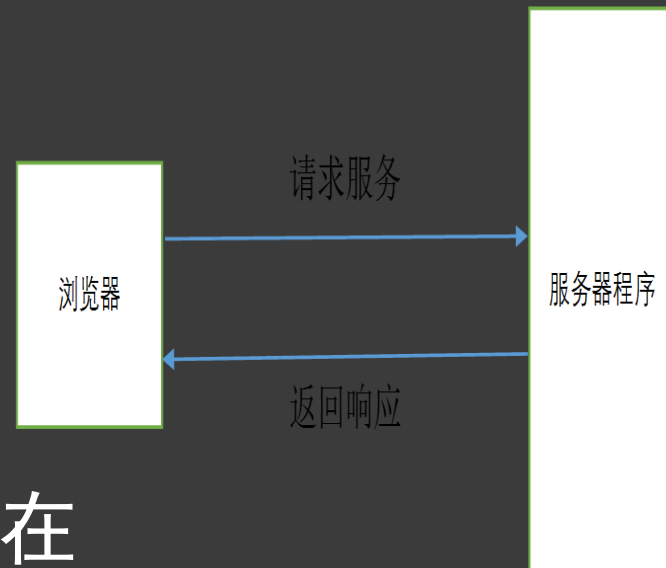


## 9.2 Web应用框架Flask

- Flask是一个使用 Python编写的轻量级 Web 应用框架。其 WSGI工具箱采用 Werkzeug，模板引擎则使用 Jinja2。Flask使用 BSD 授权。
- Flask也被称为“microframework”，因为它使用简单的核心，用扩展增加其他功能。
- Flask是第三方模块，需要先安装才能使用。
- 本章程序适用Flask 0.12版本

# URL分析和对应的处理函数

- ① 运行Python服务程序
- ② 用户在浏览器输入URL访问某个资源
- ③ Flask接受用户请求分析URL
- ④ 为这个URL找到对应的处理函数
- ⑤ 执行函数并生成响应，返回给浏览器
- ⑥ 浏览器接受并解析响应，将信息显示在页面中



# 多个URL对应同一个函数

- ◎ 多个URL可对应同一个函数，这样在浏览器中输入不同的URL返回结果是一样的
- ◎ 在浏览器输入：`http://127.0.0.1:5000`或`http://127.0.0.1:5000/hello`，结果一样

```
from flask import Flask      #导入 flask 模块

app = Flask(__name__)        #创建 flask 对象 app

@app.route("/")              #装饰器，定义目录"/"的操作

@app.route("/hello")

def hello():                  #定义函数，遇到网址目录为"/"或"/hello"执行

    return "Hello World! 这是一个 Web 应用程序"    #返回字符串

if __name__ == "__main__":

    app.run()                  #运行程序
```

# HTTP协议常用请求

- GET: 从服务器程序获取信息

POST: 通常用作向服务器程序发送数据, 一般用  
表单实现

POST请求可能会导致新的资源的建  
立和/或已有资源的修改

# 浏览器和服务端程序交互

- 浏览器和服务端程序交互可以通过get和post方法实现。
- 在装饰器上可以加上http方法，常用的是get和post方法，缺省的是get方法。
- `@app.route("/")`就等同于  
`@app.route("/", methods=["get"])`
- 如需要用两个方法，可写成：
- `@app.route("/", methods=["get", "post"])`
- **get 和 post是站在浏览器角度看的**

# 执行加法程序

```

① @app.route('/add',methods=['GET', 'POST'])
② def add():
③     n1 = int(request.form['n1'])
④     n2 = int(request.form['n2'])
⑤     n3 = n1 + n2
⑥     return ""
⑦     <!DOCTYPE html>
⑧     <html>
⑨     <body>
⑩     <h2>输入两个数求和</h2>
⑪     <form method='post' action='/add'>
⑫     <table>
⑬     <p>使用表单计算两个数的和</p>
⑭     <tr><td>被加数</td><td><input type='text' name='n1' value = ""+str(n1)+" " /></td></tr>
⑮     <tr><td>加数</td><td><input type='text' name='n2' value = ""+str(n2)+" " /></td></tr>
⑯     <tr><td>和</td><td><input type='text' name='n3' value= ""+str(n3)+" " /></td></tr>
⑰     </table>
⑱     <button name="sum" type="submit" value="sum">求和</button>
⑲     </form>
⑳     </body>
㉑     </html>
㉒     ""
㉓ if __name__ == '__main__':
㉔     app.run()
```

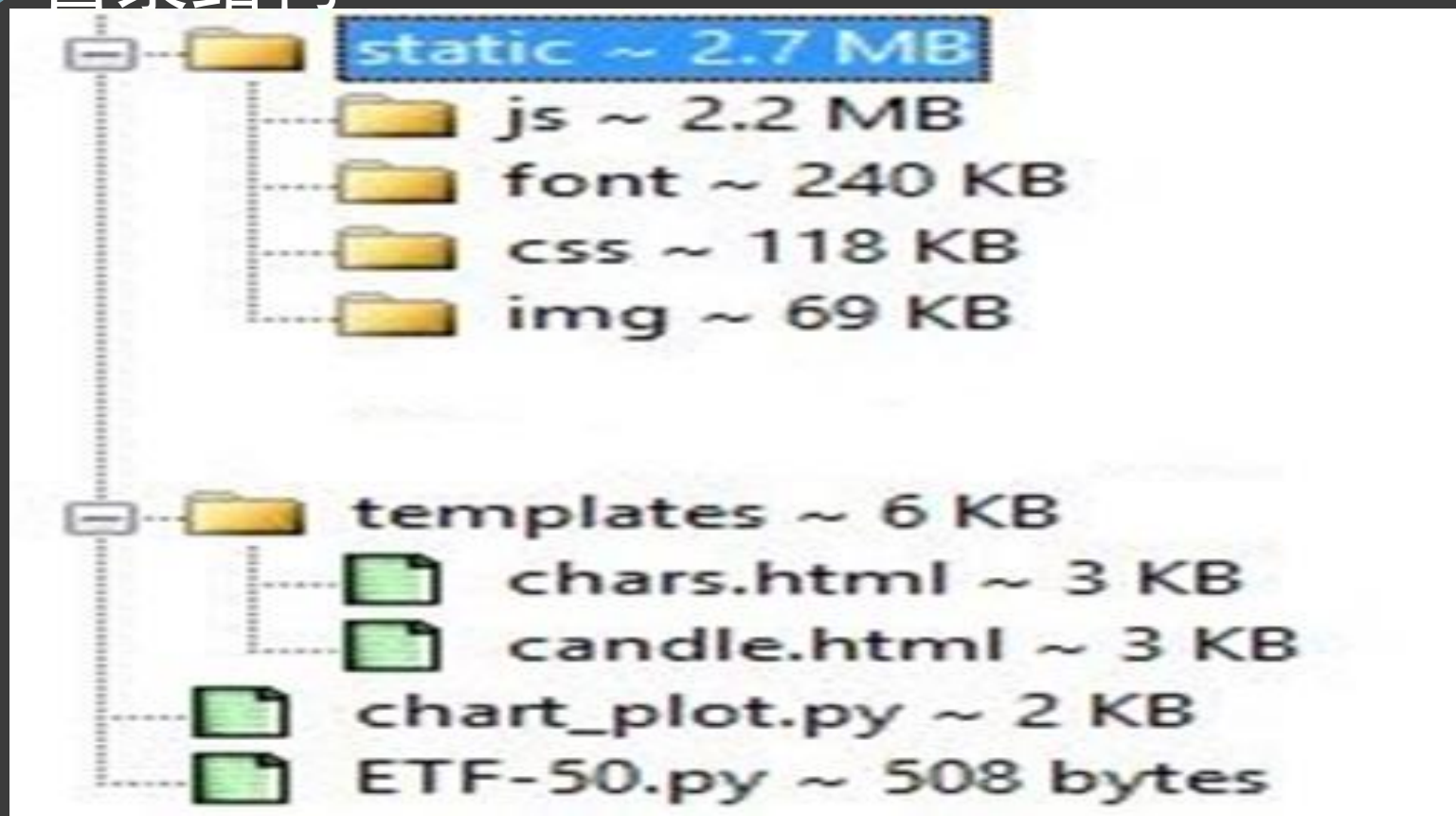
# 模板的使用

- 返回的HTML字符串可以放在一个HTML文件中，这个文件就是模板。
- 目录结构：
- add.py
- templates
- |-----index.html
- |-----result.html

# 在本机构建B/S程序

## -----绘制上证50指数K线图和半年线图

### ◎ 目录结构





# url\_for()函数

- ◎ url\_for()函数获取静态文件的URL
- ◎ url\_for('static', filename='css/bootstrap.css')
- ◎ 表示文件的URL是:
- ◎ **`/static/css/bootstrap.css`**
- ◎ 根目录就是Python主文件所在的目录

# 浏览器显示

——http://127.0.0.1:5000

