简单数据类型

字符串

```
'Hello, World!'
```

```
In : 'Hello,' + ' World!'

Out: 'Hello, World!'
```

```
In : """
...: This
...: is
...: a Test
...: """
Out: '\nThis\nis\na Test\n'


In : print('\nThis\nis\na Test\n')

This
is
a Test
```

```
In : 'abc'.upper()
Out: 'ABC'
```

endswith/startswith

```
In : 'this'.startswith('t')
Out: True

In : 'this'.endswith('t')
Out: False
```

lower/upper

```
In : 'abc'.upper()
Out: 'ABC'

In : 'ABC'.lower()
Out: 'abc'
```

split/rsplit

```
In : 'a b c'.split()
Out: ['a', 'b', 'c']

In : 'a,b,c'.split(',')
Out: ['a', 'b', 'c']

In : 'a,b,c'.split()
Out: ['a,b,c']

In : 'a b c'.rsplit(None, 1)
Out: ['a b', 'c']
```

strip/lstrip/rstrip

```
In : ' abc '.strip()
Out: 'abc'

In : ' abc '.lstrip()
Out: 'abc '

In : ' abc '.rstrip()
Out: ' abc'
```

replace

```
In : 'abc'.replace('a', 'd')
Out: 'dbc'

In : 'abc'.replace('a', 'd').replace('d', 'e')
Out: 'ebc'
```

partition/rpartition

```
In : '/a/b/c'.partition('/')
Out: ('', '/', 'a/b/c')

In : '/a/b/c'.rpartition('/')
Out: ('/a/b', '/', 'c')
```

join

```
In : ', '.join(['a', 'b', 'c'])
Out: 'a, b, c'
```

不要这样写 👇

```
s = ''

for p in parts:
    s += p
```

format

```
In : 'thi{}'.format('s')
Out]: 'this'
```

Python 2 旧写法

```
In : 'thi%s' % 's'
Out: 'this'
```

```
>>> dir('')
['__add__', '__class__', '__contains__', '__delattr__', '__dir__', '__doc__',
'__eq__', '__format__', '__ge__', '__getattribute__', '__getitem__', '__getnewargs__',
'__gt__', '__hash__', '__init__', '__init_subclass__', '__iter__', '__le__',
'__len__', '__lt__', '__mod__', '__mul__', '__ne__', '__new__', '__reduce__',
'__reduce_ex__', '__repr__', '__rmod__', '__rmul__', '__setattr__', '__sizeof__',
'__str__', '__subclasshook__', 'capitalize', 'casefold', 'center', 'count',
'encode', 'endswith', 'expandtabs', 'find', 'format', 'format_map', 'index',
'isalnum', 'isalpha', 'isdecimal', 'isdigit', 'isidentifier', 'islower',
'isnumeric', 'isprintable', 'isspace', 'istitle', 'isupper', 'join', 'ljust',
'lower', 'lstrip', 'maketrans', 'partition', 'replace', 'rfind', 'rindex',
'rjust', 'rpartition', 'rsplit', 'rstrip', 'split', 'splitlines', 'startswith',
'strip', 'swapcase', 'title', 'translate', 'upper', 'zfill']
```

转义字符串

```
'他说: "你好"'
```

```
In : '他说: \'你好\''
Out]: "他说: '你好'"
```

f字符串

```
In : name = 'Fred'

In : f'He said his name is {name}.'
Out: 'He said his name is Fred.'

In : import datetime  # 可以先不了解
In : anniversary = datetime.date(1991, 10, 12)

In : f'my anniversary is {anniversary:%A, %B %d, %Y}.'
Out: 'my anniversary is Saturday, October 12, 1991.'
```

数值 - 整型（int）

```
In : 2 + 3 加号
Out: 5

In : 2 - 3  连字号
Out: -1

In : 2 * 3 星号
Out: 6

In : 2 / 3 前斜杠
Out: 0.6666666666666666

In : 2 ** 3
Out: 8

In : 3 % 2
Out[19]: 1
```

数値 - 浮点数（float）

```
In : 1.1 + 0.2
Out: 1.3

In : 1.1 / 0.2
Out: 5.5
```

```
In : 1.1 - 0.2
Out: 0.9000000000000001

In : 1.1 * 0.2
Out: 0.22000000000000003
```

Decimal

```
In : from decimal import Decimal

In : Decimal('1.1') * Decimal('0.2')
Out: Decimal('0.22')
```

类型转换

```
In : str(1)
Out: '1'

In : int('1')
Out: 1

In : float(1)
Out: 1.0

In : int('a')
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call la
<ipython-input-16-b3c3f4515dd4> in <module>()
----> 1 int('a')

ValueError: invalid literal for int() with base 10: 'a'
```

Python 2的除法

```
>>> 2 / 3
0

>>> 3 / 2
1
```

```
>>> float(3) / 2  # 1. 让分子或者分母成为浮点数
1.5

>>> from __future__ import division  # 使用新的除法特性
>>> 3 / 2
1.5
```

数值 - 布尔值（bool）

```
1. None # null
2. False（布尔型）
3. 0
4. 0.0 # 浮点型0
5. '' # 空字符串
6. [] # 空列表
7. () # 空元组
8. {} # 空字典
```

延伸阅读

1. https://www.python.org/dev/peps/pep-3101/
2. https://www.python.org/dev/peps/pep-0498/
3. https://docs.python.org/3/library/stdtypes.html#numeric-types-int-float-complex