通过使用Flask-Script扩展，我们可以在Flask服务器启动的时候，通过命令行的方式传入参数。跟django中的runserver类似，可以在命令行下制定运行服务器额参数

Flask-Script是一个让你的命令行支持自定义命令的工具，它为Flask程序添加一个命令行解释器。可以让我们的程序从命令行直接执行相应的程序。

通过使用Flask-Script扩展，我们可以在Flask服务器启动的时候，通过命令行的方式传入参数。而不仅仅通过app.run()方法中传参，比如我们可以通过python hello.py runserver –host ip地址，告诉服务器在哪个网络接口监听来自客户端的连接。默认情况下，服务器只监听来自服务器所在计算机发起的连接，即localhost连接。

Flask-Script插件为在Flask里编写额外的脚本提供了支持。包括了一个开发服务器，一个定制的Python命令行，用于执行初始化数据库、定时任务和其他属于web应用之外的命令行任务的脚本

需要先安装 flask-script 扩展模块

```
pip install flask-script
```

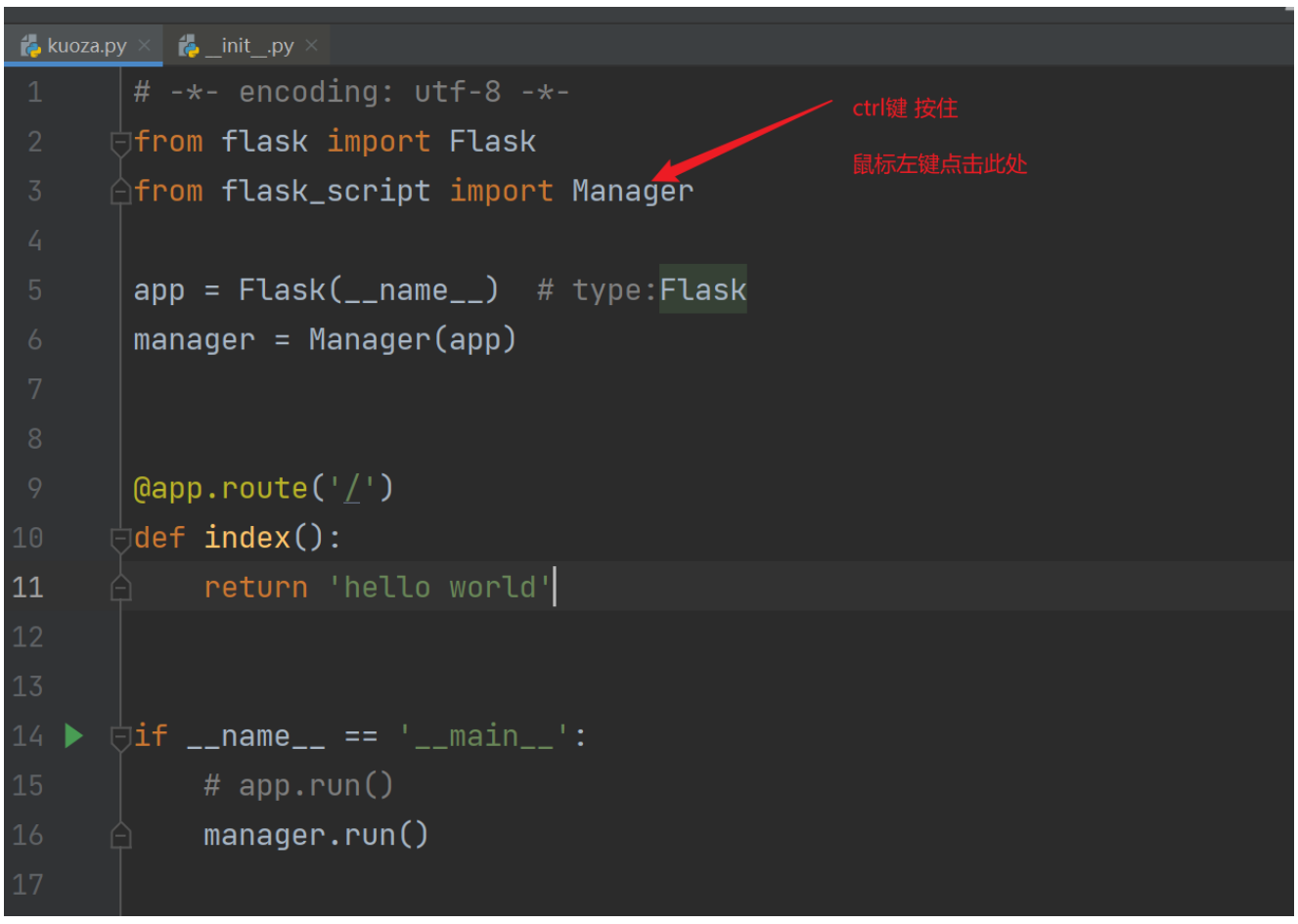
```
# -*- encoding: utf-8 -*-
from flask import Flask
from flask_script import Manager

app = Flask(__name__) # type:Flask
manager = Manager(app)


@app.route('/')
def index():
    return 'hello world'
```

```
if __name__ == '__main__':
    # app.run()
    manager.run()
```

```
D:\pythonfile\pythonProject>workon flask_bei
(flask_bei) D:\pythonfile\pythonProject>python kuoza.py runserver --help
Traceback (most recent call last):
  File "kuoza.py", line 3, in <module>
    from flask_script import Manager
  File "D:\EVNS\flask_bei\lib\site-packages\flask_script\__init__.py", line 15, in <module>
    from flask._compat import text_type
ModuleNotFoundError: No module named 'flask._compat'
```

此问题就别百度降版本乱搞了

出现这个问题是因为flask_script包2版上初始化导包有问题导致的，所以直接了当的调整下就好



kuoza.py × __init__.py ×

```python
# -*- coding: utf-8 -*-
from __future__ import absolute_import

import os
import re
import sys
import types
import warnings
from gettext import gettext as _
from collections import OrderedDict

import argparse

from flask import Flask
# from flask._compat import text_type
from flask_script._compat import text_type

from ._compat import iteritems
from .commands import Group, Option, Command, Server, Shell
from .cli import prompt, prompt_pass, prompt_bool, prompt_choices

```

这是原来的导入语法

调整后的导入语法

记得 ctrl s 保存

可以使用下面命令查看可选参数

```
python kuoza.py runserver --help
```

```
(flask_bei) D:\pythonfile\pythonProject>python kuoza.py runserver --help
usage: kuoza.py runserver [-?] [-h HOST] [-p PORT] [--threaded]
                          [--processes PROCESSES] [--passthrough-errors] [-d]
                          [-D] [-r] [-R] [--ssl-crt SSL_CRT]
                          [--ssl-key SSL_KEY]

Runs the Flask development server i.e. app.run()

optional arguments:
  -?, --help            show this help message and exit
  -h HOST, --host HOST
  -p PORT, --port PORT
  --threaded
  --processes PROCESSES
  --passthrough-errors
  -d, --debug           enable the Werkzeug debugger (DO NOT use in production
                        code)
  -D, --no-debug        disable the Werkzeug debugger
  -r, --reload          monitor Python files for changes (not 100% safe for
                        production use)
  -R, --no-reload       do not monitor Python files for changes
  --ssl-crt SSL_CRT     Path to ssl certificate
  --ssl-key SSL_KEY     Path to ssl key
```

扩展--为当前应用添加脚本命令

```
"""自定义flask_script终端命令"""
from flask_script import Command
class HelloCommand(Command):
    """命令的相关描述"""
    def run(self):
        with open("text.txt", "w") as f:
            f.write("hello\r\nhello")
            pass
        print("这是执行了hello命令")

manage.add_command('hello', HelloCommand())
```