1.运算符的使用

2.内置函数的使用

1. 会使用help()查看内置函数的用法

```
In [1]: help()

Welcome to Python 3.7's help utility!

If this is your first time using Python, you should definitely check out
the tutorial on the Internet at https://docs.python.org/3.7/tutorial/.

Enter the name of any module, keyword, or topic to get help on writing
Python programs and using Python modules.  To quit this help utility and
return to the interpreter, just type "quit".

To get a list of available modules, keywords, symbols, or topics, type
"modules", "keywords", "symbols", or "topics".  Each module also comes
with a one-line summary of what it does; to list the modules whose name
or summary contain a given string such as "spam", type "modules spam".


help>
```

2. 基本输入函数和输出函数的使用：input(),print()

```
In [2]: x=input('Please input:')

Please input:12345

In [3]: x
Out[3]: '12345'

In [4]: type(x)
Out[4]: str

In [5]: int(x)
Out[5]: 12345

In [6]: eval(x)
Out[6]: 12345

In [7]: x=input('Please input:')

Please input:[1,2,3,4]

In [8]: x
Out[8]: '[1,2,3,4]'

In [9]: type(x)
Out[9]: str

In [10]: eval(x)
Out[10]: [1, 2, 3, 4]

In [11]: x=input('Please input:')

Please input:'Hello World!'
```

```
In [12]: x
Out[12]: "'Hello World!'"

In [13]: eval(x)
Out[13]: 'Hello World!'

In [15]: print(1,3,5,7,sep='\t',end='\n',flush=False)
1       3       5       7

In [16]: with open('test.txt','a+') as fp:
    ...:     print('HELLO WORLD!',file=fp)
```

```
$ cat test.txt
HELLO WORLD!
```

3. 内置函数map(),filter(),range()的使用

- map()

```
In [25]: def add5(v):
    ...:     return v+5

In [26]: list(map(add5,range(10)))
Out[26]: [5, 6, 7, 8, 9, 10, 11, 12, 13, 14]

In [27]: def add(x,y):
    ...:     return x+y

In [28]: list(map(add,range(5),range(5,10)))
Out[28]: [5, 7, 9, 11, 13]

In [29]: list(map(lambda x,y:x+y,range(5),range(5,10)))
Out[29]: [5, 7, 9, 11, 13]

In [31]: def myMap(lst,value):
    ...:     return map(lambda item: item+value,lst)

In [32]: list(myMap(range(5),5))
Out[32]: [5, 6, 7, 8, 9]

In [37]: def myMap(iterable,op,value):
    ...:     if op not in '+-*/':
    ...:         return 'Error operator'
    ...:     func=lambda i:eval(repr(i)+op+repr(value))
    ...:     return map(func,iterable)

In [38]: list(myMap(range(5),'+',5))
Out[38]: [5, 6, 7, 8, 9]

In [41]: x
Out[41]: 75129847601417259179569349870 9

In [42]: list(map(int,str(x)))
Out[42]:
[7,
 5,
 1,
 2,
 9,
 8,
```

```
 4,
 7,
 6,
 0,
 1,
 4,
 1,
 7,
 2,
 5,
 9,
 1,
 7,
 9,
 5,
 6,
 9,
 3,
 4,
 9,
 8,
 7,
 0,
 9]
```

- filter()

```
In [43]: from functools import reduce

In [44]: seq=list(range(1,10))

In [45]: reduce(add,seq)
Out[45]: 45

In [47]:reduce(lambda x,y:x+y,seq)
Out[47]: 45

In [48]: seq=['foo','x41','?!','***']

In [49]: def func(x):
    ...:     return x.isalnum()

In [50]: filter(func,seq)
Out[50]: <filter at 0x7f6ce564aeb8>

In [51]: list(filter(func,seq))
Out[51]: ['foo', 'x41']

In [52]: seq
Out[52]: ['foo', 'x41', '?!', '***']

In [53]: list(filter(lambda x:x.isalnum(),seq))
Out[53]: ['foo', 'x41']
```

- range()

```
In [54]: list(range(1,10,2))
Out[54]: [1, 3, 5, 7, 9]
```

4. 求值函数eval()的使用

```
In [55]: eval(b'3+5')
Out[55]: 8
```

3.综合应用

1. 编写程序，输入任意大小的自然数，输出各位数字之和

```
# -*- coding: utf-8 -*-
"""
Spyder Editor

This is a temporary script file.
"""
from functools import reduce

def add(x,y):
return x+y

x=input("Please input a normal number:")

eval(x)

print(reduce(add,list(map(int,str(x)))))
```

2. 编写程序，输入两个集合setA和setB，分别输出两个集合的交集、并集和差集。

```
# -*- coding: utf-8 -*-
"""
Spyder Editor

This is a temporary script file.
"""
x=input("Please input setA:")
y=input("Please input setB:")

x=eval(x)
y=eval(y)

print('the union of x and y',x|y,end='\n')
print('the intersection of x and y',x&y,end='\n')
print('the minus of x and y',x-y,end='\n')
```

3. 编写程序，输入一个自然数，输入它的二进制、八进制和十六进制表示。

```
# -*- coding: utf-8 -*-
"""
Spyder Editor

This is a temporary script file.
"""
x=input("Please input a normal number:")


print(bin(int(x)),end='\n')
print(int(x),end='\n')
print(hex(int(x)),end='\n')
```