

PYTHON

编程基础

包

概述

- Python中的包（Package）的作用与操作系统中文件夹的作用相似，利用包可以将多个关系密切的模块组成在一起，一方面方便进行各脚本文件的管理，另一方面可以有效避免模块命名冲突问题。
- 定义一个包，就是创建一个文件夹并在该文件夹下创建一个__init__.py文件，文件夹的名字就是包名。
- 另外，可以根据需要在该文件夹下再创建子文件夹，子文件夹中创建一个__init__.py文件，则又形成了一个子包。
- 模块可以放在任何一个包或子包中，在导入模块时需要指定所在的包和子包的名字。例如，如果要导入包A中的模块B，则需要使用“import A.B”。

概述

提示：__init__.py可以是一个空文件，也可以包含包的初始化代码或者设置__all__列表。

示例



例如:

sound/ 顶级包

__init__.py 初始化这个声音包

formats/ 文件格式转换子包

__init__.py

wavread.py

wavwrite.py

aiffread.py

aiffwrite.py

auread.py

auwrite.py

...

示例



effects/ 音效子包

__init__.py
echo.py
surround.py
reverse.py
...



filters/ 过滤器子包

__init__.py
equalizer.py
vocoder.py
karaoke.py
...

示例

如果要使用sound包的effects子包的echo模块，则可以通过下面方式导入：

```
import sound.effects.echo
```

假设在echo模块中有一个echofilter函数，则调用该函数时必须指定完整的名字（包括各层的包名和模块名），即：

```
sound.effects.echo.echofilter(实参列表)
```

示例

也可以使用from import方式导入包中的模块，如：

```
from sound.effects import echo
```

通过这种方式，也可以正确导入sound包的effects子包的echo模块，而且在调用echo模块中的函数时不需要加包名，如：

```
echo.echofilter(实参列表)
```

使用from import也可以直接导入模块中的标识符，如：

```
from sound.effects.echo import echofilter
```

此时调用echofilter函数可直接写作：

```
echofilter(实参列表)
```