

HTML

李晶  lijingjing@bupt.edu.cn

网站设计：了解受众（用户）

- 用户都是浏览器
 - 用户不会完整阅读网页
- 少即是多
 - 使所有的句子尽可能地短。
- 导航
 - 尽量创建通用于网站中所有页面的导航结构。
- 下载速度
 - 加载速度要快
- 允许用户发言
 - 得到来自用户的反馈是件好事情。

网站设计：了解受众（用户）

- 用户使用的显示器
 - 不同的显示器分辨率会影响网页的显示效果
- 用户使用的浏览器
 - 不同的浏览器会影响网页的显示效果
- 用户安装的插件
 - 特殊的网页元素可能需要特殊的插件支持运行
- 怎样帮助特殊人群

参考资料

- 标签查询手册：
 - <http://www.w3school.com.cn/tags/>
- 标签效果测试：
 - [http://www.w3school.com.cn/ty/t.asp](http://www.w3school.com.cn/tiy/t.asp) 
- 工具：
 - DreamWeaver
 - 记事本
 - FrontPage

参考资料

提交代码

编辑您的代码:

```
<html>
<head>
<title>我的第一个 HTML 页面</title>
</head>

<body>
<p>body 元素的内容会显示在浏览器中。</p>
<p>title 元素的内容会显示在浏览器的标题栏中。</p>
</body>

</html>
```

查看结果:

body 元素的内容会显示在浏览器中。

title 元素的内容会显示在浏览器的标题栏中。

代码编辑区域

查看效果区域

主要内容

- HTML实例
- HTML概述
- HTML常用标签
 - 网页结构类标签
 - 段落类标签
 - 列表类标签
 - 格式化类标签
 - 超链接类标签
 - 图像类标签
 - 表格类标签
 - 表单类标签
 - 框架类标签

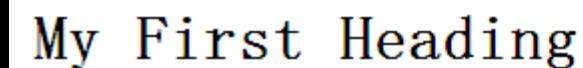
HTML实例

```
<html>
<body>

<h1>My First Heading </h1>
<p>My first paragraph. </p>

</body>
</html>
```

- 保存这个文件，命名为 '*firstHtml.html*'
- 使用浏览器打开



My First Heading

My first paragraph.

主要内容

- HTML实例
 - HTML概述
 - HTML常用标签
 - 网页结构类标签
 - 段落类标签
 - 列表类标签
 - 格式化类标签
 - 超链接类标签
 - 图像类标签
 - 表格类标签
 - 表单类标签
 - 框架类标签
1. HTML概述
 2. HTML标签
 3. HTML标签属性
 4. HTML语法规则

HTML概述

- HTML 是一种描述WEB页面的语言。
 - HTML 全名 **Hyper Text Markup Language**(超文本标签语言)
 - HTML 并不是一种编程语言, 而是一种标签语言。
 - 标签语言中包含了一组标签。
 - 这组标签存在的目的是为了描述网页内容。

HTML 标签

- HTML 标签(tags) 通常简称为标签。
 - HTML标签是由尖括号包围的关键词，比如 <html>
 - HTML 标签分为单标签和双标签
 - 双标签由开始标签和结束标签两部分构成，必须成对使用。
 - 基本语法：<标签名>内容</标签名>
 - 语法说明：“内容”是要被这对标签施加作用的部分。
 - 例子：粗体显示表示文本应以粗体显示。
 - 单标签仅单独使用就可以表达完整的意思。
 - 基本语法：<标签名 />
 - 例子：
表示换行。

HTML标签属性

- HTML可以为某些元素(标签)附加一些信息，这些附加信息被称为属性 (attribute)。
 - 基本语法 : <标签名称 属性名1="属性值" 属性名2="属性值">
 - 语法说明 : 属性应写在开始标签内，并且和标签名之间有一个空格分隔，多个属性之间要用空格分隔，除此之外其它任何字符不得加入。
 - 例子 : <hr />用于在网页中插入一条水平线, 这条水平线的粗细、对齐方式等就是该标签的属性，如：
`<hr size= "5px" align= "center " /> (建议)`
`<hr size=5px align=center />`

HTML语法规则

- HTML标签**不区分大小写**
 - 如<p>与<P>是一样的
- 多个HTML标签间可以循环嵌套，**但不可以交叉嵌套。**
- HTML文件一行可以写多个标签，一个标签也可分多行书写，不用加任何续行符。
- HTML文件由浏览器解释时只认标签，并不解释源文件中的换行、回车和多个连续空格。若要显示换行则用
，换段则用<p>，而要显示多个空格，则需使用多个&nbsp(实体字符)。

主要内容

- HTML实例
- HTML概述
- HTML常用标签
 - 网页结构类标签
 - 段落类标签
 - 列表类标签
 - 格式化类标签
 - 超链接类标签
 - 图像类标签
 - 表格类标签
 - 表单类标签
 - 框架类标签

网页结构类标签

```
<html> ←  
<head>  
  <meta http-equiv="Content-Type" content="text/html; charset=gb2312">  
  <title>.....</title>  
</head>  
<body>  
</body>  
</html> ←
```



- HTML文档标签
- 格式：`<html>.....</html>`
- 说明：`<html>`处于文档的最前面，表示HTML文档的开始，即浏览器从`<html>`开始解释，直到`</html>`为止。
- 在`<html>`和`</html>`标签对之间是HTML的所有内容，一般由头部和主体构成。

网页结构类标签

```
<html>
<head> ←
  <meta http-equiv="Content-Type" content="text/html; charset=gb2312">
  <title>.....</title>
</head> ←
<body>
</body>
</html>
```



- HTML文件头标签
- 格式：`<head>.....</head>`
- 说明：`<head>`紧跟`<html>`标签的后面，表示文件头部开始。包含了HTML页面的标题和说明信息，在浏览器窗口中，头部信息是不被显示在正文中的，在此标签中可以插入其它用以说明文件的标题和一些公共属性的标签。如编码方式、关键字、编辑软件、作者信息等。

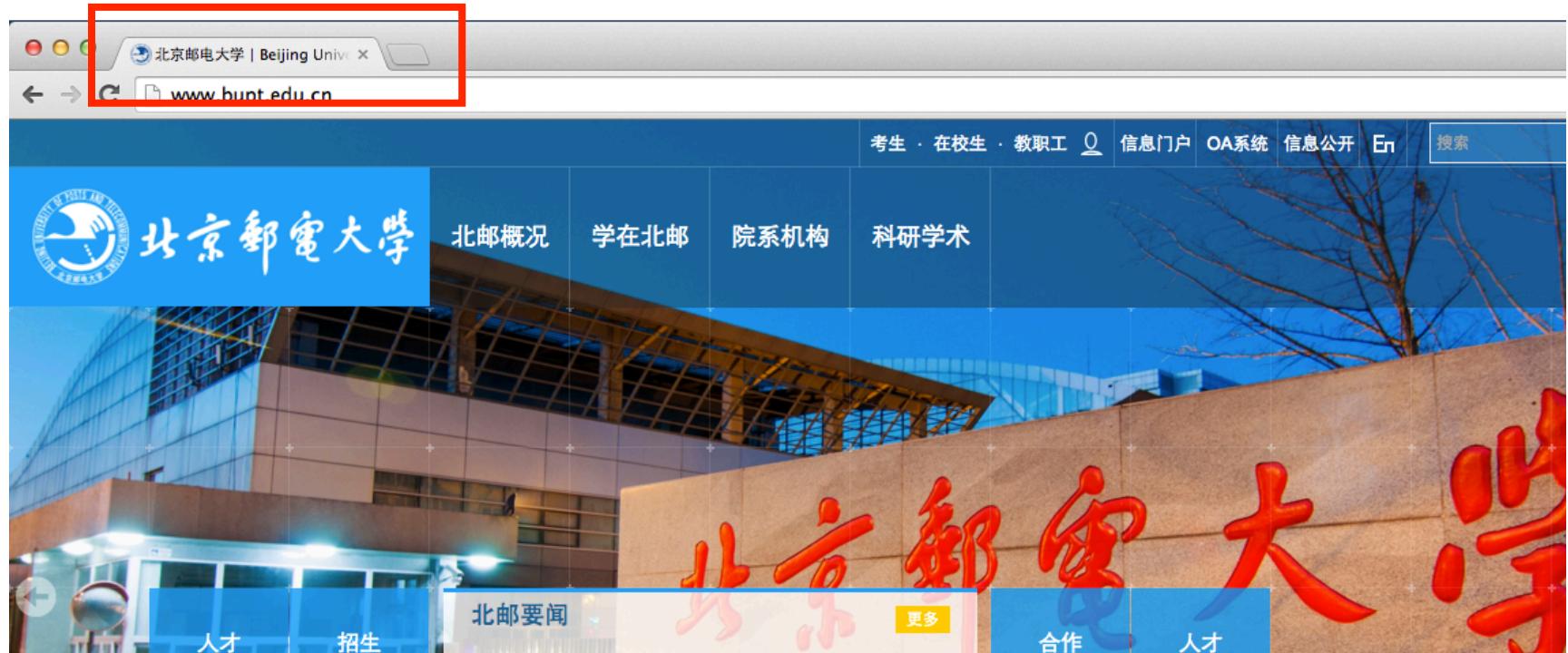
网页结构类标签

```
<html>
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=gb2312">
  <title>.....</title> ←
</head>
<body>
</body>
</html>
```



- HTML文件标题标签
- 格式：`<title>.....</title>`
- 说明：该标签定义的标题内容不在浏览器窗口中显示，而是在浏览器的标题栏中显示。通过该标题内容的设定，可以使读者迅速了解该网页的主要内容以及在使用搜索引擎时可以找到该页。
。 `<head>` 中**唯一必须出现**的标签。

网页结构类标签



网页结构类标签

```
<html>
<head>
  <meta http-equiv="Content-Type" content="text/html;charset=gb2312"/> ←
  <title>.....</title>
</head>
<body>
</body>
</html>
```



- **HTML文件说明信息标签**
- **格式** : <meta **http-equiv="" content="" />** **必须的属性**
- <meta **name="" content="" />** **可选的属性**
- **说明** : 标签用于描述不包含在标准HTML里的一些文档信息，例如开发工具，作者，网页关键字，网页描述等。这些定义的内容并不在网页页面中显示，但是一些搜索引擎可以检索这些信息，浏览者可以根据这些关键字或描述查找到该网页。

网页结构类标签

- HTML文件说明信息标签
- 作用1：描述语言字符集及刷新等信息
- 例子：
 - ✓ 描述网页的字符集 **MIME (Multipurpose Internet Mail Extensions)** 是描述消息内容类型的因特网标准。MIME 消息能包含文本、图像、音频、视频以及其他应用程序专用的数据。格式：type/subtype
 - <meta http-equiv="Content-Type" content="text/html; charset=gbk" />
 - ✓ 刷新页面并跳转到一个新的页面
 - <meta http-equiv="Refresh" content="3;url=http://www.baidu.com" />
 - ✓ 网页的过期时间，一旦过期必须到服务器上请求
 - <meta http-equiv="Expires" content="2016/4/20" />
 - ✓ 禁止从本地机的缓存中调阅内容
 - <meta http-equiv="pragma" content="no-cache" />

MIME

Type	Subtype
text	plain, rich text, html, tab-separated-values, xml
message	Email, news
Application	Octet-stream (can be used for transferring Java .class files, for example), Adobe-postscript, Mac-binhex40, xml
image	jpeg, gif
audio	basic,midi,mp3
video	mpeg, quicktime

网页结构类标签

- **HTML文件说明信息标签**
- **作用1：描述语言字符集及刷新等信息**
- **例子：**
 - ✓ 不允许IE对网页进行缓存
`<meta http-equiv="cache-control" content="no-store" />`
 - ✓ 设置cookie过期时间，当过期时，删除存储在计算机上的cookie。必须是GMT的时间格式。
`<meta http-equiv="set-cookie" content="cookiename=cookievalue;expires=Wednesday,9-oct-2011 18:25:30 GMT;path=/" />`
 - ✓ 强制页面在当前窗口中以独立的形式打开，防止别人在框架中调用页面。
`<meta http-equiv="Winodws-Target" content="_top" />`

网页结构类标签

- HTML文件说明信息标签
- 作用2：描述文档自身信息，如作者信息、过期时间、关键词列表等，是为搜索引擎服务的。

- 例子：

- ✓ 网页的关键字

- <meta name="Keywords" content="asp学习,网页设计" />

- ✓ 网页的描述

- <meta name="Description" content="html标签学习" />

- ✓ 网页的设计者信息

- <meta name="Author" content="30338***@qq.com,张三" />

- ✓ 版权声明

- <meta name="Copyright" content="自由版权" />

- ✓ 是否允许搜索引擎抓取该页面

- <meta name="Robots" content="all" />

- ✓ 说明本页是由什么软件编辑的

- <meta name="Generator" content="vim" />

网页结构类标签

- **BASE标签**
- 作用：指定一个显式URL，用于解析对外部源的链接或引用，简单说就是基址设置。
- 例子：`<base href="http://www.test.com" target="_self" />`
- 属性：
 - href表示基址地址
 - Target表示窗口打开的位置，_blank, _parent, _self, _top
- 通常情况下，浏览器会从当前文档的 URL 中提取相应的元素来填写相对 URL 中的空白。使用 `<base>` 标签可以改变这一点。浏览器随后将不再使用当前文档的 URL，而使用指定的基本 URL 来解析所有的相对 URL。这其中包括 `<a>`、``、`<link>`、`<form>` 标签中的 URL。

BASE标签实例

- <html>
- <head>
- <base href="http://www.bupt.edu.cn/" target="_blank" />
- </head>
- <body>
- <p>W3School</p>
- </body>
- </html>

[W3School](#)

- 该超链接的地址为：

[http:// www.bupt.edu.cn
/test](http://www.bupt.edu.cn/test)，单击超链接在新窗口打开页面。

网页结构类标签

- **LINK标签**
- 作用：设定IE地址栏、收藏夹里的小图标以及引用外部样式表（主要用途）。



- 属性：
 - rel：规定当前文档与被链接文档之间的关系。
 - href：指定被链接文档的位置。
 - type：指定被链接文档的 MIME 类型。

- 例子：

✓ 设置收藏夹里的小图标

```
<link rel="icon" href="favicon.ico" type="image/x-icon" />
```

✓ 引用外部样式表

```
<link rel="stylesheet" href="样式表文件地址" type="text/css" />
```

网页结构类标签

```
<html>
<head>
  <meta http-equiv="Content-Type" content="text/html;charset=gb2312">
  <title>.....</title>
</head>
<body>←
</body>←
</html>
```



- **HTML文件主体标签**
- **格式**：`<body>...</body>`
- **说明**：位于头部之后，以`<body>`开始，直到`</body>`结束。定义了网页上显示的主要内容与显示格式，**是整个网页的核心**，浏览器窗口中所能显示的内容全部被包含在该标签中，如图片、文字、表格、表单、超链接等元素。

网页结构类标签

- **注释标签**
- **格式**：`<!--.....-->`或`<comment>...</comment>`
- **说明**：在HTML文件中加入注释可以使程序清晰，容易理解。该标签中的内容在被浏览器解释时会被忽略，也不会被显示。
 - **注**：注释里面不能再有注释！

主要内容

- HTML实例
- HTML概述
- HTML常用标签
 - 网页结构类标签
 - 段落类标签
 - 列表类标签
 - 格式化类标签
 - 超链接类标签
 - 图像类标签
 - 表格类标签
 - 表单类标签
 - 框架类标签

段落类标签

- **段落标签**

格式 : <p align=“ ”>.....</p>

说明 : 段落就是格式上统一的内容的组合。

属性 : align用来设定段落的对齐方式。取值 : left、right和center。

IE	Firefox	Chrome	Safari	Opera
				

段落标签实例

- <html>
- <body>
- <p align="center">第一段居中</p>
- <p align="right">第二段居右</p>
- <p align="left">第三段居左</p>
- </body>
- </html>

第一段居中

第二段居右

第三段居左

段落类标签

- **换行标签**

格式：`
`

说明：控制段落中文字的换行显示。一般浏览器会根据窗口的宽度自动将文本进行换行显示，如果想强制浏览器不换行显示，可以使用`<nobr>`标签。

IE	Firefox	Chrome	Safari	Opera
				

- **水平线标签**

格式：`<hr align="" size="" width="" color="" />`

说明：该标签可以用来在页面上插入一条水平线。

属性：`align`：设定横线放置的位置，取值：`left`、`right`和`center`。

`size`：设定线条粗细，取值为n个像素，默认为2。

`width`：设定线段宽度，可为绝对值（n个像素）或相对值（相对于水平线所在的标签宽度的百分比），默认值为100%

`color`：设置线条的颜色

主要内容

- HTML实例
- HTML概述
- HTML常用标签
 - 网页结构类标签
 - 段落类标签
 - **列表类标签**
 - 格式化类标签
 - 超链接类标签
 - 图像类标签
 - 表格类标签
 - 表单类标签
 - 框架类标签

列表类标签

- 编号(有序)列表标签(ordered list)

格式：`<ol type="" start="">.....`

说明：该编号列表中每一项由``开头以``结束。

属性：

✓ type，用来设定编号方案。

type=1 默认值，用数字对应编号符，如1，2，3等。

type=A 用大写字母编号表项，如A，B，C等。

type=a 用小写字母编号表项，如a，b，c等。

type=I 用大写罗马字母编号表项，如I，II，III，IV等。

type=i 用小写罗马字母编号表项，如i，ii，iii，iv等。

✓ Start 设置从哪一个数字或字母开始编号，属性值是一个整数。例如，设置start="3"，则有序列表的列表项编号将从“3”、“c”、“C”、“iii”或“Ⅲ”开始编号，也就是说其值只能为数字。

IE	Firefox	Chrome	Safari	Opera
				

列表类标签

- 项目（无序）列表标签（unordered list）

格式：`<ul type="">.....`

说明：该编号列表中每一项由``开头以``结束。

属性：

✓ `type`，用来项目符号的形状。

`type=1` `type=disc` 实心圆点（默认值）

`type=circle` 空心圆点

`type=square` 实心方块

IE	Firefox	Chrome	Safari	Opera
				

列表类标签

- **列表项标签**

格式 : <li type="" value="">.....

属性 :

- 如果设置type属性，则会覆盖掉父级设置的属性，并只影响当前项，后续的列表项标签将恢复遵循列表中设置的type属性。
- value属性，可以改变当前列表项的编号大小，并会影响其后所有列表项的编号大小，该属性只适用于有序列表。

IE	Firefox	Chrome	Safari	Opera
				

列表类标签

- 关于列表的说明
 - 列表可以嵌套使用，即一个列表中可以包含有多层子列表。
 - 对于内容层次较多的网页，使用嵌套列表不仅使网页的内容布局更加合理美观，而且使其内容看起来更加清晰、明了。
 - 嵌套的列表可以是无序列表的嵌套，可以是有有序列表的嵌套，还可以是无序列表和有序列表的混合嵌套。

有序/无序列表实例

```
<ul>
  <li>Coffee</li>
  <li>Milk</li>
</ul>
```

```
<ol>
  <li>Coffee</li>
  <li>Milk</li>
</ol>
```

- Coffee
- Milk

1. Coffee
2. Milk

主要内容

- HTML实例
- HTML概述
- HTML常用标签
 - 网页结构类标签
 - 段落类标签
 - 列表类标签
 - 格式化类标签
 - 超链接类标签
 - 图像类标签
 - 表格类标签
 - 表单类标签
 - 框架类标签

格式化类标签

- 字符风格标签
- 格式：

..... 粗体(bold) <i>.....</i> 斜体(italic)
<u>.....</u> 下划线 <strike>.....</strike> 删 除 线
^{.....} 上 标 _{.....} 下 标
<big>.....</big> 以 较 大 字 体 显 示
<small>.....</small> 以 较 小 字 体 显 示

IE	Firefox	Chrome	Safari	Opera
				

格式化类标签

- **字符格式标签**
- **格式**：`.....`
- **说明**：该标签可以设置文字的字体、字号和颜色。
- **属性**：
 - `size` 设置文字大小，取值范围是从1到7，其中3是默认值。
 - `face` 设置文字字体，如宋体、楷体_GB2312、Roman等。
 - `color` 设置文字颜色，取值是颜色名（如red）或颜色数值（如#003366）



```
<html>
<body>
<h1><font face="verdana">A heading</font></h1>
<p><font size="5" face="arial" color="red">A
paragraph.</font></p>
</body>
</html>
```

A heading

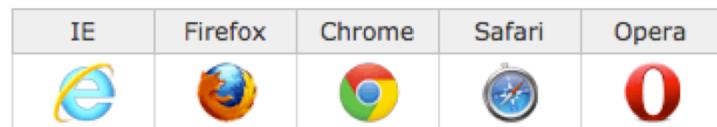
A paragraph.

主要内容

- HTML实例
- HTML概述
- HTML常用标签
 - 网页结构类标签
 - 段落类标签
 - 列表类标签
 - 格式化类标签
 - 超链接类标签
- 图像类标签
- 表格类标签
- 表单类标签
- 框架类标签

超链接标签

- 格式：`...`
- 属性：
 - href:用来设定超链接的目的资源/目标地址，即路径。
 - 创建内部超链接时，不需要为每一个链接都输入完整的地址，只需确定当前文档同站点根目录之间的相对路径关系即可。文档路径可以分为绝对路径、相对路径
 - 根相对路径 “/” 表示站点所在的文件夹
 - 当前文件夹相对路径
 - “..”表示当前目录的上一级目录
 - “.”表示当前目录，一般省略不写
 - target:当点击时在指定的窗口打开目标端资源`_blank`、`_top`、`_parent`、`_self`。
 - title:当鼠标指针放到超链接上时出现的提示。



超链接类标签

- 分类：
 - 按目的位置不同分内部链接和外部链接
 - 内部链接，在同一网站内部，不同页面之间的链接关系。
 - 明确哪个是主链接文件（即当前页），哪个是被链接文件。
 - 内部链接一般采用相对路径链接比较好。
 - 外部链接，跳转到当前网站外部，与其它网站中页面或其它元素之间的链接关系。
 - 一般要用绝对路径，要有完整的URL地址。

超链接实例

- <html>
- <body>
- <p>
- 指向网站内部页面的链接 </p>
- <p>
- 指向网站外部页面的链接 </p>
- </body>
- </html>

指向网站内部页面的链接

指向网站外部页面的链接

- 第一个超链接将在新窗口中打开一个相对于当前页面的路径的页面
- 第二个超链接将在当前窗口中打开百度页面。

超链接实例

- <html>
 - <body>
 - <p>
 -

 -
 - </p>
 - </body>
 - </html>
-
- 将图像做成超链接

下一页 ➔

主要内容

- HTML实例
- HTML概述
- HTML常用标签
 - 网页结构类标签
 - 段落类标签
 - 列表类标签
 - 格式化类标签
 - 超链接类标签
 - 图像类标签
 - 表格类标签
 - 表单类标签
 - 框架类标签

图像标签

- **格式**：``
- **说明**：用来在文档中嵌入图像。
- **属性**：
 - `src`：设定要被包含的图像的位置和文件名，其路径名使用规则同超链接中的`href`属性。
 - `alt`：规定图像的替代文本。
 - `title`：当鼠标指针放到图象上时出现的对图象的描述。
 - `width`、`height`：设定图像的尺寸，可取绝对值或相对值，如果只设置了其中的一个属性，则另一个属性会根据已设置的属性按原图等比例显示。如果对两个属性都进行了设置，且其比例和原图大小的比例不一致的话，那么显示的图像会相对于原图变形或失真。
 - `border`：设定图像边框的宽度，其默认值为0，无图像边框。
 - `align`：设定图像的位置

IE	Firefox	Chrome	Safari	Opera
				

图像标签

- Align属性的取值：
 - Left 水平方向上左对齐
 - Center 水平方向上居中对齐
 - Right 水平方向上右对齐
 - Top 图片顶部与同行其他元素顶部对齐
 - Middle 图片中部与同行其他元素中部对齐
 - Bottom 图片底部与同行其他元素底部对齐
 - Absbottom 图片的底部与同行最低元素的底部对齐，常用于Netscape浏览器
 - Absmiddle 图片的中部与同行最大元素的中部对齐，常用于Netscape浏览器
 - Baseline 图片的底部与文本基准线对齐，常用于Netscape浏览器
 - Texttop 图片的顶部与同行最高元素的顶部对齐，常用于Netscape浏览器

图像标签

- 图像格式与注意事项
 - 图像格式可以是GIF、JPEG、PNG、BMP、TIFF、XBM和PICT等。
 - 其中：GIF格式文件最多只能显示256种颜色，一般用于制作透明、隔行和动画效果图片等；而JPEG格式文件可以拥有计算机所能提供的最多种颜色，适合存放高质量的彩色图片、照片。
- 注意：
 - 高质量的图片一般用TIFF格式保存，但其图片体积过大，不太适合网络传输。
 - 网页中常用的图片格式为GIF、JPEG和PNG。
 - 一般在网页设计中选择的图片不要过大，如必须选用较大图片时，可先将其分成若干小图片，显示时再通过表格将这些小图片拼合起来。
 - 如果在网页中使用所在站点内的图片，最好使用相对路径。

图像实例

- <html>
- <body>
- <p>来自另一个文件夹的图像:
-
- </p>
- <p>来自另一个网站的图像:
-
- </p>
- </body>
- </html>

来自另一个文件夹的图像:



来自另一个网站的图像:



- 来自同一个网站的图像一般使用相对地址
- 来自其它网站的图像一般使用绝对地址

图像实例

- <html>
- <body background="/i/eg_background.jpg">
- <h3>图像背景</h3>
- <p>gif 和 jpg 文件均可用作 HTML 背景。</p>
- <p>如果图像小于页面，图像会进行重复。</p>
- </body>
- </html>

图像背景

gif 和 jpg 文件均可用作 HTML 背景。

如果图像小于页面，图像会进行重复。

- 使用body标签的background属性
- 如果图像小于页面，图像会进行重复

图像实例

- <html>
- <body>
- <p>把鼠标指针移动到图像上，大多数浏览器会显示 "alt" 文本</p>
-
- <p>如果无法显示图像，将显示 "alt" 属性中的文本：</p>
-
- </body>
- </html>

把鼠标指针移动到图像上，大多数浏览器会显示 "alt" 文本



如果无法显示图像，将显示 "alt" 属性中的文本：

向左转

- 在浏览器无法载入图像时，替换文本属性告诉读者她们失去的信息。为页面上的图像都加上替换文本属性是个好习惯。

主要内容

- HTML实例
- HTML概述
- HTML常用标签
 - 网页结构类标签
 - 段落类标签
 - 列表类标签
 - 格式化类标签
 - 超链接类标签
 - 图像类标签
 - 表格类标签
 - 表单类标签
 - 框架类标签

表格标签

- **描述**：由三个标签来构成<table>、<tr>、<td>
- **格式**：

```
<table>
  <tr>
    <td></td>
    ...
  </tr>
  ...
</table>
```

IE	Firefox	Chrome	Safari	Opera
				

- **说明**：
 - 表格标签除了可以显示大量数据列表格式外，往往还可以用于网页的布局。
 - 可以用<th></th>（只能用作列，行不可以）做表头，<caption></caption>（放在所有行之前Table之后）做表格的标题

表格标签

- <tr>设置表格中某一行
 - align设置单元格中内容的水平对齐(center、left、right、justify)
 - valign设置一整行单元格内容的垂直对齐(top、middle、bottom、baseline)vertical
 - bgcolor
 - background
 - width\height
 - bordercolor\bordercolorlight\bordercolordark
- <td>设置表格的单元格
 - 具有tr标签的属性
 - rowspan属性可实现单元格的跨行合并（纵向合并）。属性值为整数n，表格跨几行合并。
 - colspan属性可以进行单元格的跨列合并（横向合并）。属性值同上
 - 所有行属性都适用

表格实例

```
• <html>
•   <body>
•     <p>两行三列 : </p>
•     <table border="1">
•       <tr>
•         <td>100</td>
•         <td>200</td>
•         <td>300</td>
•       </tr>
•       <tr>
•         <td>400</td>
•         <td>500</td>
•         <td>600</td>
•       </tr>
•     </table>
•   </body>
• </html>
```

两行三列：

100	200	300
400	500	600

- 表格<table>内部必须以<tr>开始，<tr>内部嵌入<td>，哪怕只有一行一列。
- <td>标签内部放入在表格单元内要显示的内容。
- 由于设置了border属性，因此表格带有边框，如果这里设置为“0”，那表格就是没有边框的形式。

100	200	300
400	500	600

表格标签

- <table>属性：
 - Border 用来设定表格边框粗细。若省略，则无边框默认为0。
 - borderColor 设定表格边框的颜色
 - width 和 height 用来设定表格宽度和高度，可取绝对值或相对值。
 - Cellspacing 用来设定单元格之间的间隙。
 - Cellpadding 用来设定单元格边沿与格内内容之间的空白。
 - Bgcolor 设置表格的背景颜色
 - Background 设置表格的背景图像
 - Align 设置表格的对齐方式：center, left 和 right
 - Frame 设置外侧边框的哪个部分是可见的，值有 above、 below、 border 、 box、 hsides、 lhs、 rhs、 void、 vsides
 - Rules 设置内侧边框的哪个部分是可见的，值有 all 、 rows 、 cols、 groups 、 none

表格标签

- 表格嵌套：
 - 根据需要，在一个表格的某个单元格里再插入一个若干行和列的表格。
 - 对嵌套表格，可以像对任何其他表格一样进行格式设置，但是其宽度受它所在单元格的宽度的限制。
 - 利用表格的嵌套，一方面可以编辑出复杂而精美的效果，另一方面可根据布局需要来实现精确的编排。
 - 嵌套层次越多，网页的载入速度就会越慢。
- 注意：
 - 在创建复杂的表格之前，最好对它进行规划，比如可以先用笔在纸上设计页面；
 - 使用表格排版网页时，要尽量细化表格，不要把整个网页放在一个大的表格里，因为表格内有一些载入较慢的元素（比如计数器）时，往往会延迟整个表格的显示。这是由IE的显示特性决定的，只有当表格内所有元素全部载入后，整个表格才得以显示。针对这个特点，我们可以将整个页面分成几块，例如通用的上(放置Logo、Banner、Menu等等)、中(放置页面内容)、下(放置版权信息等等)结构，将每一部分由单独的表格来实现，这样比较好。

主要内容

- HTML实例
- HTML概述
- HTML常用标签
 - 网页结构类标签
 - 段落类标签
 - 列表类标签
 - 格式化类标签
 - 超链接类标签
 - 图像类标签
 - 表格类标签
 - 表单类标签
 - 框架类标签

HTTP协议的发展

- Http协议诞生之初是为了获取和显示简单的文本文件
- 现在Http协议以被扩展用以支持各种类型的文档内容的传输
- Http协议的版本HTTP/0.9、HTTP/1.0、HTTP/1.1
- HTTP 协议是一种**面向连接的无状态的请求-应答协议。**
- 默认工作在TCP的80端口上。

HTTP 请求/应答消息

- HTTP 协议是一种基于文本的请求-应答协议：请求和应答都是字符串。
- 每个请求/应答都由三部分组成：
 - 请求/应答行
 - 头部
 - 空行
 - 请求/应答消息体

HTTP 请求消息

- 客户请求在客户与服务器之间建立连接后被发送给服务器。
- **请求行的形式：**
 - <HTTP 方法><空格><请求路径><空格><协议规范>
 - <HTTP 方法> 协议规定的可以使用的数据传递方法
 - <请求路径> 请求的web文档或者web对象的地址
 - <协议规范> 客户端遵守的协议规范
 - 实例：
 - GET /index.html HTTP/1.0



HTTP请求消息

- **HTTP** 方法是一个保留字，规定了客户希望服务器执行的操作。
- 常见方法：
 - **GET**:获取URI指定的web对象的内容；
 - **POST**:用于向服务器主机上的某进程发送数据；
 - **PUT**:用于请求服务器保存请求附带的内容，数据保存在服务器上由URI所指定的文件位置中。



HTTP请求消息

- 请求头是在请求行之后的一个可选项。请求头的域允许客户向服务器传递有关请求及有关客户的额外信息。
- 可选用请求体来结束请求，它包含需要传送给服务器的与请求相关的数据。
 - 例如，如果在请求行中定义了POST方法，那么请求体中就会包含将传递给目标进程的数据。

HTTP响应消息

- 状态行格式：
 - <协议><空格><状态码><空格><状态码描述>
 - <协议>表示服务器支持的HTTP版本。
 - <状态码>是一个三个数字的结果代码，用于机器自动识别请求的处理结果。
 - <状态码描述>给状态码提供一个简单的文本描述，帮助用户理解请求的处理结果。

HTTP响应消息

- 状态码的第一个数字定义响应的类别，后两个数字没有分类的作用。
- 第一个数字可能取5个不同的值。
 - 100-199信息响应类，表示接收到请求并且继续处理。
 - 200-299处理成功响应类，表示动作被成功接收、理解和接受。
 - 300-399重定向响应类，为了完成指定的动作，必须接受进一步处理。
 - 400-499客户端错误，客户请求包含语法错误或者是不能正确执行。
 - 500-599服务端错误，服务器不能正确执行一个正确的请求。

HTTP响应消息

- 应答头跟在状态行之后。描述关于服务器或者服务器返回的内容或对象的相关信息。
- 应答头后面跟的是应答消息体，里面包含了从服务器返回的客户请求的内容。

动态内容生成

- 最初HTTP主要用来传输静态的信息内容，即以固定不变的形式表现的信息内容
 - 如普通文本或图像文件等。
- 随着web的演变，一些应用开始将HTTP用于其他目的了：应用允许浏览器用户在HTTP会话过程中，根据动态输入的信息来检索数据。
- 我们要依靠的第一个技术：表单

表单类标签

- **描述**：为用户输入创建 HTML 表单。把来自用户的信息提交给服务器，是服务器与客户端之间沟通的桥梁。表单有两个重要组成部分：一是描述表单的HTML源代码；二是用于处理用户在表单域中输入的信息的服务器端应用程序、客户端脚本，如JS、ASP、PHP、JSP等。

- **格式**：

```
<form name="" action="URL" method="get/post">
    <input type=" " name=" " value=" " />
    .....
    <input type=" " name=" " value=""" />

    <select name="" size="" >
        <option>.....</option>
        .....
    </select>

    <textarea></textarea>
</form>
```

IE	Firefox	Chrome	Safari	Opera
				

表单类标签

- 格式：

```
<form name=" " action="URL" method="get/post">  
    <input type=" " name=" " value=" " />  
    <input type=" " name=" " value=" " />  
    <select name=" " size=" " >  
        <option>.....</option>  
        .....  
    </select>  
    <textarea></textarea>  
</form>
```

- 属性：

- name 设定该表单项名，主要在处理表单时起作用。
- action 设定将要处理表单的URL。
- method 设定客户与服务器的数据交换方式（get、post默认值get）。
- target 设定提交页面打开的窗口

- 注意此标签不能嵌套使用

表单标签实例

- <html>
- <body>
- <form action="/example/html/form_action.asp" method="get">
- <p>First name: <input type="text" name="fname" /></p>
- <p>Last name: <input type="text" name="lname" /></p>
- <input type="submit" value="Submit" />
- </form>
- </body>
- </html>

The image shows a screenshot of a web browser displaying a form. The form consists of three elements: a text input field labeled "First name:", another text input field labeled "Last name:", and a "Submit" button. The "First name:" field contains the placeholder text "First name:". The "Last name:" field is empty. The "Submit" button is a standard rectangular button with the text "Submit" centered on it.

表单类标签

- 网页之间数据的传递方式
 - 表单(form)传递方式
 - Post方式
 - Get方式
 - URL地址传递
 - 在URL后面加 “?”，问号后是要传递的数据
 - 传递数据的形式是：变量名=值
 - 如果要传递多个数据，数据与数据之间用 “&”连接
- 例如: `http://www.x.com/dir/page.html?a=1&b=2&c=3`
- URL地址传递是用GET方式传递数据的，GET方式传递的数据量一般限制在1KB (1024个字节) 以下。

表单类标签

- <**input** />输入域标签
- 描述：单标签，它一般嵌套在表单标签中使用，用于定义一个用户的输入项。
- 属性：主要有四个：name，value，type，disabled。
 - name：标签的名字。
 - type主要有九种类型：text，password，checkbox，radio，file，hidden，submit，reset，image。用来设置用户不同的输入方式。
 - value：设置或获取<input />标签的值。
 - disabled：设置当前状态不可用，既以灰色表现。
 - 在不同的输入方式下，<input />标签的属性略有不同，其他属性因type类型的不同而不同，其含义也不尽同；

The image shows a portion of a web page with a light gray background. It features two input fields: one labeled "First name" with a text input box, and another labeled "Last name" with a text input box. Below these fields is a blue "Submit" button.

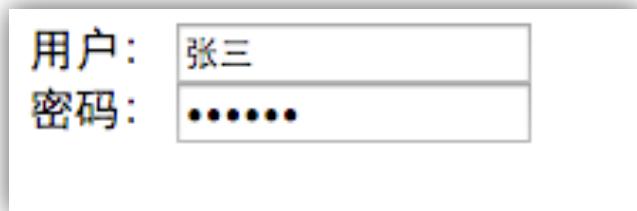
表单类标签

- <input type="text" name="" maxlength="" size="" value="">
- 当type=text时：表示该输入项的输入信息是字符串。此时，浏览器会在相应的位置显示一个文本框供用户输入信息，属性如下：
 - maxlength：设置单行输入框可以输入的最大字符数，例如限制邮政编码为6个数字、密码最多为10个字符等等，最大值是255；
 - size：设置单行输入框可显示的最大字符数，这个值总是小于等于maxlength属性的值，当输入的字符数超过文本框的长度时，用户可以通过移动光标来查看超出的内容；
 - value：文本框的值，可以通过设置value属性的值来指定当表单首次被载入时显示在输入框中的值。
 - Readonly:不接受用户的输入。
- 当type=password时：表示密码输入框，与单行文本输入框text使用起来非常相似，所不同的只是当用户在输入内容时，用特殊符号来代替显示每个输入的字符，以保证密码的安全性。

表单标签实例

- <html>
- <body>
- <form>
- 用户: <input type="text" name="user">

- 密码: <input type="password" name="password">
- </form>
- </body>
- </html>



用户名: 张三
密码: *****

- 用户的<input> type属性为text，则接收正常的字符串
- 密码的<input> type属性为password，则接收的字符串将用圆点代替。

表单类标签

- <input type="checkbox" name=" " value="">
- 当type= checkbox时：表示是一个复选框用户可以同时选中表单中的一个或多个复选项作为输入信息。
 - 由于选项可以有多个，属性name应取不同的值，如果相同则会形成一个数组；
 - 属性value的参数值就是在该选项被选中并提交后，浏览器要传送给服务器的数据。因此，value属性的参数值必须与选项内容相同或基本相同，该属性是必选项；
 - checked属性用于指定该选项在初始时是否被选中，值为true时默认选中。

表单类标签

- <input type="radio" name="" value="">
- 当type=radio时：是一个单选项，即用户只能选中表单中所有单选项中的一项作为输入信息，
 - 由于只能允许用户选择其中一项，所以所有选项的name属性都应取相同的值；
 - 不同的选项其属性value的值应是不同的；
 - checked属性用于指定该选项在初始时是被选中的。
- <input type="hidden" name="" value="">
 - 当type=hidden时：表示输入项将不在浏览器中显示。也就是说不能接受用户的输入。
 - 属性值name、value同前述几种形式。

表单标签实例

- <html>
- <body>
- <form>
- 我喜欢自行车: <input type="checkbox" name="Bike">

- 我喜欢汽车: <input type="checkbox" name="Car">

- 我是男生:
 <input type="radio" checked="checked" name="Sex" value="male" />

- 我是女生: <input type="radio" name="Sex" value="female" />
- <input type="hidden" name="age" value="19">
- </form>
- </body>
- </html>

我喜欢自行车:

我喜欢汽车:

我是男生:

我是女生:

前两项是复选框

后两项是单选框

最后一个input是一个隐藏项，在页面上不会显示。

表单类标签

- <input type="file" name=" " >
- 当type="file"时：是文件选择输入框，用户可以选择要提交的文件。
 - Accept：用来设置浏览文件时默认能看到的文件类型
 - Disabled:设置当前状态不能用
 - Readonly:设置当前不接受用户的输入
- 注意：
 - 当表单中有这项时<form>标签中的enctype属性值必须为multipart/form-data，这是指定form编码的数据方式。
 - 此表单项不能设置初始值。

表单类标签

- <input type="submit" name="button" value="提交">
- 当type=submit时：产生一个提交按钮，当用户单击该按钮时，浏览器就会将表单的输入信息传送给服务器。
 - 提交按钮的name属性是可以默认的。除name属性外，它还有一个可选的属性value，用于指定显示在提交按钮上的文字，value属性的默认值是“submit”。在一个表单中必须有提交按钮，否则将无法向服务器传送信息，disabled设置当前状态不能用；
- <input type="reset" name="button" value="重置">
- 当type=reset时：产生一个重置按钮，当用户单击该按钮时，浏览器就会清除表单中所有的输入信息而恢复到初始状态。一般情况下，提交与重置按钮经常同时出现。
 - 重置按钮的name属性也是可以默认的。value属性与submit类似，用于指定显示在清除按钮上的文字，value的默认值为“reset”，disabled设置当前状态不能用。

表单标签实例

- <html>
- <body>
- <form action="/example/html/form_action.asp" method="get">
- <p>姓名<input type="text" name="name" /></p>
- <p>密码<input type="password" name="pass" /></p>
- <p>上传照片<input type="file" name="pic" /></p>
- <input type="submit" value="提交" />
- <input type="reset" value="重置" />
- </form>
- </body>
- </html>



姓名 张三

密码

上传照片 [选择文件] 未选择任何文件

提交 重置

- 第三个input用来让用户选择文件
- 提交按钮可以将表单内容提交给后台
- 重置按钮可以将表单内填好的内容清空。

表单类标签

- 菜单列表项标签
- 描述：下拉式的列表或带有滚动条的列表，用户可以在列表中选中一个或多个选项。这一点与<input>标签中的单选框和多选框的使用方法相似，只是形式不同。
- 格式：

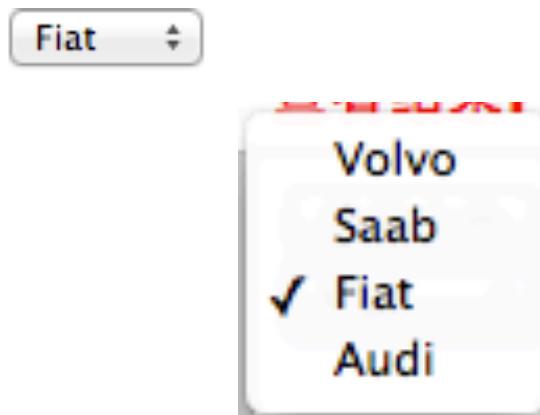
```
<select name="">
  <option selected>菜单项</option>
  <option>菜单项</option>
  ...
</select>
```
- 属性：
 - name：设定下拉列表名字。
 - size：可选项，用于改变列表框的大小。size属性的值是数字，表示显示在列表框中选项的数目，当size属性的值小于列表框中的列表项数目时，浏览器会为该列表框添加滚动条，用户可以使用滚动条来查看所有的选项。size默认值为1。
 - multiple：如果加上该属性，表示允许用户从列表中选择多项。

表单类标签

- **<option>标签**
- **描述**：用来定义列表中的选项，设置列表中显示的文字和列表条目的值，列表中每个选项有一个显示的文本和一个value值（当选项被选择时传送给处理程序的数据）。
- **说明**：`<option>`必须嵌套在`<select>`标签中使用。一个列表中有多少个选项，就要有多少个`<option>`标签与之相对应，选项的具体内容写在每个`<option>`之后，它也是一个双标签，用`</option>`结束一个选项。
- **属性**：
 - `value`: 设置当该选项被选中并提交后，浏览器传送给服务器的数据。如果是默认状态，浏览器将传送选项的内容。
 - `selected` :指定选项的初始状态，表示该选项在初始时被选中。

表单标签实例

- <html>
 - <body>
 - <form>
 - <select name="cars">
 - <option value="volvo">Volvo</option>
 - <option value="saab">Saab</option>
 - <option value="fiat" selected="selected">Fiat</option>
 - <option value="audi">Audi</option>
 - </select>
 - </form>
 - </body>
 - </html>
- select是一个下拉列表，列表中有四项，默认选项是fiat。



表单类标签

- **文本输入框标签**
- **格式**：`<textarea name="" rows="" cols=""></textarea>`
- **属性**：
 - `name`：指定文本输入框的名字。
 - `rows`：设置多行文本输入框的行数，此属性的值是数字，浏览器会自动为高度超过一行的文本输入框添加垂直滚动条。但是，当输入文本的行数小于或等于`rows`属性的值时，滚动条将不起作用。
 - `cols`：设置多行文本输入框的列数。
 - `wrap`：默认值(soft)是文本自动换行，当输入内容超过文本域的右边界时会自动转到下一行，而数据在被提交处理时自动换行的地方不会有换行符出现，其值有(hard、off、physical、soft、virtual)。

主要内容

- HTML实例
- HTML概述
- HTML常用标签
 - 网页结构类标签
 - 段落类标签
 - 列表类标签
 - 格式化类标签
 - 超链接类标签
 - 图像类标签
 - 表格类标签
 - 表单类标签
 - 框架类标签

框架窗口

- 框架技术
- 描述：

- 可以将浏览器分割成多个小窗口，并且在每个小窗口中，可以显示不同的网页。
- 当浏览器分割成多个窗口后，各窗口就会扮演不同的角色，实现不同的功能。
- 框架的基本结构主要分为**框架**和**框架集**两个部分。它是利用<frame>标签与<frameset>标签来定义。其中<frame>标签用于定义框架，而<frameset>标签则用于定义框架集。
- 框架窗口不能有<body></body>标签

框架窗口

- **框架技术**

- **格式：**

```
<frameset ...>
    <frame ... />
    <frame ... />
</frameset>
```



- **说明：**

- 常见的对窗口的分割包括：**水平分割、垂直分割和嵌套分割。**
具体采用哪种分割方式，取决于实际需要，可用`<frameset>`标签中的`rows`（水平分割）或`cols`（垂直分割）属性来进行分割。

框架窗口

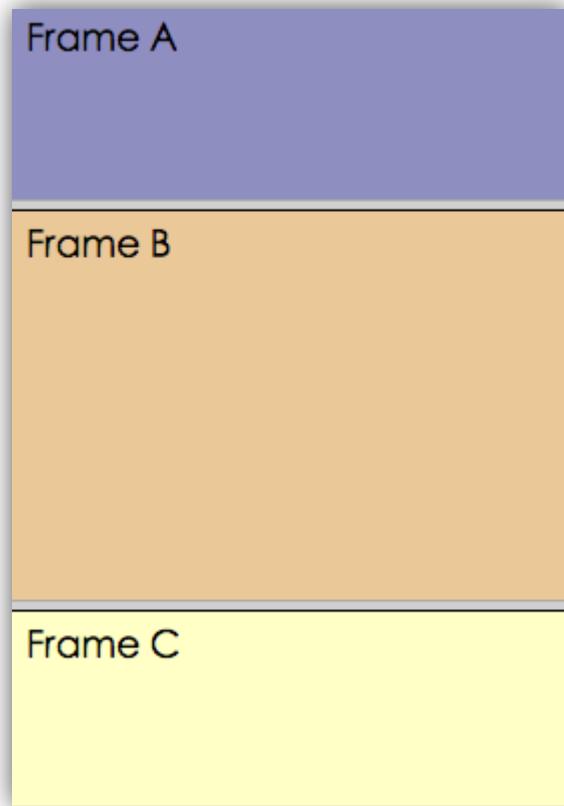
- **水平分割**：rows属性可定义一个水平分割的窗口框架。
- **格式：**

```
<frameset rows="高度1,高度2,...,*">  
  <frame src="url" />  
  <frame src="url" />  
  
  ...  
</frameset>
```

- **说明：**
 - rows属性的值代表各子窗口的高度，第一个子窗口高为高度1，第二个子窗口高为高度2，依此类推，而最后一个*，则代表最后一个子窗口的高度，值为其他子窗口高度分配后所剩余的高度。
 - 设置高度数值的方式有两种：
 - 采用整数设置，单位为像素（px），语法：`<frameset rows="100, 200, *">`
 - 用百分比设置，语法：`<frameset rows="20%, 50%, *">`

框架窗口实例

- <html>
 - <frameset rows="25%,50%,25%">
 - <frame src="/example/html/frame_a.html">
 - <frame src="/example/html/frame_b.html">
 - <frame src="/example/html/frame_c.html">
 - </frameset>
 - </html>
-
- 整个窗口分成了三个平行的区域



框架窗口

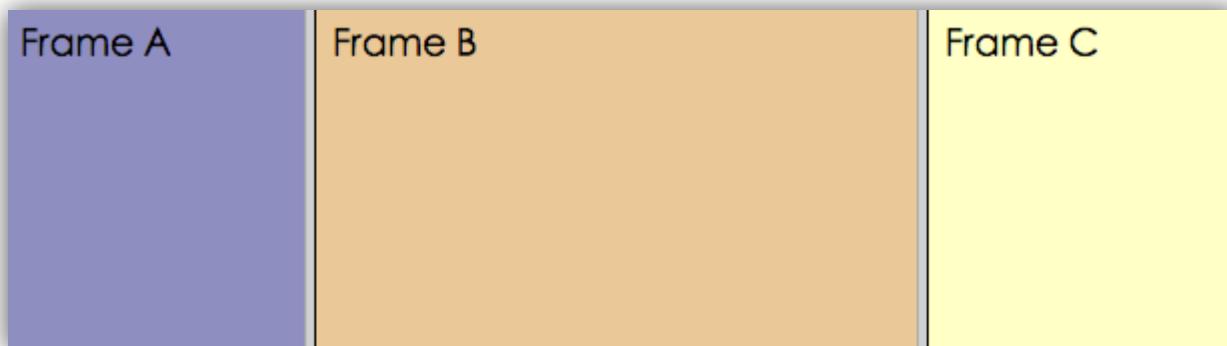
- **垂直分割**：cols属性可定义一个垂直分割的窗口框架。
- **格式：**

```
<frameset cols="宽度1,宽度2,...,*">  
  <frame src="url" />  
  <frame src="url" />  
  
  ...  
</frameset>
```

- **说明：**
 - 用法同水平分割

框架窗口实例

- <html>
 - <frameset cols="25%,50%,25%">
 - <frame src="/example/html/frame_a.html">
 - <frame src="/example/html/frame_b.html">
 - <frame src="/example/html/frame_c.html">
 - </frameset>
 - </html>
- 整个窗口分成了三个垂直的区域



框架窗口

- **嵌套分割**：在实际应用中，“厂”字型框架使用极为广泛。“厂”字型的网页就是窗口中包含水平和垂直分割。
- **格式：**

```
<frameset rows="30%,*">  
<frame />  
<frameset cols="20%,*">  
    <frame />  
    <frame />  
</frameset>  
</frameset>
```

- **说明：**
 - 用法同水平分割



框架窗口

- <frameset>标签
- 属性：
 - Frameborder设置窗口框架的周围是否显示框架，(no)0表示不显示，(yes)1表示显示边框。
 - Framespacing 框架与框架之间的距离，值为整数，单位为像素(px)，当frameborder设置为no时此属性无效
 - border：设置分割窗口框架的边框宽度，值为整数，单位为像素(px)，如果framespacing为0时此属性不起作用。
 - Bordercolor 设置框架边框的颜色。

框架窗口

- <frame />标签
- 属性：
 - Src:指定子窗口中显示的页面
 - Name:用来指定窗口的名称，可用来做为超链接的target属性值，以指定在该窗口中打开目标文件。
 - scrolling属性用于控制窗口框架中是否显示滚动条，使用此属性，可以避免HTML文件因内容过多而无法完全显示，其值为yes|no|auto。
 - Noresize : <frameset>标签的rows和cols属性来水平或垂直分割窗口。但设置后，各窗口框架的大小并非固定无法更改，当我们想更改窗口框架大小时，可以将鼠标指针移到要更改的框架上，待指针呈现双箭头符号时，再用鼠标进行拖拽，就可调整窗口的大小，当然我们不希望用户能随意地改变子窗口的大小，那么这时我们可以在<frame>标签中，添加noresize属性。