

- ▼ [1 模块](#)
 - ▼ [1.1 模块的加载方式](#)
 - [1.1.1 import 模块名 as 别名](#)
 - [1.1.2 import 包名.模块名 as 别名](#)
 - [1.1.3 import 模块名1,模块名2, 模块名3...](#)
 - [1.1.4 from 模块名 import 函数名 as 别名](#)
 - [1.2 使用自定义模块](#)

```
In [1]: #设置全部行输出
from IPython.core.interactiveshell import InteractiveShell
InteractiveShell.ast_node_interactivity = "all"

import matplotlib as mpl
import matplotlib.pyplot as plt
# 解决坐标轴刻度负号乱码
plt.rcParams['axes.unicode_minus'] = False

# 解决中文乱码问题
plt.rcParams['font.sans-serif'] = ['Simhei']
```

1 模块

在Python中有一个概念叫做模块（module）。

说的通俗点：模块就好比是工具包，要想使用这个工具包中的工具(就好比函数)，就需要导入这个模块

1.1 模块的加载方式

比如我们想对数字开平方根，就需要用到sqrt函数，如果我们直接使用就会报错：

```
In [2]: sqrt(4)
```

```
-----
NameError                                Traceback (most recent call last)
<ipython-input-2-317e033d29d5> in <module>
----> 1 sqrt(4)

NameError: name 'sqrt' is not defined
```

引用sqrt函数需要先加载math模块：

```
In [3]: import math
        math.sqrt(4)
```

```
Out[3]: 2.0
```

为什么sqrt(4)前面要加"math."呢？这是因为如果不指名哪个模块，解释器并不知道你调用的是哪个sqrt函数。

另外，如果同时加载多个模块，不同的模块可能存在相同的sqrt函数，不指明模块名，可能会加载错函数。

```
In [10]: print(dir(math))
```

```
['__doc__', '__loader__', '__name__', '__package__', '__spec__', 'acos', 'acosh', 'asin', 'asinh', 'atan', 'atan2', 'atanh', 'ceil', 'copysign', 'cos', 'cosh', 'degrees', 'e', 'erf', 'erfc', 'exp', 'expm1', 'fabs', 'factorial', 'floor', 'fmod', 'frexp', 'fsum', 'gamma', 'gcd', 'hypot', 'inf', 'isclose', 'isfinite', 'isinf', 'isnan', 'ldexp', 'lgamma', 'log', 'log10', 'log1p', 'log2', 'modf', 'nan', 'pi', 'pow', 'radians', 'remainder', 'sin', 'sinh', 'sqrt', 'tan', 'tanh', 'tau', 'trunc']
```

1.1.1 import 模块名 as 别名

有些模块名字比较长，想用别名作简化别名，可以在模块名后面加as+别名。

```
In [6]: import pandas as pd
```

```
pandas.DataFrame([1, 2, 3])
```

```
-----
NameError                                Traceback (most recent call last)
<ipython-input-6-38f444dd8b83> in <module>
      1 import pandas as pd
      2
----> 3 pandas.DataFrame([1, 2, 3])

NameError: name 'pandas' is not defined
```

```
In [7]: pd.DataFrame([1, 2, 3])
```

```
Out[7]:
```

	0
0	1
1	2
2	3

但是需要注意的是:一旦你定义了别名，就一定要使用该别名，如果继续使用原模块名就会报错:

```
In [2]: pandas.DataFrame([1, 2, 3])
```

```
-----
NameError                                Traceback (most recent call last)
<ipython-input-2-ea8fad1d1839> in <module>()
----> 1 pandas.DataFrame([1, 2, 3])

NameError: name 'pandas' is not defined
```

1.1.2 import 包名.模块名 as 别名

对于有些在一些工具包的模块，想引用的话需要在模块名前加上工具包的名字:

```
In [8]: import matplotlib
import matplotlib.pyplot
```

```
In [9]: type(matplotlib)
type(matplotlib.pyplot)
type(matplotlib.pyplot.plot)
```

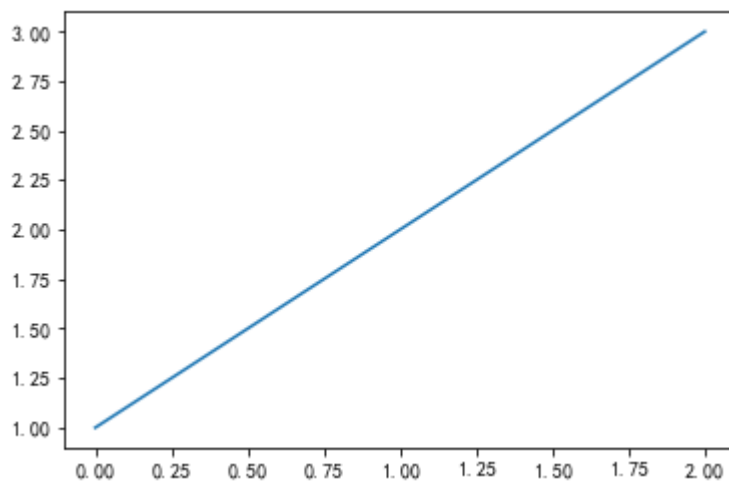
Out[9]: module

Out[9]: module

Out[9]: function

```
In [10]: import matplotlib.pyplot as plt

plt.plot([1, 2, 3]);
```



1.1.3 import 模块名1,模块名2, 模块名3...

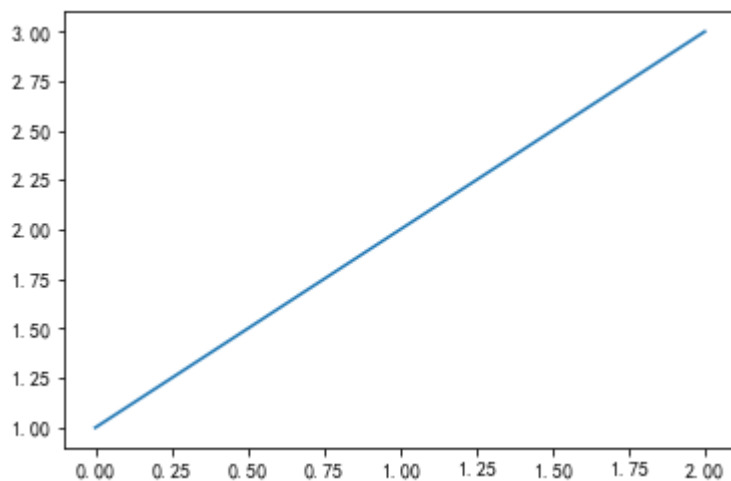
```
In [1]: import pandas, numpy, matplotlib
```

```
In [79]: import pandas as pd, numpy as np, matplotlib as mp
```

1.1.4 from 模块名 import 函数名 as 别名

from...import...的from语句让你从模块中导入一个指定的部分到当前命名空间中
import后面不单可以指定函数，还可以指定变量和类。

```
In [11]: from matplotlib.pyplot import plot as pl
pl([1, 2, 3]);
```



当然，如果我们想要加载模块里面的所有函数可以直接用星号“*”代替用户名。

```
In [1]: from math import*
sin(pi/2)
```

1.2 使用自定义模块

在Python中，每个Python文件都可以作为一个模块，模块的名字就是文件的名字。

比如有这样一个文件my_module.py，在my_module.py中定义了一个类Cal_fun

```
class Cal_fun:
    "这个类专门用来定义常见的二元算术运算"
    num_of_fun=3
    def add(self, x, y):
        return x+y
    def dedu(self, x, y):
        return x-y
    def mul(self, x, y):
        return x*y
    def div(self, x, y):
        return x/y
```

将my_module.py放在搜索模块路径中。

- sys.path是python的搜索模块的路径集，是一个list

```
In [12]: import sys
print(sys.path)

['C:\\Users\\CDA\\54期Python编程基础课件\\54期Python编程基础课件\\代码', 'C:\\ProgramData\\Anaconda3\\python37.zip', 'C:\\ProgramData\\Anaconda3\\DLLs', 'C:\\ProgramData\\Anaconda3\\lib', 'C:\\ProgramData\\Anaconda3', '', 'C:\\ProgramData\\Anaconda3\\lib\\site-packages', 'C:\\ProgramData\\Anaconda3\\lib\\site-packages\\win32', 'C:\\ProgramData\\Anaconda3\\lib\\site-packages\\win32\\lib', 'C:\\ProgramData\\Anaconda3\\lib\\site-packages\\Pythonwin', 'C:\\ProgramData\\Anaconda3\\lib\\site-packages\\IPython\\extensions', 'C:\\Users\\CDA\\.ipython']
```

这种模块名和其他模块、包名重合，将test.py改名就可以。

```
In [13]: %pwd
```

```
Out[13]: 'C:\\Users\\CDA\\54期Python编程基础课件\\54期Python编程基础课件\\代码'
```

```
In [14]: from my_module import Cal_fun
```

```
In [16]: a=Cal_fun()

a.add(100, 50)

a.dedu(100, 50)

a.mul(2, 3)
```

```
Out[16]: 150
```

```
Out[16]: 50
```

```
Out[16]: 6
```