模块代码编写基础：

模块的使用：

方法一：

导入模块：import module

使用模块属性：module.name

优势：此种方法用在一个父模块要导入两个子模块，而两个子模块里面的属性（如变量）会有同名的情况下使用。如modulea.name1 module2.name2 这样就可以区别出调用的是哪一个。

缺点：使用此种方法相当于是将模块内全部的东西都导入了，比较麻烦

方法二：

具体导入模块中的某个值：

from module import name1,name2,name3…….等

使用模块属性：name1 = xxxx

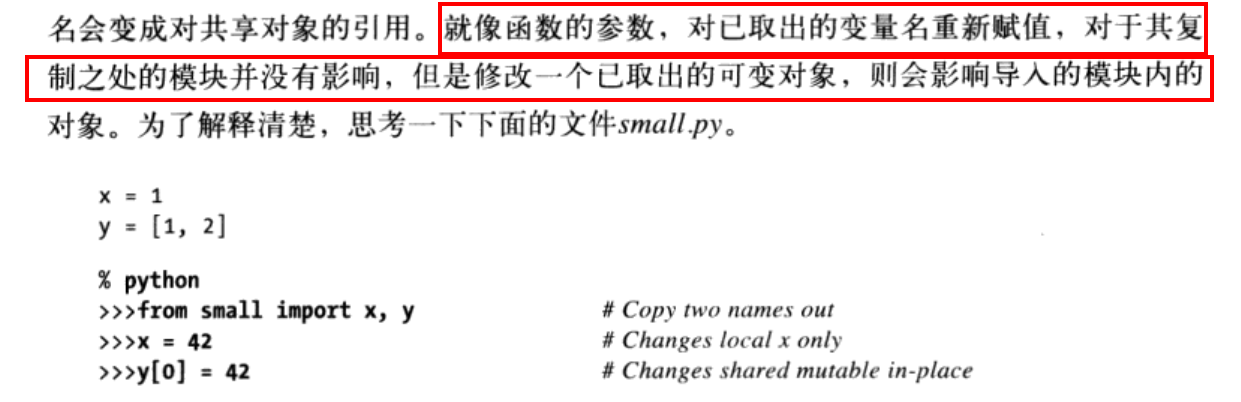
优点：可以比较精确的导入另外一个模块里面的内容，

缺点：如果在导入的父模块里面有相同命名的属性的话，会造成覆盖，具体谁覆盖谁，这个得看导入的位置与父模块相同名称的对象所在的位置。

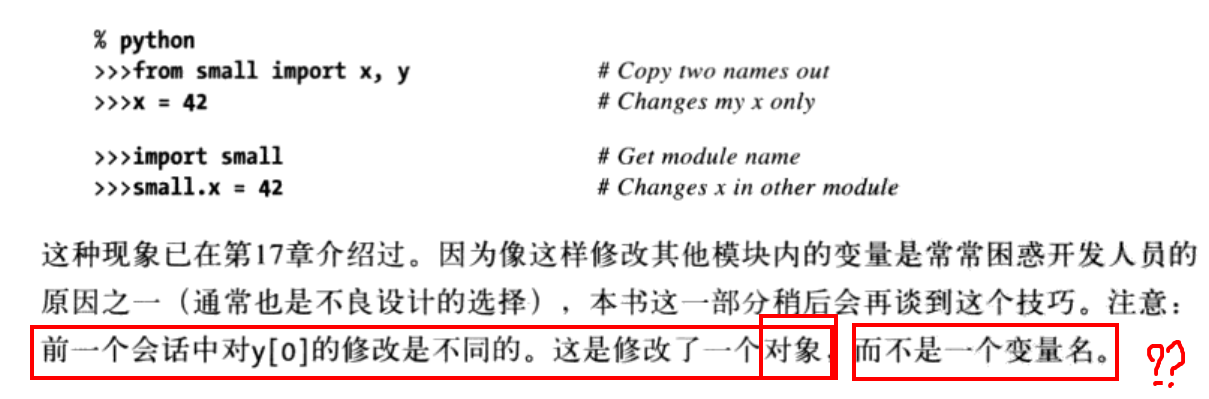
Import与from是赋值语句。另外import只发生一次。

上面两种导入方式对变量覆盖的影响：

使用from方式导入的时候，只是将被导入module里面的变量名拷贝了一份到父模块里面，所以修改的话是对变量名的修改，如下面的例子就是这种情况。这种导入方式的修改分可变变量与不可变变量的修改。



另外一种使用import方式导入的话，修改的话为对模块对象的修改，因为这种方式的变量是通过module.name的方式访问被导入模块的属性，所以修改的为模块对象，注意是模块对象，模块对象那个。注意这两种方式都可以修改另外一个模块里面的属性。



查看模块的属性：module.\_\_dict\_\_(keys)

模块的命名空间的嵌套

Reload对模块进行重载：使用reload的原因：因为使用import的话只会对导入的模块执行一次，如果我想在一个顶层代码里面对一个模块进行多次导入的话就得使用reload了。

需求：在整个工程代码运行的过程期间，动态的修改程序的某一些=参数而不用让整个工程停止。

语法：Reload（module）