

# 技术解剖手

[博客园](#)[首页](#)[联系](#)[管理](#)

随笔 - 0 文章 - 388 评论 - 12

昵称：小天儿  
园龄：4年11个月  
粉丝：17  
关注：22  
[+加关注](#)

## 文章分类(386)

[00-工程\(10\)](#)[01-光电芯](#)[02-程序\(6\)](#)[03-结构与算法\(4\)](#)[04-内核\(4\)](#)[05-模式\(24\)](#)[06-语言-C++\(11\)](#)[06-语言-dot net\(2\)](#)[06-语言-Golang\(10\)](#)

## Python 大型项目基础：包、模块、类的组织

在Python 中引用是非常简单的事情，这里需要清楚三个概念就可以了包、模块、类。类这个就不用说了。

模块对应的是一个.py 文件，那么module\_name 就是这个文件去掉.py 之后的文件名，py 文件中可以直接定义一些变量、函数、类。

那么包我们可以看作一个包含\_\_init\_\_.py 和一系列.py 文件的文件夹，这样做的目的是为了区别包和普通字符串。

```
import module_name
```

```
from package_name import module_name
```

```
from package_name import *
```

那么如何import 时，python 解释器如何才能找到该模块放置的文件位置呢？python 寻找模块的优先级如下：

- 1、 当前文件目录
- 2、 环境变量PYTHONPATH
- 3、 sys.path(list 类型)

sys.path 是list 类型，我们可以通过insert(), append() 方法来增加模块导入的搜索路径，如：

```
import sys
```

```
path = "....." # 需要增加的路径
```

```
sys.path.insert(0, path)
```

在import模块时，该模块的顶层代码将会被执行一次。如果该模块被import多次，例如import A, import B.其中B模块本身也有import A，那么只在第一次被import的时候顶层代码会被执行。

模块能像包含函数定义一样，可包含一些可执行语句。这些可执行语句通常用来进行模块的初始化工作。这些语句只在模块第一次被导入时被执行。这非常重要，有些人以为这些语句会多次导入多次执行，其实不然。

模块在被导入执行时，python解释器为加快程序的启动速度，会在与模块文件同一目录下生成.pyc文件。我们知道python是解释性的脚本语言，而.pyc是经过编译后的字节码，这一工作会自动完成，而无需程序员手动执行。

什么时候你应该使用 `from module import` ?

- 如果你要经常访问模块的属性和方法，且不想一遍又一遍地敲入模块名，使用 `from module import`。
- 如果你想要有选择地导入某些属性和方法，而不想要其它的，使用 `from module import`。
- 如果模块包含的属性和方法与你的某个模块同名，你必须使用 `import module` 来避免名字冲突。

除了这些情况，剩下的只是风格问题了，你会看到用两种方式编写的 Python 代码。

**\*\* 尽量少用 `from module import *`，因为判定一个特殊的函数或属性是从哪来的有些困难，并且会造成调试和重构都更困难。**

其它要点：

## 包

在创建许许多多模块后，我们可能希望将某些功能相近的文件组织在同一文件夹下，这里就需要运用包的概念了。包对应于文件夹，使用包的方式跟模块也类似，唯一需要注意的是，当文件夹当作包使用时，文件夹需要包含 `__init__.py` 文件，主要是为了避免将文件夹名当作普通的字符串。`__init__.py` 的内容可以为空，一般用来进行包的某些初始化工作或者设置 `__all__` 值，`__all__` 是在 `from package-name import *` 这语句使用的，全部导出定义过的模块。

那么我们平常新建文件的时候，都是建一个.py文件（类似java，但是python不用建立类，是以模块为载体）至于怎样组织包就要功能需要了

在C/C++/Java中，main是程序执行的起点，Python中，也有类似的运行机制，但方式却截然不同：Python使用缩进对齐组织代码的执行，所有没有缩进的代码（非函数定义和类定义），都会在载入时自动执行，这些代码，可以认为是Python的main函数。

每个文件（模块）都可以任意写一些没有缩进的代码，并且在载入时自动执行，为了区分主执行文件还是被调用的文件，Python引入了一个变量\_\_name\_\_，当文件是被调用时，\_\_name\_\_的值为模块名，当文件被执行时，\_\_name\_\_为'\_\_main\_\_'。这个特性，为测试驱动开发提供了极好的支持，我们可以在每个模块中写上测试代码，这些测试代码仅当模块被Python直接执行时才会运行，代码和测试完美的结合在一起。

典型的Python文件结构：

08-专题-分布式(3)

08-专题-嵌入式(11)

08-专题-区块链(1)

08-专题-容器(3)

08-专题-数据库(22)

09-项目-Elasticsearch(14)

09-项目-redis(11)

10-其他(10)

方法操作(38)

技术思考(25)

网络(6)

## 最新评论

1. Re:玩转linux——Linux性能监控：Top命令详解【转】

厉害，学习了

--张小七soso

2. Re:plantuml使用教程【转】

这个很有意思

--想不到

3. Re:Eclipse闪退/打不开/无法启动/一闪而过

方法6很有效，谢谢啦！！

--半夏微凉\*\*\*\*\*

4. Re:python为什么叫胶水语言？python为什么是系统脚本？

mark

--愤怒的TryCatch

5. Re:Elasticsearch系统概念及架构图

浅入深出ElasticSearch构建高性能搜索架构  
课程观看地址：

--无名小卒的苦

```
#!/usr/bin/env python
```

(1) 起始行

```
"this is a test module"
```

(2) 模块文档（文档字符串）

```
import sys
import os
```

(3) 模块导入

```
debug = True
```

(4) (全局) 变量定义

```
class FooClass (object):
    "Foo class"
    pass
```

(5) 类定义（若有）

```
def test():
    "test function"
    foo = FooClass()
    if debug:
        print 'ran test()'
```

(6) 函数定义（若有）

```
if __name__ == '__main__':
    test()
```

(7) 主程序

[python import 模块方法](#)



python包含子目录中的模块方法比较简单，关键是能够在sys.path里面找到通向模块文件的路径。

下面将具体介绍几种常用情况：

(1) 主程序与模块程序在同一目录下：

如下面程序结构：

```
`-- src
```

```
    |-- mod1.py
```

```
    `-- test1.py
```

若在程序test1.py中导入模块mod1, 则直接使用**import mod1**或**from mod1 import \***;

(2) 主程序所在目录是模块所在目录的父(或祖辈)目录

如下面程序结构：

```
`-- src
```

```
    |-- mod1.py
```

```
    |-- mod2
```

```
    | `-- mod2.py
```

```
    `-- test1.py
```

若在程序test1.py中导入模块mod2, 需要在mod2文件夹中建立空文件\_\_init\_\_.py文件(也可以在该文件中自定义输出模块接口); 然后使用 **from mod2.mod2 import \*** 或**import mod2.mod2**.

(3) 主程序导入上层目录中模块或其他目录(平级)下的模块

如下面程序结构：

```
`-- src
```

```
    |-- mod1.py
```

```
    |-- mod2
```

```
    | `-- mod2.py
```

```
    |-- sub
```

```
    | `-- test2.py
```

```
    `-- test1.py
```

若在程序test2.py中导入模块mod1和mod2。首先需要在mod2下建立\_\_init\_\_.py文件(同(2)), src下不必建立该文件。然后调用方式如下：

下面程序执行方式均在程序文件所在目录下执行，如test2.py是在cd sub;之后执行python test2.py而test1.py是在cd src;之后执行python test1.py; 不保证在src目录下执行python sub/test2.py成功。

```
import sys
```

```
sys.path.append("../")
```

```
import mod1
import mod2.mod2
```

(4) 从(3)可以看出，导入模块关键是能够根据sys.path环境变量的值，找到具体模块的路径。这里仅介绍上面三种简单情况。

原文：<http://blog.chinaunix.net/uid-26602509-id-3499026.html>

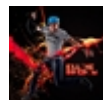
## 用以致学，学以致用

分类: [06-语言-python](#)

好文要顶

关注我

收藏该文



小天儿

关注 - 22

粉丝 - 17

[+加关注](#)

0

0

posted @ 2016-11-03 11:45 小天儿 阅读(120) 评论(0) 编辑 收藏

[刷新评论](#) [刷新页面](#) [返回顶部](#)

注册用户登录后才能发表评论，请 [登录](#) 或 [注册](#)，[访问网站首页](#)。

【推荐】50万行VC++源码: 大型组态工控、电力仿真CAD与GIS源码库

【推荐】腾讯云上实验室 1小时搭建人工智能应用

【推荐】可嵌入您系统的“在线Excel”！SpreadJS 纯前端表格控件

【推荐】阿里云“全民云计算”优惠升级



#### 最新IT新闻:

- 世界首条量子通信干线开通 首次洲际量子通信实现
  - 天猫承认说明和文案抄袭 永久下线"智能测肤"功能
  - 巨人网络豪掷8.2亿控股投哪网 签严苛对赌协议
  - 饿了么科技值要爆炸：继无人机后，语音点餐系统横空出世
  - 继中国后，韩国成为第二个明确禁止ICO的国家
- » 更多新闻...



#### 最新知识库文章:

- 如何阅读计算机科学类的书
  - Google 及其云智慧
  - 做到这一点，你也可以成为优秀的程序员
  - 写给立志做码农的大学生
  - 架构腐化之谜
- » 更多知识库文章...