# 一Python核心内容

## 1 Python高级编程

### 1.1高级1

#### 1.1.1私有化

1.相对路径，绝对路径区别

2.不要在子模块之间相互调用

主模块直接调用子模块

子模块

子模块

子模块

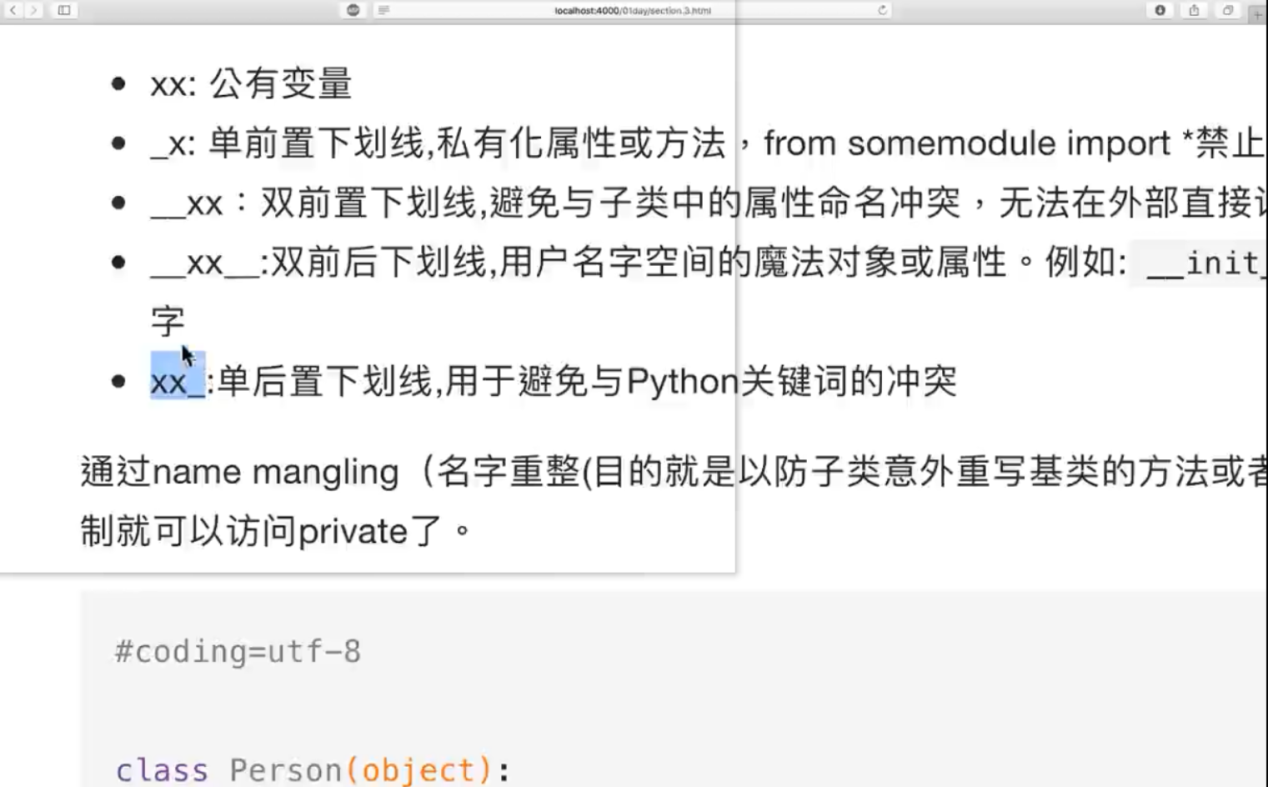
3. “==“” 用来判断内容是否相同，“is” 用来判断是否指向同一个地址（内容）

4.“from 模块名 import \* ” ：导入这个模块，但是变量名前面加单个\_或者两个\_\_的变量不能使用

另一种，直接导入模块“ import 模块名”，使用变量的时候

“模块名.变量名”就可以使用，这个变量。



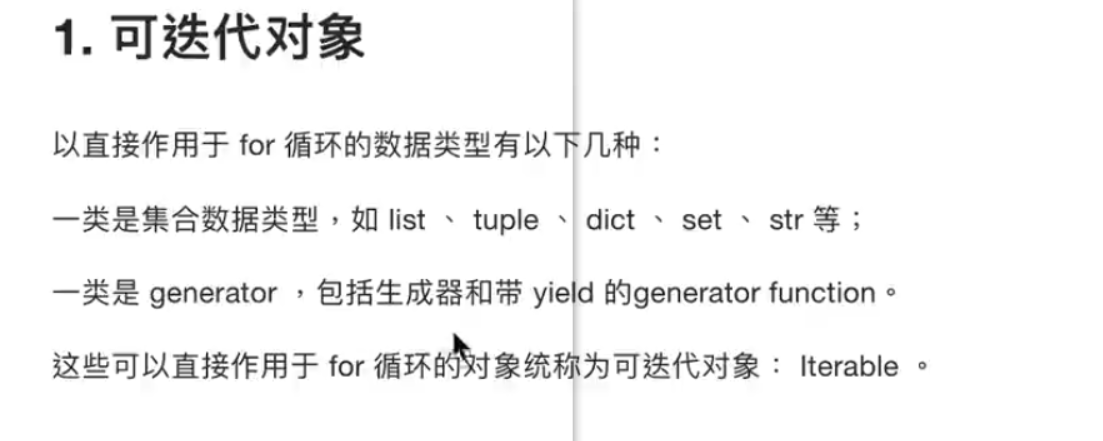


### 1.2 python高级2-生成器、闭包、装饰器

#### 1.2.1.迭代器

1．迭代器

迭代器是访问几何元素的一种方式，迭代器是一个可以记住便利的位置的对象，迭代器对象从集合的第一个元素开始访问，知道所有的元素被访问完结束，迭代器只能往前不能后退。

2 判断是否可迭代



3.迭代器

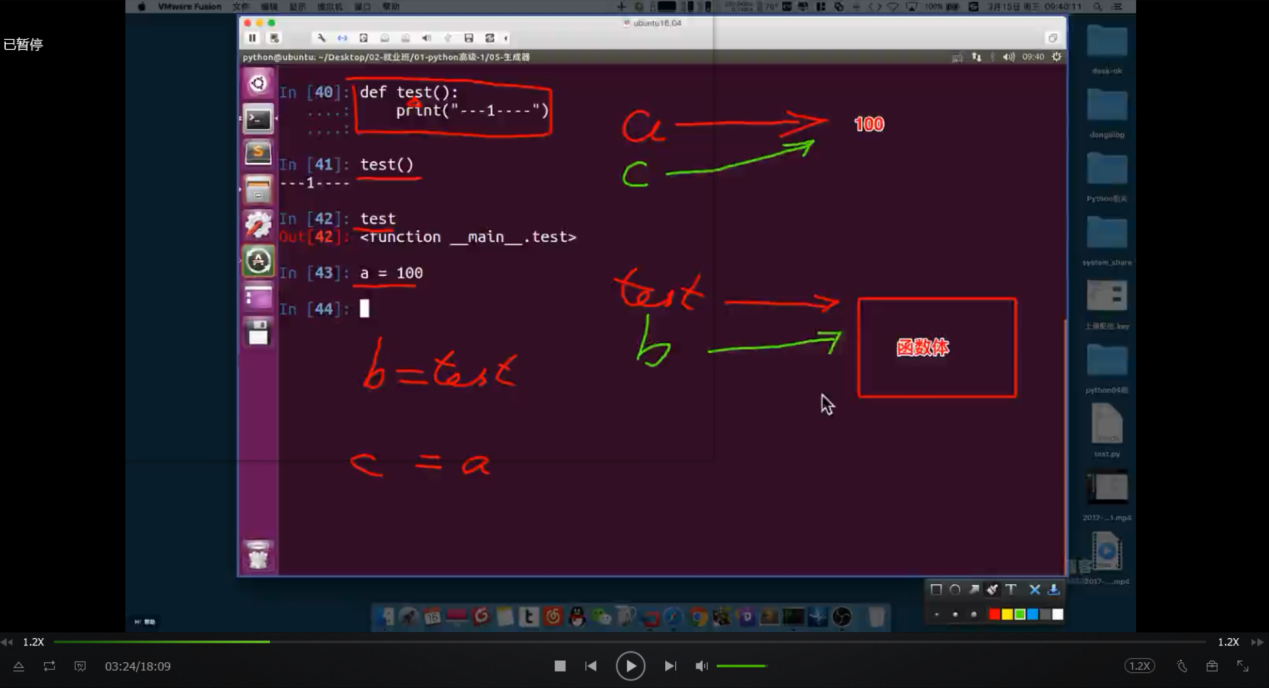
**3 iter（函数）



#### 1.2.2 闭包

1 函数引用

函数引用就是找到函数地址找到以后就可以直接用。



只要变量指向一个函数（变量名=函数名），那么就可以直接调用这个函数，只有python中可以这么用。C++中需要用到函数指针，python根本不需要。Python 中的函数和 Java、C++不太一样，Python 中的函数可以像普通变量一样当做参数传递给另外一个。

2 闭包



3 闭包应用

（1）

def test (number):

def test\_in(number):

print(number+100)

return test\_in #返回的是test\_in 函数

ret = test(100) # 另ret指向test\_in函数，100是给number赋值

ret() #最终结果为200

def test (number):

def test\_in(number2):

print(number+number2)

return test\_in #返回的是test\_in 函数

ret = test(100) # 另ret指向test函数，100是给number赋值

ret(100) #100是给number2的值，因为ret指向的是test\_in函数

（2）



4 装饰器

装饰器是程序开发中经常用的功能，所以这也是python面试中经常面试的问题，非常重要。（生成器是生成用的，装饰器是装扮用的）

注意：python中尽量避免定义函数名相同，因为如果相同调用的时候只会调用后边的那个函数，忽略前边的那个函数。C和C++会告诉你有重名的，但是python不会告诉你。

例子：

def w1(func):

def inner():

print(“----正在验证---”)

if False:

func()

else:

print(“---没有权限---“)

return inner  
#f1 = w1(f1)

@w1 #此语句就相当于f1 = w1(f1)

def f1()：

Print(“----f1---”)

#f2 = w1(f2)

@w2 #此语句就相当于f2 = w1(f2)

Def f2():

Print(“---f2---”)

f1() #调用函数f1

f2() #调用函数f2