python□□□□□module□□(packag
ort

python□□□□□□□□□□□□□□□□□□□□□

□□□□□

```
project/
│
├── main.py
├── routers/
│   ├── __init__.py
│   ├── items.py
│   └── users.py
├── models/
│   ├── __init__.py
│   ├── item.py
│   └── user.py
├── schemas/
│   ├── __init__.py
│   ├── item.py
│   └── user.py
├── utils/
│   ├── __init__.py
│   ├── string_util.py
│   └── log_factory.py
└── templates/
    └── index.html
```

package□import

- □□import py_name □□□□

```
project/
│
├── main.py
├── vp_smtp.py
└──
```

```
# vp_smtp


def send_mail(msg: str):
    print(msg)
    pass


# main.py

# 模块全部导入
import vp_smtp
# 使用模块名称访问
vp_smtp.send_mail("使用模块名称访问")

# 导入模块并重别名
import vp_smtp as vpsmtp
# 使模块别名调用
vpsmtp.send_mail("使模块别名调用")

# 导入模块中的具体的方法
from vp_smtp import send_mail
# 调用方法
send_mail("直接调用方法")
```

- 将源文件写入到一个文件中，并将文件夹下面所有包(一组模块)

```
# utils\vp_smtp.ly

def send_mail(msg: str):
    print(msg)
```

```
    pass


# main.py



#   按源文件存储路径 package.moduel 方式导入，自动查找对应的py文件
import utils.vp_smtp
# 完全限定调用 package.moduel.fun
utils.vp_smtp.send_mail("完全限定调用 package.moduel.fun")

#   按源文件存储路径 package.moduel 方式导入，并增加别名，自动查找对应的py文件
import utils.vp_smtp as vpsmtp
# 别名调用
vpsmtp.send_mail("别名调用 alias.fun")

# 从包中导入 vp_smtp
from utils import vp_smtp
vp_smtp.send_mail("从包中导入 vp_smtp")
```

__init__.py

__init__.py 是一个特殊的文件

- 标记文件夹是一个package
- 可以加__version__,例如大三都喜欢加入1
- 定义包级别的变量，函数，等,用作访问控制

```
# __init__.py


# 先在__init__.py 初始化包和组织包的结构
from .vp_smtp import send_mail


# main.py
```

```
# 推荐使用这种方式 便于组织和管理源文件的公开
from utils import send_mail
send_mail("从__init__导入')
```

版本号 #12
由 董列涛 创建于 24 三月 2025 13:02:59
由 董列涛 更新于 18 九月 2025 03:45:38