

python编译py文件为pyc文件

参考: <https://blog.csdn.net/kongkongqixi/article/details/106207658>

1.编译文件

编译py文件为pyc文件：

```
import py_compile
py_compile.compile('./test.py')
```

结果会在test.py同文件夹下生成__pycache__文件夹：生成test.cpython-36.pyc
[]cpython-36,36数字是python版本，不同版本的数字不同）文件；

编译文件夹下所有py文件为pyc文件：

```
import compileall
path = './'
compileall.compile_dir(path)
```

结果会在每一文件夹下都会生成__pycache__文件夹，把当前文件夹下的py文件编译为xxx.cpython-36.pyc文件保存；

2.文件读取

pyc文件运行：需要把文件xxx.cpython-36.pyc重命名为xxx.pyc这样就可以正常文件一样导入，如果是文件夹下所有文件，都需要修改为xxx.pyc文件形式，不然可能会出现找不到包的导入报错，导入如下：

```
import test
```

运行pyc文件形式：

```
import pyc
```

3.全项目替换pyc文件实例代码

编译：

```
import compileall
path = './'
compileall.compile_dir(path)
```

pyc文件替换py文件，并修改后缀：

```
import os
import shutil
path = './'
def move_pyc(path):
    for i in os.listdir(path):
        if os.path.isdir(os.path.join(path, i)):
            move_pyc(os.path.join(path, i))
    if os.path.exists(os.path.join(path, '__pycache__')):
        for name in os.listdir(os.path.join(path, '__pycache__')):
            file_name = name.split('.')[0]+'.py'
            if os.path.exists(os.path.join(path, file_name)):
```

```
print(os.path.join(path, file_name))
# os.remove(os.path.join(path, file_name)) #删除py文件, 慎重
shutil.move(os.path.join(path, '__pycache__', name),
os.path.join(path, name.replace('cpython-36.', '')))
```

版权声明：本文为CSDN博主kongkongqixi的原创文章，遵循CC 4.0 BY-SA版权协议，转载请附上原文出处链接及本声明。

原文链接<https://blog.csdn.net/kongkongqixi/article/details/106207658>