python中eval()与int()的区别  
1,针对数字，eval()与int() 里面数字的字符串，能否返回相同的结果(9999)。  
print eval('9999')  
print int('9999')

**int中只能是整数, int(5.5)会报错**

eval可以是小数

eval去掉引号，str加上引号，两者相对应  
2,针对字符串的表达式，eval能接受字符串，并把"9999+1"作为表达式求值，而int()调用失败，由于字符串参数不能代表整数的字符串，有非法的文字"+"  
print eval('9999+1')   //返回10000  
print int('9999+1')     //返回ValueError: invalid literal for int() with base 10: '9999+1'

**pip安装和conda安装的不同**

1.pip是从PyPI上直接下载，他仅仅是简单地下载而已，不能包含某些包必要的依赖文件。

2.Conda没有语言限制，可以构建和管理任何语言的任何类型的软件，这其中也包括Python。conda旨在管理任何软件堆栈中的包和依赖关系，其安装过程中，会提示哪些包与当前所要安装的包之间有依赖关系，并且会自动安装和更新，用起来会更靠谱。Conda从自己的知识库出发，通常是与Conda社区共同的图书馆的便利构建。Conda的存储库有一个用于安装的BaseMax包的版本，所以它成功了。

3.这并不是说Pip在这个例子中比Conda“更坏”，因为你可以很容易地下载包并在本地用PIP安装它。这个特定的图书馆刚刚选择不向PyPI添加版本。

4. python和C语言一样，在函数内未声名局部变量，则调用外部的全局变量（即指针），若函数内真实创建了局部变量，则不会运算外部全局变量,即使与外部全局变量同名

a=50

def fun():

global a #调用全局变量a

5. a^b (or)是位运算中异或的意思，相同为0不同为1  
a&b (and)是且运算，也是位运算，相同为1不同为0

6.shift+alt+F VSCODE中自动调整空格和间距

7.python的import（包括from…import…）在运行前会自己运行一边，因此若module内有class外的print之类，会打印出来。