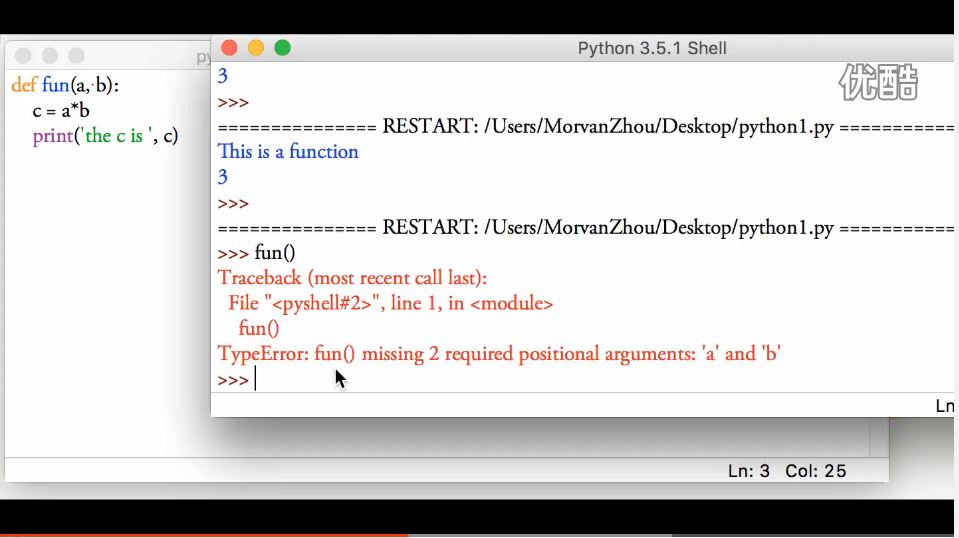
**Python基础 11 函数 参数**

大部分操作与C没有区别，就是创建函数所用的公式不同。

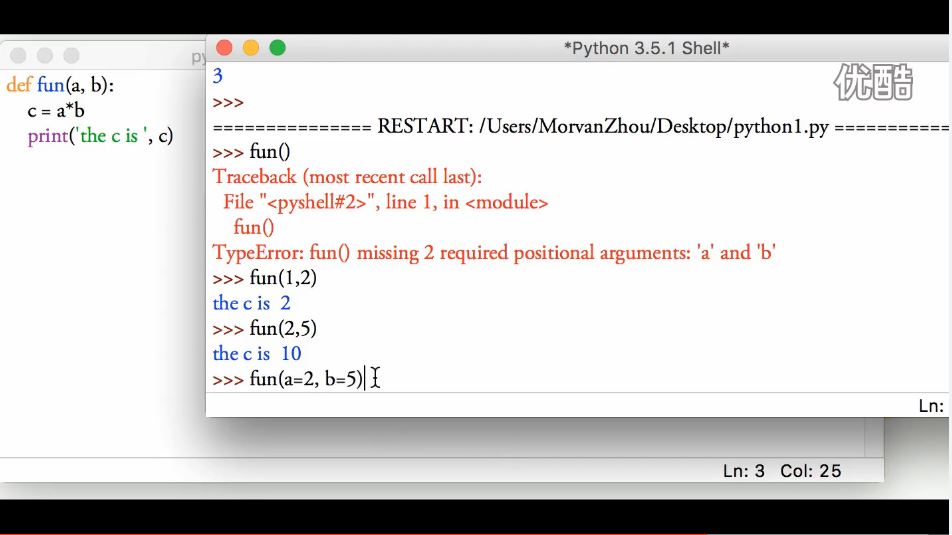
def fun(a,b);

c=a\*b

1. 调用函数时，若不带参数或者参数不完整，则会报错。

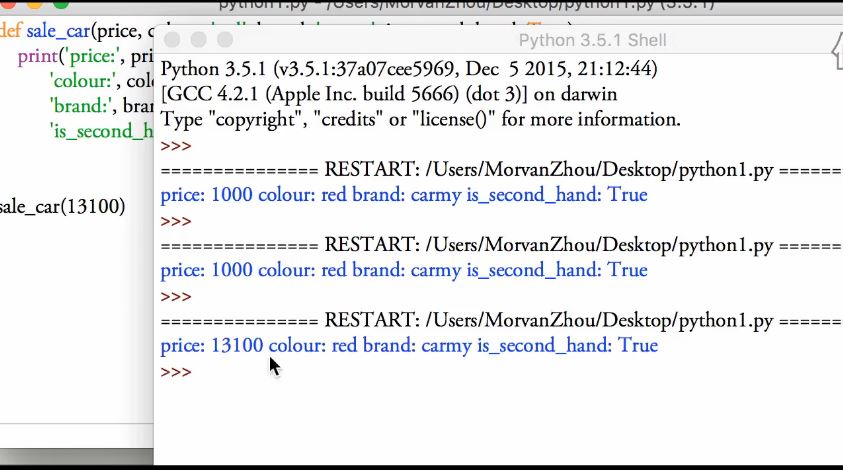
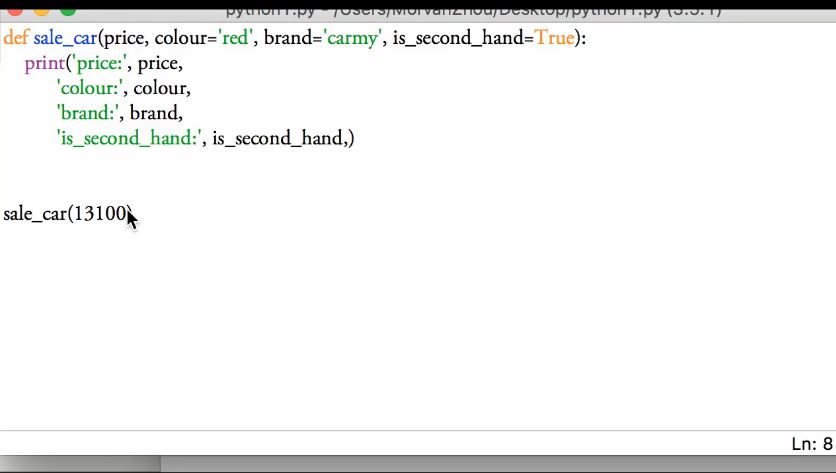


1. 如果忘记形参变量，可以直接在括号里重新命名形参。【如下】



**Python基础 12 函数设置默认参数**

直接在构造函数时，给形参赋值。

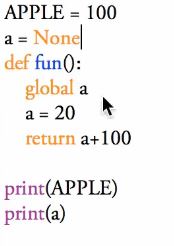


* 参数赋默认值时，必须从右往左，顺序不可乱，否则报错。

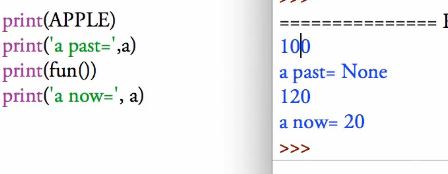
56

**Python基础13 全局变量和局部变量**

* 大部分与C规则一样
* 如果一定要在构造函数内部定义全局变量，如下：



运行结果如下



**Python基础 14 模块安装【Mac和Linux，windows直接找exe安装】**

模块安装语句：

10

Mac用户必用

Python模块安装工具，3.0及其以上版本适用

安装模块名称

模块卸载语句：

11

是否卸载（yes/no）？

12

成功卸载：

13

选择安装某版本模块：

14

版本号

升级版本：

15

升级

**Python基础 15文件的读写（1）:文件的读写**

例：

Text=’This is my first test’

my\_file=open(‘my file.txt’,’w’) --------’w’读写 ’r’只读

my\_file.write(text)-------编写文件

my\_file.close()---------关闭文件，勿忘

**Python基础 16文件的读写（2）：追加文件内容**

append\_text=’\nThis is append file.’

my\_file=open(‘my file.txt’,’a’) --------’a’ append

my\_file.write(append\_text)-------编写文件

my\_file.close()---------关闭文件，勿忘

**Python基础 17文件的读写（3）：只读文件**

file=open(‘’my file.txt,’r’)-------存储文件

content=file.read()------读取文件内容

print(content)------打印内容

**Python基础 18 class类**

class Caculation:

name=’Good caculator’

price=18

def add(self,x,y):

print(self.name)

result=x+y

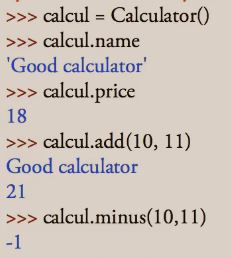
print(result)

def add(self,x,y):

result=x-y

print(result)

运行结果：



**Python基础 19 类init功能**

def\_\_init\_\_(self,name,price,hight,width,weight):

self.name=name

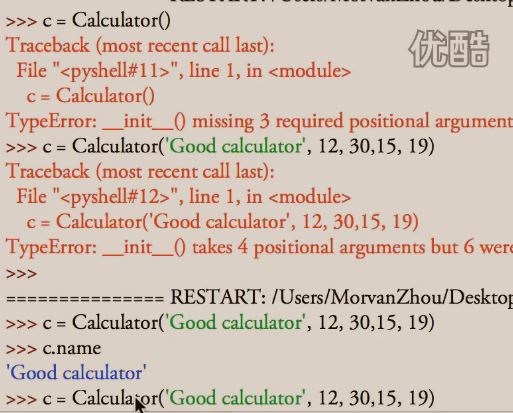
self.price=price

self.h=hight

self.wi=width

self.we=weight

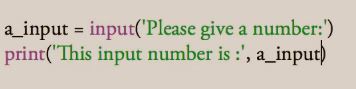
运行结果：



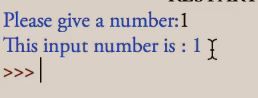
* 定义时，参数必须按顺序赋值

**Python基础 20 input输入**

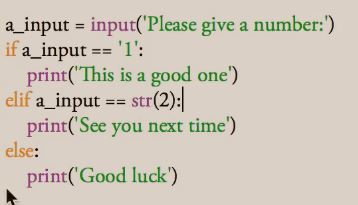
程序1内容：



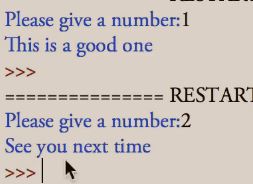
输出结果：



程序2内容:



输出内容：



* Input-------返回的是string