用来方便查找

表单的目的是为了收集用户信息

一个完整的表单通常由表单域，表单控件（也叫表单元素），提示信息三个部分构成

表单域是一个包含表单元素的区域

<form></form>标签用于定义表单域，以实现用户信息的收集和传递

form会把它范围内的表单元素信息提交给服务器

<form action="要提交到哪里" method="提交方式"name="表单域名字">

</form>

表单元素：在表单域中可以定义各种表单元素，这些表单元素就是允许用户在表单中输入或者选择的内容控件

input输入表单元素

select下拉表单元素

textarea文本域元素

input标签是一个单标签，是输入的意思，在表单元素中用来收集用户信息

<input type="属性值">

在input标签中，包含一个type属性，这个type是用来根据不同的type属性来选择各种形式的控件（可以是文本字段，复选框，按钮）

text文本

button按钮

checkbox复选框

file 输入字段和"浏览"按钮，供文件上传

hidden 隐藏输入字段

image图片形式的提交按钮

password密码字段

radio单选按钮

reset重置按钮会清除表单中的所有数据

submit提交按钮会把表单数据发送到服务器

input除了type属性外还有

name定义input元素名，单选框和复选框要有相同的name值；

value规定input元素的值，在text文本中可以看到直接效果，其他的都是传给后台 ,

checked规定input元素首次加载时应当被选中，主要针对单选按钮和复选按钮，默认选择 checked="checked"

maxlength规定输入字段中字符的最大长度 maxlength="长度"

<label></label>标签为input元素定义标注

label标签用于绑定一个表单元素，当点击label标签内的文本时，浏览器就会自动将焦点（光标）转到或者选择对应的表单元素上，用来增加用户体验

语法：

<label for="sex">要点击的内容</label>

<input type="radio"name=""id="sex">

label标签的for属性要和相关元素的id属性相同

<select>表单元素：

有多个选项选择是，想要节省网页空间，就可以使用select标签定义下拉列表

select语法：

<select>

<option>选项1</option>

<option>选项2</option>

<option>选项3</option>

</select>

在option中定义selected="selected"时，当前项默认选中项

<textarea>表单元素：

当用户输入内容较多的情况下，就不能使用文本框表单了，要用<textarea>

textarea标签是用于多行文本输入的控件

使用多行文本输入控件，可以输入更多的文字，该控件常见于留言板，评论

textarea语法：

<textarea name="#" id="" cols="30" rows="10"> //框的大小cols是框的列 rows是框的行

</textarea>

<!-- action 要提交的到哪里--> <!--method用什么方式传送-->

<form action="form.js" method="POST"name="名字">

<div>

<label>你的名字</label><!--标注-->

<!--type 是选择框的形式--> <!--name是用什么名字传输--> <!--placeholder是框里的字-->

<input type="text" name="name"placeholder="输入你的名字">

<label>你的性别</label>

<input type="radio" name="sex">男

<input type="radio" name="sex">女

<input type="email" name="email" placeholder="你的邮箱">

</div>

<input type="submit" name="submit" value="按钮">

</form>

列表有三种无序列表，有序列表，自定义列表

列表是用来布局的

ul标签标示无序列表，一般以项目符号呈现列表项，而列表项使用li标签定义

<ul>

<li></li>

</ul>

无序列表的个列表项之间没有顺序级别之分，是并列的

ul中只能嵌套li，直接在ul标签中输入其他标签或其他文字的做法是不被允许的

li之间相当于一个容器，可以容纳所有元素

ol标签表示有序列表，一般以项目符号呈现列表项，而列表项使用li标签定义

<ol>

<li></li>

</ol>

ol中只能嵌套li，直接在ol标签中输入其他标签或其他文字的做法是不被允许的

li之间相当于一个容器，可以容纳所有元素

自定义列表

dl标签用于定义描述列表，该标签会与dt标签（定义项目或名字）和dd标签（描述项目或名字）一起使用

<dl>

<dt></dt>

<dd><dd>

<dd><dd>

</dl>

dl中只能嵌套dt和dd，直接在ol标签中输入其他标签或其他文字的做法是不被允许的

dt和dd之间相当于一个容器，可以容纳所有元素

写在其他标签外面，不要写在里面

<a href="https://www.baidu.com">显示点击的名字</a>

在新的网页打开：

<a href="https://www.baidu.com"target="\_blank">显示点击的名字</a>

href是连接的路径

target是打开方式 默认为\_sel是在当前页面打开，\_blank是在新页面打开

内部连接 <a href="内部本地路径"></a>

空链接 如果当时没有确定链接的目标是什么时， <a href="#"></a>

下载链接 如果href里面地址是一个文件或者压缩包，点击就会下载这个文件

网页元素的链接 在网页中的各种网页元素，如文本，图像，表格，音频，视频等都可以添加插连接

以图片为例<a href=""><img src=""></img></a>

锚点链接 点击链接就可以快速定位到页面中的某个位置

<a href="#跳转名字"></a>

跳转目标位置写id="名字"好像只能在其他标签里用 例如<h1 id="跳转名字"></h1>

要写在别的标签外面，不要写在里面

<img src="" alt="" title=""> src是图片路径,可以是链接也可以是文件路径，alt是图片显示不出的时候的，提示文字,title是提示文本，鼠标放到这个位置，提示文字

width="" 设置宽度

height="" 设置高度

border="" 边框粗细

图像标签可以拥有多个属性，必须写在标签名后面

属性之间不分先后顺序， 标签名和属性，属性和属性之间均以空格分开

属性采取的格式 属性="属性值"

同一级路径 可以直接写文件名

下一级路径 / 进入下一级目录

上一级路径 ../ 返回上一级目录

绝对路径：

完整的文件夹所在位置

完整的网络地址

文字加粗 <string></string>或<b></b>

斜体<em></em>或<i></i>

删除线<del></del>或<s></s>

下划线<ins><ins>或<u></u>

div和span是没有语义的，就是一个盒子，用来装内容的

<div></div>

<span></span>

注释 <!-- -->

特殊字符

空格 &nbsp

< &lt

> &gt

其他的可以网上找

表格标签是实际开发中非常常用的标签

表格的主要作用 用来显示，展示数据的，因为可以让数据显示的非常规整

表格的基本语法

<table> //定义表格

<thead> //表格的头部

<tr>//行

<th>名字</th>//th是表头的单元格

<th>年龄</th>

<th>身高</th>

<th>体重</th>

</tr>

</thead>

<tbody>//表格的身体

<tr>//行

<td>Xman </td>//td是表格里的单元

<td>24</td>

<td>178</td>

<td>65kg</td>

</tr>

</tbody>

</table>

表格的属性，修改外观形态的，！！！这些属性要写到表格标签table里面去！！！

属性名 属性值 效果

align left，center，right 对齐方式，往哪边对齐

border 1或"" 规定表格单元是否拥有边框，默认是""表示没有

cellpadding 像素值 规定单元边沿与其内容之间的空白，默认像素为1

cellspacing 像素值 规定单元格之间的空白，默认像素为2

width 像素值或百分比 规定表格的宽度

跨行合并：rowspan="合并单元格的个数" 以要合并的最上侧单元格为目标单元格，在里面写合并代码

跨列合并：colspan="合并单元格的个数" 以要合并的最左侧单元格为目标单元格，在里面写合并代码

合并单元格三步曲

1 先确定要跨行还是跨列

2找到目标单元格，写上合并方式=“合并数量”

3删除被合并的单元格

HTML5 的新增特性主要是针对于以前的不足，增加了一些新的标签、新的表单和新的表单属性等。

这些新特性都有兼容性问题，基本是 IE9+ 以上版本的浏览器才支持，如果不考虑兼容性问题，可以大量使用这

些新特性。

新特性增加了很多，但是我们专注于开发常用的新特性

HTML5 新增的语义化标签

以前布局，我们基本用 div 来做。div 对于搜索引擎来说，是没有语义的。

HTML5 新增的语义化标签

<header>：头部标签

 <nav>：导航标签

 <article>：内容标签

 <section>：定义文档某个区域

 <aside>：侧边栏标签

 <footer>：尾部标签

这种语义化标准主要是针对搜索引擎的

 这些新标签页面中可以使用多次

 在 IE9 中，需要把这些元素转换为块级元素

 其实，我们移动端更喜欢使用这些标签

 HTML5 还增加了很多其他标签

1.2 HTML5 新增的多媒体标签

新增的多媒体标签主要包含两个：

1. 视频：<video>

2. 音频：<audio>

使用它们可以很方便的在页面中嵌入音频和视频，而不再去使用 flash 和其他浏览器插件。

HTML5 在不使用插件的情况下，也可以原生的支持视频格式文件的播放，当然，支持的格式是有限的。当前 <video> 元素支持三种视频格式： mp4，webm，ogg 尽量使用 mp4格式

HTML5 在不使用插件的情况下，也可以原生的支持音频格式文件的播放，当然，支持的格式是有限的。

音频：<audio> ：

语法：

<video src="文件地址" controls="controls"></video>

video的常见属性：autoplay=autoplay //视频自动播放，谷歌浏览器需要添加muted来解决自动播放问题

controls=controls //向用户显示播放控件

width=多少px //设置播放器的宽度

height=多少px //设置播放器的高度

loop=loop //是否循环播放

preload=auto //预先加载视频，如果有了autoplay就忽略这个属性

preload=none //不预先加载视频

src=url（）//视频路径

poster=imgurl //加载等待的画面图片

muted=muted //静音播放

HTML5 在不使用插件的情况下，也可以原生的支持音频格式文件的播放，当然，支持的格式是有限的。

音频<audio>：当前 <audio> 元素支持三种音频格式：mp3，wav，ogg

语法：

<audio src="文件地址" controls="controls"></audio>

常见属性： autoplay=autoplay //音频就绪后自动播放

controls=controls //向用户显示控件，比如播放按钮

loop=loop //循环播放

src=url //音频路径

谷歌浏览器把音频和视频自动播放禁止了

多媒体标签总结

音频标签和视频标签使用方式基本一致

 浏览器支持情况不同

 谷歌浏览器把音频和视频自动播放禁止了

 我们可以给视频标签添加 muted 属性来静音播放视频，音频不可以（可以通过JavaScript解决）

 视频标签是重点，我们经常设置自动播放，不使用 controls 控件，循环和设置大小属性

HTML5 新增的 input 类型

type="email" //限制用户输入必须为email类型

type="url" //限制用户输入必须为url类型

type="date" //限制用户输入必须为日期类型

type="time" //限制用户输入必须为时间类型

type="month" //限制用户输入必须为月份类型

type="week" //限制用户输入必须为周类型

type="number" //限制用户输入必须为数字类型

type="tel" //手机号码

type="search" //搜索框

type="color" //生成一个颜色选择表单

重点记住： number tel search 这三个

HTML5 新增的表单属性：

required=required //表单拥有该属性表示其内容不能为空，必填

placeholder="" //提示信息

autofocus=autofocus //自动聚焦属性，页面加载完成自动聚焦到指定表单

autocomplete=off/on //当用户在字段开始键入时，浏览器基于之前键入过的值，应该显示出在字段中填写的选项，默认已经打开on 关闭off ，需要放在表单内，同时加上name属性，同时成功提交

multiple=multiple //配合文件文本，可以多选文件提交

可以通过以下设置方式修改placeholder里面的字体颜色：

input::placeholder {

color: ;

}