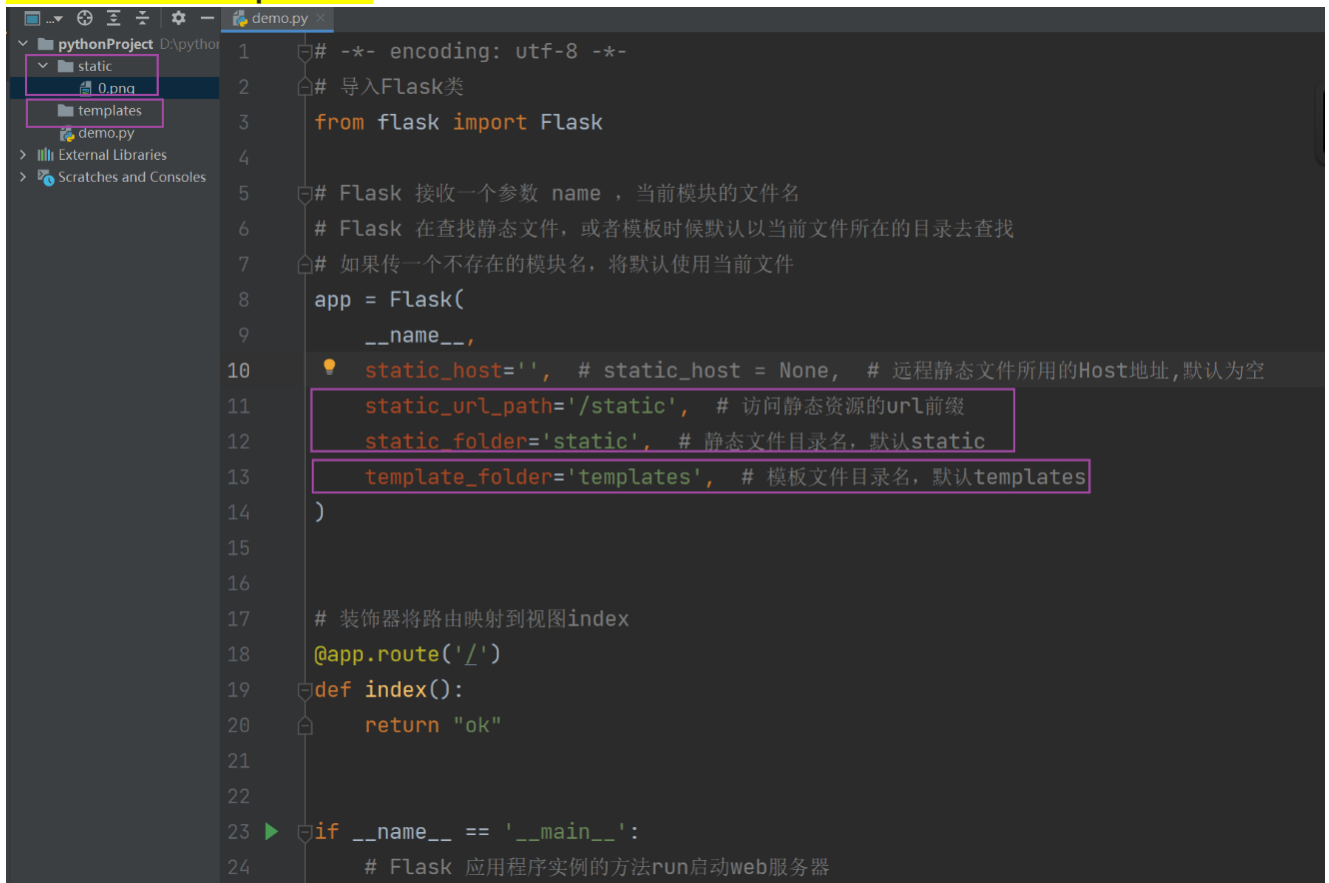


# 1.视图调用模板

首先我们在项目目录下创建一个templates目录统一管理模板文件。

## 注意注册templates



```
1  # -*- encoding: utf-8 -*-
2  # 导入Flask类
3  from flask import Flask
4
5  # Flask 接收一个参数 name，当前模块的文件名
6  # Flask 在查找静态文件，或者模板时候默认以当前文件所在的目录去查找
7  # 如果传一个不存在的模块名，将默认使用当前文件
8  app = Flask(
9      __name__,
10     static_host='', # static_host = None, # 远程静态文件所用的Host地址,默认为空
11     static_url_path='/static', # 访问静态资源的url前缀
12     static_folder='static', # 静态文件目录名,默认static
13     template_folder='templates', # 模板文件目录名,默认templates
14 )
15
16
17 # 装饰器将路由映射到视图index
18 @app.route('/')
19 def index():
20     return "ok"
21
22
23 if __name__ == '__main__':
24     # Flask 应用程序实例的方法run启动web服务器
```

```
# -*- encoding: utf-8 -*-
# 导入Flask类
from flask import Flask

app = Flask(__name__,
            template_folder='templates')

if __name__ == '__main__':
```

```
# Flask 应用程序实例的方法run启动web服务器
app.run(debug=True)
```

在视图中使用 `render_templates` 函数调用模板文件。

在`templates`目录下创建一个模板文件`index.html`

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>首页</title>
</head>
<body>
    <h1>首页</h1>
</body>
</html>
```

## 2.render\_templates 调用模板

---

```
# -*- encoding: utf-8 -*-
# 导入Flask类
from flask import Flask, render_template

app = Flask(__name__,
            template_folder='templates')

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

```
# 视图中调用模板
return render_template('index.html')

if __name__ == '__main__':

    # Flask 应用程序实例的方法run启动web服务器
    app.run(debug=True)
```

← ↻ 🏠 ⓘ 127.0.0.1:5000

# 首页

## 3.模板上下文

```
# -*- encoding: utf-8 -*-
# 导入Flask类
from flask import Flask, render_template

app = Flask(__name__,
            template_folder='templates')

@app.route('/')
def index():
    # render_template 函数第一个参数是模板名称, 后面的参数以
    key=value的形式, 传给模板的变量
    return render_template('index.html', name='python',
                           age=26)
```

```
if __name__ == '__main__':  
  
    # Flask 应用程序实例的方法run启动web服务器  
    app.run(debug=True)
```

```
<!DOCTYPE html>  
<html lang="en">  
<head>  
    <meta charset="UTF-8">  
    <title>首页</title>  
</head>  
<body>  
    <h1>首页-- {{ name }}</h1>  
</body>  
</html>
```

← ↻ 🏠 ⓘ 127.0.0.1:5000

# 首页-- python

注意：jinja2上下文传递，跟django中的是有区别的，django模板可以接收一个字典作为上下文，但是jinja2中只能以 `key=value` 的形式传递。

解决办法：

```
# -*- encoding: utf-8 -*-
# 导入Flask类
from flask import Flask, render_template

app = Flask(__name__,
            template_folder='templates')

@app.route('/')
def index():
    # render_template 函数第一个参数是模板名称, 后面的参数以
    # key=value的形式, 传给模板的变量
    # return render_template('index.html', name='python',
    # age=26)

    context = {
        'name': 'python',
        'age': 26
    }
    # 将上下文定义成一个字典, 在传递上下文的时候, 先将其解包
    return render_template('index.html', **context)

if __name__ == '__main__':

    # Flask 应用程序实例的方法run启动web服务器
    app.run(debug=True)
```

将上下文参数写成字典, 传参时先解包。