

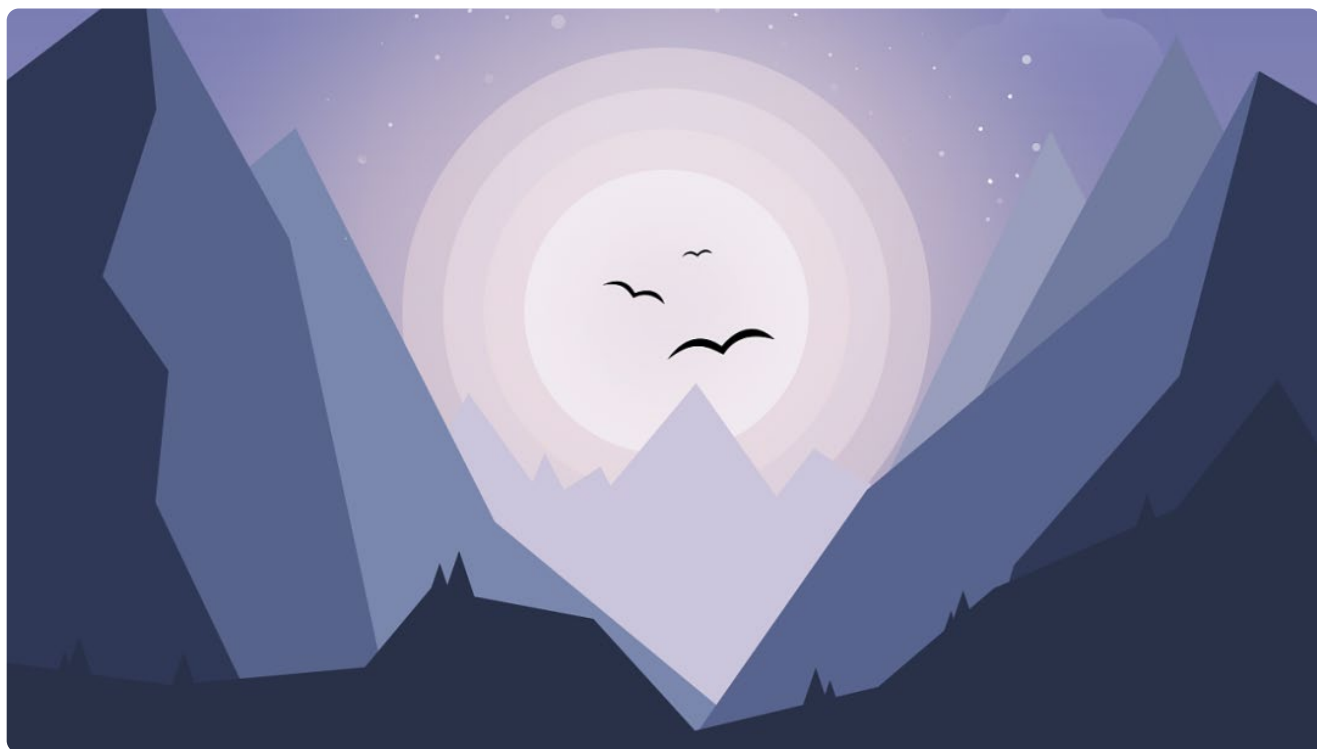
「学习笔记」HTML基础

原创 饭老板 前端fan 2020-09-07

收录于话题

#前端入门

4个



「学习笔记」HTML基础

前言

勤做笔记不仅可以让自己学的扎实，更重要的是可以让自己少走弯路。有人说："再次翻开笔记是什么感觉"，我的回答是："初恋般的感觉"。或许笔记不一定十全十美，但肯定会让你有种初恋般的怦然心动。◆◆◆◆

本章着重复习Html的基础内容，学习Html究竟要学些什么呢？主要是学习各种标签，来搭建网页的“结构”。

本篇文章主要由五个章节构成，从WEB标准到初识HTML，接着学习HTML常用标签，最后学习表格列表和表单。◆◆◆◆开始充电之旅啦~~~

一、认识WEB

「网页」主要是由 **文字**、**图像** 和 **超链接** 等元素构成，当然除了这些元素，网页中还可以包括音频、视频以及Flash等。

「浏览器」是网页显示、运行的平台。

「浏览器内核」(排版引擎、解释引擎、渲染引擎)

负责读取网页内容，整理讯息，计算网页的显示方式并显示页面。

浏览器	内核	备注
IE	Trident	IE、猎豹安全、360极速浏览器、百度浏览器
firefox	Gecko	可惜这几年已经没落了，打开速度慢、升级频繁、猪一样的队友fl ash、神一样的对手chrome。
Safari	webkit	现在很多人错误地把 webkit 叫做 chrome内核（即使 chrome内核已经是 blink 了）。苹果感觉像被别人抢了媳妇，都哭晕在厕所里面了。
chrome	Chromium/Blink	在 Chromium 项目中研发 Blink 渲染引擎（即浏览器核心），内置于 Chrome 浏览器之中。Blink 其实是 WebKit 的分支。大部分国产浏览器最新版都采用Blink内核。二次开发
Opera	blink	现在跟随chrome用blink内核。

Web标准

「构成」◆◆ 结构标准，表现标准和行为标准

- 结构标准用于对网页元素进行整理和分类(HTML)
- 表现标准用于设置网页元素的版式、颜色、大小等外观属性(CSS)
- 行为标准用于对网页模型的定义及交互的编写(JavaScript)

「Web标准的优点」◆◆

- 易于维护：只需更改CSS文件，就可以改变整站的样式
- 页面响应快：HTML文档体积变小，响应时间短
- 可访问性：语义化的HTML（结构和表现相分离的HTML）编写的网页文件，更容易被屏幕阅读器识别
- 设备兼容性：不同的样式表可以让网页在不同的设备上呈现不同的样式
- 搜索引擎：语义化的HTML能更容易被搜索引擎解析，提升排名

二、HTML初识

HTML初识

「HTML」(Hyper Text Markup Language):超文本标记语言

「所谓超文本，有2层含义：」

- 因为它可以加入图片、声音、动画、多媒体等内容（超越文本限制）
- 不仅如此，它还可以从一个文件跳转到另一个文件，与世界各地主机的文件连接（超级链接文本）。

「HTML骨架格式」



<!-- 页面中最大的标签 根标签 -->

<html>

<!-- 头部标签 -->

<head>

<!-- 标题标签 -->

<title></title>

</head>

<!-- 文档的主体 -->

<body>

</body>

</html>

「团队约定大小写」

- HTML标签名、类名、标签属性和大部分属性值统一用小写

「HTML元素标签分类」

- 常规元素(双标签)
- 空元素(单标签)



常规元素(双标签)

<标签名> 内容 </标签名> 比如<body>我是文字</body>

空元素(单标签)

<标签名 /> 比如
或

「HTML标签关系」

- **嵌套关系**父子级包含关系
- **并列关系**兄弟级并列关系
 - 如果两个标签之间的关系是**嵌套关系**，子元素最好缩进一个tab键的身位（一个tab是4个空格）。如果是**并列关系**，最好上下对齐。

文档类型<!DOCTYPE >

「**文档类型**」用来说明你用的XHTML或者HTML是什么版本。`<!DOCTYPE html>`告诉浏览器按照HTML5标准解析页面。

页面语言lang

lang指定该html标签内容所用的语言



```
<html lang="en">
```

en 定义语言为英语 zh-CN定义语言为中文

「lang的作用」

- 根据lang属性来设定不同语言的css样式，或者字体
- 告诉搜索引擎做精确的识别
- 让语法检查程序做语言识别
- 帮助翻译工具做识别
- 帮助网页阅读程序做识别

字符集

「**字符集**」(Character set)是多个字符的集合,计算机要准确的处理各种字符集文字，需要进行字符编码，以便计算机能够识别和存储各种文字。

- UTF-8是目前最常用的字符集编码方式
- 让 html 文件是以 UTF-8 编码保存的，浏览器根据编码去解码对应的html内容。



```
<meta charset="UTF-8" />
```

「meta viewport的用法」

通常viewport是指视窗、视口。浏览器上(也可能是一个app中的webview)用来显示网页的那部分区域。在移动端和pc端视口是不同的,pc端的视口是浏览器窗口区域,而在移动端有三个不同的视口概念:布局视口、视觉视口、理想视口

meta有两个属性name 和 http-equiv

name属性的取值

- keywords(关键字) 告诉搜索引擎, 该网页的关键字
- description(网站内容描述) 用于告诉搜索引擎, 你网站的主要内容。
- viewport(移动端的窗口)
- robots(定义搜索引擎爬虫的索引方式) robots用来告诉爬虫哪些页面需要索引, 哪些页面不需要索引
- author(作者)
- generator(网页制作软件)
- copyright(版权)

http-equiv有以下参数

http-equiv相当于http的文件头作用, 它可以向浏览器传回一些有用的信息, 以帮助正确和精确地显示网页内容

- content-Type 设定网页字符集(Html4用法, 不推荐)
- Expires(期限), 可以用于设定网页的到期时间。一旦网页过期, 必须到服务器上重新传输。
- Pragma(cache模式), 是用于设定禁止浏览器从本地机的缓存中调阅页面内容, 设定后一旦离开网页就无法从Cache中再调出
- Refresh(刷新), 自动刷新并指向新页面。
- cache-control (请求和响应遵循的缓存机制)



```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

HTML标签的语义化

- 方便代码的阅读和维护, 样式丢失的时候能让页面呈现清晰的结构。
- 有利于SEO, 搜索引擎根据标签来确定上下文和各个关键字的权重。
- 方便其他设备解析, 如盲人阅读器根据语义渲染网页

「拓展」 标签：规定页面上所有链接的默认 URL 和设置整体链接的打开状态



```
<head>
  <base href="http://www.baidu.com" target="_blank">
  <base target="_self">
</head>
<body>
  <a href="">测试</a> 跳转到 百度
</body>
```

HTML常用标签

常用标签

「1. 排版标签」主要和css搭配使用，显示网页结构的标签，是网页布局最常用的标签。

- 标题标签h(h1~h6)
- 段落标签p,可以把 HTML 文档分割为若干段落
- 水平线标签hr
- 换行标签br
- div和span标签:是没有语义的,是我们网页布局最主要的2个盒子。

「2. 排版标签」

- b和strong 文字以粗体显示
- i和em 文字以斜体显示
- s和del 文字以加删除线显示
- u和ins 文字以加下划线显示

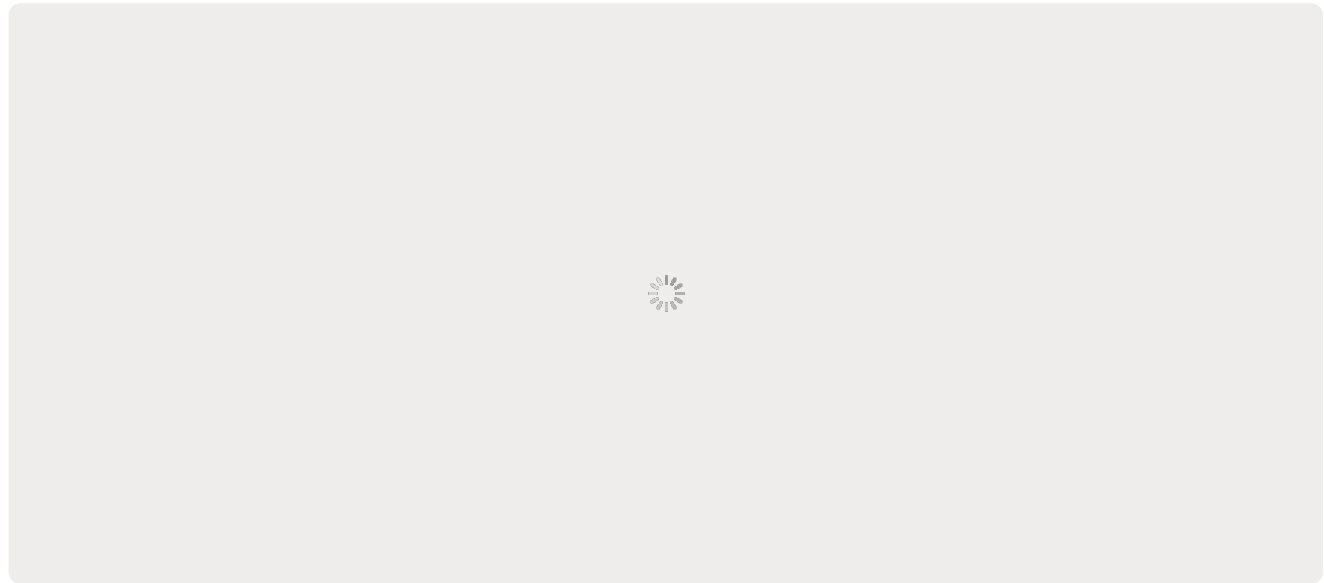
「3. 标签属性(行内式)」

使用HTML制作网页时，如果想让HTML标签提供更多的信息，可以使用HTML标签的属性加以设置。



```
<标签名 属性1="属性值1" 属性2="属性值2" ...> 内容 </标签名>
<手机 颜色="红色" 大小="5寸"> </手机>
```

「4. 图像标签img」



注意：

- 标签可以拥有多个属性，必须写在开始标签中，位于标签名后面。
- 属性之间不分先后顺序，标签名与属性、属性与属性之间均以空格分开。
- 采取 键值对 的格式 key="value" 的格式



```

```

「5. 链接标签(重点)」



```
<a href="跳转目标" target="目标窗口的弹出方式">文本或图像</a>
target="_self" 默认窗口弹出方式
target="_blank" 新窗口弹出
```

属性	作用
href	用于指定链接目标的url地址，（必须属性）当为标签应用href属性时，它就具有了超链接的功能
target	用于指定链接页面的打开方式，其取值有_self和_blank两种，其中_self为默认值，_blank为在新窗口中打开方式。

src 和 href 的区别

一句话概括:**src 是引入资源的 href 是跳转url的**

1. src用于替换当前元素, href用于在当前文档和引用资源之间确立联系。
2. src是source的缩写, 指向外部资源的位置, 指向的内容将会嵌入到文档中当前标签所在位置; 在请求src资源时会将其指向的资源下载并应用到文档内, 例如js脚本, img图片和frame等元素。当浏览器解析到该元素时, 会暂停其他资源的下载和处理, 直到将该资源加载、编译、执行完毕, 图片和框架等元素也如此, 类似于将所指向资源嵌入当前标签内。这也是为什么将js脚本放在底部而不是头部。
3. href是Hypertext Reference的缩写, 指向网络资源所在位置, 建立和当前元素(锚点)或当前文档(链接)之间的链接。如果我们在文档中添加那么浏览器会识别该文档为css文件, 就会并行下载资源并且不会停止对当前文档的处理。这也是为什么建议使用link方式来加载css, 而不是使用@import方式。

注意:

1. 外部链接 需要添加 http:// www.baidu.com
2. 内部链接 直接链接内部页面名称即可 比如 < a href="index.html"> 首页
3. 如果当时没有确定链接目标时, 通常将链接标签的href属性值定义为“#”(即href="#"), 表示该链接暂时为一个空链接。
4. 不仅可以创建文本超链接, 在网页中各种网页元素, 如图像、表格、音频、视频等都可以添加超链接。

锚点定位: 通过创建锚点链接, 用户能够快速定位到目标内容。



1. 使用相应的id名标注跳转目标的位置。(找目标)
2. 使用链接文本创建链接文本(被点击的)

```
<h3 id="two">第2集</h3>
```

```
<a href="#two">
```

「6. 注释标签」



```
<!-- 注释语句 -->
```

```
快捷键是:      ctrl + /  
或者 ctrl + shift + /
```

团队约定: 注释内容前后各一个空格字符, 注释位于要注释代码的上面, 单独占一行

「7. 路径」



「8. 其他知识」

预格式化文本pre标签元素中的文本通常会保留空格和换行符。而文本也会呈现为等宽字体。格式化文本就是，按照我们预先写好的文字格式来显示页面，保留空格和换行等。

特殊字符



什么是XHTML

- XHTML 指「可扩展超文本标签语言」(EXtensible HyperText Markup Language)。
- XHTML 的目标是取代 HTML。
- XHTML 与 HTML 4.01 几乎是相同的。
- XHTML 是更严格更纯净的 HTML 版本。
- XHTML 是作为一种 XML 应用被重新定义的 HTML,是严格版本的HTML。例如它要求标签必须小写, 标签必须被正确关闭, 标签顺序必须正确排列, 对于属性都必须使用双引号等。
- XHTML 是一个 W3C 标准。

写HTML代码时应注意什么？

- 尽可能少的使用无语义的标签div和span；
- 在语义不明显时，既可以使用div或者p时，尽量用p，因为p在默认情况下有上下间距，对兼容特殊终端有利；
- 不要使用纯样式标签，如：b、font、u等，改用css设置。
- 需要强调的文本，可以包含在strong或者em标签中（浏览器预设样式，能用CSS指定就不用他们），strong默认样式是加粗（不要用b），em是斜体（不用i）；
- 使用表格时，标题要用caption，表头用thead，主体部分用tbody包围，尾部用tfoot包围。表头和一般单元格要区分开，表头用th，单元格用td；
- 表单域要用fieldset标签包起来，并用legend标签说明表单的用途；
- 每个input标签对应的说明文本都需要使用label标签，并且通过为input设置id属性，在lable标签中设置for来让说明文本和相对应的input关联起来。

表格

「1. 表格」

现在还是较为常用的一种标签，但不是用来布局，常见显示、展示表格式数据。因为它可以让数据显示的非常的规整，可读性非常好。特别是后台展示数据的时候表格运用是否熟练就很重要，一个清爽简约的表格能够把繁杂的数据表现得很有条理。

「2. 创建表格」



```
<table>
```

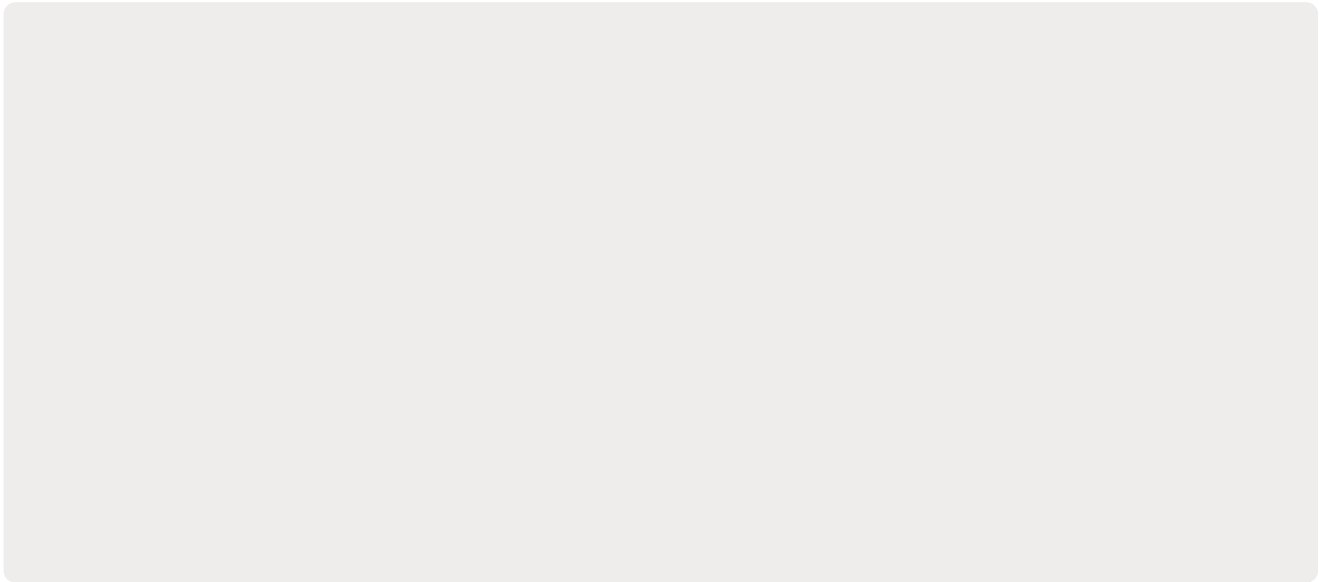
```
<tr>
  <td>单元格内的文字</td>
  ...
</tr>
...
</table>
```

`table`、`tr`、`td`，他们是创建表格的基本标签，缺一不可

- `table`用于定义一个表格标签。
- `tr`标签 用于定义表格中的行，必须嵌套在 `table` 标签中。
- `td` 用于定义表格中的单元格，必须嵌套在`<tr></tr>`标签中。
- 字母 `td` 指表格数据（table data），即数据单元格的内容，现在我们明白，表格最合适的地方就是用来存储数据的。`td`像一个容器，可以容纳所有的元素。



表头单元格标签**th**:一般表头单元格位于表格的第一行或第一列，并且文本加粗居中,只需用表头标签`<th></th>`替代相应的单元格标签`<td></td>`即可。

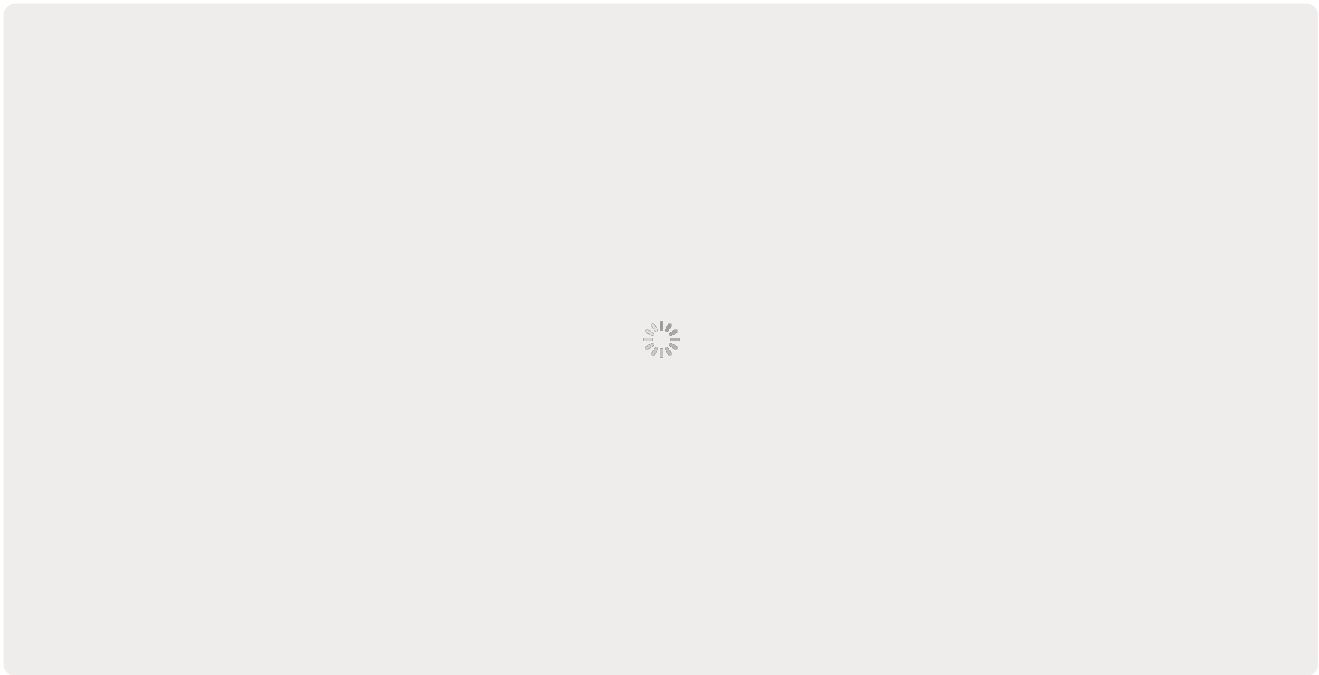


表格标题caption通常这个标题会被居中且显示于表格之上。`caption` 标签必须紧随 `table` 标签之后。这个标签只存在 表格里面才有意义。你是风儿我是沙

●●●

```
<table>
  <caption>我是表格标题</caption>
</table>
```

「3. 表格属性」



三参为0，平时开发的我们这三个参数 `border` `cellpadding` `cellspacing` 为 0

4. 单 ,合并的顺序我们按照 先上 后下 先左 后右 的顺序 ,合并完之后需

「 合并 元格」

要删除多余的单元格。

- 跨行合并：rowspan="合并单元格的个数"
- 跨列合并：colspan="合并单元格的个数"

「5. 总结表格」

标签名	定义	说明
<table></table>	表格标签	就是一个四方的盒子
<tr></tr>	表格行标签	行标签要再table标签内部才有意义
<td></td>	单元格标签	单元格标签是个容器级元素，可以放任何东西
<th></th>	表头单元格标签	它还是一个单元格，但是里面的文字会居中且加粗
<caption></caption>>	表格标题标签	表格的标题，跟着表格一起走，和表格居中对齐
colspan 和 rowspan	合并属性	用来合并单元格的

「6. 表格划分结构」

对于比较复杂的表格，表格的结构也就相对的复杂了，所以又将表格分割成三个部分：题头、正文和脚注。而这三部分分别用:thead,tbody,tfoot来标注， 这样更好的分清表格结构。

注意：

- 1.<thead></thead>：用于定义表格的头部。用来放标题之类的东西。<thead> 内部必须拥有<tr> 标签！
2. <tbody></tbody>：用于定义表格的主体。放数据本体 。
3. <tfoot></tfoot>放表格的脚注之类。
4. 以上标签都是放到table标签中。

列表

「列表ul」容器里面装载着结构，样式一致的文字或图表的一种形式，叫列表。

列表最大的特点就是整齐 、整洁、 有序，跟表格类似，但是它可组合自由度会更高。

「1. 无序列表 ul」

- 中只能嵌套，直接在标签中输入其他标签或者文字的做法是不被允许的。

- 与之间相当于一个容器，可以容纳所有元素。



```
<ul>
  <li>列表项1</li>
  <li>列表项2</li>
  <li>列表项3</li>
  . . . . .
</ul>
```

「2. 有序列表 ol」

- 标签中的type属性值为排序的序列号，不添加type属性时，有序列表默认从数字1开始排序。
- 常用的type属性值分别为是1, a, A, i, I
- <ol reversed="reversed">中的reversed属性能够让有序列表中的序列倒序排列。
- <ol start="3">中的start属性值为3，有序列表中的第一个序列号将从3开始排列。



```
<ol type="A">
  <li>列表项1</li>
  <li>列表二</li>
  <li>列表三</li>
</ol>
```

「2. 自定义列表 dl」

- 定义列表常用于对术语或名词进行解释和描述，定义列表的列表项前没有任何项目符号。

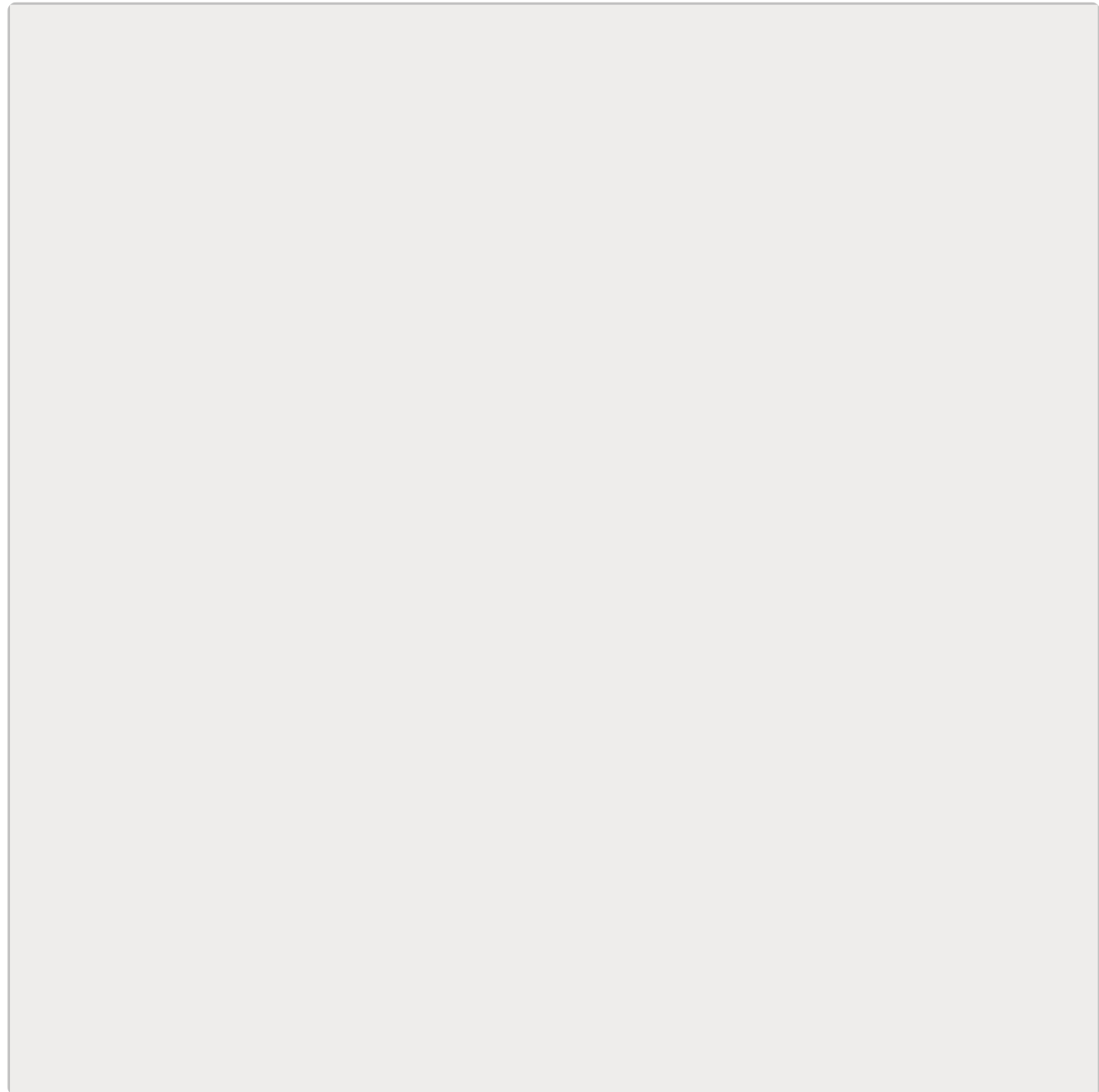


```
<dl>
  <dt>名词1</dt>
  <dd>名词1解释1</dd>
  <dd>名词1解释2</dd>
  . . .
  <dt>名词2</dt>
  <dd>名词2解释1</dd>
  <dd>名词2解释2</dd>
  . . .
</dl>
```



表单

在HTML中，一个完整的表单通常由**表单控件**（也称为表单元素）、**提示信息**和**表单域**3个部分构成。表单目的是为了收集用户信息。



表单控件：

包含了具体的表单功能项，如单行文本输入框、密码输入框、复选框、提交按钮、重置按钮等。

提示信息：

一个表单中通常还需要包含一些说明性的文字，提示用户进行填写和操作。

表单域：

它相当于一个容器，用来容纳所有的表单控件和提示信息，可以通过他定义处理表单数据所用程序的url地址，以及数据提交到服务器的方法。如果不定义表单域，表单中的数据就无法传送到后台服务器。

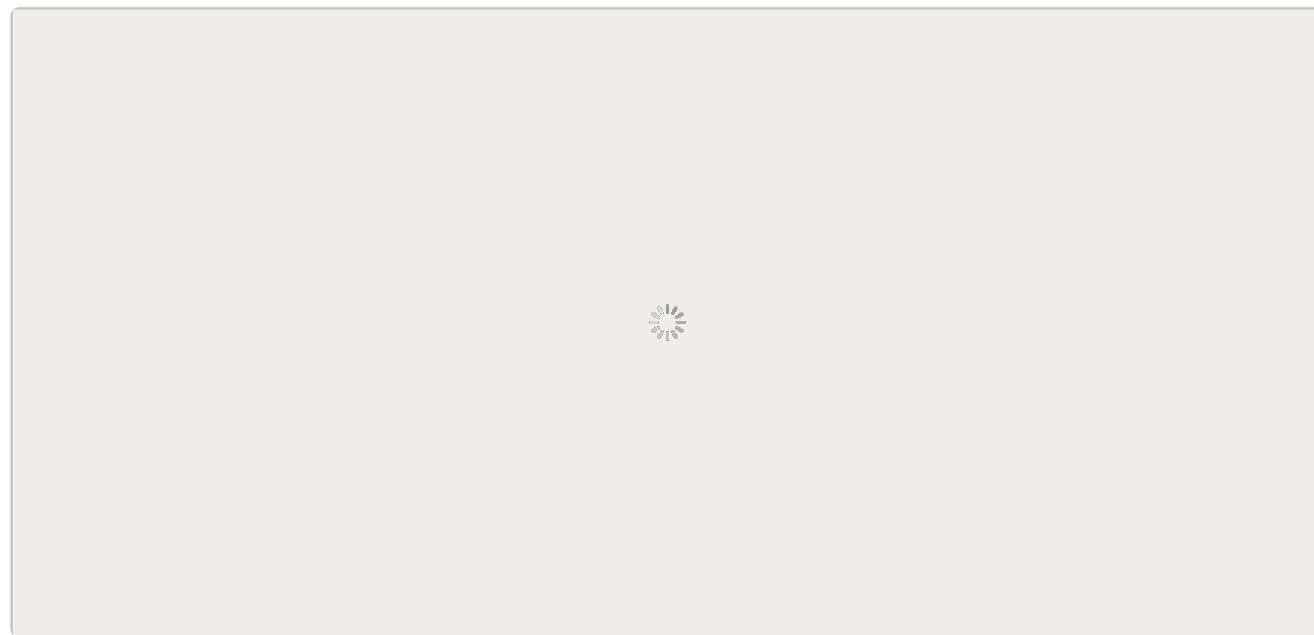
「1. input 控件」



```
<input type="属性值" value="你好">
```


- input 输入的意思
- <input /> 标签为单标签
- type属性设置不同的属性值用来指定不同的控件类型
- 除了type属性还有别的属性

常用属性：



用户名：<input type="text" />

密 码：<input type="password" />

value属性

- value 默认的文本值。有些表单想刚打开页面就默认显示几个文字，就可以通过这个value来设置。



用户名:<input type="text" name="username" value="请输入用户名">

name属性

- name表单的名字，这样，后台可以通过这个name属性找到这个表单。页面中的表单很多，name主要作用就是用于区别不同的表单。
 - name属性后面的值，是我们自己定义的。

radio 如果是一组，我们必须给他们命名相同的名字 name 这样就可以多个选其中的一个啦

- name属性，我们现在用的较少，但是，当我们学ajax 和后台的时候，是必须的。



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

checked属性

- 表示默认选中状态。 较常见于 单选按钮和复选按钮。



性 别：

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

input 属性小结

属性	说明	作用
type	表单类型	用来指定不同的控件类型
value	表单值	表单里面默认显示的文本
name	表单名字	页面中的表单很多，name主要作用就是用于区别不同的表单。
checked	默认选中	表示那个单选或者复选按钮一开始就被选中了

「2. label标签」

- label 标签为 input 元素定义标注（标签）。
- label标签主要目的是为了提高用户体验。为用户提高最优秀的服务。

作用：用于绑定一个表单元素，当点击label标签的时候，被绑定的表单元素就会获得输入焦点。

如何绑定元素呢

- 第一种用法就是用label标签直接包含input表单， 适合单个表单选择
- 第二种用法 for 属性规定 label 与哪个表单元素绑定(通过id)。



第一种

```
<label> 用户名：
```

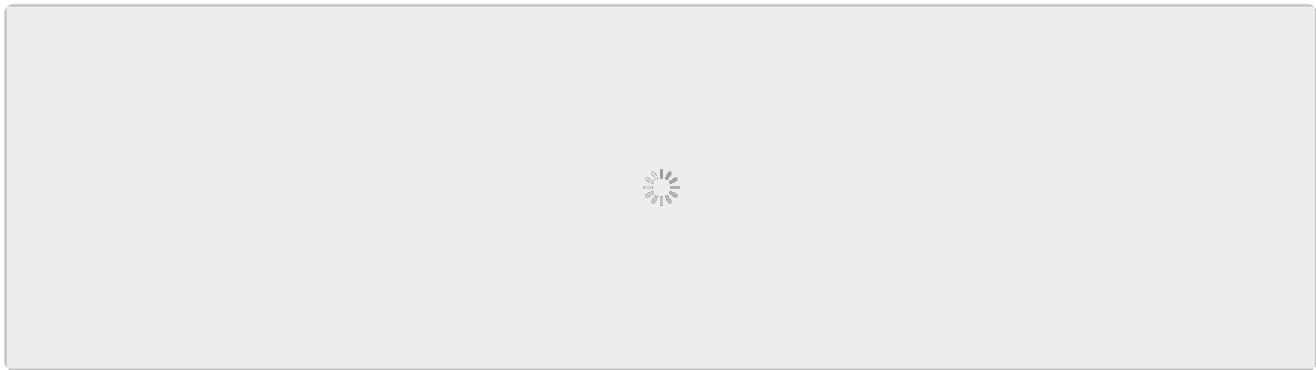
```
<input type="radio" name="username" value="请输入用户名">
</label>
```

第二种

```
<label for="sex">男</label>
<input type="radio" name="sex" id="sex">
```

「3. textarea控件(文本域)」

- 通过textarea控件可以轻松地创建多行文本输入框。
- cols="每行中的字符数" rows="显示的行数" 我们实际开发不用



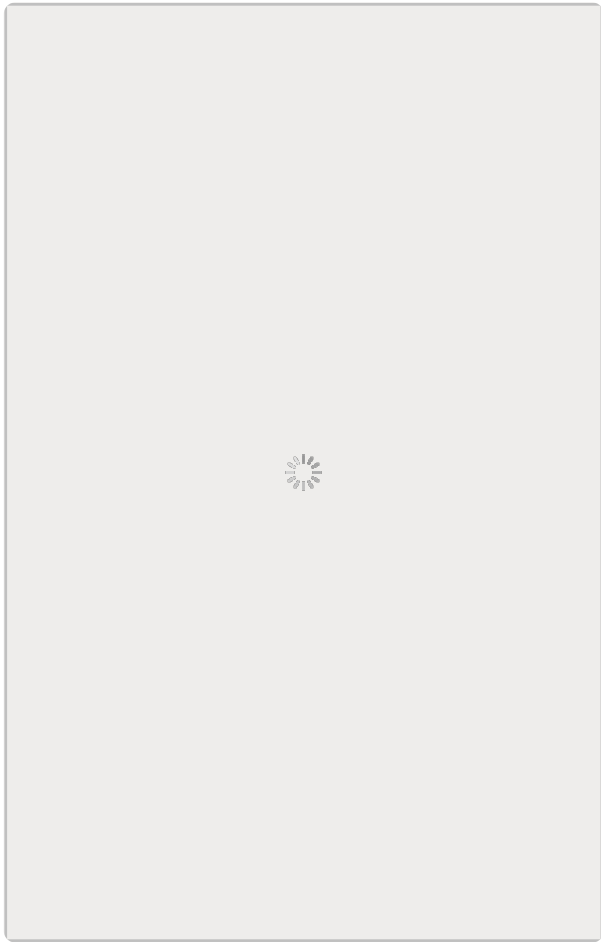
```
<textarea >
  文本内容
</textarea>
```

文本框和文本域区别

表单	名称	区别	默认值显示	用于场景
input type="text"	文本框	只能显示一行文本	单标签，通过value显示默认值	用户名、昵称、密码等
textarea	文本域	可以显示多行文本	双标签，默认值写到标签中间	留言板

「4. select下拉列表」

- 如果有多个选项让用户选择，为了节约空间，我们可以使用select控件定义下拉列表。
- 在option 中定义selected =" selected "时，当前项即为默认选中项。
- 我们实际开发会用的比较少



```
<select>

  <option>选项1</option>
  <option>选项2</option>
  <option>选项3</option>
  ...
</select>
```

form 表单域

- 收集的用户信息怎么传递给服务器？
 - 通过 **form 表单域**
- 目的：
 - 在HTML中，form标签被用于定义表单域，以实现用户信息的收集和传递，form中的所有内容都会被提交给服务器。



```
<form action="url地址" method="提交方式" name="表单名称">
  各种表单控件
</form>
```

常用属性：

- 每个表单都应该有自己表单域。后面学 **ajax 后台交互** 的时候，必须需要form表单域。

属性	属性值	作用
action	url地址	用于指定接收并处理表单数据的服务器程序的url地址。
method	get/post	用于设置表单数据的提交方式，其取值为get或post。
name	名称	用于指定表单的名称，以区分同一个页面中的多个表单。

GET 和 POST 的区别

- GET在浏览器回退时是无害的，而POST会再次提交请求。
- GET请求会被浏览器主动cache，而POST不会，除非手动设置。
- GET请求只能进行url编码，而POST支持多种编码方式。
- GET请求参数会被完整保留在浏览器历史记录里，而POST中的参数不会被保留。
- GET请求大小一般是(1024字节)，http协议并没有限制，而与服务器，操作系统有关，POST理论上来说没有大小限制，http协议规范也没有进行大小限制，但实际上post所能传递的数据量根据取决于服务器的设置和内存大小。
- 对参数的数据类型，GET只接受ASCII字符，而POST没有限制。
- GET比POST更不安全，因为参数直接暴露在URL上，所以不能用来传递敏感信息。

团队约定：

- 元素属性值使用双引号语法
- 元素属性值可以写上的都写上

●●●

推荐

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

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

从输入url到页面展示发生了什么(面试)

作者：Twinkle_

链接：<https://juejin.im/post/6869279683230629896>

来源：掘金

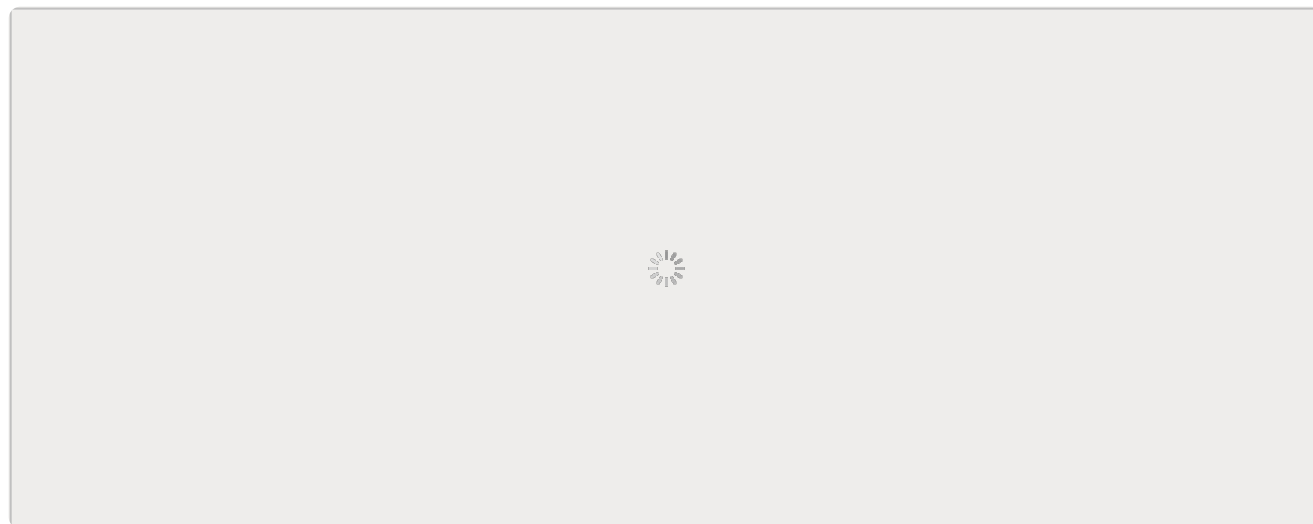
浏览器的多进程架构

从浏览器输入 **URL** 到页面渲染的整个过程都是由 浏览器架构中的各个进程之间的配合完成。

1. **浏览器主进程**：管理子进程、提供服务功能
2. **渲染进程**：将HTML、CSS、JS渲染成界面，js引擎v8和排版引擎Blink就在上面，他会为每一个tab页面创建一个渲染进程
3. **GPU进程**：本来是负责处理3Dcss的，后来慢慢的UI界面也交给GPU来绘制
4. **网络进程**：就是负责网络请求，网络资源加载的进程
5. **插件进程**：负责插件的运行的，因为插件很容易崩溃，把它放到独立的进程里不要让它影响别人

浏览器的多进程架构

从用户输入信息到页面展示的不同阶段，是不同的进程在发挥作用，示意图如下：



从图中可以看出，整个过程是需要各个进程之间相互配合完成的，过程大致可以描述为：

1. 用户输入url,处理输入信息，主进程开始导航，交给网络进程干活
2. 网络进程发起网络请求，其中有可能会发生重定向
3. 服务器响应URL之后，主进程就要通知渲染进程，你要开始干活了
4. 渲染进程准备好了，要想渲染进程提交数据，这个时间叫做提交文档
5. 渲染进程接受到数据，完成页面渲染。

具体过程

1. 输入url

用户输入url，处理输入信息：

- 如果为非url结构的字符串，交给浏览器默认引擎去搜索改字符串；
- 若为url结构的字符串，浏览器主进程会交给网络进程，开始干活。

2.1 查找浏览器缓存

- 网络进程会先看看是否存在本地缓存，如果有就直接返回资源给浏览器进程，无则下一步
DNS-> IP -> TCP

2.2 DNS解析

- 网络进程拿到url后，先会进行DNS域名解析得到IP地址。如果请求协议是HTTPS，那么还需要建立TLS连接。

2.2 建立TCP连接，三次握手

- 接下来就是利用IP地址和服务器建立TCP连接。连接建立之后，向服务器发送请求。

3. 服务器响应

- 服务器收到请求信息后，会根据请求信息生成响应行、响应头、响应体，并发给网络进程。网络进程接受了响应信息之后，就开始解析响应头的内容。
- 网络进程解析响应行和响应头信息的过程：

3.1 重定向

- 如果响应行状态码为301（永久重定向）和302（临时），那么说明需要重定向到其他url。这时候网络进程会从响应头中的Location字段里读取重定向的地址，并重新发起网络请求。

3.2 响应数据处理

- 导航会通过请求头的Content-type字段判断响应体数据的类型。浏览器通过这个来决定如何显示响应体的内容。比如：若为application/octet-stream，则会按照下载类型来处理这个请求，导航结束。若为text/html，这就告诉浏览器服务器返回的是html格式，浏览器会通知渲染进程，你要干活了。

4. 准备渲染进程

- 默认情况，每个页面一个渲染进程。但若处于同一站点（同根域名+协议），那么渲染进程就会复用。

5. 提交文档

- 渲染进程准备好后，浏览器进程发出“提交文档的消息”，渲染进程接受了消息之后，会跟网络进程简历传输数据的管道。

等数据传输完成了，渲染进程会告诉浏览器进程，**确认文档提交**，这时候浏览器会更新页面，安全状态，url，前进后退的历史。

- 到这里导航结束，进入渲染阶段。

注：当浏览器刚开始加载一个地址之后，标签页上的图标便进入了加载状态。但此时图中页面显示的依然是之前打开的页面内容，并没立即替换为百度首页的页面。因为需要等待提交文档阶段，页面内容才会被替换。

前端HTML基础面试题

iframe有哪些缺点？

iframe是一种框架，也是一种很常见的网页嵌入方式。

「iframe的优点」

- iframe能够原封不动的把嵌入的网页展现出来。
- 如果有多个网页引用iframe，那么你只需要修改iframe的内容，就可以实现调用的每一个页面内容的更改，方便快捷。
- 网页如果为了统一风格，头部和版本都是一样的，就可以写成一个页面，用iframe来嵌套，可以增加代码的可重用。
- 如果遇到加载缓慢的第三方内容如图标和广告，这些问题可以由iframe来解决。

「iframe的缺点」

- 会产生很多页面，不容易管理。
- iframe框架结构有时会让人感到迷惑，如果框架个数多的话，可能会出现上下、左右滚动条，会分散访问者的注意力，用户体验度差。
- 代码复杂，无法被一些搜索引擎索引到，这一点很关键，现在的搜索引擎爬虫还不能很好的处理iframe中的内容，所以使用iframe会不利于搜索引擎优化。
- 很多的移动设备（PDA 手机）无法完全显示框架，设备兼容性差。
- iframe框架页面会增加服务器的http请求，对于大型网站是不可取的。
- 现在基本上都是用Ajax来代替iframe，所以iframe已经渐渐的退出了前端开发。

label的作用是什么？是怎么用的？

例子1：点击" 用户名:" 就可以定位光标到输入框



用 名


```
<form><label for="myid "> 户 :</label>
<input type="text" id="myid" /></form>
```

例子2: 点击" 用户名:" 或按键alt+1, 都可以定位光标到输入框



```
<form>
  <label for="myid" accesskey="1"> 用户名:</label>
  <input type="text" id="myid" tabindex="1" />
</form>
```

for 属性功能：表示Label 标签要绑定的HTML 元素，你点击这个标签的时候，所绑定的元素将获取焦点。

accesskey 属性

功能：表示访问Label 标签所绑定的元素的热键，当您按下热键，所绑定的元素将获取焦点。

局限性：accessKey 属性所设置的快捷键不能与浏览器的快捷键冲突，否则将优先激活浏览器的快捷键。

HTML5的form如何关闭自动完成功能？

HTML的输入框可以拥有自动完成的功能，当你往输入框输入内容的时候，浏览器会从你以前的同名输入框的历史记录中查找出类似的内容并列在输入框下面，这样就不用全部输入进去了，直接选择列表中的项目就可以了。

但有时候我们希望关闭输入框的自动完成功能，例如当用户输入内容的时候，我们希望使用AJAX技术从数据库搜索并列举而不是在用户的历史记录中搜索。

关闭输入框的自动完成功能有3种方法：

1. 在IE的Internet选项菜单里的内容--自动完成里面设置
2. 设置form的autocomplete为"on"或者"off"来开启或者关闭自动完成功能
3. 设置输入框的autocomplete为"on"或者"off"来开启或者关闭该输入框的自动完成功能

将 HTML5 看成开放的网络平台

「什么是 HTML5 的基本构件 (building block) ？」

1. 语义 - 提供更准确地描述内容。
2. 连接 - 提供新的方式与服务器通信。

3. 离线和存储 - 允许网页在本地存储数据并有效地离线运行。
4. 多媒体 - 在 Open Web 中，视频和音频被视为一等公民（first-class citizens）。
5. 2D/3D 图形和特效 - 提供更多种演示选项。
6. 性能和集成 - 提供更快访问速度和性能更好的计算机硬件。
7. 设备访问 - 允许使用各种输入、输出设备。
8. 外观 - 可以开发丰富的主题。

浏览器是怎么对HTML5的离线储存资源进行管理和加载的呢？

在浏览器的html头部加上manifest属性，如果是第一次访问浏览器会根据manifest的内容进行下载存储离线内容，如果已经访问过则从离线存储中进行加载，然后在比对服务器如果有新内容在更新离线存储

离线情况下，浏览器就直接使用离线存储的资源。

浏览器的渲染过程？



- 1、将获取的html解析成dom树
- 2、处理css，构成层叠样式表模型CSSOM
- 3、将dom树和CSSOM合并为渲染树
- 4、根据CSSOM将渲染树的节点布局计算
- 5、将渲染树节点样式绘制到页面上

// 注意

在渲染的过程中是自上而下渲染，
js会阻塞页面的渲染，优先等js执行完成
如果在渲染的过程中改变了样式，会造成回流需要重新渲染

link和@import的区别？



- 1、从属关系区别：

link属于html标签，而@import是css提供的。

- 2、加载顺序区别：

页面被加载时，link会同时被加载，而@import引用的css会等到页面被加载完再加载。

- 3、兼容性区别：

import只在IE5以上才能识别，而link是html标签，无兼容问题。

- 4、dom可操作性区别：

可以通过JS 操作 DOM ，插入link标签来改变样式；由于 DOM 方法是基于文档的，无法使用@import的方式插入样式

5、权重区别：

如果已经存在相同样式，@import引入的这个样式将被该 CSS 文件本身的样式层叠掉，表现出link方式的样式权重高于@import的权重这样的直观效果。
(简而言之，link和@import，谁写在后面，谁的样式就被应用，后面的样式覆盖前面的样式。)

src与href的区别？



- 1、href 是指向网络资源所在位置，建立和当前元素（锚点）或当前文档（链接）之间的链接，用于超链接。
- 2、src是指向外部资源的位置，指向的内容将会嵌入到文档中当前标签所在位置；
在请求src资源时会将其指向的资源下载并应用到文档内，例如js脚本，img图片和frame等元素。当浏览器解析到该元素时，会暂停其他资源的下载和处理，直到将该资源加载、编译、执行完毕，图片和框架等元素也如此，类似于将所指向资源嵌入当前标签内。这也是为什么将js脚本放在底部而不是头部。

♥ 感谢大家

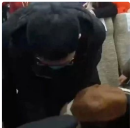
如果你觉得这篇内容对你挺有有帮助的话：

- 1. 点赞支持下吧，让更多的人也能看到这篇内容（收藏不点赞，都是耍流氓 -_-）
- 2. 关注公众号「前端fan」，定期为你推送好文。
- 3. 下一篇将着重记录CSS的学习笔记
- 4. HTML5与CSS3新增特性单独写一章

喜欢此内容的人还喜欢

这4名学生，找到了！
广东共青团

这4名学生，找到了！



创业分享 | 56创带你一起剖析第十一届“创业分享”赛
中央民族大学56创



Vue超好玩的新特性：在CSS中使用JS变量
web前端学习圈



