



# B/S体系软件设计

- HTML

胡晓军



# A brief history

- **HTML 1.0 & Mosaic(1993.6)**
- **W3C: HTML 2.0**
- **W3C: HTML 3.0?、HTML3.2**
- **W3C: HTML 4.01(1999.12)**
- **W3C: XML**
- **W3C: XHTML 1.0**
- **W3C: HTML 5(2014.10)**
- **W3C <http://www.w3.org>**



# Why learn HTML?

- **HTML is the Web's core language for creating documents and applications for everyone to use, anywhere.**
- **WYSIWYG?**
- **for CGI, JSP**
- **The best begin: Notepad/Visual Studio Code**
- **DreamWeaver**



# HTML tag

- **<tag>**
- **<tag>...</tag>**
- **<tag attribute1=...> ... </tag>**
- **<!-- ... -->**



# HTML structure

**<html>**

**<head>**

**<title>...</title>**

**</head>**

**<body>**

**</body>**

**</html>**



# Tags in head

- **<title>**
- **<meta>**
- **<script>**
- **<style>**



# **<body>**

- **bgcolor**

➤ **"#RRGGBB"**

- **Text, link, vlink, alink**

- **background**

➤ **"URL"**



# Basic format

- **head**

- `<h1>...</h1>`
- ...
- `<h6>...</h6>`

- **Char. Format**

- `<i>...</i>`
- `<b>...</b>`

- **paragraph**

- `<br>`
- `<nobr>`
- `<div> ... </div>`
- `<p>...</p>`
- `<pre> ... </pre>`
- `<plaintext> ... </plaintext>`
- `<blockquote>... </blockquote>`



- **align**

➤ = left, right, center

- **id**

- **title**



- **Space, tab and new-line**会被忽略
- 重复的**<p>**会被忽略
- **<** 和 **>**会被当作**tag**的符号而不能正确显示
  - **&lt;** → **<**
  - **&gt;** → **>**
  - **&amp;** → **&**
  - **&#97** → **a**



- **<cite>**
- **<code>**
- **<em>**
- **<strong>**
- **<address>**

- **<b>, <i>**
- **<big>, <small>**
- **<blink>**
- **<sub>, <sup>**
- **<tt>**



# lists

- **<ul>**  
    **<li>...</li>**
- **</ul>**
- **<ol>**  
    **<li>...</li>**
- **</ol>**

- **<dl>**  
    **<dt>...</dt>**  
    **<dd>...</dd>**
- **</dl>**

- **width = "10%"**
- **size=**
- **align=**
- **color=**



- **`<a href="URL"> ... </a>`**
- **`<a name="name"> ... </a>`**
- **`<a href="URL#name"> ... </a>`**
- **`target=_top, _blank`**

- **src=**
- **lowsrc=**
- **alt=**
- **align=top, middle, bottom, left, right**
- **border=**
- **height=, width=**



# Principles for images

- 保持简单
- 重用图片
- 分割大文档 (**50k**)
- 缩略图
- 指定尺寸



- ****
- **<map name="...">...</map>**
- **<area shape=circle coords="..." href="...">**



# table

- 表格 **<table>...</table>**
- 标题 **<caption>**
- 行 **<tr>...</tr>**
- 表头 **<th>...</th>**
- 元素 **<td>...</td>**



# table

列



表格标题



在 HTML 文档中，广泛使用表格来存放网页上的文本和图像



## **<table>**

- **align=center**
- **bgcolor, background**
- **border**
- **frame = box, void, above, below, lhs, rhs, hside, vside**
- **cellspacing**
- **cellpadding**
- **cols**
- **width, height**

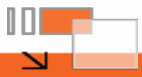


**<tr><th><td>**

- **align**
- **valign**
- **width =400 =40%**
- **height**
- **colspan=3**
- **rowspan**
- **bgcolor, backgroud**



- 将浏览器的窗口分割为几个独立的框架
  - **<frameset>**
  - **<frame>**
  - **<noframes>**
  - **<iframe>**



# **<framset>**

- **rows="150,300,150"**
- **rows="25%,50%,25%"**
- **rows="150,\*"**
- **cols=**
- **border**



# **<frame>**

- **name=**
- **src=**
- **noresize**
- **marginheight,marginwidth**
- **frameborder**





# **<iframe>**

- **name**
- **src**
- **align**
- **height**
- **weight**

- 表单是一个或多个文本输入框、可点击的按钮、多选框等组成
- 一个页面中可以有多多个表单
- 表单由**<form>...</form>**组成
- 其中也可以放常规文本
- 读者填写后提交，数据由服务器端程序处理



## <form>

- **action = <http://server/cgi-bin/sthlikethat>**
- **enctype="multipart/form-data"/"text/plain"**
- **method=post/get**
- **target**



# Form的例子

```
<form method=post action="...">
```

Name:

```
<input type=text name=name size=32 maxlength=80>
```

```
<p>
```

Sex:

```
<input type=radio name=sex value="M"> Male
```

```
<input type=radio name=sex value="F"> Female
```

```
<p>
```

Incomer:

```
<select name=income size=1>
```

```
<option>Under $25,000
```

```
<option>$25,001 to %50,000
```

```
<option>$50,001 and higher
```

```
</select>
```

```
<p>
```

```
<input type=submit>
```

```
</form>
```



# input

- **text**
- **password**
- **file**
- **checkbox**
- **radio**



- **submit**

`<input type=submit name=action value="ok">`

- **reset**

- **image**

- **button**

- **multiple**属性

- **size**属性

- **option**

**<option value=Dog>Dog**

**<option>Dog**



- **HTML**注重内容而不是形式
- **CSS**是管理形式的方式
- **1996, W3C, CSS 1**
- **1998, W3C, CSS 2**
- **2011, W3C, CSS 2.1**
- **2001, W3C, CSS 3 Draft**





- 样式是一个规则，告诉浏览器如何**render**一个特定的标签内容
- 内联样式
- 文档级样式
- 外部样式表



- **style** 属性
- **<h1 style="color:blue; font-style:italic">It's a blue text!</h1>**



- 在**head**区的**<style>...</style>**

**<head>**

**<style type= "text/css" media="screen,print">**

**<!--**

**h1 {color: blue; font-style: italic}**

**-->**

**</style>**

**</head>**



- 一个独立的文件，包含样式规则

**H1 {color: blue; font-style: italic }**

- 不需要其他头尾标记
- 可以放在网络的任意地方
- 一个样式表文件可以为很多页面使用



**<head>**

**<link rel=stylesheet type="text/css"  
href="URL">**

**</head>**



# 引入外部样式表

```
<head>
<style>
  <!--
    @import url(http://....);
    @import "URL"
    body:
      {background: url(backgrounds/marble.gif)}
  -->
</style>
```

**@import**可以出现在独立样式表文件中



## 连接 vs. 引用

- 多个<link>时，**CSS2**标准要求浏览器要求读者选择，但是现在的浏览器采取覆盖的策略，新引入的**CSS**覆盖已引入的，内联的覆盖外部的



- 基本格式

**Selector {property1:value1; property2:value1 value 2 value 3; ...}**

- 多个选择符

**h1, h2, h3, h4, h5, h6 {text-align: center}**





**ol li {list-style: upper-roman}**

**ol ol li {list-style: upper-alpha}**

**ol ol ol li {list-style: decimal}**

**h1 em {color: red}**

**h1 em, p strong, address { color: blue}**



# 特殊选择符

**\*{color: purple}**

**h1+h2 {margin-top: +4mm}**



```
<style>
```

```
<!--
```

```
p.abstract { font-style: italic}
```

```
-->
```

```
</style>
```

```
...
```

```
<p class=abstract>
```

```
...
```

```
</p>
```



```
.italic {font-style: italic}
```

```
<p class=italic>
```

```
...
```

```
</p>
```



- 字体
- 颜色
- 背景
- 文本
- 框
- 布局

- 关键字: **underline, small...**
- 长度: **1.5cm, +0.25mm, -3pt, +100em, -2ex**
- 百分数
- **URL: url(service://....)**
- 颜色: **RGB(red,green,blue), #rrggbb**



- **Graphics: Canvas 2D、Canvas 3D (WebGL)**
- **Multimedia: video、audio**
- **Offline Storage**
- **New tags: article、footer、header、nav、section**
- **New form types: date、time、email、url、search**
- **WebSocket**
- **CSS 3**



- **HTML5 的 canvas 元素使用 JavaScript 在网页上绘制图像。**
- **Canvas**是一个矩形区域，您可以控制其每一像素。
- **canvas** 拥有多种绘制路径、矩形、圆形、字符以及添加图像的方法。
- **<canvas id="myCanvas" width="200" height="100"></canvas>**
- **Javascript操作canvas**
  - **<script type="text/javascript">**
  - **var c=document.getElementById("myCanvas");**
  - **var cxt=c.getContext("2d");**
  - **cxt.fillStyle="#FF0000";**
  - **cxt.fillRect(0,0,150,75);**
  - **</script>**





# Video/Audio

- **`<video src="movie.mp4" controls="controls"> </video>`**
- **`<audio src="song.mp3" controls="controls"> </audio>`**



- **HTML5** 提供了两种在客户端存储数据的新方法：
  - **localStorage** - 没有时间限制的数据存储
  - **sessionStorage** - 针对一个 **session** 的数据存储
- 之前，这些都是由 **cookie** 完成的。但是 **cookie** 不适合大量数据的存储，因为它们由每个对服务器的请求来传递，这使得 **cookie** 速度很慢而且效率也不高。
- 在 **HTML5** 中，数据不是由每个服务器请求传递的，而是只有在请求时使用数据。它使在不影响网站性能的情况下存储大量数据成为可能。
- 对于不同的网站，数据存储于不同的区域，并且一个网站只能访问其自身的数据。
- **HTML5** 使用 **JavaScript** 来存储和访问数据。



# New Input types

- email
- url
- number
- range
- Date pickers (date, month, week, time, datetime, datetime-local)
- search
- color



# New Form Element

- datalist
- keygen
- output