

WEB SERVICE

Web 服务器项目

DAY02

内容

上午	09:00 ~ 09:30	作业讲解回顾
	09:30 ~ 10:20	HTML文件
	10:30 ~ 11:20	
	11:30 ~ 12:00	
下午	14:00 ~ 14:50	HTTP协议
	15:00 ~ 15:50	
	16:00 ~ 16:50	
	17:00 ~ 17:30	总结和答疑



HTML文件

HTML文件

HTML文件

HTML文件简介

HTML文件结构

HTML文件元素

HTML文件



HTML文件简介

- HTML是用来制作超文本文档的一种标记语言。用其编写的超文本文档称为HTML文档，可以运行在各种操作系统的平台上。
- 自1990年以来，HTML一直都是www的信息表示语言，用于描述homepage的格式设计和其他homepage的连接信息。使用HTML语言描述的文件，需要通过浏览器显示效果，是一种最为基础的语言。
- 所谓超文本，因为它可以加入图片、声音、动画、影视等内容，并可以从一个文件跳转到另外一个文件，与世界各地主机上的文件连接。



HTML文件简介(续1)

- HTML的作用是用来展示页面的表现形式，如页面的布局、页面的颜色、页面中的内容等。
- 所谓标记，是因为他采用了一系列的指令和符号来控制输出效果，这些指令符号用<标签名字和属性>来表示。
- HTML的焦点主要集中在信息如何显示上，而不是信息的内容和结构。
- 可以直接使用记事本编写HTML文档，存成.htm或者.html即可。网站是由很多个html文件组成的一个集合。
- 首页文件名默认为index.htm或者index.html。



HTML文件结构

- `<html>`
- `<head>`
- `<title>...</title>`
- `<meta>`
- `</head>`
- `<body>`
- html文件的正文
- `</body>`
- `</html>`

- 元素是html语言的基本部分,元素总是成对出现的.每一对元素都有开始的标记,也有结束的标记.元素的标记要用尖括号括起来.结束的标记总是在开始的标记前加上/。

HTML文件结构(续1)

- 第一张网页如下：
 - `<html>`
 - `<head>`
 - `<title>the first page</title>`
 - `</head>`
 - `<body>`
 - `this is my first homepage.`
 - `</body>`
 - `</html>`
- 将该文件保存为first.html



HTML文件元素

- HTML元素可以有自己相关属性，每一个属性可以有我们网页编制者赋一定的值。元素属性出现在元素的<>内，并且和元素之间有一个空格分割，属性值使用“属性”引起来。

- <html>
- <head>
- <title>the first page</title>
- </head>
- <body>
- <p align=" center" >this is my first homepage.</p>
- </body>
- </html>



HTML文件元素(续1)

- 将文件保存为second.html
- <head>元素出现在文档的开头部分。<head>和</head>之间的内容不会出现在浏览器的文档窗口显示，但是其间的元素有特殊重要的意义。
- <title>元素定义html文档的标题。<title>和</title>之间的内容将显示在浏览器窗口的标题栏。
- <body>元素表明是html文档的主体部分。在<body>和</body>之间会有很多其他的元素，这些元素和元素的属性构成html文档的主体部分。



HTML文件元素(续2)

- 标题 (Heading) 设置
- 标题是通过 `<h1>` - `<h6>` 标签进行定义的.`<h1>` 定义最大的标题。 `<h6>` 定义最小的标题。
 - `<html>`
 - `<body>`
 - `<h1>`这是标题 1`</h1>`
 - `<h2>`这是标题 2`</h2>`
 - `<h3>`这是标题 3`</h3>`
 - `<h4>`这是标题 4`</h4>`
 - `<h5>`这是标题 5`</h5>`
 - `<h6>`这是标题 6`</h6>`
 - `</body>`
 - `</html>`



HTML文件元素(续3)

- HTML注释
- 可以将注释插入 HTML 代码中，这样可以提高其可读性，使代码更易被人理解。浏览器会忽略注释，也不会显示它们。
- 注释写法如下：
 - `<!-- 这是一个注释 -->`
- HTML水平线
- `<hr>` 标签在 HTML 页面中创建水平线。
- hr 元素可用于分隔内容。
- 代码如下：
 - `<p>这是一个段落。 </p>`
 - `<hr>`
 - `<p>这是一个段落。 </p>`



HTML文件元素(续4)

- HTML 图像- 图像标签 (``) 和源属性 (`Src`)
- 在 HTML 中，图像由 `` 标签定义。
- `` 是空标签，意思是说，它只包含属性，并且没有闭合标签。
- 要在页面上显示图像，你需要使用源属性 (`src`)。 `src` 指 "source"。源属性的值是图像的 URL 地址。
- 定义图像的语法是：
- ``
- URL 指存储图像的位置。如果名为 "pulpit.jpg" 的图像位于 `www.runoob.com` 的 `images` 目录中，那么其 URL 为 `http://www.runoob.com/images/pulpit.jpg`。



HTML文件元素(续5)

- 无序列表
- 无序列表是一个项目的列表，此列项目使用粗体圆点（典型的小黑圆圈）进行标记。
- 无序列表始于 `` 标签。每个列表项始于 ``。
 - ``
 - `Coffee`
 - `Milk`
 - ``
- 列表项内部可以使用段落、换行符、图片、链接以及其他列表等等。



HTML文件元素(续6)

- 有序列表
- 同样，有序列表也是一列项目，列表项目使用数字进行标记。
- 有序列表始于 `` 标签。每个列表项始于 `` 标签。
 - ``
 - `Coffee`
 - `Milk`
 - ``
- 列表项内部可以使用段落、换行符、图片、链接以及其他列表等等。



HTML文件元素(续7)

- 自定义列表
- 自定义列表不仅仅是一列项目,而是项目及其注释的组合。
- 自定义列表以 `<dl>` 标签开始。每个自定义列表项以 `<dt>` 开始。每个自定义列表项的定义以 `<dd>` 开始。
 - `<dl>`
 - `<dt>Coffee</dt>`
 - `<dd>Black hot drink</dd>`
 - `<dt>Milk</dt>`
 - `<dd>White cold drink</dd>`
 - `</dl>`
- 定义列表的列表项内部可以使用段落、换行符、图片、链接以及其他列表等等。



HTTP协议

HTTP协议

HTTP协议

HTTP简介

HTTP报文

HTTP报文结构

HTTP协议



HTTP简介

- 什么是HTTP

- 超文本传输协议 (HTTP , HyperText Transfer Protocol) 是互联网上应用最为广泛的一种网络协议。所有的WWW文件都必须遵守这个标准。设计HTTP最初的目的是为了提供一种发布和接收HTML页面的方法。1960年美国人 Ted Nelson构思了一种通过计算机处理文本信息的方法，并称之为超文本 (hypertext) ,这成为了HTTP超文本传输协议标准架构的发展根基。
- Ted Nelson组织协调万维网协会 (World Wide Web Consortium) 和互联网工程工作小组 (Internet Engineering Task Force) 共同合作研究，最终发布了一系列的RFC，其中著名的RFC 2616定义了HTTP 1.1。



HTTP简介(续1)

• HTTP协议的特点

- 1. 支持客户/服务器模式。
- 2. 简单快速：客户向服务器请求服务时，只需传送请求方法和路径。请求方法常用的有GET、HEAD、POST。每种方法规定了客户与服务器联系的类型不同。由于HTTP协议简单，使得HTTP服务器的程序规模小，因而通信速度很快。
- 3. 灵活：HTTP允许传输任意类型的数据对象。正在传输的类型由Content-Type加以标记。



HTTP简介(续2)

- 4. 无连接：无连接的含义是限制每次连接只处理一个请求。服务器处理完客户的请求，并收到客户的应答后，即断开连接。采用这种方式可以节省传输时间。
- 5. 无状态：HTTP协议是无状态协议。无状态是指协议对于事务处理没有记忆能力。缺少状态意味着如果后续处理需要前面的信息，则它必须重传，这样可能导致每次连接传送的数据量增大。另一方面，在服务器不需要先前信息时它的应答就较快。



HTTP简介(续3)

• 什么是超文本(HyperText) ?

- 包含有超链接(Link)和各种多媒体元素标记(Markup)的文本。这些超文本文件彼此链接, 形成网状(Web), 因此又被称为网页(Web Page)。这些链接使用**URL**表示。最常见的超文本格式是超文本标记语言HTML。

• 什么是URL ?

- URL即统一资源定位符(**U**niform **R**esource **L**ocator), 用来唯一地标识万维网中的某一个文档。URL由协议、主机和端口(默认为80)以及文件名三部分构成。如:

http://www.aqtc.edu.cn:80/news/index.html

The diagram shows the URL http://www.aqtc.edu.cn:80/news/index.html with three brackets underneath. The first bracket is under 'http://' and points to the label '协议://'. The second bracket is under 'www.aqtc.edu.cn:80' and points to the label '主机:端口(80)'. The third bracket is under '/news/index.html' and points to the label '文件名及其路径'.

协议:// 主机:端口(80) 文件名及其路径



HTTP简介(续4)

- HTML 统一资源定位器(Uniform Resource Locators)。
- URL 是一个网页地址。Web浏览器通过URL从Web服务器请求页面。
- URL可以由字母组成，如"runoob.com"，或互联网协议（IP）地址：192.68.20.50。大多数人进入网站使用网站域名来访问，因为名字比数字更容易记住。
- 当您点击 HTML 页面中的某个链接时，对应的 <a> 标签指向万维网上的一个地址。
- 一个统一资源定位器(URL) 用于定位万维网上的文档。
- 一个网页地址实例：
<http://www.runoob.com/html/html-tutorial.html>



HTTP简介(续5)

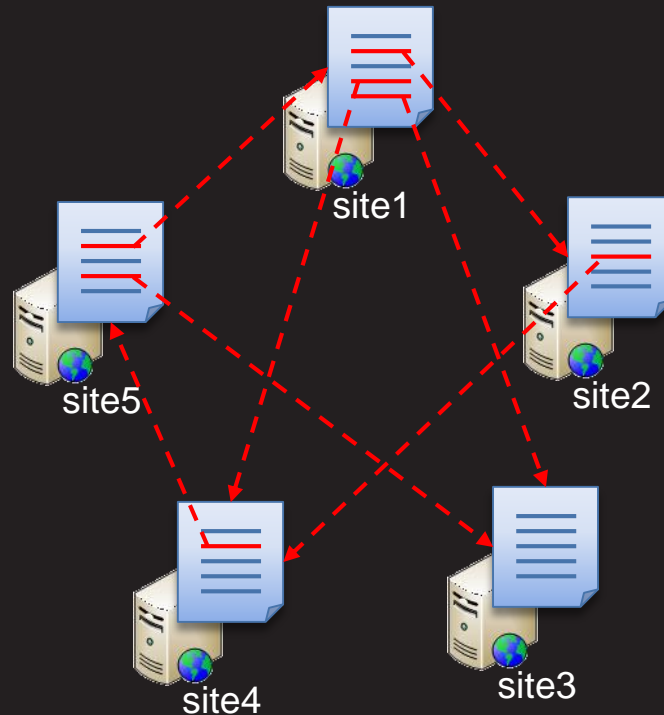
- 语法规则:
 - scheme://host.domain:port/path/filename
- 说明:
 - scheme - 定义因特网服务的类型。最常见的类型是 http
 - host - 定义域主机 (http 的默认主机是 www)
 - domain - 定义因特网域名 , 比如 runoob.com
 - :port - 定义主机上的端口号 (http 的默认端口号是 80)
 - path - 定义服务器上的路径 (如果省略 , 则文档必须位于网站的根目录中) 。
 - filename - 定义文档/资源的名称



HTTP简介(续6)

- 什么是超文本传输协议HTTP？

- 是一种按照URL指示，将超文本文档从一台主机(Web服务器)传输到另一台主机(浏览器)的应用层协议，以实现超链接的功能。



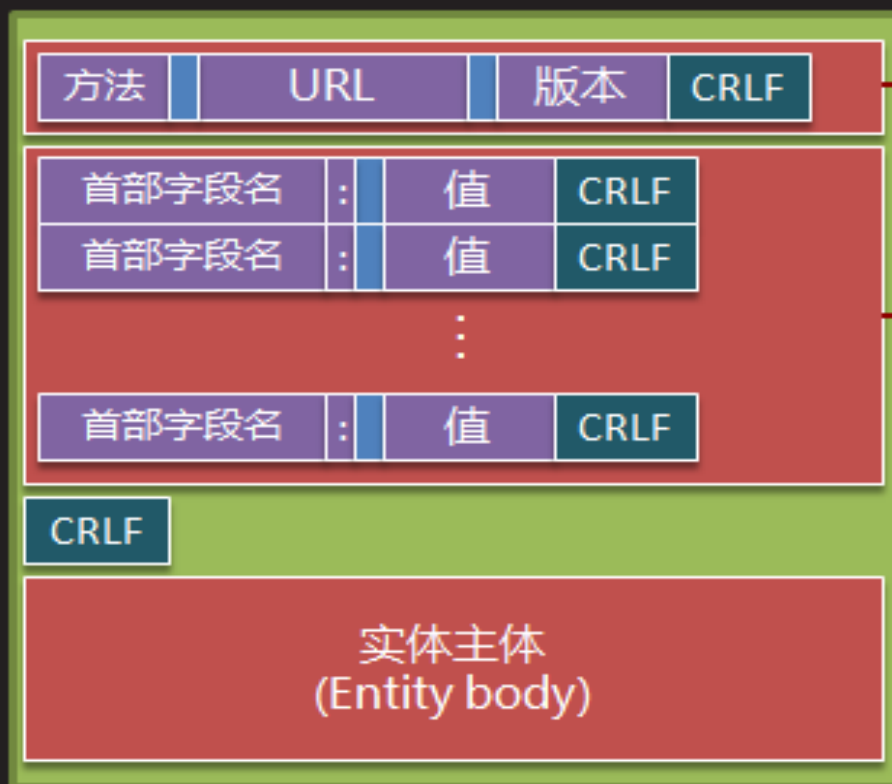
HTTP报文

- HTTP报文是在HTTP应用程序之间发送的数据块。这些数据块以一些文本形式的元信息开头，这些信息描述了报文的内容和含义，后面跟着可选的数据部分。这些报文在客户端、服务器和代理之间流动。
- HTTP使用术语流入和流出来描述事物处理的方向。报文流入源端服务器，工作完成之后，会流回用户的Agent代理中。
- HTTP报文是简单的格式化数据块。每条报文都包含一条来自客户端的请求，或者一条来自服务器的响应。他们有三个部分组成：对报文进行描述的起始行、包含属性的首部块，以及可选的、包含数据的主体部分。



HTTP报文结构

- 请求报文
 - 即从客户端(浏览器)向Web服务器发送的请求报文。
报文的所有字段都是ASCII码



请求行 如: GET /index.html HTTP/1.1

首部行:用来说明浏览器、服务器或报文主体的一些信息。如:

Host: www.sxtyu.com
Connection: close
User-Agent: Mozilla/5.0
Accept-Language: cn



空格 CRLF 回车换行

HTTP报文结构(续1)

- 响应报文
 - 即从Web服务器到客户机(浏览器)的应答。报文的所有字段都是ASCII码。



HTTP报文结构(续2)

- 请求报文中的方法
 - 方法(Method)是用来告知服务器要做什么,也就是一些命令。请求报文中的操作有:

方法(操作)	含义	方法(操作)	含义
GET	请求读取一个Web页面	HEAD	请求读取一个Web页面的首部
POST	附加一个命名资源(如Web页面)	PUT	请求存储一个Web页面
DELETE	删除Web页面	TRACE	用于测试,要求服务器送回收到的请求
CONNECT	用于代理服务器	OPTION	查询特定选项



HTTP报文结构(续3)

- 响应报文中的状态码
 - 状态码(Status-Code)是响应报文状态行中包含的一个3位数字，指明特定的请求是否被满足，如果没有满足，原因是什么。状态码分为以下五类：

状态码	含义	例子
1xx	通知信息	100=服务器正在处理客户请求
2xx	成功	200=请求成功(OK)
3xx	重定向	301=页面改变了位置
4xx	客户错误	403=禁止的页面；404=页面未找到
5xx	服务器错误	500=服务器内部错误；503=以后再试

具体各状态码的含义，请参考W3C的HTTP1.1标准规范RFC2616
[<http://www.w3.org/Protocols/rfc2616/rfc2616.html>]



HTTP报文结构(续4)

- 首部字段或消息头

头(header)	类型	说明
User-Agent	请求	关于浏览器和它平台的信息，如Mozilla5.0
Accept	请求	客户能处理的页面的类型，如text/html
Accept-Charset	请求	客户可以接受的字符集，如Unicode-1-1
Accept-Encoding	请求	客户能处理的页面编码方法，如gzip
Accept-Language	请求	客户能处理的自然语言，如en(英语)，zh-cn(简体中文)
Host	请求	服务器的DNS名称。从URL中提取出来，必需。
Authorization	请求	客户的信息凭据列表
Cookie	请求	将以前设置的Cookie送回服务器，可用来作为会话信息
Date	双向	消息被发送时的日期和时间
Server	响应	关于服务器的信息，如Microsoft-IIS/6.0
Content-Encoding	响应	内容是如何被编码的（如gzip）
Content-Language	响应	页面所使用的自然语言
Content-Length	响应	以字节计算的页面长度
Content-Type	响应	页面的MIME类型
Last-Modified	响应	页面最后被修改的时间和日期，在页面缓存机制中意义重大
Location	响应	指示客户将请求发送给别处，即重定向到另一个URL
Set-Cookie	响应	服务器希望客户保存一个Cookie



HTTP报文结构(续5)

- 实例

The screenshot displays the 'Network' tab of a web browser's developer tools, showing the 'Headers' section for an HTTP response and request. The response headers are highlighted with a red box, and the request headers are also highlighted with a red box. Annotations in yellow speech bubbles point to specific parts of the headers.

响应头信息 (Response Headers):

- 状态行 (Status Line):** HTTP/1.1 200 OK
- 首部字段(行) (Header Fields):**
 - Date: Fri, 19 Mar 2010 06:44:05 GMT
 - Server: Microsoft-IIS/6.0
 - X-Powered-By: ASP.NET
 - Content-Length: 31470
 - Content-Type: text/html
 - Cache-Control: private

请求头信息 (Request Headers):

- 请求行 (Request Line):** GET /index.asp HTTP/1.1
- 首部字段(行) (Header Fields):**
 - Host: www.sxtyu.com
 - User-Agent: Mozilla/5.0 (Windows; U; Windows NT 6.1; zh-CN; rv:1.9.2) Gecko/20100115 Firefox/3.6
 - Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
 - Accept-Language: zh-cn,zh;q=0.5
 - Accept-Encoding: gzip, deflate
 - Accept-Charset: GB2312,utf-8;q=0.7,*;q=0.7
 - Keep-Alive: 115
 - Connection: keep-alive
 - Cookie: online=true; ASPSESSIONIDCSDRCDTD=FPNEMIBBJKJFMJLDDFIAHEM



总结和答疑

