

# Python程序设计

## 第九讲 Web应用和网络爬虫 开发Web应用程序



张 华

WHU

# Web应用程序开发概述

## Web应用程序

- ✿ Web应用程序是一种可以通过浏览器访问的应用程序，程序的最大好处是用户很容易访问应用程序，终端只需要有浏览器即可，不需要再安装其他软件。

## 主要采用浏览器/服务器架构

- ✿ 简称B/S（Browser/Server）
- ✿ 它能够很好地应用在广域网上，成为越来越多企业的选择。

# Web应用程序开发概述

## Web应用程序的运行过程

- ✿ 运行用Python编写的服务程序；
- ✿ 用户在浏览器输入URL访问某个资源；
- ✿ 服务程序接受用户请求分析URL；
- ✿ 为这个URL找到对应的处理函数；
- ✿ 执行函数并生成响应，返回给浏览器；
- ✿ 浏览器接受并解析响应，将信息显示在页面中。



# 统一资源定位符

## URL (Uniform Resource Locator)

✿ 一般格式如下：

**protocol** :// **hostname[:port]** / **path** / [;parameters][?query]#fragment

✿ 由三部分组成：

- **protocol**: 第一部分是协议，例如百度使用的就是**https**协议；
- **hostname[:port]**: 第二部分是主机名，端口号为可选参数；
  - 百度的主机名就是**www.baidu.com**
  - **Python**的**Flask**服务程序默认端口号为**5000**
- **path**: 第三部分是主机资源的具体地址，如目录和文件名等。

# 创建网页的语言

## 超级文本标记语言 (HTML)

✿ HTML是为创建网页而设计的一种标记语言。

➤ 网页的本质就是超级文本标记语言，通过结合使用其他的技术，可以创造出功能强大的网页。

✿ 一个HTML文件对应一个网页，以.htm或.html为扩展名。

➤ 可以使用任何能够生成TXT类型的文本编辑器来产生超文本标记文件，然后修改文件扩展名即可。

# HTML

## HTML常用标签

标 签	属 性	用 途
HTML		整个HTML文档
HEAD		文档头
TITLE		文档标题
BODY	background、bgcolor	文档主体内容
H1、H2、H3等		标题
I, EM		斜体强调样式
B, STRONG		加粗强调样式
PRE		格式化文本
P, SPAN, DIV		段落, 块元素, 行内元素
BR		换行
A	href	超链接
IMG	src, width, height	图片
TABLE	width, border	表格
TR		表格中的一行
TH, TD		表头/单元格
OL, UL		有序/无序列表
LI		列表
DL		描述列表
INPUT	name	用户输入域
SELECT	name	下拉菜单

# 层叠样式表

## CSS (Cascading Style Sheets)

- ✿ **CSS**为HTML标记语言提供了一种样式描述，定义了其中元素的显示方式。
- ✿ **CSS**提供了丰富的文档样式外观。
  - **CLASS**属性允许向一组在**CLASS**属性上具有相同值的元素应用声明。使用**CLASS**属性来分类元素，在样式表中创建规则来引用**CLASS**属性的值，然后浏览器自动将这些属性应用到该组元素。**CLASS**选择器以标志符“.”开头，用于指示后面是哪种类型的选择器。
  - **ID**属性操作类似于**CLASS**属性，但有一点重要的不同之处：**ID**属性的值在整篇文档中必须是唯一的。这使得**ID**属性可用于设置单个元素的样式规则。包含**ID**属性的选择器称为**ID**选择器。**ID**选择器的标志符是符号“#”。

# Web应用框架Flask

## Flask



- ✿ <https://palletsprojects.com/p/flask/>
- ✿ Flask是一个使用 Python编写的轻量级 Web应用框架。
- ✿ 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。
  - Anaconda3已集成Flask，不需要自己安装。
  - 如果你的Python环境没有集成Flask，则用下面的命令安装
  - `pip install flask`



# 基于Flask创建Web应用

## 基于Flask创建一个极简单的服务器程序

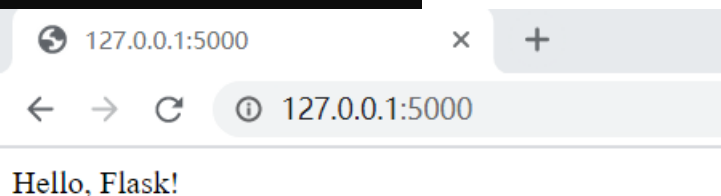
✿ 接收用户请求的URL，为这个URL找到对应的处理函数。

```
# hello.py
from flask import Flask
app = Flask(__name__)           # 创建Flask对象

@app.route('/')                 # 通过装饰器定义请求URL目录 "/" 的操作
def hello_flask():             # 定义函数对URL目录 "/" 请求进行处理
    return 'Hello, Flask!'     # 返回字符串（网页的HTML代码）

if __name__ == '__main__':
    app.run()                  # 运行应用程序
```

```
* Serving Flask app "hello" (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production
  Use a production WSGI server instead.
* Debug mode: off
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```



# 基于Flask创建Web应用

## 浏览器和服务器程序交互

✿ 通过HTTP协议的GET和POST方法实现。

➤ 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。

✿ 在装饰器上可以加上HTTP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'])`

# 基于Flask创建Web应用

 **案例：从浏览器输入2个数，求它们的和。**

 **创建前端网页**

➤ **index.html**

输入两个数求和

使用表单计算两个数的和

被加数	<input type="text"/>
加数	<input type="text"/>
<input type="button" value="求和"/>	

输入两个数求和

使用表单计算两个数的和

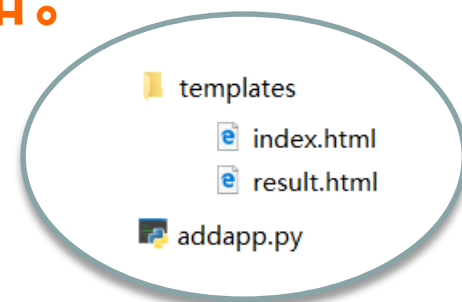
被加数	<input type="text" value="123"/>
加数	<input type="text" value="456"/>
<input type="button" value="求和"/>	

➤ **result.html**

输入两个数求和

使用表单计算两个数的和

被加数	123
加数	456
和	579



# 基于Flask创建Web应用

案例：从浏览器输入2个数，求它们的和。

创建前端网页：index.html

```
<html>
<head>
  <title>Add</title>
  <meta charset="UTF-8">
  <meta name="description" content="" />
  <meta name="keywords" content="" />
</head>
<body>
  <h2>输入两个数求和</h2>
  <form method='post' action='/add'>
    <table>
      <p>使用表单计算两个数的和</p>
      <tr><td>被加数</td><td><input type='text' name='n1' /></td></tr>
      <tr><td>加数</td><td><input type='text' name='n2' /></td></tr>
    </table>
    <button name="sum" type="submit" value="sum">求和</button>
  </form>
</body>
</html>
```

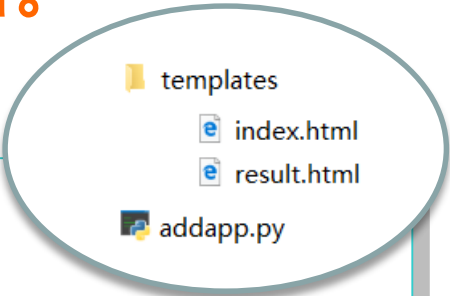
templates  
index.html  
result.html  
addapp.py

# 基于Flask创建Web应用

案例：从浏览器输入2个数，求它们的和。

创建前端网页：result.html

```
<html>
  <body>
    <h2>输入两个数求和</h2>
    <table>
      <p>使用表单计算两个数的和</p>
      <tr><td>被加数</td><td> {{n1}} </td></tr>
      <tr><td>加数</td><td> {{n2}} </td></tr>
      <tr><td>和</td><td> {{n3}} </td></tr>
    </table>
  </body>
</html>
```



- templates
  - index.html
  - result.html
- addapp.py

# 基于Flask创建Web应用

案例：从浏览器输入2个数，求它们的和。

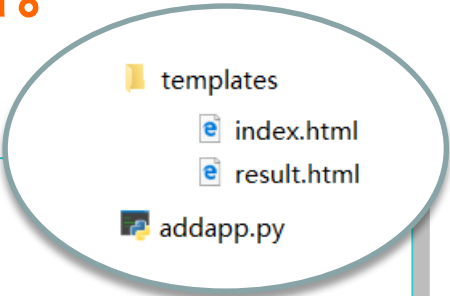
创建后端Python程序：addapp.py

```
from flask import Flask, request
import flask

app = Flask(__name__)

@app.route('/', methods=['GET', 'POST'])
@app.route('/add', methods=['GET', 'POST'])
def add():
    if flask.request.method == 'GET':
        return flask.render_template('index.html')
    else:
        n1 = int(request.form['n1'])
        n2 = int(request.form['n2'])
        s = n1 + n2
        return flask.render_template('result.html', n1=n1, n2=n2, n3=s)

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



templates

- index.html
- result.html

addapp.py