

本文存下来作为备忘。

HTML HTML 指 **超文本标签语言**。

XHTML XHTML 是更严谨更纯净的 HTML 版本。

CSS CSS 指层叠样式表（Cascading Style Sheets）。

TCP/IP TCP/IP 是针对因特网的通信协议。

什么是 HTML 文件？

- HTML 指超文本标签语言。
- HTML 文件是包含一些标签的文本文件。
- 这些标签告诉 WEB 浏览器如何显示页面。
- HTML 文件必须使用 htm 或者 html 作为文件扩展名。
- HTML 文件可以通过简单的文本编辑器来创建。

HTML 文件中的第一个标签是 <html>。这个标签告诉浏览器这个 HTML 文件的开始点。文件中最后一个标签是 </html>。这个标签告诉您的浏览器，这是 HTML 文件的结束点。

位于 <head> 标签和 </head> 标签之间的文本是头信息。头信息不会显示在浏览器窗口中。

<title> 标签中的文本是文件的标题。标题会显示在浏览器的标题栏。

<body> 标签中的文本是将被浏览器显示出来的文本。

HTML文档是由 HTML 元素构成的文本文件。

HTML元素是通过使用 HTML 标签进行定义的。

HTML 标签

- HTML 标签是用来 **标记 HTML 元素** 的。
- HTML 标签 **被 < 和 > 符号包围**。
- 这些包围的符号叫作。
- HTML 标签是 **成对出现** 的。例如 和 。
- 位于 **起始标签** 和 **终止标签** 之间的文本是元素的内容。
- HTML 标签 **对大小写不敏感**， 和 的作用的相同的。

HTML中最重要的标签是定义标题、段落和换行的标签。

标题

标题使用 <h1> 至 <h6> 标签进行定义。<h1> 定义最大的标题。<h6> 定义最小的标题。HTML 会自动在标题 **前后添加 一个 额外的 折行**。

段落

段落使用 <p> 标签进行定义。HTML 会自动在段落 **前后添加 一个 额外的 空行**。

换行符

当你打算结束一行，而又不想开始一个新段落时，
 标签就派上用场了。无论你将它置于何处，
 标签都会产生一个强制的换行。

 标签是空白标签，由于关闭标签没有任何意义，因此它没有类似 </br> 的终止标签。

HTML 注释

注释标签用于在 HTML 源码中插入注释。注释会被浏览器忽略。您可以使用注释对您的代码进行解释，这么做在以后的日子里会对您的代码编辑产生帮助。

<!-- This is a comment -->

注意：左括号后需要写一个惊叹号，右括号前就不需要了。

基本的 HTML 标签

标签	描述
<html>	定义 HTML 文档。
<body>	定义文档的主体。
<h1> to <h6>	定义标题 1 至标题 6。
<p>	定义段落。
 	插入折行。
<hr>	定义水平线。
<!-->	定义注释。

HTML 标签的属性

HTML 标签拥有属性。**属性为 HTML 元素提供附加信息。**

属性总是以 **名称/值对** 的形式出现，比如：name="value"

属性总是在 HTML 元素的 **开始标签中** 规定。

使用小写属性

属性和属性值对大小写敏感。不过，万维网联盟在其 HTML 4 推荐标准中推荐小写的属性/属性值，而 XHTML 要求使用小写属性/属性值。

始终为属性值加引号

属性值应该始终被包括在引号内。双引号是最常用的，不过使用单引号也没有问题。

文本格式化标签

标签	描述
	定义粗体文本。
<big>	定义大号字。
	定义着重文字。
<i>	定义斜体字。
<small>	定义小号字。
	定义加重语气。
<sub>	定义下标字。
<sup>	定义上标字。
<ins>	定义插入字。
	定义删除字。
<s>	<i>不赞成使用。</i> 使用 代替。
<strike>	<i>不赞成使用。</i> 使用 代替。
<u>	<i>不赞成使用。</i> 使用样式（style）代替。

“计算机输出”标签

标签	描述
<code>	定义计算机代码。
<kbd>	定义键盘码。

<samp>	定义计算机代码样本。
<tt>	定义打字机代码。
<var>	定义变量。
<pre>	定义预格式文本。
<listing>	不赞成使用。使用 <pre> 代替。
<plaintext>	不赞成使用。使用 <pre> 代替。
<xmp>	不赞成使用。使用 <pre> 代替。

引用、引用和术语定义

标签	描述
<abbr>	定义缩写。
<acronym>	定义首字母缩写。
<address>	定义地址。
<bdo>	定义文字方向。
<blockquote>	定义长的引用。
<q>	定义短的引用语。
<cite>	定义引用、引证。
<dfn>	定义一个定义项目。

字符实体

一些字符在 HTML 中拥有特殊的含义，比如小于号 (<) 用于定义 HTML 标签的开始。如果我们希望浏览器正确地显示这些字符，我们必须在 HTML 源码中插入字符实体。

字符实体 有三部分：一个 **和号 (&)**，一个 **实体名称**，或者 **#** 和一个 **实体编号**，以及一个 **分号 (;)**。

要在 HTML 文档中显示小于号，我们需要这样写：< 或者 <

使用实体名称而不是实体编号的好处在于，名称相对来说更容易记忆。而这么做的坏处是，并不是所有的浏览器都支持最新的 **实体名称**，然而几乎所有的浏览器对 **实体编号** 的支持都很好。

注意：实体对大小写敏感。

最常用的字符实体

显示结果	描述	实体名称	实体编号
	空格	 	
<	小于号	<	<
>	大于号	>	>
&	和号	&	&
"	引号	"	"
'	撇号	' (IE不支持)	'

其他一些常用的字符实体

显示结果	描述	实体名称	实体编号
¢	分	¢	¢
£	镑	£	£
¥	日圆	¥	¥
§	节	§	§

©	版权	©	©
®	注册商标	®	®
×	乘号	×	×
÷	除号	÷	÷

锚标签和 Href 属性

HTML使用 <a>（锚）标签来创建连接另一个文档的链接。

锚可以指向网络上的任何资源：一张 HTML 页面，一幅图像，一个声音或视频文件等等。

Target 属性

使用 Target 属性，你可以定义被链接的文档在何处显示。

锚标签和 Name 属性

Name 属性用于 **创建被命名的锚**（named anchors）。当使用命名锚 (named anchors) 时, 我们可以创建 **直接跳至页面中某个节** 的链接，这样使用者就无需不停的滚动页面来寻找他们需要的信息。

链接标签

标签	描述
<a>	定义锚。

框架

通过使用框架，你可以在同一个浏览器窗口中显示不止一个页面。每份HTML文档称为一个框架，并且每个框架都独立于其他的框架。

使用框架的坏处：

- 开发人员必须同时跟踪更多的HTML文档
- 很难打印整张页面

框架结构标签（<frameset>）

- 框架结构标签（<frameset>）定义如何将窗口分割为框架
- 每个 frameset 定义了一系列行 或 列
- rows/columns 的值规定了每行或每列占据屏幕的面积

编者注：frameset标签也被某些文章和书籍译为框架集。

框架标签（Frame）

Frame 标签定义了放置在每个框架中的 HTML 文档。

基本的注意事项 - 有用的提示：

假如一个框架有可见边框，用户可以 **拖动** 边框来改变它的大小。为了 **避免** 这种情况发生，可以在<frame> 标签中加入：**noresize="noresize"**。

为不支持框架的浏览器添加 **<noframes>标签**。

重要提示： 不能将 <body></body> 标签与 <frameset></frameset> 标签同时使用!不过,假如你添加包含一段文本的 <noframes> 标签,就必须将这段文字嵌套于 <body></body> 标签内。

表格

表格由 <table> 标签来定义。每个表格均有若干行（由<tr> 标签定义），每行被分割为若干单元格（由 <td> 标签定义）。字母 td 指表格数据（table data），即数据单元格的内容。数据单元格可以包含文本、图片、列表、段落、表单、水平线、表格等等。

表格和边框属性

如果不用定义边框属性，表格将不显示边框。有时这很有用，但是大多数时候，我们希望显示边框。

基本的注意事项 - 有用的提示：

<thead>, <tbody> 和<tfoot>很少被用到，这是由于浏览器对它们的支持不太好。希望这种情况在未来版本的 XHTML 中会有所改观。

表格标签

表格	描述
<table>	定义表格
<caption>	定义表格标题。
<th>	定义表格的 表头。
<tr>	定义表格的行。
<td>	定义表格单元。
<thead>	定义表格的页眉。
<tbody>	定义表格的主体。
<tfoot>	定义表格的页脚。
<col>	定义用于表格列的属性。
<colgroup>	定义表格列的组。

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 。

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。

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。

列表标签

标签	描述
	定义有序列表。
	定义无序列表。
	定义列表项。
<dl>	定义定义列表。
<dt>	定义定义项目。
<dd>	定义定义的描述。
<dir>	已废弃。使用 代替它。
<menu>	已废弃。使用 代替它。

表单

表单是一个包含表单元素的区域。

表单元素是允许用户在表单中（比如：文本域、下拉列表、单选框、复选框等等）输入信息的元素。表单使用表单标签（<form>）定义。

输入

多数情况下被用到的表单标签是输入标签（<input>）。输入类型是由 **类型属性（type）** 定义的。大多数经常被用到的输入类型如下：

文本域（Text Fields）

注意，表单本身并不可见。同时，在大多数浏览器中，文本域的缺省宽度是 **20 个字符**。

单选按钮（Radio Buttons）

当用户从若干给定的的选择中选取其一时，就会用到单选框。

注意，只能从中选取其一。

复选框（Checkboxes）

当用户需要从若干给定的选择中选取一个或若干选项时，就会用到复选框。

表单的动作属性（Action）和确认按钮

当用户单击确认按钮时，表单的内容会被传送到另一个文件。表单的动作属性定义了目的文件的文件名。由动作属性定义的这个文件通常会对接收到的输入数据进行相关的处理。

表单标签

标签	描述
<form>	定义供用户输入的表单
<input>	定义输入域
<textarea>	定义文本域 (一个多行的输入控件)
<label>	定义一个控制的标签
<fieldset>	定义域
<legend>	定义域的标题
<select>	定义一个选择列表
<optgroup>	定义选项组
<option>	定义下拉列表中的选项
<button>	定义一个按钮
<isindex>	已废弃。有<input>代替。

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

alt 属性 用来为图像定义一串 **预备的可替换的文本**。替换文本属性的值是由用户定义的。 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。此时，浏览器将显示这个替代性的文本而不是图像。为页面上的图像都加上替换文本属性是个好习惯，这样有助于更好的显示信息，并且对于那些使用纯文本浏览器的人来说是非常有用的。

图像标签

标签	描述
	定义图像。
<map>	定义图像地图。
<area>	定义图像地图中的可点击区域。

背景颜色（Bgcolor）

背景颜色属性将背景设置为某种颜色。属性值可以是十六进制数、RGB 值或颜色名。

```
<body
bgcolor="#000000"
>
```

```
<body
bgcolor="rgb(0,0,0)"
>
```

```
<body
bgcolor="black"
>
```

以上的代码均将背景颜色设置为黑色。

背景（Background）

背景属性将背景设置为图像。属性值为图像的URL。如果图像尺寸小于浏览器窗口，那么图像将在整个浏览器窗口进行复制。

颜色由 **红色、绿色、蓝色** 混合而成。

颜色值

颜色由一个十六进制符号来定义,这个符号由红色、绿色和蓝色的值组成 (RGB)。每种颜色的最小值是 0 (十六进制:#00)。最大值是255 (十六进制:#FF)。

在HTML中, **字体标签(font)**是不被支持的。一般都认为,在今后版本的HTML中,这个标签会被清除出去。即使很多人都在用它,我们都应该尽量避免它,而使用**样式**取而代之。**正确的方法 - 使用样式**

通过使用HTML4.0,所有的格式化代码均可移出HTML文档,然后移入一个独立的样式表即可。

如何使用样式

当浏览器读到一个样式表,它就会按照这个样式表来对文档进行格式化。有以下三种方式来插入样式表：

外部样式表

当样式需要被应用到很多页面的时候,外部样式表将是理想的选择。使用外部样式表,你可以通过更改一个文件来改变整个站点的外观。

```
<head>

<link rel="stylesheet" type="text/css" href="mystyle.css">

</head>
```

内部样式表

当单个文件需要特别样式时,就可以使用内部样式表。你可以在head部分通过<style>标签定义内部样式表。

```
<head>

<style type="text/css">

body {background-color: red}

p {margin-left: 20px}

</style>

</head>
```

内联样式

当特别的样式需要被应用到个别的元素时,就可以使用内联样式。使用内联样式的方法是在相关的标签中使用样式属性。样式属性可以包含任何CSS属性。以下实例显示出如何改变段落的颜色和左边界。

```
<p
style="color: red; margin-left: 20px"
>
```

This is a paragraph

```
</p>
```

标签	描述
<style>	定义样式定义。

<link>	定义资源索引。
<div>	定义文档中的一个节。
	定义文档中的一个节。
	不赞成使用。请使用样式。
<basefont>	不赞成使用。请使用样式。
<center>	不赞成使用。请使用样式。

头元素

头元素包含关于文档的概要信息，也成为元信息（meta-information）。Meta意为“关于某方面的信息”。可以这么说，元数据（meta-data）是关于数据的信息，或者说元信息是关于信息的信息。

头元素内的信息

头元素内的元素不会被浏览器显示出来。

根据HTML的标准，仅仅几个标签在HTML的头部分是合法的。它们是：<base>, <link>, <meta>, <title>,<style>,<script>。

Head标签

标签	描述
<head>	定义关于文档的信息。
<title>	定义文档标题。
<base>	定义页面中所有链接的基准URL。
<link>	定义资源索引。
<meta>	定义元信息。

标签	描述
<!DOCTYPE>	定义文档类型。此标签须位于html标签之前。

文章标签：[html](#) [学习](#) [标签](#) [xml](#)

个人分类：[J2EE](#)

此PDF由[spygg](#)生成,请尊重原作者版权!!!

我的邮箱:liushidc@163.com