

01 HTML简单介绍

HTML介绍

HTML 是用来描述网页的一种语言。

- HTML 指的是超文本标记语言 (Hyper Text Markup Language)
- HTML 不是一种编程语言，而是一种标记语言 (markup language)
- 标记语言是一套标记标签 (markup tag)
- HTML 使用标记标签来描述网页

HTML书写规范

HTML 标记标签通常被称为 HTML 标签 (HTML tag)。

- HTML 标签是由尖括号包围的关键词，比如 `<html>`
- HTML 标签通常是成对出现的，比如 `` 和 ``
- 标签对中的第一个标签是开始标签，第二个标签是结束标签
- 开始和结束标签也被称为开放标签和闭合标签
- 绝大多数标签都具有属性，建议属性值用引号引起，如：`<body id="content">`
- 大多数标签是可以嵌套使用的，只允许包裹嵌套，不允许交叉嵌套
- HTML大小写不敏感。建议使用小写

HTML创建

HTML文件可以使用记事本，Dreamweaver等编辑器来进行创建，保存时后缀名为html或者htm，整个文件在`<html>`和`</html>`标签之间，html标签内有`<head>`和`<body>`等子标签。如：

```
<html>
  <head></head>
  <body>
    <!-- 页面显示内容 -->
  </body>
</html>
```

`<!-- 注释 -->` 为html中的注释

02 HTML的常用标签

基础标签(结构类标签)

基础标签

`<!DOCTYPE>`标签

定义与作用

`<!DOCTYPE>` 声明必须是 HTML 文档的第一行，位于 `<html>` 标签之前。

<!DOCTYPE> 声明不是 HTML 标签；它是指示 web 浏览器关于页面使用哪个 HTML 版本进行编写的指令。

在 HTML 4.01 中，<!DOCTYPE> 声明引用 DTD，因为 HTML 4.01 基于 SGML。DTD 规定了标记语言的规则，这样浏览器才能正确地呈现内容。

HTML5 不基于 SGML，所以不需要引用 DTD。

常用的 DOCTYPE 声明：

HTML 5

```
<!DOCTYPE html>
```

HTML 4.01 Strict

该 DTD 包含所有 HTML 元素和属性，但不包括展示性的和弃用的元素（比如 font）。不允许框架集（Framesets）。

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
```

HTML 4.01 Transitional

该 DTD 包含所有 HTML 元素和属性，包括展示性的和弃用的元素（比如 font）。不允许框架集（Framesets）。

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
```

HTML 4.01 Frameset

该 DTD 等同于 HTML 4.01 Transitional，但允许框架集内容。

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN"
"http://www.w3.org/TR/html4/frameset.dtd">
```

<html>标签

此元素可告知浏览器其自身是一个 HTML 文档。

<html> 与 </html> 标签限定了文档的开始点和结束点，在它们之间是文档的头部和主体。正如您所了解的那样，文档的头部由 <head> 标签定义，而主体由 <body> 标签定义。

<head>标签

<head> 标签用于定义文档的头部，它是所有头部元素的容器。<head> 中的元素可以引用脚本、指示浏览器在哪里找到样式表、提供元信息等等。

下面这些标签可用在 head 部分：<base>，<meta>，<link>，<script>，<style>，以及 <title>。

<title> 定义文档的标题，它是 head 部分中唯一必需的元素。

<base>标签

<base> 标签为页面上的所有链接规定默认地址或默认目标。

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。

<base> 标签必须位于 head 元素内部。

base属性

href：规定页面中所有相对链接的基准 URL，必选。

target：在何处打开页面中所有的链接。

```
<base href="http://www.baidu.com"/>
<base target="_blank"/>
```

<meta>标签

<meta> 元素可提供有关页面的元信息（meta-information），比如针对搜索引擎和更新频度的描述和关键词。

<meta> 标签位于文档的头部，不包含任何内容。<meta> 标签的属性定义了与文档相关联的名称/值对。

meta属性

content：定义与 http-equiv 或 name 属性相关的元信息，必选

http-equiv：把 content 属性关联到 HTTP 头部。

name：把 content 属性关联到一个名称。

charset：定义文档的字符编码。

scheme：定义用于翻译 content 属性值的格式。HTML5不支持。

```
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<meta name="keywords" content="HTML,ASP,PHP,SQL">
```

<title> 标签

<title> 元素可定义文档的标题

<title> 标签是 <head> 标签中唯一要求包含的东西

```
<head>
<title>文档标题</title>
</head>
```

<link>标签

<link> 标签定义文档与外部资源的关系。

<link> 标签最常见的用途是链接样式表。

link属性

href：规定被链接文档的位置。

rel：规定当前文档与被链接文档之间的关系。

type：规定被链接文档的 MIME 类型。

```
<head>
<link rel="stylesheet" type="text/css" href="theme.css"/>
</head>
```

<script>标签

<script> 标签用于定义客户端脚本，比如 JavaScript。

script 元素既可以包含脚本语句，也可以通过 src 属性指向外部脚本文件。

必需的 type 属性规定脚本的 MIME 类型。

script属性

type：指示脚本的 MIME 类型。

charset：规定在外部脚本文件中使用的字符编码。

src：规定外部脚本文件的 URL。

xml:space：规定是否保留代码中的空白。

```
<head>
  <script type="text/javascript"src="js/script.js"></script>
  <script type="text/javascript">
    alert("内部导入，弹出框");
  </script>
</head>
```

<style>标签

<style> 标签用于为 HTML 文档定义样式信息。

type 属性是必需的，定义 style 元素的内容。唯一可能的值是 "text/css"。

```
<head>
  <style type="text/css">
    h1{color:red}
    p{color:blue}
  </style>
</head>
```

<body>标签

body 元素定义文档的主体

```
<body>
  文档的内容... ..
</body>
```


标签

 可插入一个简单的换行符。

 标签是空标签（意味着它没有结束标签，因此这是错误的：
</br>）。在 XHTML 中，把结束标签放在开始标签中，也就是
。

<h1> to <h6>标签

<h1> - <h6> 标签可定义标题。<h1> 定义最大的标题。<h6> 定义最小的标题

```
<h1>This is heading 1</h1>
<h2>This is heading 2</h2>
<h3>This is heading 3</h3>
<h4>This is heading 4</h4>
<h5>This is heading 5</h5>
<h6>This is heading 6</h6>
```

<hr />标签

<hr> 标签在 HTML 页面中创建一条水平线。

水平分隔线可以在视觉上将文档分隔成各个部分。

在 HTML 中，<hr> 标签没有结束标签。

在 XHTML 中，<hr> 必须被正确地关闭，比如 <hr />。

<p>标签

<p> 标签定义段落。

```
<p>这是段落内容</p>
```

案例展示

```
<!DOCTYPE html>
<html>
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8"/>
  <title>基础标签</title>
</head>
<body>
```

```
<!-- br换行 -->
hello, <br/>java
<!-- 水平线 -->
<hr>
<!-- h1 to h6 -->
<h1>标题1</h1>
<h2>标题2</h2>
<h3>标题3</h3>
<h4>标题4</h4>
<h5>标题5</h5>
<h6>标题6</h6>
<!-- p标签 -->
<p>段落1</p><p>段落2</p>
</body>
</html>
```

图片

标签

img 元素向网页中嵌入一幅图像

在 HTML 中， 标签没有结束标签。

在 XHTML 中， 标签必须被正确地关闭。

img属性

alt：规定图像的替代文本。

src：规定显示图像的 URL。

height：定义图像的高度。

ismap：将图像定义为服务器端图像映射。

usemap：将图像定义为客户器端图像映射。

width：设置图像的宽度。

```

```

链接

<a>标签

<a> 标签定义超链接，用于从一张页面链接到另一张页面。

<a> 元素最重要的属性是 href 属性，它指示链接的目标。

a属性

href：规定链接指向的页面的 URL。

target：规定在何处打开链接文档。

综合案例

```
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type"content="text/html; charset=UTF-8"/>
<title>链接标签</title>
</head>
<body>
<a href="http://www.baidu.com" target="_blank">这是一个a标签的跳转链接</a>
</body>
</html>
```

列表

<dd>标签

<dd> 在定义列表中定义条目的定义部分。

<dt>标签

<dt> 标签定义了定义列表中的项目。

<dl>标签

<dl> 标签定义了定义列表（ definition list ）。

<dl> 标签用于结合 <dt>（ 定义列表中的项目 ）和 <dd>（ 描述列表中的项目 ）

```
<dl>
  <dt>水果</dt>
    <dd>苹果</dd>
    <dd>西瓜</dd>
  <dt>蔬菜</dt>
    <dd>白菜</dd>
    <dd>西红柿</dd>
</dl>
```

标签

 标签定义列表项目。

 标签可用在有序列表 () 和 unordered 列表 () 中。

标签

 标签定义有序列表

ol属性

reversed：规定列表顺序为降序。(9,8,7...)

start：规定有序列表的起始值。

type：规定在列表中使用的标记类型。(1 , A , a , I , i)

表标签

 标签定义无序列表

```
<body>
<p>无序列表: </p>
<ul>
<li>雪碧</li>
<li>可乐</li>
<li>橙汁</li>
</ul>
<hr>
<p>有序列表: </p>
<ol type="I">
<li>睁开眼睛</li>
<li>穿上衣服</li>
<li>洗漱上班</li>
</ol>
</body>
```

表格类标签

<table>标签

<table> 标签定义 HTML 表格。

简单的 HTML 表格由 table 元素以及一个或多个 tr、th 或 td 元素组成。

tr 元素定义表格行，th 元素定义表头，td 元素定义表格单元。

table属性

border：规定表格边框的宽度。

cellspacing：规定单元格之间的空白。

frame：规定外侧边框的哪个部分是可见的。

rules：规定内侧边框的哪个部分是可见的。

width：规定表格的宽度。

<th>标签

定义表格内的表头单元格。

th 元素内部的文本通常会呈现为居中的粗体文本

<td>标签

<td> 标签定义 HTML 表格中的标准单元格。

HTML 表格有两类单元格：

表头单元 - 包含头部信息（由 th 元素创建）

标准单元 - 包含数据（由 td 元素创建）

td 元素中的文本一般显示为正常字体且左对齐。

td 属性

abbr：规定单元格中内容的缩写版本。

align：规定单元格内容的水平对齐方式。

bgcolor：规定单元格的背景颜色(不赞成使用。请使用样式取而代之)

colspan:规定单元格可横跨的列数。

height:规定表格单元格的高度(不赞成使用。请使用样式取而代之)

rowspan：规定单元格可横跨的行数。

valign：规定单元格内容的垂直排列方式。

width：规定表格单元格的宽度(不赞成使用。请使用样式取而代之)

<tr>标签

<tr> 标签定义 HTML 表格中的行

tr 元素包含一个或多个 th 或 td 元素

tr 属性

align：定义表格行的内容对齐方式。

bgcolor：规定表格行的背景颜色(不赞成使用。请使用样式取而代之)

valign：规定表格行中内容的

<thead>标签

<thead> 标签定义表格的表头。该标签用于组合 HTML 表格的表头内容。

thead 元素应该与 tbody 和 tfoot 元素结合起来使用。

tbody 元素用于对 HTML 表格中的主体内容进行分组，而 tfoot 元素用于对 HTML 表格中的表注（页脚）内容进行分组。

<thead> 内部必须拥有 <tr> 标签！

<tbody>标签

<tbody> 标签表格主体（正文）。该标签用于组合 HTML 表格的主体内容。

<tbody> 内部必须拥有 <tr> 标签！

<tfoot>标签

<tfoot> 标签定义表格的页脚（脚注或表注）。该标签用于组合 HTML 表格中的表注内容。

<tfoot> 内部必须拥有 <tr> 标签！

<caption>标签

caption 元素定义表格标题。

caption 标签必须紧随 table 标签之后。您只能对每个表格定义一个标题。通常这个标题会被居中于表格之上。

综合案例

```
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8"/>
<title>表格类标签</title>
</head>

<body>
<table border="1" width="90%" cellspacing="0" rules="rows">
<caption>商品列表显示</caption>
<thead>
  <tr bgcolor="green" align="center">
    <th>商品编号</th>
    <th>商品名称</th>
    <th>种类</th>
    <th>价格</th>
  </tr>
</thead>
<tbody>
  <tr>
    <td>001</td>
    <td>白菜</td>
    <td rowspan="2">蔬菜类</td>
    <td align="right">1.2</td>
  </tr>
  <tr>
    <td>002</td>
    <td>西红柿</td>
    <td align="right">3.5</td>
  </tr>
  <tr>
    <td>003</td>
    <td>白条肉</td>
    <td>肉类</td>
    <td align="right">15.6</td>
  </tr>
</tbody>
<tfoot>
  <tr>
    <td colspan="4">总共3种商品</td>
  </tr>
</tfoot>
</table>
</body>
</html>
```

表单类标签

<form>标签

<form> 标签用于为用户输入创建 HTML 表单。

表单能够包含 input 元素，比如文本字段、复选框、单选框、提交按钮等等。

表单还可以包含textarea、label等元素。

表单用于向服务器传输数据。

form属性

accept-charset：规定服务器可处理的表单数据字符集。

action：规定当提交表单时向何处发送表单数据。

autocomplete：规定是否启用表单的自动完成功能。

enctype：规定在发送表单数据之前如何对其进行编码。

enctype 属性可能的值：

application/x-www-form-urlencoded

multipart/form-data

text/plain

method：规定用于发送 form-data 的 HTTP 方法。

name：规定表单的名称。

novalidate：如果使用该属性，则提交表单时不进行验证。

target：规定在何处打开 action URL。

<input >标签

<input> 标签用于搜集用户信息。

根据不同的 type 属性值，输入字段拥有很多种形式。输入字段可以是文本字段、复选框、掩码后的文本控件、单选按钮、按钮等等。

input标签基于H5增加许多日期类型的控件，具体可参见type属性，

input属性

checked：规定此 input 元素首次加载时应当被选中。

disabled：当 input 元素加载时禁用此元素。

max：规定输入字段的最大值。

maxlength：规定输入字段中的字符的最大长度。

min：规定输入字段的最小值。

- name：定义 input 元素的名称。
- pattern：规定输入字段的值的模式或格式。
- placeholder：规定帮助用户填写输入字段的提示。
- readonly：规定输入字段为只读。
- required：指示输入字段的值是必需的。
- value：规定 input 元素的值。
- type：规定 input 元素的类型。

type值介绍

Type值	说明	浏览器支持 (空表示所有浏览器)
button	定义可点击的按钮（通常与 JavaScript 一起使用来启动脚本）。	
checkbox	定义复选框。	
file	定义文件选择字段和 "浏览..." 按钮，供文件上传。	
hidden	定义隐藏输入字段。	
image	定义图像作为提交按钮。	
password	定义密码字段（字段中的字符会被遮蔽）	
radio	定义单选按钮。	
reset	定义重置按钮（重置所有的表单值为默认值）。	
submit	定义提交按钮。	
text	默认。定义一个单行的文本字段（默认宽度为 20 个字符）。	

综合案例

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>input标签</title>
  </head>
  <body>
    <table>
      <tr>
        <td align="right">按钮</td>
        <td><input type="button" value="保存" onclick="save()"></td>
      </tr>
      <tr>
        <td align="right">复选框</td>
        <td><input type="checkbox" name="fav[]" value="basketball"> 篮球<br>
```

```

        <input type="checkbox"name="fav[]"value="football"> 足球</td>
    </tr>
    <tr>
        <td align="right">文件</td>
        <td><input type="file"name="fileImg"></td>
    </tr>
    <tr>
        <td align="right">隐藏框</td>
        <td><input type="hidden"name="id"value="001"></td>
    </tr>
    <tr>
        <td align="right">图片按钮</td>
        <td><input type="image"src="img_submit.gif"alt="submit"></td>
    </tr>

    <tr>
        <td align="right">密码框</td>
        <td><input type="password"name="pwd"></td>
    </tr>
    <tr>
        <td align="right">性别</td>
        <td><input type="radio"name="gender"value="male"> 男<br>
            <input type="radio"name="gender"value="female">女</td>
    </tr>
    <tr>
        <td align="right">重置按钮</td>
        <td><input type="reset"value="重置"></td>
    </tr>
    <tr>
        <td align="right">提交按钮</td>
        <td><input type="submit"></td>
    </tr>
    <tr>
        <td align="right">文本框</td>
        <td><input type="text"name="username"></td>
    </tr>
</table>
</body>
</html>

```

<label>标签

<label> 标签为 input 元素定义标注（标记）。

label 元素不会向用户呈现任何特殊效果。不过，它为鼠标用户改进了可用性。如果您在 label 元素内点击文本，就会触发此控件。就是说，当用户选择该标签时，浏览器就会自动将焦点转到和标签相关的表单控件上。

<label> 标签的 for 属性应当与相关元素的 id 属性相同。

label属性

for：规定 label 绑定到哪个表单元素。

```
<label for="dept">所在部门:</label>
```

<button>标签

<button> 标签定义一个按钮。

这是该元素与使用 input 元素创建的按钮之间的不同之处在于，元素内部可以防止文本或图像。

button属性

autofocus：规定当页面加载时按钮应当自动地获得焦点。

disabled：规定应该禁用该按钮。

name：规定按钮的名称。

type：规定按钮的类型(button,reset,submit)

value：规定按钮的初始值。可由脚本进行修改。

```
<button type="button">Click Me!</button>
```

<option>标签

option 元素定义下拉列表中的一个选项（一个条目）。

option 元素位于 select 元素内部。

option属性

selected：规定选项（在首次显示在列表中时）表现为选中状态。

value：定义送往服务器的选项值。

<select>标签

select 元素可创建单选或多选菜单。

<select> 元素中的 <option> 标签用于定义列表中的可用选项。

select属性

disabled：规定禁用该下拉列表。

multiple：规定可选择多个选项。

name：规定下拉列表的名称。

required：规定文本区域是必填的。

size：规定下拉列表中可见选项的数目。

```
<select name="dept">
  <option value="0">--select--</option>

  <option value="1">develop</option>

  <option value="2" selected>test</option>

  <option value="3">QA</option>

</select>
```

<textarea>标签

<textarea> 标签定义多行的文本输入控件。

可以通过 cols 和 rows 属性来规定 textarea 的尺寸。

textarea属性

cols：规定文本区内的可见宽度。

disabled：规定禁用该文本区。

maxlength：规定文本区域的最大字符数。

name：规定文本区的名称。

placeholder：规定描述文本区域预期值的简短提示。

readonly：规定文本区为只读。

required：规定文本区域是必填的。

rows：规定文本区内的可见行数。

wrap：规定当在表单中提交时，文本区域中的文本如何换行。

```
<textarea rows="4"cols="20"name="remark"id="remark">
个人介绍。。。
</textarea>
```

综合案例

```
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type"content="text/html; charset=UTF-8"/>
<title>表单类标签</title>
</head>
<body>
  <form method="post"action="#">
    <fieldset>
      <legend>用户注册</legend>
      <table>
        <tr>
```

```

        <td align="right"><label for="username">用户名: </label></td>
        <td><input type="text" name="username" id="uername"></td>
    </tr>
    <tr>
        <td align="right"><label for="password">密码: </label></td>
        <td><input type="password" name="password" id="password"></td>
    </tr>
    <tr>
        <td align="right"><label>性别: </label></td>
        <td><label for="man">男</label><input
type="radio" name="sex" id="man">
        <label for="woman">女</label><input
type="radio" name="sex" id="woman"></td>
    </tr>
    <tr>
        <td><label>专业爱好: </label></td>
        <td><label for="java">java语言</label><input
type="checkbox" name="major" id="java">
        <label for="bigN">大数据</label><input
type="checkbox" name="major" id="bigN">
        <label for="python">python语言</label><input
type="checkbox" name="major" id="python">
        <label for="C++">C++程序设计</label><input
type="checkbox" name="major" id="C++"></td>
    </tr>
    <tr>
        <td><label for="dept">所在部门:</label></td>
        <td><select name="dept">
            <option value="0">--请选择--</option>
            <option value="1">开发部</option>
            <option value="2" selected>测试部</option>
            <option value="3">人事部</option>
        </select></td>
    </tr>
    <tr>
        <td><label for="photo">上传头像</label></td>
        <td><input type="file" name="photo" id="photo"></td>
    </tr>
    <tr>
        <td><label for="remark">个人介绍</label></td>
        <td><textarea rows="4" cols="20" name="remark" id="remark">
            </textarea></td>
    </tr>
    <tr>
        <td colspan="2"><input type="submit" value="注册">
            <input type="reset" value="重置">
            <input type="button" value="返回"></td>
    </tr>
</table>
</fieldset>
</form>
</body>
</html>

```

其他标签

<div>标签

<div> 可定义文档中的分区或节 (division/section) 。

<div> 标签可以把文档分割为独立的、不同的部分。

<div> 是一个块级元素。这意味着它的内容自动地开始一个新行。可以通过 <div> 的 class 或 id 应用额外的样式

标签

 用于对文档中的行内元素进行组合。

 标签没有固定的格式表现。当对它应用样式时，它才会产生视觉上的变化。如果不对 应用样式，那么 元素中的文本与其他文本不会任何视觉上的差异。

 标签提供了一种将文本的一部分或者文档的一部分独立出来的方式。

<iframe>标签

iframe 元素会创建包含另外一个文档的内联框架 (即行内框架) 。

iframe属性

frameborder：规定是否显示框架周围的边框。

height：规定 iframe 的高度。

marginheight：定义 iframe 的顶部和底部的边距。

marginwidth：定义 iframe 的左侧和右侧的边距。

name：规定 iframe 的名称。

scrolling：规定是否在 iframe 中显示滚动条。

src：规定在 iframe 中显示的文档的 URL。

srcdoc：规定在 <iframe> 中显示的页面的 HTML 内容(IE不支持)。

width：定义 iframe 的宽度。

综合案例

```
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type"content="text/html; charset=UTF-8"/>
<title>iframe标签</title>
</head>
<body>
<table width="80%">
<tr>
<td><a href="基础标签.html"target="mainFrame">基础标签<a></td>
<td><a href="图片标签.html"target="mainFrame">图片标签<a></td>
<td><a href="列表标签.html"target="mainFrame">列表标签<a></td>
<td><a href="格式类标签.html"target="mainFrame">格式类标签<a></td>
<td><a href="表格类标签.html"target="mainFrame">表格类标签<a></td>
<td><a href="表单类标签.html"target="mainFrame">表单类标签<a></td>
</tr>
```

```
<tr>
  <td colspan="6">
    <div class="content">
      <iframe
name="mainFrame"frameborder="0"style="width:100%;height:550px;"src="基础标
签.html"></iframe>
    </div>
  </td>
</tr>
</table>
</body>
</html>
```