中國人民大學
RENMIN UNIVERSITY OF CHINA

第1编 Python语法基础

第5讲 模块与包

余力

buaayuli@ruc.edu.cn

中国人民大学
RENMIN UNIVERSITY OF CHINA

01. 模块

模块的概念

- 一个文件是一个模块
- 减少模块之间的对偶关系(coupling)
  - 如同函数, 模块写的越独立, 越好
  - 若模块在另一个模块里, 要做到不受该模块的广域变量影响比较好
- 模组应该尽量不要更改别的模块的变量内容
  - 少用goto

基本概念

- 建立模块：
  - 只要将程序代码写到文档里, 存成.py即可
- 使用模块：
  - Import：得到一整个完整的模块文档
  - From：可以从某个模块文档里取出几个特定的名称
  - Reload：可重载模块程序代码而毋须跳离解译程序
  - 另外也可以直接于系统下执行

模块搜索路径

1、在当前目录下搜索该模块；

2、环境变量PYTHONPATH指定的路径列表中依次搜索；

模组搜寻路径：

sys.path

sys.path.append

sys.path.append('C:\\Users\\yuli')

3、在python安装路径中搜索

import和from

- import module
- import module.xx.xx
- from module.xx.xx import xx
- from module.xx.xx import xx as rename
- from module.xx.xx import *

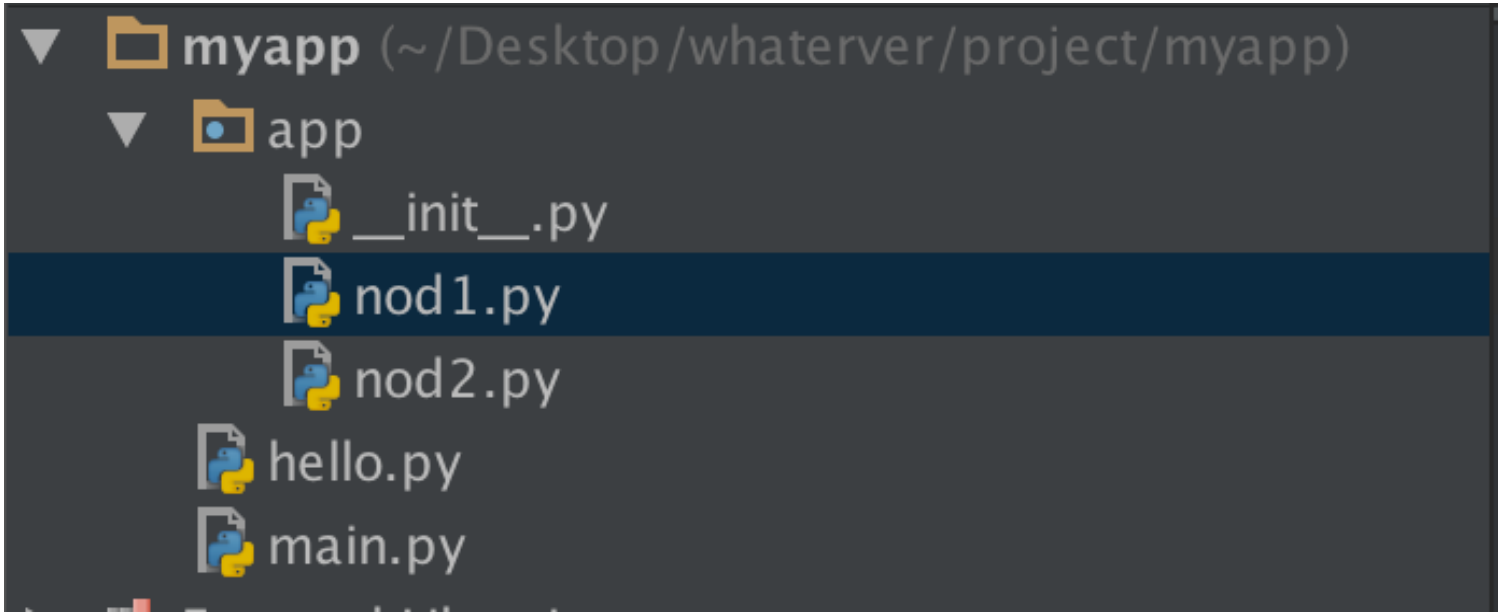
基本范例

```
import print_example
print_example.printer("Hello python!")

from print_example import printer
printer("Hello python!")

from print_example import *
printer("Hello python!")
```

不同目录之间的模块调用



Nod2.py调用hello.py

sys.path.append("c:\myapp") **#添加路径**

import hello

hello.hello1()

基本范例二

- Ex. 有一个名为"loadexample.py"的模块文档

```
import sys
name = 42
def func(): pass
print "done loading."
```

第1次导入时会自动运行

有命名空间

- import loadexample
  - done loading.
- loadexample.name → 42
- loadexample.func → <function func at 765f20>
- loadexample.__dict__.keys()
  - ["__file__", "name", "__name__", #__file__:模块的文件名
  - "sys", "__doc__", "__builtins__", "func"] #__name__:模块名称

模块的内置属性

- 模块有一些内置属性，用于完成特定的任务。
- dir(shapes)，就输出了模块shapes的属性。

```
['CHAR', '__builtins__', '__cached__', '__doc__', '__file__', '__loa
der__', '__name__', '__package__', '__spec__', 'rectangle', 'square'
, 'triangle']
```

- 如：
  - **__doc__：模块中用于描述的文档字符串**
  - **__name__：模块名**
  - **__file__：模块保存的路径**

__doc__(1/2)

- python提供__doc__来放置你想要说明的字符串(函数, 模块, 类别皆有)
  - **只要开头的部份是字符串而非语句即可**
  - python会将该字符串存到__doc__里(只有储存第一行)
- 范例：有一个module名字叫**docstr**

```
"I am:  docstr.__doc__"

class larc:

    "I am: larc.__doc__ or docstr.larc.__doc__"

    def method(self, arg):

        "I am: larc.method.__doc__ or self.method.__doc__"
```

__doc__(2/2)

```
import docstr

print( docstr.__doc__ )

    "I am: docstr.__doc__"

print( docstr.larc.__doc__ )

    "I am: larc.__doc__ or docstr.larc.__doc__"

print( docstr.larc.method.__doc__ )

    "I am: larc.method.__doc__ or self.method.__doc__"
```

__name__和__main__

- 每一个模块都有一个内建的属性：__name__
  - **当作程序来执行，__name__会设定成"__main__**
  - 当作模块来导入，__name__会设定成模块的名称
- __main__在自身测试上最常出现

```
% tester.py

def add(x,y):
    return x+y

if __name__=="__main__":
   print (add(2,4))
```

- **%python**
- **import tester**
  - **This is a module**
- **%run test.py**
  - **6**

名称引用

- 简单变量
  - "X"意指在当前的范围里搜寻名称X(LGB法则)
- 引用方法
  - "X.Y"意指在X对象中搜寻属性Y(not范围)
- 引用路径
  - "X.Y.Z"意指在X对象中搜寻Y, 然后在对象X.Y中搜寻Z

导入模式

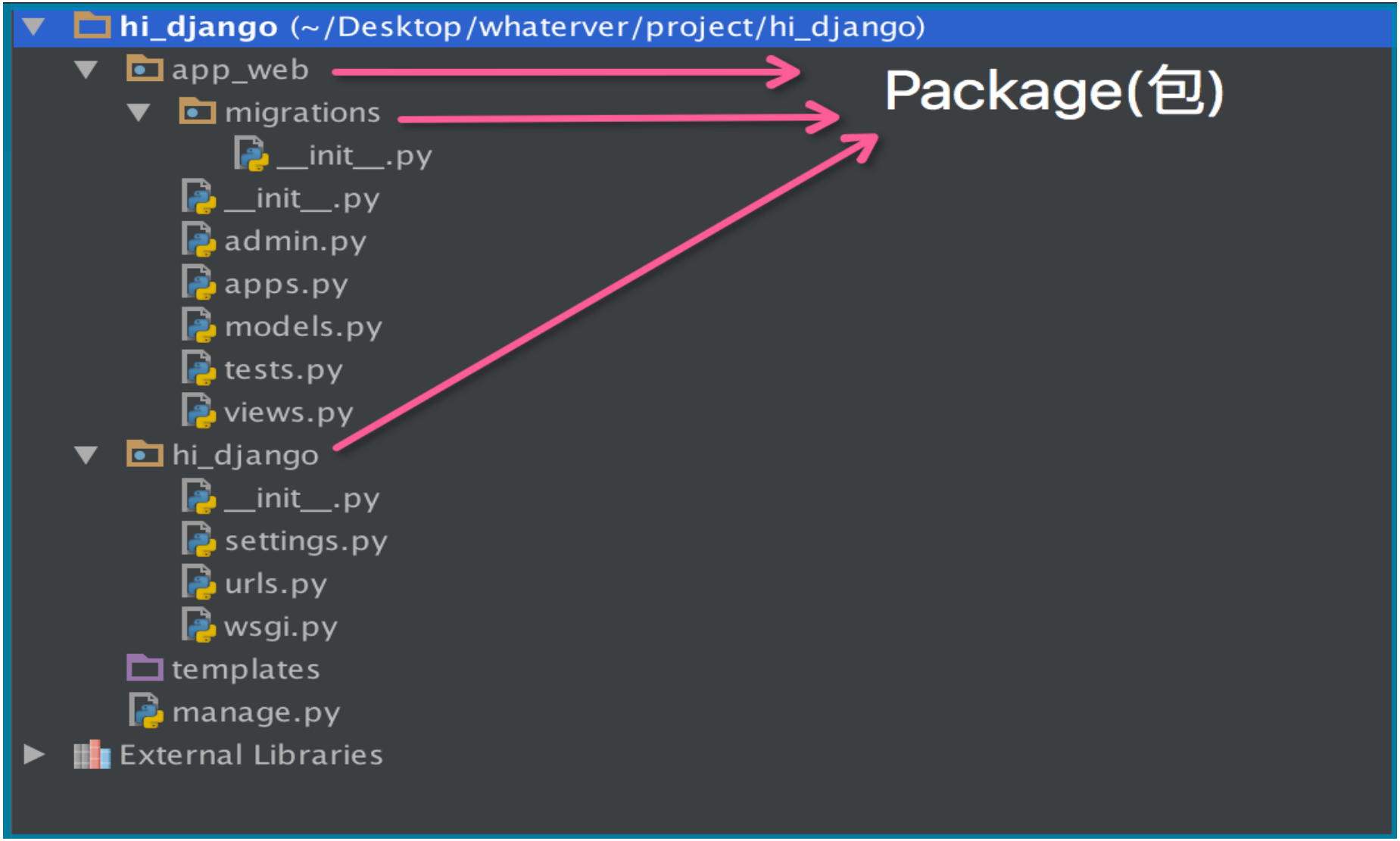
- **导入只发生一次**
  - 模组在第一次使用import或from时会载入
  - 执行模块的程序代码, 会使模块建立其名称空间
  - 之后任何的import和from**只会从已加载的模块存取东西**
- ## simple.py
- spam = 1

```
import simple
simple.spam          #→   1
simple.spam = 2
import simple
simple.spam          #→   2
```

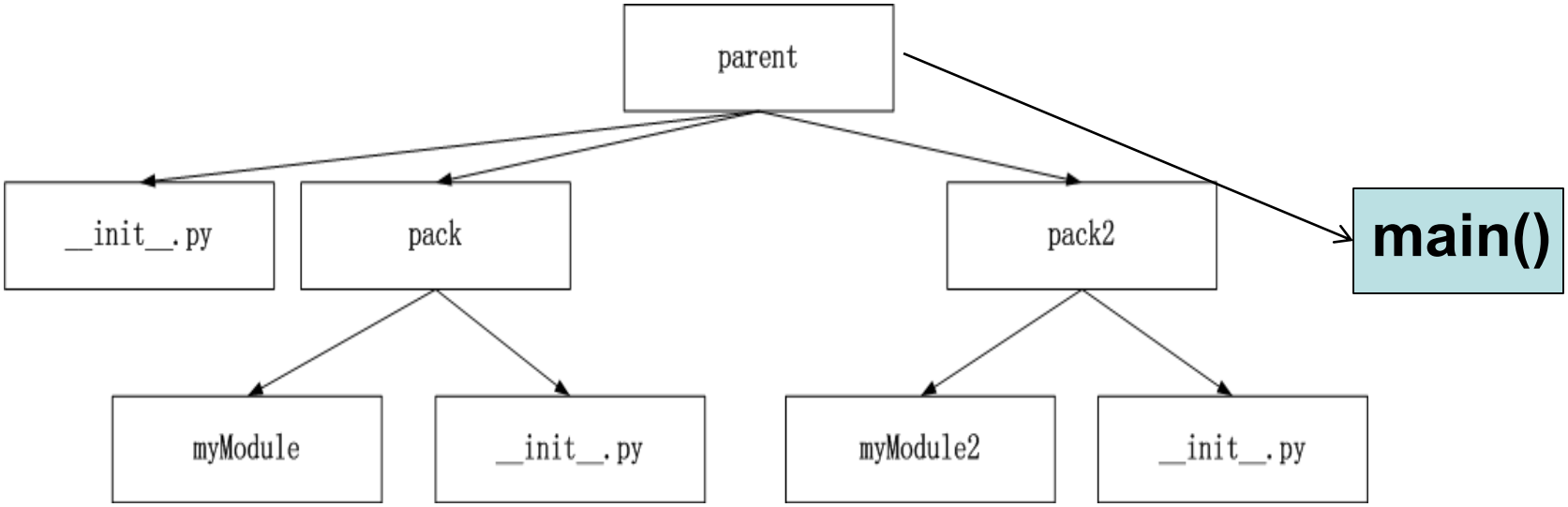
中国人民大学
RENMIN UNIVERSITY OF CHINA

02. 包

自定义包


hi_django (~/Desktop/whaterver/project/hi_django)
app_web
migrations
__init__.py
__init__.py
admin.py
apps.py
models.py
tests.py
views.py
hi_django
__init__.py
settings.py
urls.py
wsgi.py
templates
manage.py
External Libraries
Package(包)

例：一个包与模块的树形关系



- 定义一个包parent。在parent包中创建两个子包pack和pack2。
- Pack包中定义了一个模块myModule，pack2包中定义了一个模块myModule2。
- 最后在包parent中定义一个模块main，调用包pack和pack2。

导入包-1

```
Graphics/
        __init__.py
        Primitive/
            __init__.py
            lines.py
            fill.py
            text.py
              ...
        Graph2d/
            __init__.py
            plot2d.py
            ...
        Graph3d/
            __init__.py
            plot3d.py
            ...
        Formats/
            __init__.py
            gif.py
            png.py
            tiff.py
            jpeg.py
```

- import Graphics
  - Graphics.Primitive.fill.floodfill(img,x,y,color)
- from Graphics.Primitive import fill
  - fill.floodfill(img,x,y,color)
- from Graphics.Primitive.fill import floodfill
  - floodfill(img,x,y,color)

- 下边这个语句具有歧义:

  from Graphics.Primitive import *

- Import Graphics

导入包-2

- 下面这个语句**只会执行Graphics目录下的__init__.py文件，而不会导入任何模块:**

  import Graphics

  Graphics.Primitive.fill.floodfill(img,x,y,color) # 失败!

- **可以采取下面手段来解决这个问题:**

  # Graphics/__init__.py

  import Primitive, Graph2d, Graph3d

  # Graphics/Primitive/__init__.py

  import lines, fill, text, ...

  __all__ = ["lines","text","fill",...]

第三方模块的导入

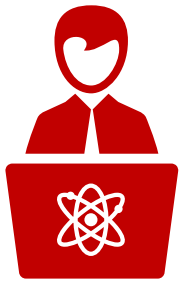
（1）单文件模块

- 直接把文件拷贝到 $python_dir/Lib

（2）多文件模块，带setup.py

- python setup.py install

中国人民大学
RENMIN UNIVERSITY OF CHINA



谢谢大家！