

一. Form表单

form 标签用于为用户输入创建 HTML 表单，用于向服务器传输数据。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 select、textarea、checked、selected。

1. form表单常用属性

- enctype: 设置表单编码方式，常用的值有application/x-www-form-urlencoded（默认）、multipart/form-data（文件上传）
- method: 设置表单请求方式，常用的值有：post、get
- action: 设置表单提交的地址

表单使用示例：

```
<!-- form表单
      action: 点击表单中的提交按钮，表单访问的java程序的路径
      method: 请求方式    get:get请求    post:post请求
      enctype: 传递数据的类型
              application/x-www-form-urlencoded: 表单提交数据
              multipart/form-data : 上传文件的类型，图片、文件
-->
<form action="https://www.baidu.com" method="get" enctype="application/x-www-
form-urlencoded">
    <!-- input: 输入框
            type:输入框的类型    text:文本类型    password:密码类型    submit:提交
按钮类型

            name:输入框的内容的名称
            value : 输入框的内容
    -->
    <p>用户名: <input type="text" name="name" value="张三"/></p>
    <p>密码: <input type="password" name="password" value="123456"/></p>
    <p><input type="submit" value="提交登录"/></p>
</form>
```

2. 表单常用组件

2.1. 文本框

```
<input name="xx" type="text" />
```

2.2. 密码框

```
<input name="xx" type="password" />
```

2.3. 提交按钮

```
<input type="submit" value="登录" />
```

2.4. 重置按钮

```
<input type="reset" value="重置" />
```

2.5. 复选框

<!-- 复选框

可以选择多个选项，将多个选项的内容提交给后台服务器java程序

注意：

1. type必须是checkbox，
2. 所有选项的name的值必须一致，
3. 需要设置选项的value属性的值，才能发送正确的数据给后台服务器java程序

为什么选项的name要保持一致呢？

就是为了让选项成为一组

传递的数据：hobby=篮球&hobby=足球&hobby=羽毛球

可以变种成：hobby:[篮球，足球，羽毛球]

-->

爱好：

```
<input type="checkbox" name="hobby" value="篮球" checked/>:篮球
```

```
<input type="checkbox" name="hobby" value="足球"/>:足球
```

```
<input type="checkbox" name="hobby" value="羽毛球"/>:羽毛球
```

2.6. 单选框

性别：

```
<input name="sex" type="radio" checked/>男
```

```
<input name="sex" type="radio"/>女
```

```
<input name="sex" type="radio"/>保密
```

2.7. 下拉框

<!-- 下拉框，默认显示第一个option内容

name属性是select和option的

value:option中如果没有value,默认提供展示的文本内容，如果设置了value,就提交value内容，value的值就是内容的标识符

disabled：设置下拉框不可用

selected：select默认展示的option

-->

性别：

```
<select name="sex">
```

```
<option value="-1">性别</option>
```

```
<option value="0" selected>男</option>
```

```
<option value="1">女</option>
```

```
<option value="2">其他</option>
```

```
</select>
```

2.8. 多行文本框

<!-- cols: 规定文本区域内可见的宽度。rows: 规定文本区域内可见的行数。 -->

```
<textarea name="bigtxt" cols="5" rows="5"></textarea>
```

不建议使用rows和cols属性，建议使用css的width和height来控制多行文本框大小

2.9. 上传文件

```
<input type="file" accept="image/jpeg" name="file"/>
```

3. HTML5中新增的表单元素

- email

```
<input type="email" name="user_email" />
```

- number

```
<!-- min: 最小值    max: 最大值 -->
<input type="number" name="points" min="1" max="10" />
```

- range

```
<!-- 进度条 -->
<input type="range" name="points" min="1" max="10" />
```

- 日期选择 (date, month, week, time, datetime, datetime-local)

```
<input type="date" name="user_date" />
```

HTML5 拥有多个可供选取日期和时间的新输入类型：

- date - 选取日、月、年
 - month - 选取月、年
 - week - 选取周和年
 - time - 选取时间（小时和分钟）
 - datetime-local - 选取时间、日、月、年（本地时间）
- 颜色选择器

颜色选择器用来让用户通过调色板自主选择颜色

```
<input type="color" name="select_color"/>
```

4. HTML5新增的表单属性

autocomplete 属性

autocomplete 属性规定 form 或 input 域应该拥有自动完成功能。

当用户在自动完成域中开始输入时，浏览器应该在该域中显示填写的选项。

提示：autocomplete 属性有可能在 form 元素中是开启的，而在input元素中是关闭的。

HTML form 中开启 autocomplete（一个 input 字段关闭 autocomplete）：

```
<form action="" autocomplete="on">
  First name:<input type="text" name="fname"><br>
  Last name: <input type="text" name="lname"><br>
  E-mail: <input type="email" name="email" autocomplete="off"><br>
  <input type="submit">
</form>
```

novalidate 属性

novalidate 属性是一个 boolean(布尔) 属性。

novalidate 属性规定在提交表单时不应该验证 form 或 input 域。

无需验证提交的表单数据：

```
<form action="demo-form.php" novalidate>
  E-mail: <input type="email" name="user_email">
  <input type="submit">
</form>
```

placeholder 属性

placeholder 属性提供一种提示，描述输入域所期待的值。

简短的提示在用户输入值前会显示在输入域上。

注意：placeholder 属性适用于以下类型的 `<input>` 标签：text, search, url, tel, email 以及 password。

required 属性

required 属性规定必须在提交之前填写输入域（不能为空）。

注意：required 属性适用于以下类型的 `<input>` 标签：text, search, url, tel, email, password, date, number, checkbox, radio 以及 file。

step 属性

step 属性为输入域规定合法的数字间隔。

如果 step="3"，则合法的数是 -3,0,3,6 等

注意：step 属性与以下type类型一起使用：number, range, date, datetime-local, month, time 和 week。

autofocus 属性

autofocus 属性是一个 boolean 属性。

autofocus 属性规定在页面加载时，域自动地获得焦点。

form 属性

form 属性规定输入域所属的一个或多个表单。

```
<form action="demo-form.php" id="form1">
  First name: <input type="text" name="fname"><br>
  <input type="submit" value="提交">
</form>

Last name: <input type="text" name="lname" form="form1">
```

list 属性

list 属性规定输入域的 datalist。

min 和 max 属性

min、max 和 step 属性用于为包含数字或日期的 input 类型规定限定（约束）。

注意： min、max 和 step 属性适用于以下类型的 `<input>` 标签：date、number 以及 range。

`<input>` 元素最小值与最大值设置：

```
Enter a date before 1980-01-01:
<input type="date" name="bday" max="1979-12-31">

Enter a date after 2000-01-01:
<input type="date" name="bday" min="2000-01-02">

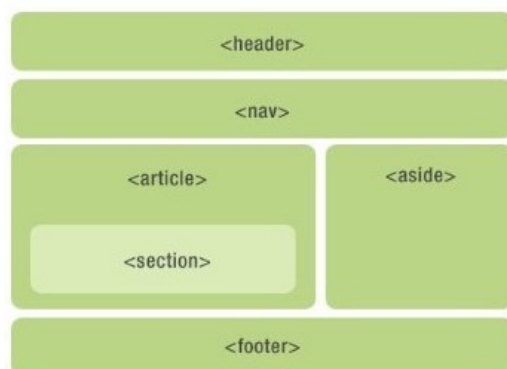
Quantity (between 1 and 5):
<input type="number" name="quantity" min="1" max="5">
```

二. HTML5新增布局标签

许多现有网站都包含以下HTML代码：`<div id="nav">`、`<div class="header">` 或者 `<div id="footer">`，来指明导航链接，头部，以及尾部。

HTML5新增的还有article、nav、header、footer.....等，其实现效果都是和div一样没有效果，但是合理使用却能让页面结构更加清晰有逻辑。

HTML5 提供了新的语义元素来明确一个Web页面的不同部分：



1. HTML5 header 元素

`<header>` 元素描述了文档的头部区域。

原来写一个页面，比如有头部和底部。我们都是定义一个div，只不过头部有一个class=header、尾部的div是class=footer；

现在html5出现以后，主张语义化标签。现在我们写样式的时候，选择器直接就写header就可以了。不需要这么写 `div.header` 这样的好处是使得文档结构层次清晰、利于代码编写和开发、还有搜索引擎优化等等的好处。

2. HTML5 nav 元素

`<nav>` 标签定义导航链接的部分。

`<nav>` 元素用于定义页面的导航链接部分区域，但是，不是所有的链接都需要包含在 `<nav>` 元素中。

```
<nav>
  <a href="/html/">HTML</a> |
  <a href="/css/">CSS</a> |
  <a href="/js/">JavaScript</a> |
  <a href="/jquery/">jQuery</a>
</nav>
```

3. HTML5 article 元素

`<article>` 标签定义独立的内容。

article元素代表文档、页面或应用程序中独立的、完整的、可以独自被外部引用的内容。它可以是一篇**博客或报刊中的文章、一篇论坛帖子、一段用户评论**或独立的插件，或其他任何独立的内容。除了内容部分，一个article元素通常有它自己的标题（一般放在一个header元素里面），有时还有自己的脚注。

注：article元素是可以嵌套使用的，内层的内容在原则上需要与外层的内容相关联。例如，一篇博客文章中，针对该文章的评论就可以使用嵌套article元素的方式；用来呈现评论的article元素被包含在表示整体内容的article元素里面。

4. HTML5 section 元素

`<section>` 标签定义文档中的节（section、区段）。

根据W3C HTML5文档：section 包含了一组内容及其标题。

5. HTML5 aside 元素

`<aside>` 标签定义页面主区域内容之外的内容（比如侧边栏）。

aside 标签的内容应与主区域的内容相关。

6. HTML5 footer 元素

`<footer>` 元素描述了文档的底部区域。

`<footer>` 元素应该包含它的包含元素。

一个页脚通常包含文档的作者，著作权信息，链接的使用条款，联系信息等。

7. HTML5 figure 和 figcaption 元素

`<figure>` 标签规定独立的流内容（图像、图表、照片、代码等等）。

`<figure>` 元素的内容应该与主内容相关，但如果被删除，则不应影响文档流产生影响。

`<figcaption>` 标签定义 `<figure>` 元素的标题。

`<figcaption>` 元素应该被置于 "figure" 元素的第一个或最后一个子元素的位置。

```
<figure>
  
  <figcaption>Fig1. - The Pulpit Pock, Norway.</figcaption>
</figure>
```

三. HTML5多媒体标签

1. HTML5 video(视频)

video - 如何工作

如需在 HTML5 中显示视频，你所有需要的是：

```
<video width="320" height="240" controls>
  <source src="movie.mp4" type="video/mp4">
  <source src="movie.ogg" type="video/ogg">
  您的浏览器不支持Video标签。
</video>
```

`<video>` 元素提供了 播放、暂停和音量控件来控制视频。

- 同时 `<video>` 元素也提供了 `width` 和 `height` 属性控制视频的尺寸，如果设置的高度和宽度，所需的视频空间会在页面加载时保留。如果没有设置这些属性，浏览器不知道大小的视频，浏览器就不能再加载时保留特定的空间，页面就会根据原始视频的大小而改变。
- `<video>` 与 `</video>` 标签之间插入的内容是提供给不支持 video 元素的浏览器显示的。
- `<video>` 元素支持多个 `<source>` 元素。 `<source>` 元素可以链接不同的视频文件，浏览器将使用第一个可识别的格式。

视频格式与浏览器的支持

当前， `<video>` 元素支持三种视频格式： MP4， WebM， 和 Ogg：

浏览器	MP4	WebM	Ogg
Internet Explorer	YES	NO	NO
Chrome	YES	YES	YES
Firefox	YES	YES	YES
Safari	YES	NO	NO
Opera	YES (从 Opera 25 起)	YES	YES

视频格式：

格式	MIME-type
MP4	video/mp4
WebM	video/webm
Ogg	video/ogg

video标签属性

属性	值	描述
muted	muted	如果出现该属性，视频的音频输出为静音。
autoplay	autoplay	如果出现该属性，则视频在就绪后马上播放。
controls	controls	添加播放控制及音量控制功能栏。
height	Pixels	指定播放器的高度，以pixel为单位。
loop	loop	如果指定，视频将重复播放。
poster	url	指定视频的预览图。
preload	auto metadata none	如果指定，视频会在加页面加载过程中被加载。当autoplay被指定时，会被忽略。 auto - 当页面加载后载入整个音频 meta - 当页面加载后只载入元数据 none - 当页面加载后不载入音频
src	url	目标视频的URL。
width	pixels	指定播放器的宽度，以pixel为单位。

2. HTML5 audio(音频)

audio - 如何工作

如需在 HTML5 中播放音频，你需要使用以下代码：

```
<audio controls>
  <source src="horse.ogg" type="audio/ogg">
  <source src="horse.mp3" type="audio/mpeg">
  您的浏览器不支持 audio 元素。
</audio>
```

control 属性供添加播放、暂停和音量控件。

- 在 `<audio>` 与 `</audio>` 之间你需要插入浏览器不支持的 `<audio>` 元素的提示文本。
- `<audio>` 元素允许使用多个 `<source>` 元素。 `<source>` 元素可以链接不同的音频文件，浏览器将使用第一个支持的音频文件。

音频格式及浏览器支持

目前， `<audio>` 元素支持三种音频格式文件：MP3，Wav，和 Ogg：

浏览器	MP3	Wav	Ogg
Internet Explorer 9+	YES	NO	NO
Chrome 6+	YES	YES	YES
Firefox 3.6+	YES	YES	YES
Safari 5+	YES	YES	NO
Opera 10+	YES	YES	YES

音频格式的MIME类型

Format	MIME-type
MP3	audio/mpeg
Ogg	audio/ogg
Wav	audio/wav

audio标签属性

属性	属性值	描述
autoplay	autoplay	如果指定, 音频会在准备好后立即播放.
controls	controls	显示播放控制工具栏..
loop	loop	如果指定, 则循环播放.
muted	muted	如果出现该属性, 则音频输出为静音。
preload	auto metadata none	如果指定, 视频会在加页面加载过程中被加载。当autoplay被指定时, 会被忽略。 auto - 当页面加载后载入整个音频 meta - 当页面加载后只载入元数据 none - 当页面加载后不载入音频
src	url	目标音频的URL.

3. source标签

video和audio元素的子元素, 可指定多个文件来源, 用来解决浏览器的兼容问题。

如果使用了source元素, 则不可以在video和audio中设置src属性。

四. 标签总结

HTML标签可以按照语法结构和特性两个维度来分类。

1. 按照语法结构划分

- 单标签

、<hr>、、<input>、<param>、<meta>、<link>

- 双标签

<html>、<head>、<title>、<body>、<table>、<tr>、<td>、、<p>、<form>、<h1>、<h2>、<h3>、<h4>、<h5>、<h6>、、、<i>、<div>、<a>、、、

2. 按照特性来划分

- 行元素

特性：

1. 行元素不会独占一行
2. 行元素支持通过css设置宽、高

代表元素：、<a>、
、、、、<input>、<textarea>、<select>

- 块元素

特性：

1. 块元素会独占一行
2. 行元素支持通过css设置宽、高

代表元素：<div>、<body>、<table>、<p>、<form>、<h1>、<h2>、<h3>、<h4>、<h5>、<h6>、、、

- 行内块元素

特性：

1. 不会独占一行
2. 支持通过css设置宽、高