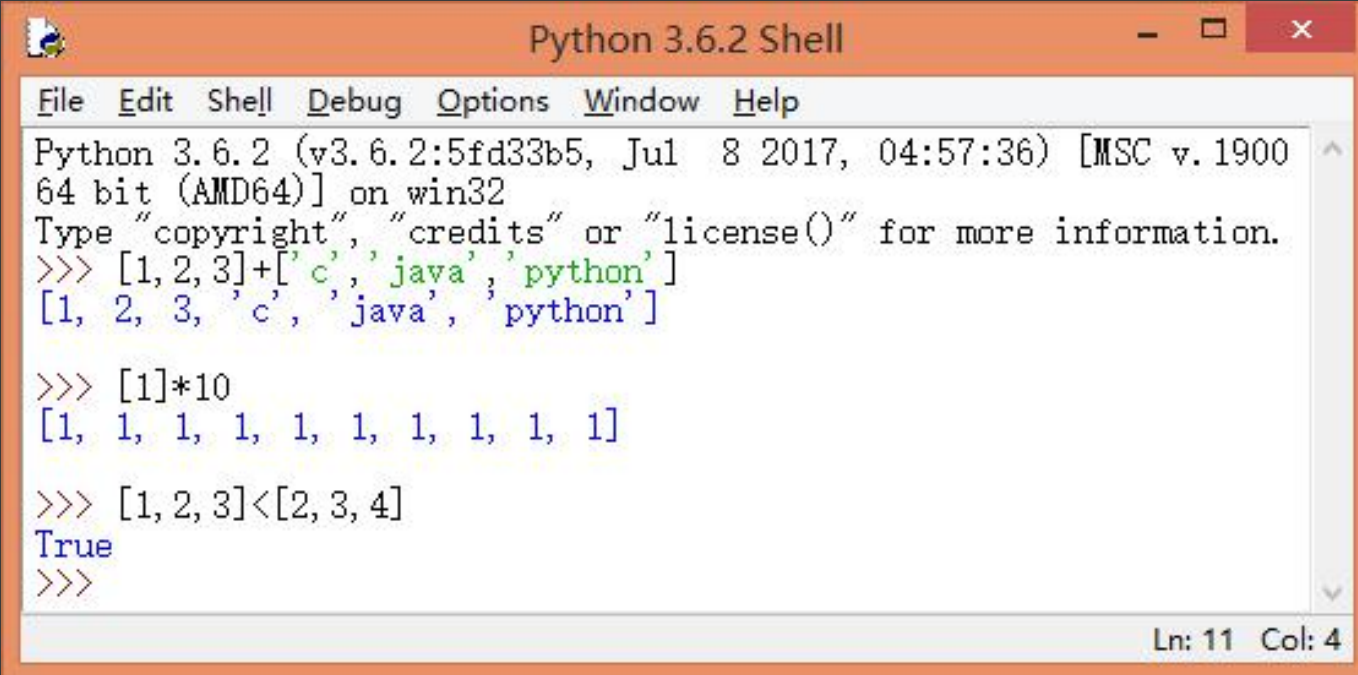
# 列表

列表是python 最常用数据类型之一：

- ◎ 由零个或多个元素组成，元素之间用逗号分开，整个列表被方括号所包裹
- ◎ 元素类型可以相同也可以不同
- ◎ 通过序号可以引用列表中的元素

# 列表

- 支持加法、乘法、比较、索引、切片操作等等



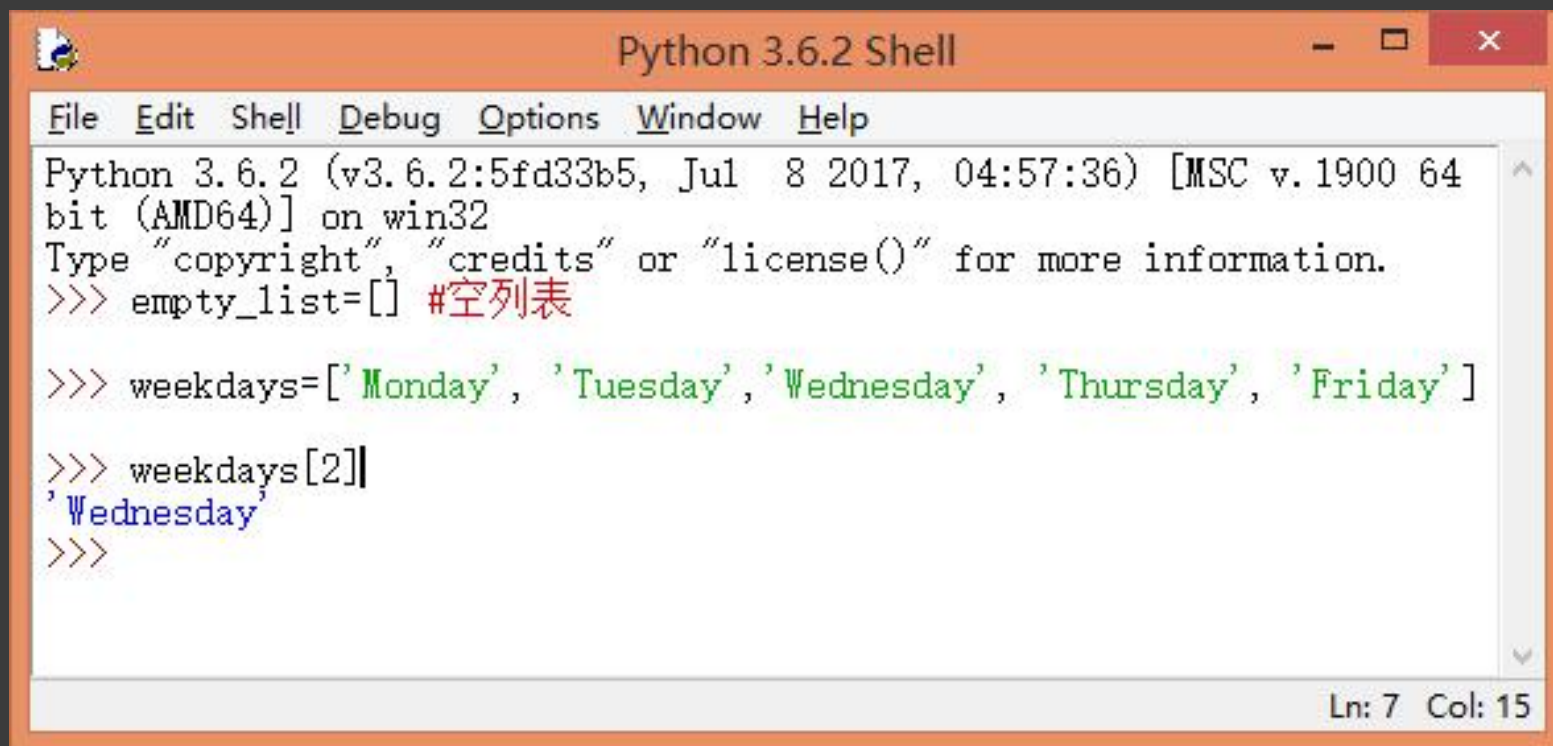
```
Python 3.6.2 Shell
File Edit Shell Debug Options Window Help
Python 3.6.2 (v3.6.2:5fd33b5, Jul 8 2017, 04:57:36) [MSC v.1900
64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> [1,2,3]+['c','java','python']
[1, 2, 3, 'c', 'java', 'python']

>>> [1]*10
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1]

>>> [1,2,3]<[2,3,4]
True
>>>
```

Ln: 11 Col: 4

# 列表运算

A screenshot of a Python 3.6.2 Shell window. The window has a title bar with the text "Python 3.6.2 Shell" and standard window controls. Below the title bar is a menu bar with "File", "Edit", "Shell", "Debug", "Options", "Window", and "Help". The main text area shows the following code and output:

```
Python 3.6.2 (v3.6.2:5fd33b5, Jul 8 2017, 04:57:36) [MSC v.1900 64
bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> empty_list=[] #空列表

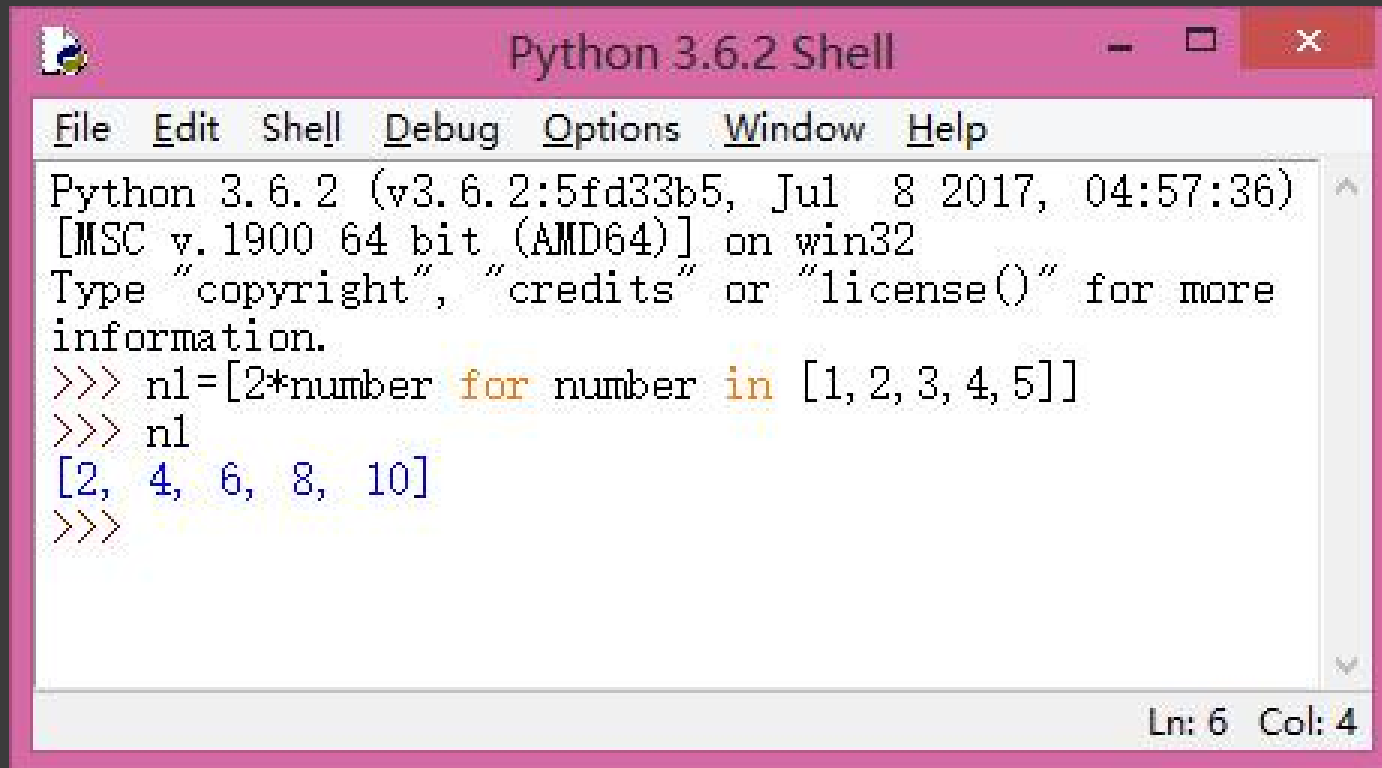
>>> weekdays=['Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday']

>>> weekdays[2]
'Wednesday'
>>>
```

The status bar at the bottom right indicates "Ln: 7 Col: 15".

# 列表推导式

- ◎ 列表推导式（又称列表解析式）提供了一种简明扼要的方法来创建列表。
- ◎ 它可以将循环和条件判断结合，从而避免语法冗长的代码，同时提高程序性能。
- ◎ 基本格式：  
[ expression for item in iterable ]

A screenshot of a Python 3.6.2 Shell window. The window has a pink title bar with the text "Python 3.6.2 Shell" and standard window controls (minimize, maximize, close). Below the title bar is a menu bar with options: File, Edit, Shell, Debug, Options, Window, and Help. The main text area shows the Python version and build information, followed by a list comprehension assignment and its output. The status bar at the bottom right indicates "Ln: 6 Col: 4".

```
Python 3.6.2 (v3.6.2:5fd33b5, Jul 8 2017, 04:57:36)
[MSC v.1900 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more
information.
>>> n1=[2*number for number in [1, 2, 3, 4, 5]]
>>> n1
[2, 4, 6, 8, 10]
>>>
```

Ln: 6 Col: 4

# 带条件的列表解析

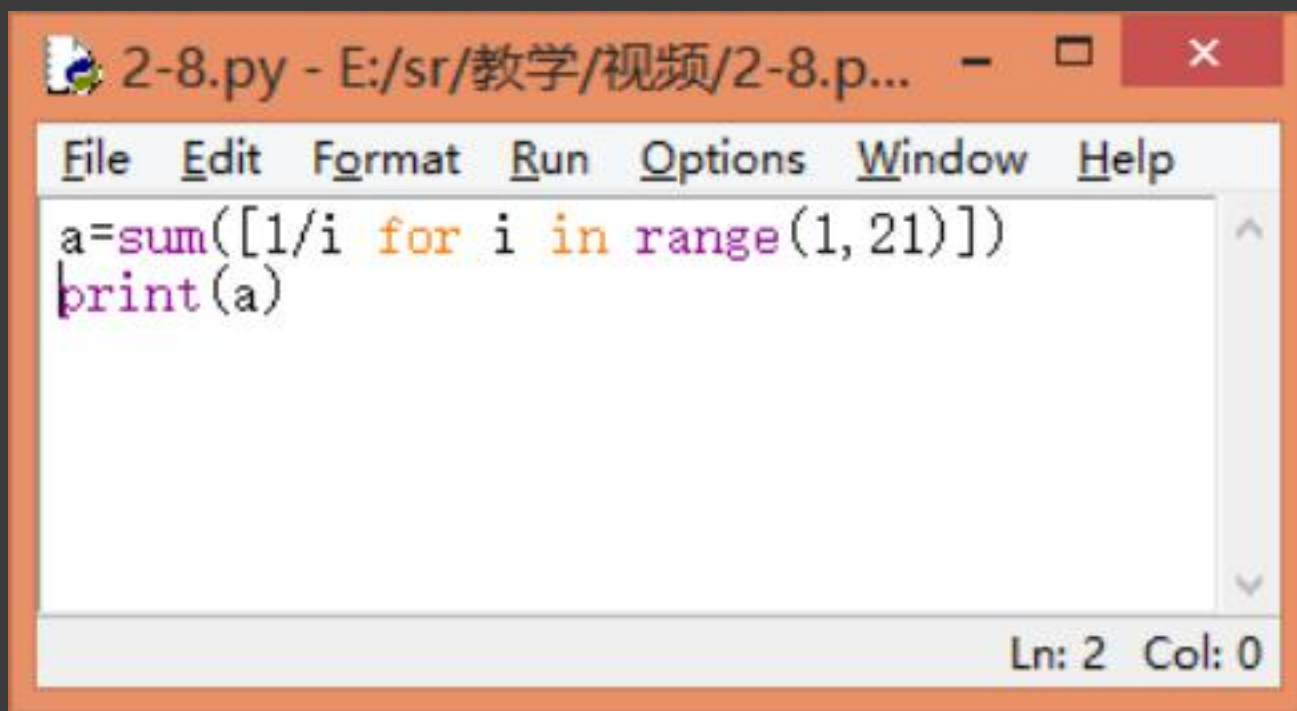
[expression for item in iterable if condition]

- ④ >>> nl=[number for number in \range(1,8) if number % 2 == 1]

- ④ >>> number\_list  
[1, 3, 5, 7]

# 列表推导式应用

求 $1+1/2+\dots+1/20$ 之和



The image shows a screenshot of a Python IDE window titled "2-8.py - E:/sr/教学/视频/2-8.p...". The window has a menu bar with "File", "Edit", "Format", "Run", "Options", "Window", and "Help". The main text area contains the following Python code:

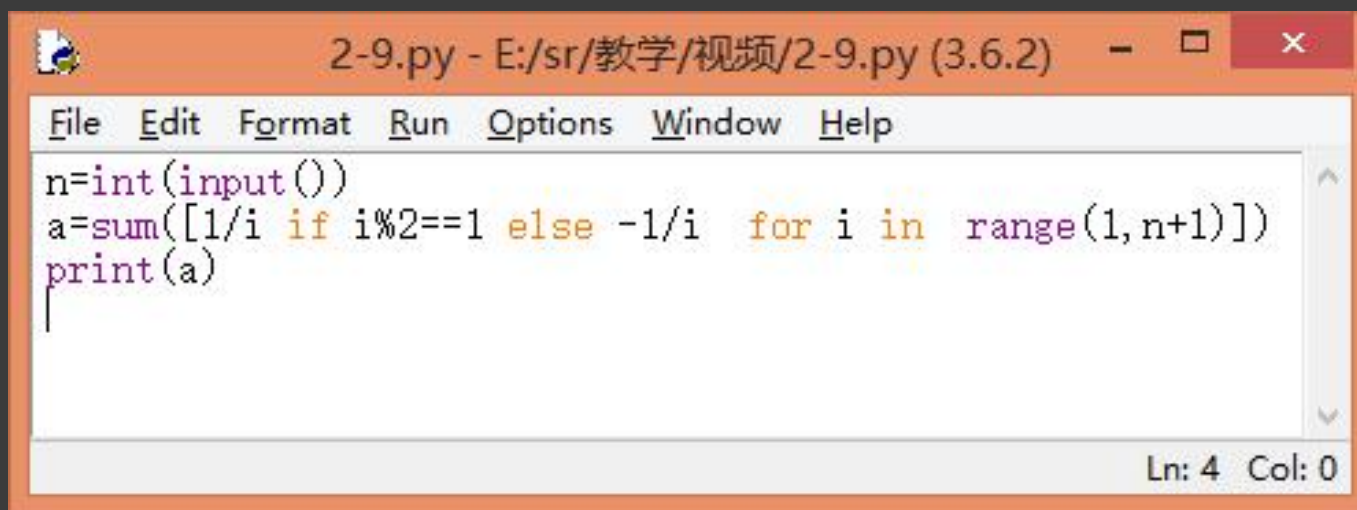
```
a=sum([1/i for i in range(1, 21)])  
print(a)
```

The status bar at the bottom right indicates "Ln: 2 Col: 0".



# 列表推导式应用

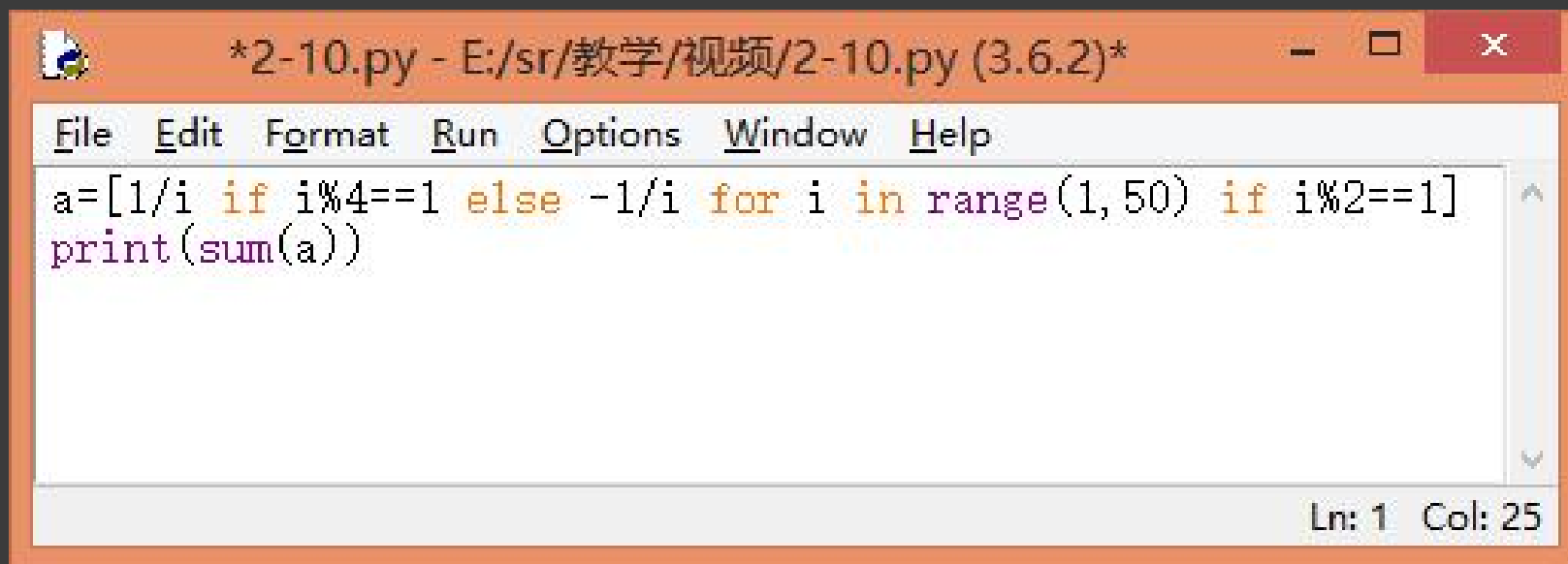
求  $1 - 1/2 + 1/3 - 1/4 + \dots$  之前n项和( $n \geq 10$ )



```
2-9.py - E:/sr/教学/视频/2-9.py (3.6.2) - □ ×
File Edit Format Run Options Window Help
n=int(input())
a=sum([1/i if i%2==1 else -1/i for i in range(1,n+1)])
print(a)
Ln: 4 Col: 0
```

# 列表推导式应用

求  $1 - 1/3 + 1/5 - 1/7 + \dots - 1/47 + 1/49$



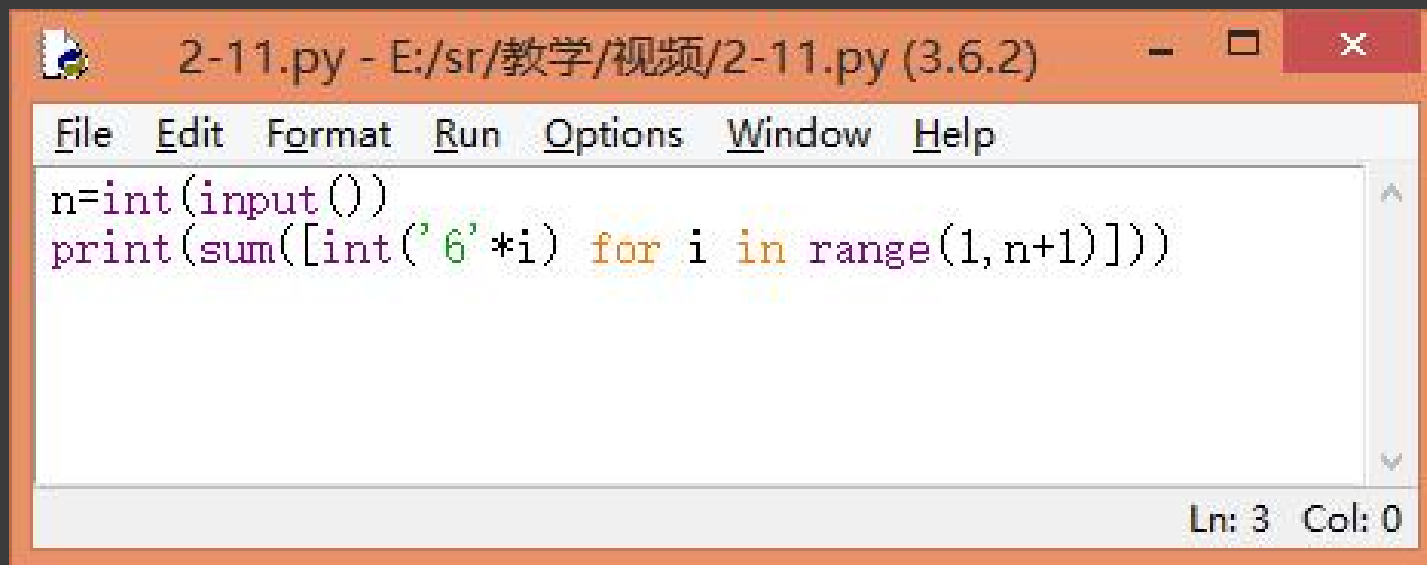
The screenshot shows a Python IDE window titled '\*2-10.py - E:/sr/教学/视频/2-10.py (3.6.2)\*'. The menu bar includes File, Edit, Format, Run, Options, Window, and Help. The code editor contains the following Python code:

```
a=[1/i if i%4==1 else -1/i for i in range(1, 50) if i%2==1]
print(sum(a))
```

The status bar at the bottom right indicates 'Ln: 1 Col: 25'.

# 列表推导式应用

求  $6+66+666+\dots+666\dots666$



The screenshot shows a Python IDE window titled "2-11.py - E:/sr/教学/视频/2-11.py (3.6.2)". The window contains the following Python code:

```
n=int(input())  
print(sum([int('6'*i) for i in range(1,n+1)]))
```

The status bar at the bottom right of the window indicates "Ln: 3 Col: 0".