**\_\_name\_\_ 是当前模块名，当模块被直接运行时模块名为 \_\_main\_\_ 。这句话的意思就是，当模块被直接运行时，以下代码块将被运行，当模块是被导入时，代码块不被运行。**

相信初学者在学习Python的过程中，不可避免的总会遇到if \_\_name\_\_ == 'main'语句，虽然一直知道它的作用，但是可能一直比较模糊，今天菜鸟分析就与大家一起举例说说我的理解。

举个例子，我们在下面在test.py中写入如下代码：

print( '恋习Python')

defmain:

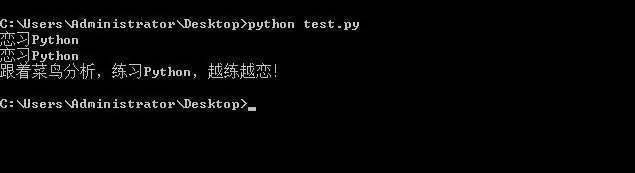
print( '恋习Python')

if\_\_name\_\_ == '\_\_main\_\_':

main

print( '跟着菜鸟分析，练习Python越练越恋')

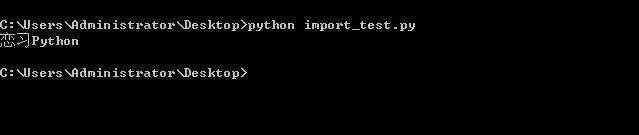
运行代码，则输出结果为：



然后，在同一文件夹新建名称为import\_test.py的脚本，只输入代码：

importtest

运行代码，则输出结果为：



与之前test.py代码运行结果比较，只有输出恋习Python，也就是if \_\_name\_\_=="\_\_main\_\_": 之前的语句被执行，之后的没有被执行。

这是为什么呢？别急，菜鸟分析继续给你举例子说明，绝对让你满意为止，最后有一种豁然开朗的感觉。

现在，我们在test.py脚本的if \_\_name\_\_=="\_\_main\_\_":之前加入print \_\_name\_\_，即将\_\_name\_\_打印出来，则最后运行结果如下：



可以看出，此时变量\_\_name\_\_的值为"\_\_main\_\_"。

再执行import\_test.py，模块内容和执行结果如下：



此时，import\_test.py中的\_\_name\_\_变量值为test，不满足\_\_name\_\_=="\_\_main\_\_"的条件，因此，无法执行其后的代码。

哈哈，是不是一下明白啦，觉得不错，给菜鸟分析点个赞哦！

再仔细想想，其运行原理也就是：

由于每个Python模块（Python文件）都包含内置的变量\_\_name\_\_，当运行模块被执行的时候，\_\_name\_\_等于文件名（包含了后缀.py）。如果import到其他模块中，则\_\_name\_\_等于模块名称（不包含后缀.py）。 而“\_\_main\_\_”等于当前执行文件的名称（包含了后缀.py）。所以当模块被直接执行时，\_\_name\_\_ == '\_\_main\_\_'结果为真；而当模块被import到其他模块中时，\_\_name\_\_ == '\_\_main\_\_'结果为假，就是不调用对应的方法。

简而言之就是：\_\_name\_\_ 是当前模块名，当模块被直接运行时模块名为 \_\_main\_\_ 。当模块被直接运行时，代码将被运行，当模块是被导入时，代码不被运行。