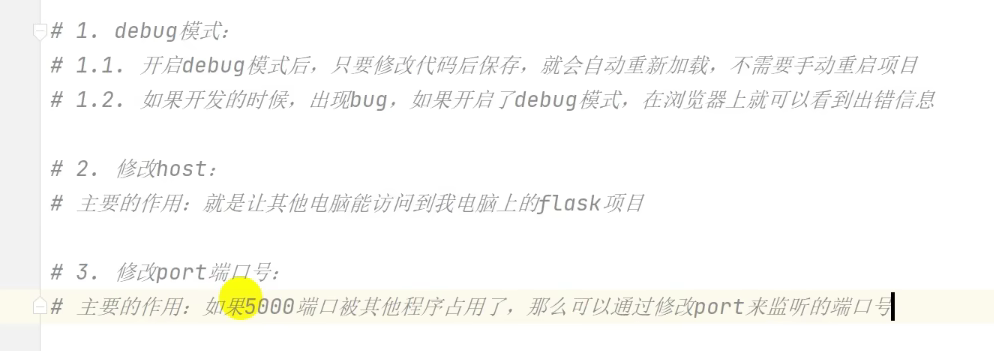
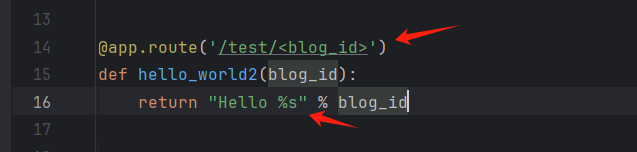
# 项目配置



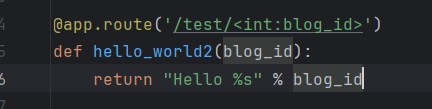
Debug模式可以热更新代码

# url与视图

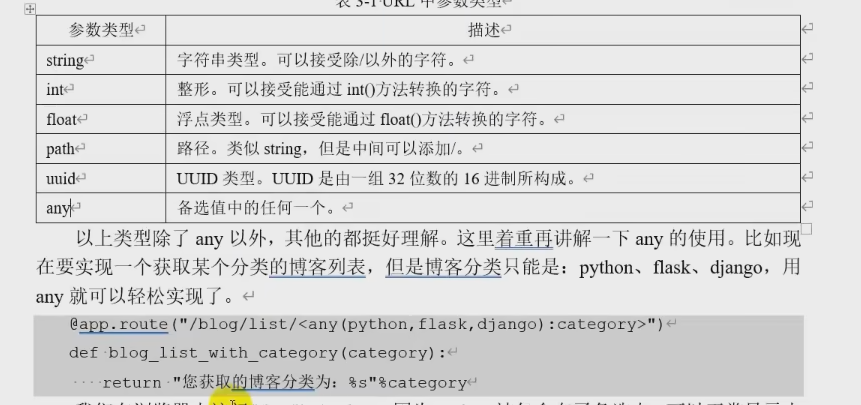
## path风格带参数

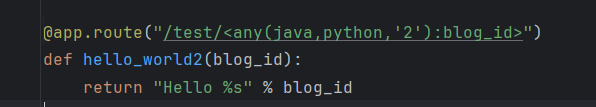


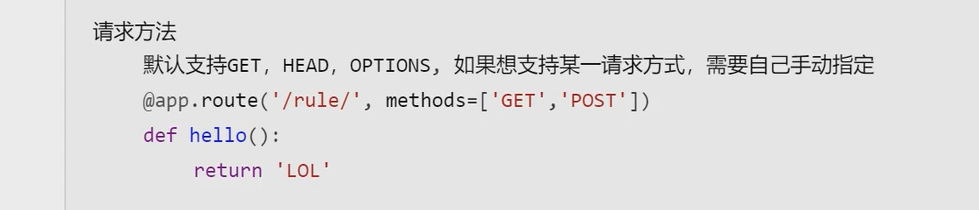
## 定义类型



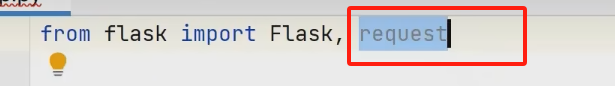
定义枚举类型：any





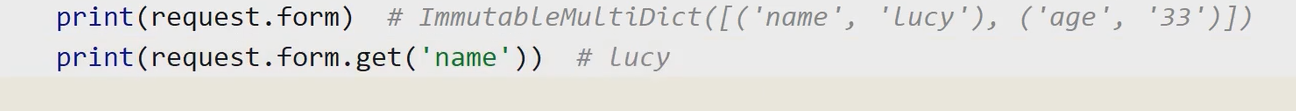


## 问号传参：

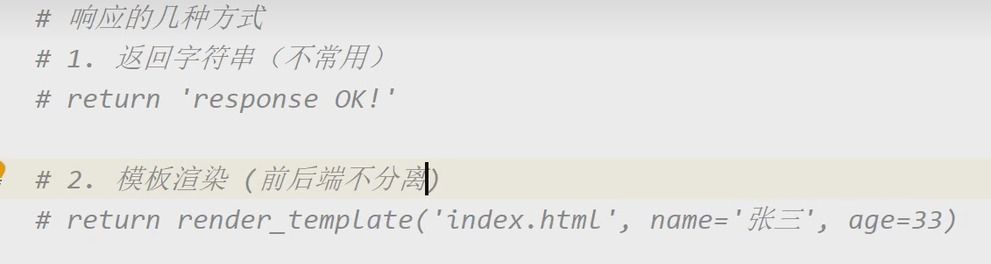




获取post参数



## 返回格式：



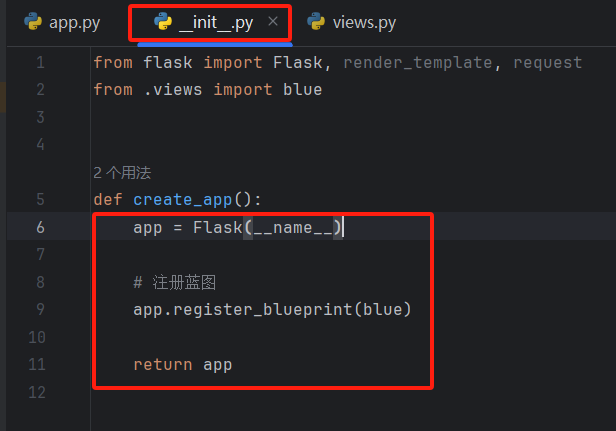


## 重定向



## 项目拆分（使用蓝图）：

先在\_\_init\_\_.py定义一个方法，类似工厂设计模式



然后再app.py调用初始化app



在view页面则使用蓝图（需要在\_\_init\_\_.py注册到app）来拆分



执行流程：

在**app.py**中点击运行

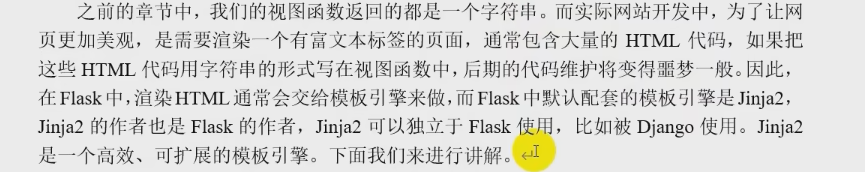
执行from App import create\_app这句代码时，会进行APP导包，然后执行**\_\_init\_\_.py**

在**\_\_init\_\_.py**内会执行from .views import blue 然后去执行**views.py**的代码

执行完views.py，再执行create\_app() 函数的创建，里面的代码并不执行，

再回到**app.py** 执行create\_app() 方法里面的内容，最后启动

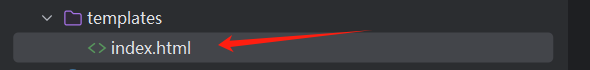
# Jinja2模板



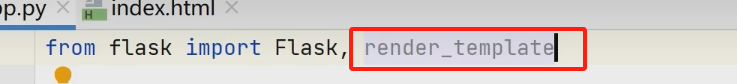
就是渲染html文件的

## 1.使用

新建html



引入：



使用：

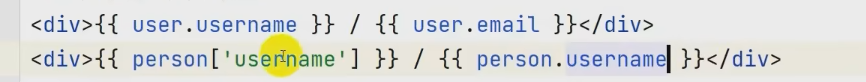




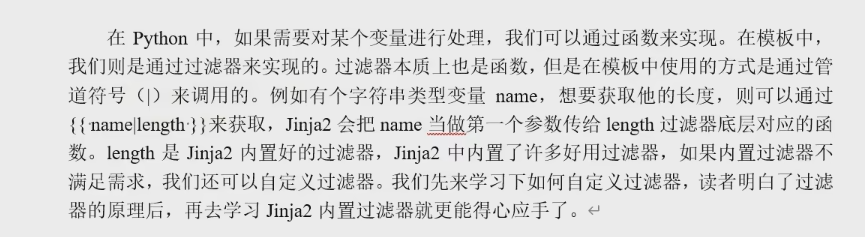
传递类属性：

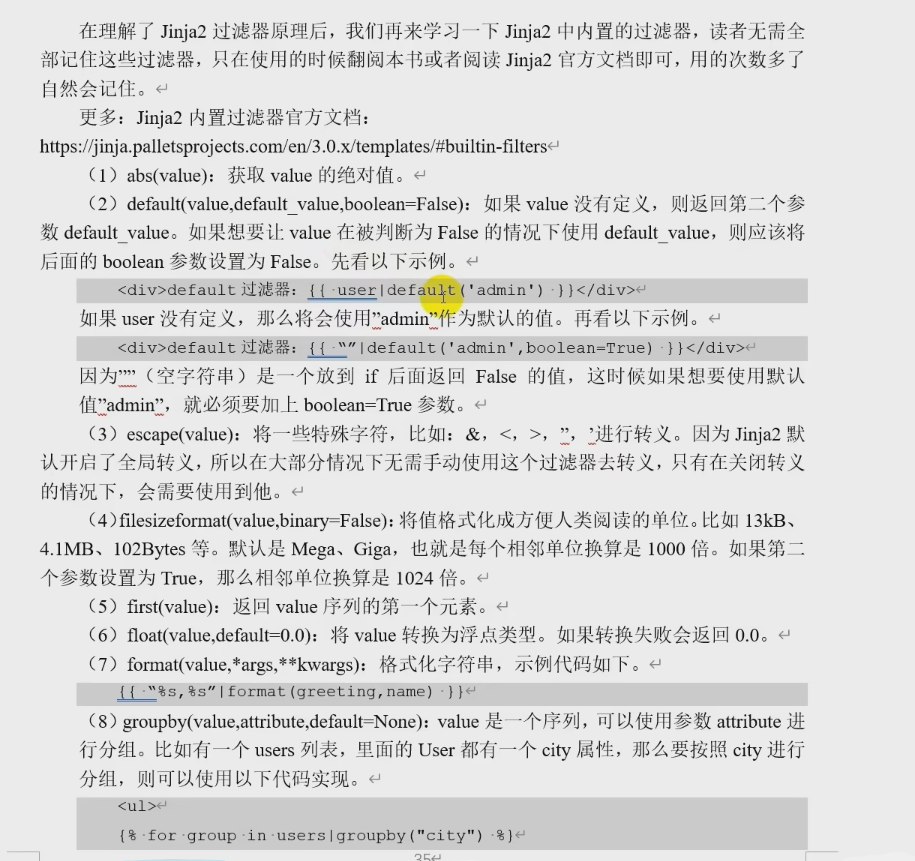


Person是字典



## 2.过滤器





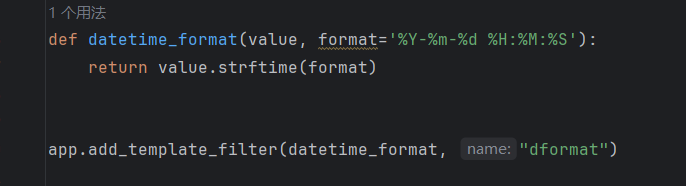
等等，有30多个内置过滤器

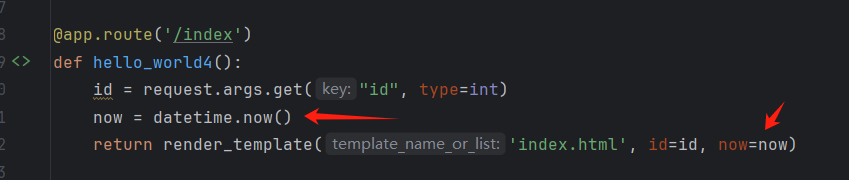


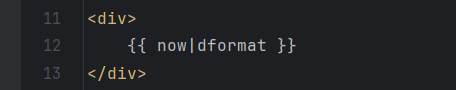


自定义过滤器：

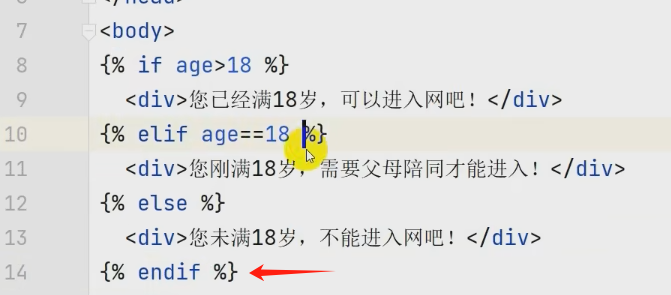








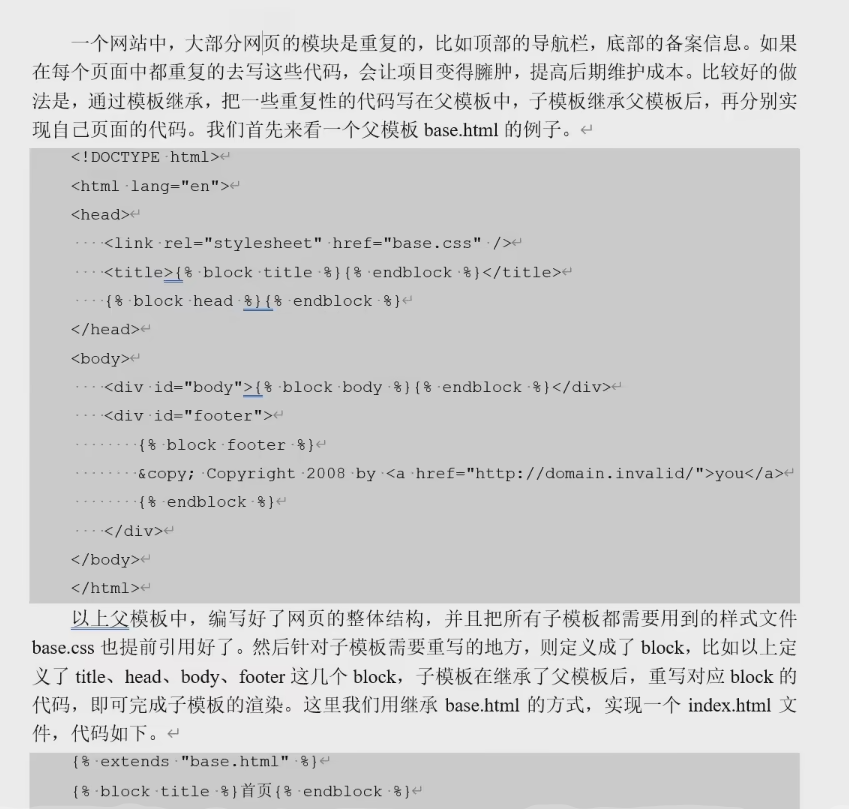
## 3.控制语句





必须全部循环处理，不能中途停止

## 4.模板继承

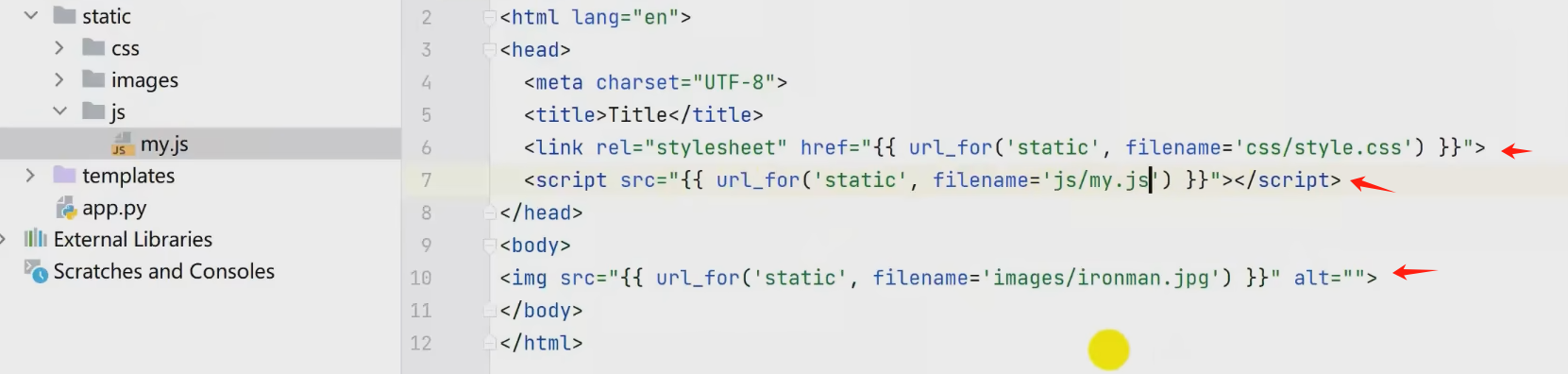








## 5.静态文件



# 数据库

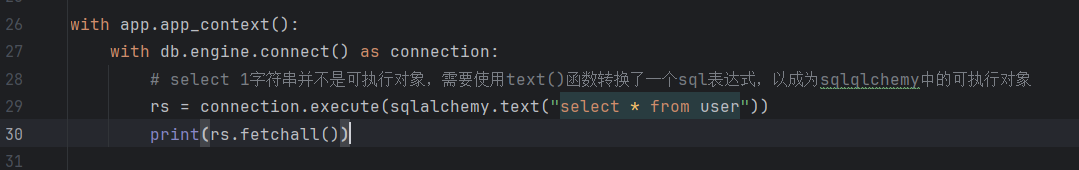
安装pymysql

安装Flash-SQLAlchemy相当于是orm，类似mybatis等





测试连接：



实体模型：

