

## 第二回：四则运算

1交互式运行窗口：灵活直观便于调试，但不易保持难以重复执行

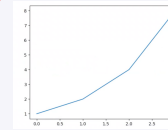
类似于计算器

举例

```
5+2  
3-4
```

```
>>> 5+2  
7  
>>> 3-4  
-1
```

```
import matplotlib.pyplot as p  
p.plot([1, 2, 4, 8]);p.show()
```



导入matplotlib库，画图

计算有关的语法

+

-

负数先写减号

\* (8)

/ (?)

除法运算的处理

p2整数与整数相除依然是整数，小数部分舍弃

p3结论精确

整除运算符//

小于真实结果的最大整数

$7//2=3-7//2=4$

幂运算\*\* 优先于乘除

$2^{**}3$

$2^{**} (1/2)$  1/2加括号

模运算% (求余数)

$7\%2$  1

运算优先级

先幂再乘除再加减

用 (一律使用) 圆括号改变“运算”，中括号大括号有其他用途

注意使用英文 (半角) 输入

未来会讲解p2p2如何快速转换

2源代码文件：无法直接看到每一步的结果，调试比较麻烦。方便保存，随时运行

保存

1.shell工具栏点击file-new file

2.copy

3.Ctrl s

按钮

run

或者F5

Q：为什么没结果

A：代码里，没有写让我把结果告诉你

```
print(2 * 3)  
print(2 ** 3)  
print(7 * 2)  
print(4 / 3 * 3.14 * 5 ** 3)  
print((((3 + 5) * 2 - 7) * 6 + 2) * 8)
```

单词全小写，括号用半角

交互式窗口自动解释每一个语句结果，不用写print

print函数

```
6  
8  
1  
523.3333333333334  
448
```

程序语句需要print

shell处

p2的print不需要括号