

# 第九章

## WEB应用程序开发及网 络爬虫

CS, ZJU  
2018年12月

# Overview

- ◎ Web应用程序开发概述
- ◎ Web应用框架Flask
- ◎ 文件模板
- ◎ 云端部署Web应用程序
- ◎ 网络爬虫

# 9.1 Web应用程序开发概述

- 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。
- 这种程序开发也称为App开发
- Web应用程序现在主要采用浏览器/服务器架构(Browser/Server,简称B/S)，它能够很好地应用在广域网上，成为越来越多企业的选择。

# 极简Web应用程序

服务器程序

浏览器

The image shows a web browser window on the left and a Python IDE window on the right. The browser window displays 'Hello World! 这是一个web应用程序' at the URL '127.0.0.1:5000'. The Python IDE window shows the source code for 'hello.py' and the output of the Python shell.

```
from flask import Flask
app = Flask(__name__)

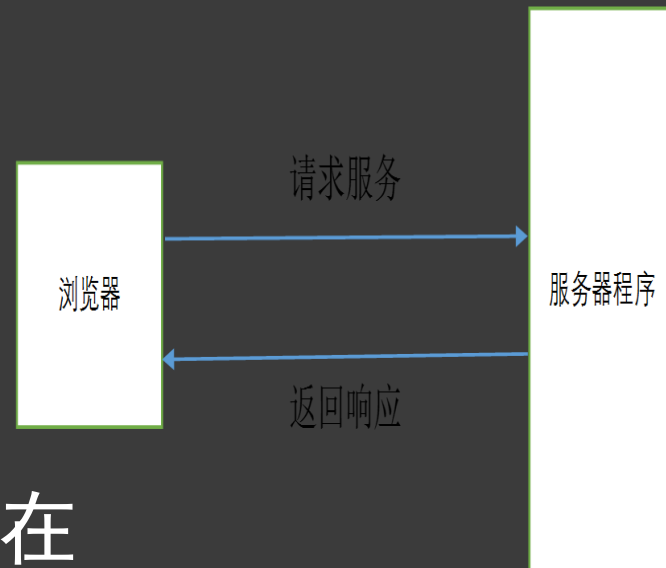
@app.route("/")
def hello():
    return "Hello World! \n 这是一个"

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

Python 3.6.3 (v3.6.3:2c5fed8, Oct 3 2017, 18:11:49) [MSC v.1900 64 bit (AMD64)] on win32  
Type "copyright", "credits" or "license()" for more information.  
>>>  
== RESTART: C:/Users/lenovo/AppData/Local/Programs/Python/Python36/hello.py ==  
\* Running on http://127.0.0.1:5000 / (Press CTRL+C to quit)

# Web应用程序的运行过程

- ① 运行用Python编写的服务程序
- 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



# URL统一资源定位符

URL就是统一资源定位符(Uniform Resource Locator)，它的一般格式如下：

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

URL的格式由三部分组成：

- (1)protocol：第一部分就是协议，例如百度使用的就是https协议；
- (2)hostname[:port]：第二部分就是主机名(端口号为可选参数)，  
Flask默认的端口号为5000，  
百度的主机名就是:www.baidu.com
- (3)path：第三部分就是主机资源的具体地址，如目录和文件名等。

# 超级文本标记语言（HTML）

- ◎ 超级文本标记语言（HTML）是为网页创建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。

# 返回网页的服务器程序

```
from flask import Flask
app = Flask(__name__)
@app.route("/")
def hello():
    return '''<!DOCTYPE HTML>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<title>div 标签</title>
</head>
<body>
    <h2>热门课程排行榜</h2>
    <ol>
        <li>Python 程序设计 </li>
        <li>零基础学习 html</li>
        <li>javascript 全攻略</li>
    </ol>
    <h2>最新课程排行</h2>
    <ol>
        <li>版本管理工具介绍-Git 篇 </li>
        <li>Canvas 绘图详解</li>
        <li>Pandas 数据分析</li>
    </ol>
</body>
</html>'''
if __name__ == "__main__":
    app.run()
```



# 程序运行

div标签

127.0.0.1:5000

## 热门课程排行榜

1. Python程序设计
2. 零基础学习html
3. javascript全攻略

## 最新课程排行

1. 版本管理工具介绍—Git篇
2. Canvas绘图详解
3. Pandas数据分析

Thonny - C:\Users\lenovo\Desktop\Python程序设计\第9章\9-2.py @ 31:1

File Edit View Run Device Tools Help

9-2.py

```
1 from flask import Flask
2 app = Flask(__name__)
3 @app.route("/")
4 def hello():
5     return '''<!DOCTYPE HTML>
6     <html>
7     <head>
8     <meta http-equiv="Content-Type" content="text/html; char
9     <title>div标签</title>
10    </head>
11    <body>
12        <h2>热门课程排行榜</h2>
13        <ol>
14            <li>Python程序设计 </li>
```

Shell

Python 3.7.2 (bundled)

```
>>> %Run 9-2.py
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
127.0.0.1 - - [21/Apr/2019 20:19:05] "GET / HTTP/1.1" 200
-
127.0.0.1 - - [21/Apr/2019 20:19:05] "GET / HTTP/1.1" 200
-
127.0.0.1 - - [21/Apr/2019 20:19:05] "GET /favicon.ico HTTP/1.1" 404 -
```

# 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

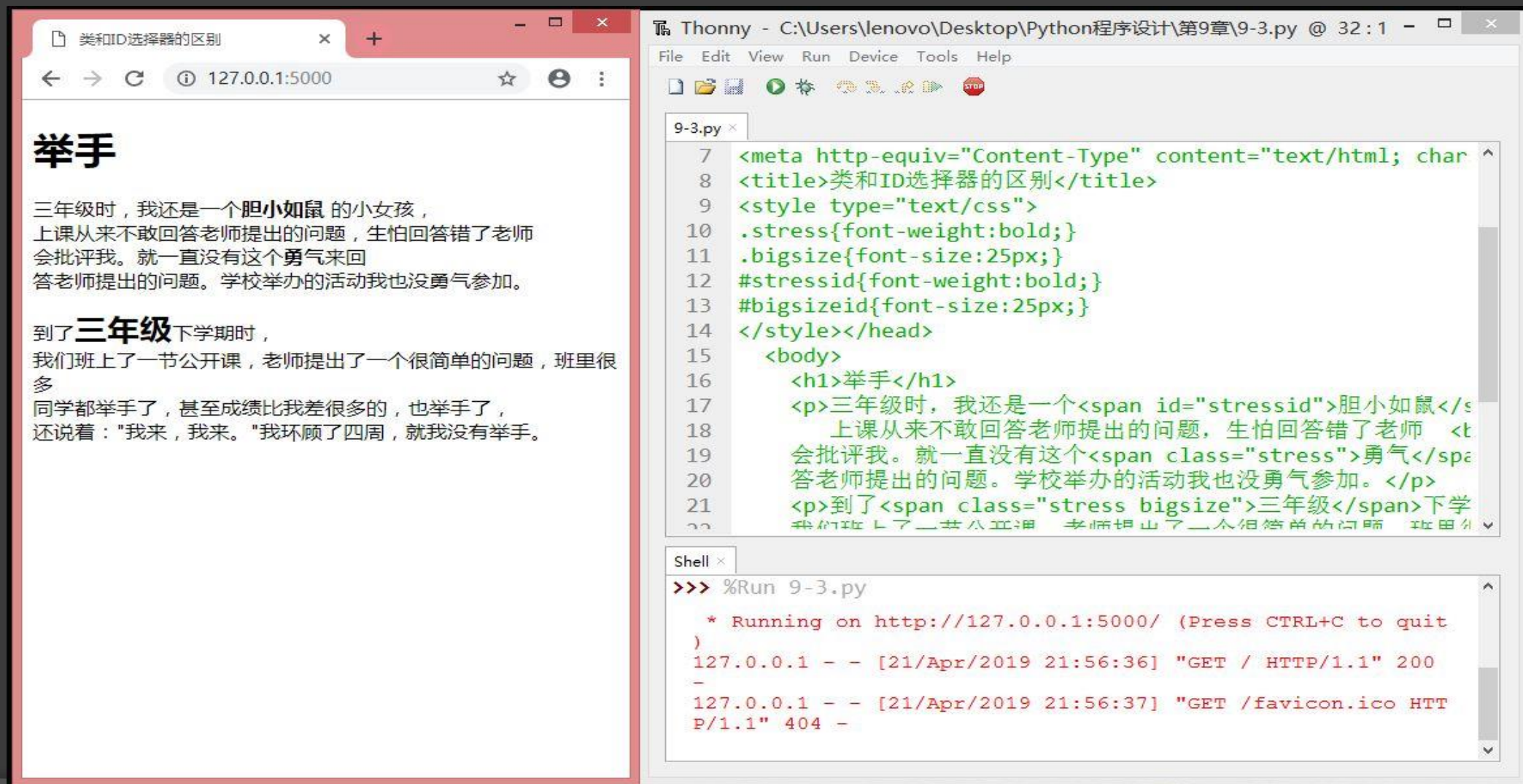
# 层叠样式表(CSS)

- ◎ 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“#”。

# 带CSS样式的服务器程序

```
from flask import Flask
app = Flask(__name__)
@app.route("/")
def hello():
    return '''<!DOCTYPE HTML>
<html><head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<title>类和 ID 选择器的区别</title>
<style type="text/css">
.stress{font-weight:bold;}
.bigsized{font-size:25px;}
#stressid{font-weight:bold;}
#bigsizedid{font-size:25px;}
</style></head>
<body>
    <h1>勇气</h1>
    <p>三年级时，我还是一个<span id="stressid">胆小如鼠</span>
的小女孩， <br/>
        上课从来不敢回答老师提出的问题，生怕回答错了老师 <br/>
        会批评我。就一直没有这个<span class="stress">勇气</span>来回 <br/> 答老师提出的问题。
学校举办的活动我也没勇气参加。</p> <br/>
    <p>到了<span class="stress bigsize">三年级</span>下学期时， <br/> 我们班上了一节公
开课，老师提出了一个很简单的问题，班里很多 <br/>
    同学都举手了，甚至成绩比我差很多的，也举手了， <br/>
    还说着："我来，我来。"我环顾了四周，就我没有举手。</p>
</body></html>'''
if __name__ == "__main__":
    app.run()
```

# 返回带CSS的网页



The image shows a web browser window on the left and a Python IDE (Thonny) window on the right. The browser window displays a web page titled "类和ID选择器的区别" (Difference between Class and ID Selectors) at the address 127.0.0.1:5000. The page content includes a heading "举手" (Raise Hand) and two paragraphs of text. The Python IDE window shows the source code of the web page (9-3.py) and the output of the shell.

**Web Browser Window:**

类和ID选择器的区别

127.0.0.1:5000

## 举手

三年级时，我还是一个胆小如鼠的小女孩，上课从来不敢回答老师提出的问题，生怕回答错了老师会批评我。就一直没有这个勇气来回答老师提出的问题。学校举办的活动我也没勇气参加。

到了**三年级**下学期时，我们班上了一节公开课，老师提出了一个很简单的问题，班里很多同学都举手了，甚至成绩比我差很多的，也举手了，还说着："我来，我来。"我环顾了四周，就我没有举手。

**Python IDE Window:**

Thonny - C:\Users\lenovo\Desktop\Python程序设计\第9章\9-3.py @ 32:1

```
7 <meta http-equiv="Content-Type" content="text/html; char
8 <title>类和ID选择器的区别</title>
9 <style type="text/css">
10 .stress{font-weight:bold;}
11 .bigsize{font-size:25px;}
12 #stressid{font-weight:bold;}
13 #bigsizeid{font-size:25px;}
14 </style></head>
15 <body>
16 <h1>举手</h1>
17 <p>三年级时，我还是一个<span id="stressid">胆小如鼠</s
18 上课从来不敢回答老师提出的问题，生怕回答错了老师 <t
19 会批评我。就一直没有这个<span class="stress">勇气</spa
20 答老师提出的问题。学校举办的活动我也没勇气参加。</p>
21 <p>到了<span class="stress bigsize">三年级</span>下学
22 我们班上了一节公开课，老师提出了一个很简单的问题，班里很
```

Shell

```
>>> %Run 9-3.py
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
127.0.0.1 -- [21/Apr/2019 21:56:36] "GET / HTTP/1.1" 200
127.0.0.1 -- [21/Apr/2019 21:56:37] "GET /favicon.ico HTTP/1.1" 404 -
```

# 常用选择器例子

选择器	例子	解释
.class	.stress	选择class=stress的所有节点
#id	#stressid	选择id=stressid的所有节点
*	*	选择的所有节点
element	span	选择所有span节点
element element	div p	选择div节点内部的所有p节点
element,element	div,p	选择所有div节点和所有p节点
element>element	div>p	选择父节点为div节点的所有p节点

# HTML DOM树

