

河南中医药大学信息技术学院（智能医疗行业学院）智能医学工程专业《互联网医疗服务开发》课程

第02章：HTML

冯顺磊

河南中医药大学信息技术学院（智能医疗行业学院）
河南中医药大学信息技术学院智能医疗教研室
<https://aitcm.hactcm.edu.cn>
2025/2/20

本章概要

- 基础
- 属性
- 结构元素
- 链接元素
- 文本元素
- 列表元素
- 图像元素
- 表格元素
- 表单元素
- 多媒体元素



1. 基础： 1.1 HTML概述

- HTML (Hypertext Markup Language, 超文本标记语言)，使用HTML元素编写的文档称为HTML文档，目前最新版本是HTML5.0。
- HTML是目前在网上应用最为广泛的语言，也是构成网页文档的主要语言。HTML文档是由HTML元素组成的描述性文本，HTML元素可以识别文字、图形、动画、声音、表格和链接等网页中常用的组成部分。

💡 HTML 元素不区分大小写。也就是说，输入元素时既可以使用大写字母也可以使用小写字母。例如，元素 `<title>` 可以写作 `<title>`、`<TITLE>`、`<Title>`、`<TITIE>` 等，也都可以正常工作。不过，从一致性、可读性来说，最好仅使用小写字母。

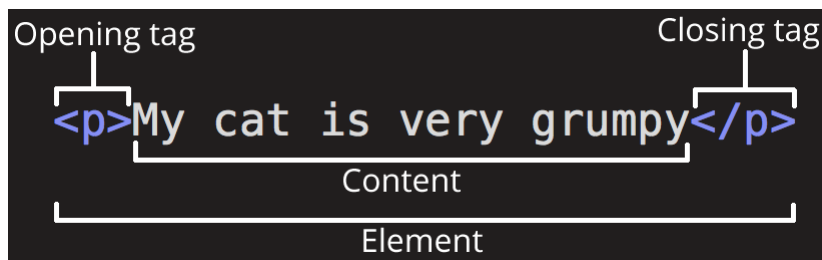
```
<!DOCTYPE html>
<html lang="zh-CN">

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>我的第一个 HTML 页面</title>
</head>

<body>
  <h1>欢迎来到我的网页</h1>
  <p>这是一个简单的 HTML 页面示例。</p>
</body>
</html>
```

1. 基础： 1.2 HTML元素结构

- HTML元素包括三个部分，即开始元素、内容、结束元素，具体如下。
 - 开始标签 (Opening tag)：包含元素的名称（以`<p>`标签为例），被左、右尖括号所包围。开头标签标志着元素开始或开始生效的地方。在这个示例中，它在段落文本的开始之前。
 - 内容 (Content)：元素的内容，本例中就是段落的文本。
 - 结束标签 (Closing tag)：与开始标签相似，只是其在元素名之前包含了一个斜杠。这标志着该元素的结束。没有包含关闭标签是初学者常见一个的错误，它可能会在页面展示时产生一些错误。





1. 基础：1.3 HTML元素分类

- 按内容分类

内容类型	描述
内嵌	在文档中添加其他类型的内容，如audio、video、canvas和iframe等
流	在文档和应用的body中使用的元素，如form、h1和small等
标题	段落标题，如h1、h2和group等
交互	与用户交互的内容，如音频和视频的控件、button和textarea等
元数据	通常出现在页面的head中，设置页面其他部分的表现和行为，如script、style和title等
短语	文本和文本标记元素，如mark、kbd、sub和sup等。



1. 基础：1.3 HTML元素分类

- HTML按功能划分可分为：基础、格式、表单、框架、图像、音频/视频、连接、列表、表格、样式/节、元信息、编程。
- 基础元素

元素名称	描述
<!DOCTYPE>	定义文档类型
<html>	定义HTML文档
<title>	定义文档的标题
<body>	定义文档的主体
<h1> to <h6>	定义HTML标题
<p>	定义段落
 	定义简单的换行
<hr>	定义水平线
<!--...-->	定义注释

1. 基础：1.3 HTML元素分类

● 格式元素

元素名称	描述	元素名称	描述	元素名称	描述	元素名称	描述
<acronym>	定义只取首字母的缩写。HTML5不再支持		定义被删除文本	<q>	定义短的引用	<time>	定义日期/时间
<abbr>	定义缩写	<dfn>	定义项目	<rp>	定义若浏览器不支持ruby元素显示的内容	<tt>	定义打字机文本。HTML5不再支持
<address>	定义文档作者或拥有者的联系信息		定义强调文本	<rt>	定义ruby注释的解释	<u>	不赞成使用。定义下划线文本
	定义粗体文本		定义文本的字体、尺寸和颜色。HTML5不再支持	<ruby>	定义ruby注释	<var>	定义文本的变量部分
<bdi>	定义文本的文本方向，使其脱离其周围文本的方向设置	<i>	定义斜体字	<s>	定义加删除线的文本	<wbr>	定义换行
<bdo>	定义文字方向	<ins>	定义被插入文本	<samp>	定义计算机代码样本		
<big>	定义大号文本	<kbd>	定义键盘文本	<small>	定义小号文本		
<blockquote>	定义长的引用	<mark>	定义有记号的文本	<strike>	定义加删除线文本。HTML5不再支持		
<center>	定义居中文本。HTML5不再支持	<meter>	定义预定义范围内的度量		定义强调文本		
<cite>	定义引用(citation)	<pre>	定义预格式文本	<sup>	定义上标文本		
<code>	定义计算机代码文本	<progress>	定义任何类型的任务的进度	<sub>	定义下标文本		

1. 基础：1.3 HTML元素分类

● 表单元素

元素名称	描述	元素名称	描述
<form>	定义供用户输入的HTML表单	<label>	定义input元素的标注
<input>	定义输入控件	<fieldset>	定义围绕表单中元素的边框
<textarea>	定义多行的文本输入控件	<legend>	定义fieldset元素的标题
<button>	定义按钮	<isindex>	不赞成使用。定义与文档相关的可搜索索引
<select>	定义选择列表（下拉列表）	<datalist>	定义下拉列表
<optgroup>	定义选择列表中相关选项的组合	<keygen>	定义生成密钥
<option>	定义选择列表中的选项	<output>	定义输出的一些类型



1. 基础： 1.3 HTML元素分类

● 框架元素

元素名称	描述
<frame>	定义框架集的窗口或框架。HTML5不再支持。
<frameset>	定义框架集。HTML5不再支持
<noframes>	定义针对不支持框架的用户的替代内容。HTML5不再支持
<iframe>	定义内联框架



1. 基础： 1.3 HTML元素分类

● 图像元素

元素名称	描述
	定义图像
<map>	定义图像映射
<area>	定义图像地图内部的区域
<canvas>	定义图形
<figcaption>	定义figure 元素的标题
<figure>	定义媒介内容的分组，以及它们的标题



1. 基础：1.3 HTML元素分类

● 音频/视频元素

元素名称	描述
<audio>	定义声音内容
<source>	定义媒介源
<track>	定义用在媒体播放器中的文本轨道
<video>	定义视频



1. 基础：1.3 HTML元素分类

● 链接元素

元素名称	描述
<a>	定义锚
<link>	定义文档与外部资源的关系
<main>	定义文档的主体部分
<nav>	定义导航链接



1. 基础：1.3 HTML元素分类

● 列表元素

元素名称	描述
	定义无序列表
	定义有序列表
	定义列表的项目
<dir>	定义目录列表。HTML5不再支持
<dl>	定义定义列表
<dt>	定义定义列表中的项目
<dd>	定义定义列表中项目的描述
<menu>	定义命令的菜单/列表
<command>	定义命令按钮



1. 基础：1.3 HTML元素分类

● 表格元素

元素名称	描述
<table>	定义表格
<caption>	定义表格标题
<th>	定义表格中的表头单元格
<tr>	定义表格中的行
<td>	定义表格中的单元
<thead>	定义表格中的表头内容
<tbody>	定义表格中的主体内容
<tfoot>	定义表格中的表注内容（脚注）
<col>	定义表格中一个或多个列的属性值
<colgroup>	定义表格中供格式化的列组



1. 基础：1.3 HTML元素分类

● 样式/节元素

元素名称	描述
<style>	定义文档的样式信息
<div>	定义文档中的节
	定义文档中的节
<header>	定义section或page的页眉
<footer>	定义section或page的页脚
<section>	定义section
<article>	定义文章
<aside>	定义页面内容之外的内容
<details>	定义元素的细节
<dialog>	定义对话框或窗口
<summary>	定义一个可见的标题。当用户点击标题时会显示出详细信息。



1. 基础：1.3 HTML元素分类

● 元信息元素

元素名称	描述
<head>	定义关于文档的信息
<meta>	定义关于HTML 文档的元信息
<base>	定义页面中所有链接的默认地址或默认目标
<basefont>	定义页面中文本的默认字体、颜色或尺寸。HTML5不再支持。

1. 基础： 1.3 HTML元素分类

- 编程元素

元素名称	描述
<script>	定义客户端脚本
<noscript>	定义针对不支持客户端脚本的用户的替代内容
<applet>	定义嵌入的applet。HTML5不再支持
<embed>	为外部应用程序（非HTML）定义容器
<object>	定义嵌入的对象
<param>	定义对象的参数

1. 基础： 1.4 HTML文档结构

- <!DOCTYPE html>： 声明文档类型。
- <html></html>： <html> 元素包裹了页面中所有的内容，有时被称为根元素。
- <head></head>： <head> 元素是一个容器，它包含了在 HTML 页面中但不在 HTML 页面中显示的内容。内容包括搜索结果中出现的关键词和页面描述、CSS 样式、字符集声明等等。
- <meta charset="utf-8">： <meta> 元素代表了不能由其他 HTML元相关元素表示的元数据。charset 属性将你的文档的字符集设置为 UTF-8。
- <title></title>： <title> 元素设置了页面的标题，也就是出现在该页面加载的浏览器元素中的内容。当页面被加入书签时，页面标题也被用来描述该页面。
- <body></body>： <body> 元素包含了你访问页面时所有显示在页面上的内容，包含文本、图片、视频、游戏、可播放音频轨道等等。

```
<!DOCTYPE html>
<!-- 声明文档结构类型-->
<html>
<!-- 声明文档文字区域-->
<head>
<!-- 文档的头部区域-->
<meta charset="utf-8">
<!-- 文档的头部区域中元数据的字符集定义，utf-8表示国际通用的字符集编码格式-->
<title>文档结构</title>
<!-- 文档的头部区域的标题，title内容对于SEO来说极其重要-->
</head>
<body>
<!-- 文档的主要内容区域-->
</body>
</html>
```



1. 基础： 1.5 块级元素和内联元素

- 块级元素在页面中以块的形式展现。一个块级元素出现在它前面的内容之后的新行上。任何跟在块级元素后面的内容也会出现在新的行上。块级元素通常是页面上的结构元素。例如，一个块级元素可能代表标题、段落、列表、导航菜单或页脚。一个块级元素不会嵌套在一个内联元素里面，但它可能嵌套在另一个块级元素里面。
- 内联元素通常出现在块级元素中并环绕文档内容的一小部分，而不是一整个段落或者一组内容。内联元素不会导致文本换行。它通常与文本一起使用，例如，<a> 元素创建一个超链接， 和 等元素创建强调。



示例2-1-1：块级元素和内联元素



1. 基础： 1.5 嵌套元素

- 在HTML中可以把元素放到其他元素之中，这种方式称为嵌套。
- 在HTML中所有的元素都需要正确地打开和关闭。
- 如果元素未正确地打开和关闭，浏览器展示可能会错误。



示例2-1-1：嵌套元素
示例2-1-2：嵌套元素-错误示范

1. 基础： 1.5 空元素

- 空元素

- 不是所有元素都拥有开始标签、内容和结束标签。
- 一些元素只有一个标签，通常用来在此元素所在位置插入/嵌入一些东西，这些元素被称为空元素。
- 例如：元素 `<img>` 是用来在页面插入一张指定的图片。



示例2-1-4：空元素

1. 基础： 1.5 HTML中的空白

- HTML中的空白

- 在 HTML元素的内容中无论使用多少个空格（包括一个或多个空白字符或换行），当渲染这些代码的时候，HTML解释器会将连续出现的空白字符减少为一个单独的空格符。



示例2-1-5：HTML中的空白

1. 基础： 1.5 HTML中的特殊字符

- HTML中的特殊字符

- 在 HTML 中，字符 <、>、"、' 和 & 是特殊字符，它们是 HTML 语法自身的一部分，在展示这些字符时必须使用字符引用——表示字符的特殊编码，每个字符引用以符号 & 开始，以分号 (;) 结束。



示例2-1-6：HTML中的特殊字符

1. 基础： 1.5 HTML中的注释

- HTML中的特殊字符

- 将 HTML 中的内容设置为注释，需将其用特殊的记号 <!-- 和 --> 包裹起来。



示例2-1-7：HTML中的注释

2. 属性： 2.1 属性介绍

- 属性包含元素的额外信息，这些信息不会出现在实际的内容中。在上述例子中，这个 class 属性是一个识别名称，以后为元素设置样式信息时更加方便。
 - 一个空格，它在属性和元素名称之间。如果一个元素具有多个属性，则每个属性之间必须由空格分隔。
 - 属性名称，后面跟着一个等于号。
 - 一个属性值，由一对引号 ("") 引起来。

Attribute

```
<p class="editor-note">My cat is very grumpy</p>
```

2. 属性： 2.2 公共属性

- 公共属性大致可分为基本属性、语言属性、键盘属性、内容属性和延伸属性等类型。
- 基本属性
 - 大部分元素都有基本属性

属性名称	描述
class	定义类规则或样式规则
id	定义元素的唯一标识
style	定义元素的样式声明

- 下面这些元素没有基本属性，这些元素一般位于文档头部，用来标识网页元信息。
html、head、title、base、meta、partam、script、style

2. 属性： 2.2 公共属性

- 语言属性
 - 语言属性主要用来定义元素的语言类型

属性名称	描述
lang	定义元素的语言代码或编码
dir	定义文本的方向，包括ltr和rtl，分别标识从左向右和从右向左

- 下面这些元素没有基本属性，这些元素一般位于文档头部，用来标识网页元信息。
frameset、frame、iframe、br、hr、base、param、script

2. 属性： 2.2 公共属性

- 键盘属性
 - 键盘属性定义元素的键盘访问方法

属性名称	描述
accesskey	定义访问某元素的键盘快捷键
tabindex	定义元素的Tab键索引编号

- accesskey属性可以使用快捷键（Alt+字母）访问指定的URL，请注意，不能和其他快捷键冲突。
- abindex属性用来定义元素的Tab键访问顺序，可以使用Tab键遍历网页中的所有链接和表单元素。遍历时会按照tabindex的大小决定顺序，当遍历到某个链接时，按Enter键即打开链接页面。



示例2-2-1: accesskey属性
示例2-2-2: tabindex属性

2. 属性：2.2 公共属性

- 内容属性
 - 内容属性定义元素包含内容的附加信息，这些信息对于元素来说具有重要补充作用，避免元素本身包含信息不全而被误解。

属性名称	描述
alt	定义元素的替换文本
title	定义元素的提示文本
longdesc	定义元素包含内容的大段描述信息
cite	定义元素包含内容的引用信息
datetime	定义元素包含内容的日期和时间

- alt和title是两个常用的属性，分别定义元素的替换文本和提示文本，在使用时请注意其语义性。
- title属性不能用在下列元素：html、head、title、base、basefont、meta、partam、script、style



示例2-2-3: alt与title属性

2. 属性：2.2 全局属性

- 全局属性
 - 全局属性是指可以对任何元素都使用的属性。

属性名称	描述	属性名称	描述
accesskey	规定激活元素的快捷键	hidden	规定元素仍未或不再相关
class	规定元素的一个或多个类名（引用样式表中的类）	id	规定元素的唯一id
contenteditable	规定元素内容是否可编辑	lang	规定元素内容的语言
contextmenu	规定元素的上下文菜单。上下文菜单在用户点击元素时显示	spellcheck	规定是否对元素进行拼写和语法检查
data-*	用于存储页面或应用程序的私有定制数据	style	规定元素的行内CSS样式
dir	规定元素中内容的文本方向	tabindex	规定元素的tab键次序
draggable	规定元素是否可拖动	title	规定有关元素的额外信息
dropzone	规定在拖动被拖动数据时是否进行复制、移动或链接	translate	规定是否应该翻译元素内容

3. 结构元素： 3.1 header

- header
 - header元素是一种具有引导和导航作用的结构元素，通常用来放置整个页面或页面内的一个内容区块的标题，但也可以包含其他内容，例如数据表格、搜索表单或相关的logo图片。
 - 一个网页内并未限制header元素的个数，可以拥有多个，可以为每个内容区块加一个header元素。
 - 仅支持全局属性。



示例2-3-1: header元素

3. 结构元素： 3.2 footer

- footer
 - footer元素表示网页、文章或章节的尾部。如果用于整张网页的尾部，就称为“页尾”，通常包含版权信息或者其他相关信息。
 - 与header元素一样，一个页面中也未限制footer元素的个数。可以同时为article元素或section元素添加footer元素。footer不能嵌套，即内部不能放置另一个footer，也不能放置header。
 - 仅支持全局属性。



示例2-3-2: footer元素

3. 结构元素： 3.3 nav

- nav
 - nav元素用于放置页面或文档的导航信息。
 - 一般来说，nav往往放置在header里面，不适合放入footer。另外，一个页面可以有多个nav，比如一个用于站点导航，另一个用于文章导航。
 - 仅支持全局属性。



示例2-3-3: nav元素

3. 结构元素： 3.4 article

- article
 - article元素表示页面里面一段完整的内容，即使页面的其他部分不存在，也具有独立使用的意义，通常用来表示一篇文章或者一个论坛帖子。它可以有自己的标题（h1到h6）。
 - 一个网页可以包含一个或多个<article>，比如包含多篇文章。
 - 仅支持全局属性。



示例2-3-4: article元素

3. 结构元素： 3.5 section

- section

- section元素用于对网站或应用程序中页面上的内容进行分块。一个section元素通常由内容及其标题组成。但section元素并非一个普通的容器元素。当一个容器需要被直接定义样式或通过脚本定义行为时，推荐使用div而非section元素。
- section很适合幻灯片展示的页面，每个section代表一个幻灯片。
- 一般来说，section都应该有标题，即包含h1~h6元素。多个section可以放在同一个article里面，一个section里面也可能包含多个article，这取决于section和article在当前页面的含义。
- 仅支持全局属性。



示例2-3-5: session元素

3. 结构元素： 3.6 aside

- aside

- aside元素用来放置与网页或文章主要内容间接相关的部分。
- 网页级别的aside，可以用来放置侧边栏，但不一定就在页面的侧边；
- 文章级别的aside，可以用来放置评论或注释。
- 仅支持全局属性。



示例2-3-6: abside元素

3. 结构元素： 3.7 hgroup

- hgroup

- hgroup元素是将标题及其子标题进行分组的元素。hgroup元素通常会将h1-h6元素进行分组，譬如一个内容区块的标题及其子标题算一组。
- 通常，如果文章只有一个主标题，是不需要hgroup元素的，但是文章有主标题，主标题下面有子标题，就需要使用hgroup。
- hgroup 元素允许将一个标题与任意次要内容（例如子标题、副标题或口号）组合在一起。在hgroup 中，这些类型的内容也表示为 p元素。
- hgroup本身对网页的文档大纲没有任何影响，而hgroup中所允许的单个标题则会被用于文档大纲。
- 仅支持全局属性。



示例2-3-6: abside元素

3. 结构元素： 3.8 h1 ~ h6

- h1 ~ h6

- h1 ~ h6 标题（Heading）元素呈现了六个不同的级别的标题，h1 级别最高，而 h6 级别最低。
- 使用时需避免跳过某级标题：始终要从h1开始，接下来依次使用h2 等等。默认情况下，浏览器会粗体显示标题。h1的字号比h2大，h2比h3大，以此类推。
- 仅支持全局属性。



示例2-3-8: 标题元素

3. 结构元素： 3.9 address

- address

- address元素用来在文档中呈现联系信息，包括文档作者或文档维护者的名字、他们的网站链接、电子邮箱、真实地址、电话号码等。address不只是用来呈现电子邮箱或真实地址，还来展示跟文档相关的联系人的所有联系信息。
- 当表示一个和联系信息无关的任意的地址时，请改用p元素而不是address元素。
- 这个元素不能包含除联系信息之外的任何信息，比如出版日期（这应当被包含在time元素之中）。
- 通常，address元素可以放在footer元素之中。
- 仅支持全局属性。



示例2-3-9: address元素

3. 结构元素： 3.10 time

- time

- time元素代表24小时中的每个时刻或者某个日期，它表示时间时允许带时差。
- 除支持全局属性，还具备下列特殊属性。
 - datetime
 - 该属性表示此元素的时间和/或日期，并且属性值必须符合下文所描述的格式。



示例2-3-10: time元素

4. 链接元素： 4.1 a

● a

- 链接在网页制作中是一个必不可少的部分，在浏览网页时，单击一张图片或者一段文字就可以弹出一个新的网页，这些功能都是通过超链接实现的。
- 绝对路径和相对路径
 - 绝对路径
 - 绝对路径是指文件的完整路径，包括文件传输的协议http、ftp等，一般用于网站的外部链接，例如：
http://news.sina.com.cn/
 - 相对路径
 - 相对路径是指相对于当前文件的路径，它包含了从当前文件指向目的文件的路径。同时只要是处于站点文件夹之外，即使不属于同一个文件目录下，相对路径建立的连接也适合。采用相对路径是建立两个文件之间的相互关系，可以不受站点和所处服务器位置的影响

相对位置	如何输入
同一目录	输入要链接的文档
链接上一目录	先输入“../”，再输入目录名称
链接下一目录	先输入目录名，后加“/”

4. 链接元素： 4.1 a

● a

- 关键属性： href
 - href属性给出链接指向的网址。它的值应该是一个 URL 或者锚点。
- 关键属性： title
 - title属性给出链接的说明信息。鼠标悬停在链接上方时，浏览器会将这个属性的值，以提示块的形式显示出来。
- 关键属性： target
 - target属性指定如何展示打开的链接，target属性的值可以是以下4个关键字之一。
 - _self: 当前窗口打开，这是默认值。
 - _blank: 新窗口打开。
 - _parent: 上层窗口打开，这通常用于从父窗口打开的子窗口，或者<iframe>里面的链接。如果当前窗口没有上层窗口，这个值等同于_self。
 - _top: 顶层窗口打开。如果当前窗口就是顶层窗口，这个值等同于_self。

4. 链接元素： 4.1 a

- a

- 关键属性：rel

- rel属性说明链接与当前页面的关系，其常见的属性值如下。
 - alternate: 当前文档的另一种形式，比如翻译。
 - author: 作者链接。
 - bookmark: 用作书签的永久地址。
 - external: 当前文档的外部参考文档。
 - help: 帮助链接。
 - license: 许可证链接。
 - next: 系列文档的下一篇。
 - nofollow: 告诉搜索引擎忽略该链接，主要用于用户提交的内容，防止有人企图通过添加链接，提高该链接的搜索排名。
 - norereferrer: 告诉浏览器打开链接时，不要将当前网址作为 HTTP 头信息的Referer字段发送出去，这样可以隐藏点击的来源。
 - noopener: 告诉浏览器打开链接时，不让链接窗口通过 JavaScript 的window.opener属性引用原始窗口，这样就提高了安全性。
 - prev: 系列文档的上一篇。
 - search: 文档的搜索链接。
 - tag: 文档的元素链接。

4. 链接元素： 4.1 a

- a

- 关键属性：download

- download属性表明当前链接用于下载，而不是跳转到另一个 URL。注意，download属性只在链接与网址同源时，才会生效。也就是说，链接应该与网址属于同一个网站。
- download属性还有一个用途，就是有些地址不是真实网址，而是数据网址，比如data:开头的网址。这时，download属性可以为虚拟网址指定下载的文件名。

- 关键属性：type

- type属性给出链接 URL 的 MIME 类型，比如到底是网页，还是图像或文件。它也是纯粹提示性的属性，没有实际功能。



示例2-4-1：a元素

5. 文本元素： 5.1 div

- div

- div是一个通用元素，表示一个区块（division）。它没有语义，如果网页需要一个块级元素容器，又没有其他合适的元素，就可以使用这个元素。
- 只有当没有其他合适的语义元素（例如 article或nav）时，才应使用div元素。
- 仅支持全局属性。



示例2-5-1：div元素

5. 文本元素： 5.2 p

- p

- p元素表示文本的一个段落。p元素可以是相关内容的任何结构分组，如图像或表格字段。
- 段落是块级元素，如果在关闭的 `</p>` 标签之前解析了另一个块级元素，则该元素将自动关闭，但不建议这样使用。
- 仅支持全局属性。



示例2-5-2：p元素

5. 文本元素： 5.3 span

- span
 - span元素是一个通用的行级容器，本身不具备特殊含义。它可被用于对元素进行编组，以便于添加样式（通过使用 class 或 id 属性），或共享属性值（例如 lang 属性）。
 - 该元素仅应在无其他合适语义元素时使用。span与div元素非常相似，但是div是块级元素，而span是一个行级元素。
 - 仅支持全局属性。



示例2-5-3: span元素

5. 文本元素： 5.4 br、wbr

- br、wbr
 - br让网页产生一个换行效果。
 - wbr元素跟br很相似，表示一个可选的断行。如果一行的宽度足够，则不断行；如果宽度不够，需要断行，就在wbr的位置的断行。它是为了防止浏览器在一个很长的单词中间，不正确地断行或者不断行，所以事先标明可以断行的位置。
 - br元素是单独使用的，没有闭合元素。
 - wbr主要用于欧洲一些单词很长的语言或者 URL 的断行。
 - 仅支持全局属性。



示例2-5-4: br、wbr元素

6. 列表元素： 6.1 ol、ul、li

- ol、ul、li
 - ol元素是一个有序列表容器（ordered list），会在内部的列表项前面产生数字编号。列表项的顺序有意义时，比如排名，就会采用这个元素。
 - ul元素是一个无序列表容器（unordered list），会在内部的列表项前面产生实心小圆点，作为列表符号。列表项的顺序无意义时，采用这个元素。
 - li表示列表项，用在ol或ul容器之中。
- ul仅支持全局属性。

6. 列表元素： 6.1 ol、ul、li

- ol、ul、li
 - ol除支持全局属性外，具有以下特殊属性。
 - reverse: 此布尔值属性指定列表中的条目是否是倒序排列的，即编号是否应从高到低反向标注。
 - start: 一个整数值属性，指定了列表编号的起始值。此属性的值应为阿拉伯数字，尽管列表条目的编号类型 type 属性可能指定为了罗马数字编号等其他类型的编号。比如说，想要让元素的编号从英文字母 "d" 或者罗马数字 "iv" 开始，都应当使用 start="4"。
 - type: 设置编号的类型。
 - a 表示小写英文字母编号
 - A 表示大写英文字母编号
 - i 表示小写罗马数字编号
 - I 表示大写罗马数字编号
 - 1 表示数字编号（默认）编号类型适用于整个列表，除非在 元素的 元素中使用不同的 type 属性。

6. 列表元素： 6.1 ol、ul、li

• ol、ul、li

通常，有序列表的条目会和它前面的编号标记一起显示，编号标记可以是数字或者字母。

- ol和ul元素两者可以无限嵌套，既可以同类嵌套，也可以互相嵌套。
- ol和ul元素都表示列表。区别在于，ol元素的有序列表的顺序是有意义的。举一些例子：
 - 烹饪食谱中的各个步骤
 - 指路时的各处转弯方向
 - 营养信息元素上按含量排序的成分列表
- 至于如何确定该选择哪一个列表元素，可以尝试更改列表项的顺序，如果其含义会发生改变，那么就应当使用ol元素，否则使用ul更合适。



示例2-6-4：列表元素

6. 列表元素： 6.2 dl、dd、dt

• dl、dd、dt

- dl元素是一个块级元素，表示一组术语的列表（description list）。
- 术语名（description term）由dt元素定义。
- 术语解释（description detail）由dd元素定义。
- dl常用来定义词汇表。
- 仅支持全局属性。



示例2-6-2：dl、dd、dt元素

7. 表格元素： 7.1 img

- **img**

- **img**元素用于插入图片。它是单独使用的，没有闭合元素。除支持全局属性外，还具有以下特殊属性。
- **关键属性：alt**
 - alt属性用来设定图片的文字说明。图片不显示时（比如下载失败，或用户关闭图片加载），图片的位置上会显示该文本。
- **关键属性：width、height**
 - 图片默认以原始大小插入网页，width属性和height属性可以指定图片显示时的宽度和高度，单位是像素或百分比。
 - 注意，一旦设置了这两个属性，浏览器会在网页中预先留出这个大小的空间，不管图片有没有加载成功。不过，由于图片的显示大小可以用 CSS 设置，所以不建议使用这两个属性。
 - 一种特殊情况是，width属性和height属性只设置了一个，另一个没有设置。这时，浏览器会根据图片的原始大小，自动设置对应比例的图片宽度或高度。举例来说，图片大小是 800像素 x 800像素，width属性设置成200，那么浏览器会自动将height设成200。

7. 图像元素： 7.1 img

- **img**

- **关键属性：referrerpolicy**
 - 导致的图片加载的 HTTP 请求，默认会带有Referer的头信息。referrerpolicy属性对这个行为进行设置。
 - no-referrer: 不会发送 Referer 标头。
 - no-referrer-when-downgrade: 若未使用 TLS (HTTPS) 导航到源站，则不发送 Referer 标头。
 - origin: 发送到源站的来源地址将被限制为：协议、主机和端口。
 - origin-when-cross-origin: 发送到其他来源的来源地址会被限制为协议、主机和端口。同源导航仍将包含路径。
 - same-origin: 仅同源请求发送来源地址，而跨源请求则不包含来源地址信息。
 - strict-origin: 仅在协议安全级别保持不变 (HTTPS→HTTPS) 的情况下将文档的来源作为来源地址发送。而在目标的安全性降低 (HTTPS→HTTP) 时则不发送来源地址。
 - strict-origin-when-cross-origin (默认值): 执行同源请求时发送完整的 URL，且仅在协议安全级别保持不变 (HTTPS→HTTPS) 时发送来源 (origin)，在目标安全性降低 (HTTPS→HTTP) 时则不发送来源。
 - unsafe-url: 来源地址包括来源 (origin) 和路径 (path) (但不包括片段、密码或用户名)。这个值是不安全的，因为它将来源和路径从受 TLS 保护的资源泄露到不安全的来源。

7. 图像元素： 7.1 img

• img

● 关键属性: crossorigin

- 有时，图片和网页属于不同的网站，网页加载图片就会导致跨域请求，对方服务器可能要求跨域认证。crossorigin属性用来告诉浏览器，是否采用跨域的形式下载图片，默认是不采用。
- 简单说，只要打开了这个属性，HTTP 请求的头信息里面，就会加入origin字段，给出请求发出的域名，不打开这个属性就不加。
- 一旦打开该属性，它可以设为两个值。
 - anonymous: 跨域请求不带有用户凭证（通常是 Cookie）。
 - use-credentials: 跨域请求带有用户凭证。

● 关键属性: loading

- 浏览器的默认行为是，只要解析到元素，就开始加载图片。对于很长的网页，这样做很浪费带宽，因为用户不一定会往下滚动，一直看到网页结束。用户很可能是点开网页，看了一会就关掉了，那些不在视口的图片加载的流量，就都浪费了。
- loading属性改变了这个行为，可以指定图片的懒加载，即图片默认不加载，只有即将滚动进入视口，变成用户可见时才会加载，这样就节省了带宽。
- loading属性可以取以下三个值。
 - auto: 浏览器默认行为，等同于不使用loading属性。
 - lazy: 启用懒加载。
 - eager: 立即加载资源，无论它在页面上的哪个位置。
- 由于行内图片的懒加载，可能会导致页面布局重排，所以使用这个属性的时候，最好指定图片的高和宽。

7. 图像元素： 7.1 img

• img

● Web 最常用的图像格式是：

- APNG（动态可移植网络图形）——无损动画序列的不错选择（GIF 性能较差）。
- AVIF（AV1 图像文件格式）——静态图像或动画的不错选择，其性能较好。
- GIF（图像互换格式）——简单图像和动画的不错选择。
- JPEG（联合图像专家组）——有损压缩静态图像的不错选择（目前最流行的格式）。
- PNG（便携式网络图形）——对于无损压缩静态图像而言是不错的选择（质量略好于 JPEG）。
- SVG（可缩放矢量图形）——矢量图像格式。用于必须以不同尺寸准确描绘的图像。
- WebP（网络图片格式）——图像和动画的绝佳选择。
- 推荐使用诸如 WebP 和 AVIF 等图像格式，因为它们在静态图像和动画的性能均比 PNG、JPEG、JIF 好得多。WebP 得到了广泛的支持，而 AVIF 则缺乏 Safari 的支持。
- 对于必须以不同尺寸准确绘制的图像，推荐使用 SVG 格式。



示例2-7-1: img元素



8. 表格元素： 7.1 table、caption、thead、tbody、tfoot、colgroup、col、tr、th、td

- table、caption、thead、tbody、tfoot、colgroup、col、tr、th、td
 - table是一个块级容器元素，所有表格内容都要放在这个元素里面。
 - caption总是table里面的第一个子元素，表示表格的标题。该元素是可选的。
 - thead、tbody、tfoot都是块级容器元素，且都是table的一级子元素，分别表示表头、表体和表尾。
 - 这三个元素都是可选的。如果使用了thead，那么tbody和tfoot一定在thead的后面。如果使用了tbody，那么tfoot一定在tbody后面。
 - 大型表格内部可以使用多个tbody，表示连续多个部分。
 - colgroup是table的一级子元素，用来包含一组列的定义。col是colgroup的子元素，用来定义表格的一列。
 - tr元素表示表格的一行（table row）。如果表格有thead、tbody、tfoot，那么tr就放在这些容器元素之中，否则直接放在table的下一级。
 - th和td都用来定义表格的单元格。其中，th是标题单元格，td是数据单元格。



8. 表格元素： 7.1 table、caption、thead、tbody、tfoot、colgroup、col、tr、th、td

- table、caption、thead、tbody、tfoot、colgroup、col、tr、th、td
 - table、caption、thead、tbody、tfoot、colgroup、tr均支持全局属性。
 - col除支持全局属性外，具有以下特殊属性。
 - 关键属性：span属性
 - 值为正整数，默认为1。如果大于1，就表示该列的宽度包含连续的多列。
 - th、td除支持全局属性外，还具有以下特殊属性。
 - 关键属性：colspan，rowspan
 - 单元格会有跨越多行或多列的情况，这要通过colspan属性和rowspan属性设置，前者表示单元格跨越的栏数，后者表示单元格跨越的行数。它们的值都是一个非负整数，默认为1。
 - 关键属性：headers
 - 如果表格很大，单元格很多，源码里面会看不清，哪个单元格对应哪个表头，这时就可以使用headers属性。

8. 表格元素： 7.1 table、caption、thead、tbody、tfoot、colgroup、col、tr、th、td

- table、caption、thead、tbody、tfoot、colgroup、col、tr、th、td
 - 关键属性：scope
 - scope属性只有th元素支持，一般不在td元素使用，表示该th单元格到底是栏的标题，还是列的标题。
 - row：该行的所有单元格，都与该标题单元格相关。
 - col：该列的所有单元格，都与该标题单元格相关。
 - rowgroup：多行组成的一个行组的所有单元格，都与该标题单元格相关。
 - colgroup：多列组成的一个列组的所有单元格，都与该标题单元格相关。
 - auto：默认值，表示由浏览器自行决定。



示例2-8-1：表格元素

9. 表单元素： 9.1 form

- form
 - form元素用于声明一个包含表单元素的区域，该元素并不会生成可视部分，却允许用户在该区域中添加可输入信息的表单控件元素，如文本域、下拉列表、单选框、复选框等。
 - from除支持全局属性外，具有以下特殊属性。
 - 关键属性：action
 - 指定点击表单内的【确认】按钮时，该表单信息被提交到哪个地址。该属性既可以指定一个绝对地址，也可以指定一个相对地址，为必填属性。
 - 关键属性：method
 - 指定提交表单时发送何种类型的请求，该属性值可以为get或post，分别用于发送GET或POST请求，表单默认以GET方式提交请求。GET请求和POST请求区别如下。
 - GET请求：GET请求把表单数据显式地放在URL中，并且对长度和数据值编码有所限制。GET请求会将请求的参数名和值转换成字符串，并附加在原URL之后，因此可以在地址栏中看到请求参数名和值。GET请求传送的数据量较小，一般不能大于2KB。
 - POST方式的请求：POST请求把表单数据放在HTTP请求体中，并且没有长度限制。POST请求传输的数据量总比GET传输的数据量大，而且POST请求参数以及对应的值放在HTML的HEADER中传输，用户不能在URL中看到请求参数值，安全性相对较高。

9. 表单元素： 9.1 form

- form
 - 关键属性：enctype
 - 可指定表单进行编码时所使用的字符集。其取值如下所示。
 - application/x-www-form-urlencoded，默认编码方式，数据被编码为名称和值的形式，在发送到服务器之前，所有字符都会进行编码，其中空格转换为“+”，特殊符号转换为对应的ASCII HEX值。
 - multipart/form-data：数据被编码为一条消息，页上的每个控件对应消息中的一部分。
 - text/plain：数据以纯文本的形式进行编码，其中不含任何控件或格式字符。其中，空格转换为“+”加号，但不特殊符号编码。
 - 关键属性：name
 - 指定表单的唯一名称
 - 关键属性：target
 - 指定使用哪种方式打开目标URL，与超链接中target可接受的属性值完全一样，该属性值可以是_blank、_parent、_self和_top四个值中之一。



示例2-9-1：form元素

9. 表单元素： 9.2 fieldset、legend

- fieldset、legend
 - fieldset元素是一个块级容器元素，表示控件的集合，用于将一组相关控件组合成一组。
 - legend元素用来设置fieldset控件组的标题，通常是fieldset内部的第一个元素，会嵌入显示在控件组的上边框里面。
- legend仅支持全局属性。
- fieldset除支持全局属性外，具有以下特殊属性。
 - disabled：布尔属性，一旦设置会使得<fieldset>内部包含的控件都不可用，都变成灰色状态。
 - form：指定控件组所属的<form>，它的值等于<form>的id属性。
 - name：该控件组的名称。



示例2-9-2：fieldset、legend元素



9. 表单元素： 9.3 label

- fieldset、legend
 - label元素是一个行内元素，提供控件的文字说明，帮助用户理解控件的目的。
- label除支持全局属性外，具有以下特殊属性。
 - for：关联控件的id属性。
 - form：关联表单的id属性。设置了该属性后，label可以放置在页面的任何位置，否则只能放在form内部。



示例2-9-3: label元素



9. 表单元素： 9.4 input

- input
 - input元素是一个行内元素，用来接收用户的输入。它是一个单独使用的元素，没有结束标志。
 - 它有多种类型，取决于type属性的值，默认值是text，表示一个输入框。
- input的属性非常多，有些属性是某个类型专用的，放在下文的“类型”部分介绍。这里介绍一些所有类型的共同属性。
 - autofocus：布尔属性，是否在页面加载时自动获得焦点。
 - disabled：布尔属性，是否禁用该控件。一旦设置，该控件将变灰，用户可以看到，但是无法操作。
 - form：关联表单的id属性。设置了该属性后，控件可以放置在页面的任何位置，否则只能放在form内部。
 - list：关联的datalist的id属性，设置该控件相关的数据列表。
 - name：控件的名称，主要用于向服务器提交数据时，控件键值对的键名。注意，只有设置了name属性的控件，才会向服务器提交，不设置就不会提交。
 - readonly：布尔属性，是否为只读。
 - required：布尔属性，是否为必填。
 - type：控件类型。
 - value：控件的值。

9. 表单元素： 9.4 input

• input

- type属性决定了input的形式。
- text
 - text是普通的文本输入框，用来输入单行文本。如果用户输入换行符，换行符会自动从输入中删除。text输入框有以下配套属性。
 - maxlength: 可以输入的最大字符数，值为一个非负整数。
 - minlength: 可以输入的最小字符数，值为一个非负整数，且必须小于maxlength。
 - pattern: 用户输入必须匹配的正则表达式，比如要求用户输入4个~8个英文字符，可以写成pattern="[a-z]{4,8}"。如果用户输入不符合要求，浏览器会弹出提示，不会提交表单。

9. 表单元素： 9.4 input

• input

- type属性决定了input的形式。
- text
 - placeholder: 输入字段为空时，用于提示的示例值。只要用户没有任何字符，该提示就会出现，否则会消失。
 - readonly: 布尔属性，表示该输入框是只读的，用户只能看，不能输入。
 - size: 表示输入框的显示长度有多少个字符宽，它的值是一个正整数，默认等于20。超过这个数字的字符，必须移动光标才能看到。
 - spellcheck: 是否对用户输入启用拼写检查，可能的值为true或false。

```
<input type="text" id="name" name="name"
required minlength="4" maxlength="8"
size="10">
```

9. 表单元素： 9.4 input

- input

- search

- type="search"是一个用于搜索的文本输入框，基本等同于type="text"。某些浏览器会在输入的时候，在输入框的尾部显示一个删除按钮，点击就会删除所有输入，让用户从头开始输入。

```
<form>
  <input type="search" id="mySearch" name="q" placeholder="输入搜索词....."
  required>
  <input type="submit" value="搜索">
</form>
```

9. 表单元素： 9.4 input

- input

- button

- type="button"是没有默认行为的按钮，通常脚本指定click事件的监听函数来使用。
 - 建议尽量不要使用这个类型，而使用<button>元素代替，一则语义更清晰，二则<button>元素内部可以插入图片或其他 HTML 代码。

```
<input type="button" value="点击">
```

9. 表单元素： 9.4 input

• input

• submit

- type="submit"是表单的提交按钮。用户点击这个按钮，就会把表单提交给服务器。
- 如果不指定value属性，浏览器会在提交按钮上显示默认的文字，通常是Submit。
- 该类型有以下配套属性，用来覆盖form元素的相应设置。
 - formaction: 提交表单数据的服务器 URL。
 - formenctype: 表单数据的编码类型。
 - formmethod: 提交表单使用的 HTTP 方法（get或post）。
 - formnovalidate: 一个布尔值，表示数据提交给服务器之前，是否要忽略表单验证。
 - formtarget: 收到服务器返回的数据后，在哪个窗口显示。

```
<input type="submit" value="提交">
```

9. 表单元素： 9.4 input

• input

• image

- type="image"表示将一个图像文件作为提交按钮，行为和用法与type="submit"完全一致，图像文件是一个可以点击的按钮，点击后会提交数据到服务器。
- 该类型有以下配套属性。
 - alt: 图像无法加载时显示的替代字符串。
 - src: 加载的图像 URL。
 - height: 图像的显示高度，单位为像素。
 - width: 图像的显示宽度，单位为像素。
 - formaction: 提交表单数据的服务器 URL。
 - formenctype: 表单数据的编码类型。
 - formmethod: 提交表单使用的 HTTP 方法（get或post）。
 - formnovalidate: 一个布尔值，表示数据提交给服务器之前，是否要忽略表单验证。
 - formtarget: 收到服务器返回的数据后，在哪个窗口显示。

```
<input type="image" alt="登录" src="login-button.png">
```

9. 表单元素： 9.4 input

• input

• checkbox

- type="checkbox"是复选框，允许选择或取消选择该选项。
- 此类型会在文字前面，显示一个可以点击的选择框，点击可以选中，再次点击可以取消。上面代码中，checked属性表示默认选中。
- value属性的默认值是on。也就是说，如果没有设置value属性，以上例来说，选中复选框时，会提交agreement=on。如果没有选中，提交时不会有该项。

```
<input type="checkbox" id="agreement" name="agreement" checked>
<label for="agreement">是否同意</label>
```

9. 表单元素： 9.4 input

• input

• checkbox

- 多个相关的复选框，可以放在fieldset里面。
- 如果用户同时选中两个复选框，提交的时候就会有两个name属性，比如interest=coding&interest=music。

```
<fieldset>
  <legend>你的兴趣</legend>
  <div>
    <input type="checkbox" id="coding" name="interest" value="coding">
    <label for="coding">编码</label>
  </div>
  <div>
    <input type="checkbox" id="music" name="interest" value="music">
    <label for="music">音乐</label>
  </div>
</fieldset>
```

9. 表单元素： 9.4 input

• input

• radio

- type="radio"是单选框，表示一组选择之中，只能选中一项。单选框通常为一个小圆圈，选中时会被填充或突出显示。
- 下面代码中，性别只能在两个选项之中，选择一项。
- 注意，多个单选框的name属性的值，应该都是一致的。提交到服务器的就是选中的那个值。
- 该类型的配套属性如下。
 - checked: 布尔属性，表示是否默认选中当前项。
 - value: 用户选中该项时，提交到服务器的值，默认为on。

```
<fieldset>
<legend>性别</legend>
<div>
<input type="radio" id="male" name="gender" value="male">
<label for="male">男</label>
</div>
<div>
<input type="radio" id="female" name="gender"
value="female">
<label for="female">女</label>
</div>
</fieldset>
```

9. 表单元素： 9.4 input

• input

• email

- type="email"是一个只能输入电子邮箱的文本输入框。表单提交之前，浏览器会自动验证是否符合电子邮箱的格式，如果不符合就会显示提示，无法提交到服务器。

```
<input type="email" pattern=".+@foobar.com" size="30" required>
```

- 上面代码会生成一个必填的文本框，只能输入后缀为foobar.com的邮箱地址。
- 该类型有一个multiple的布尔属性，一旦设置，就表示该输入框可以输入多个逗号分隔的电子邮箱。
- 注意，如果同时设置了multiple属性和required属性，零个电子邮箱是允许的，也就是该输入框允许为空。

```
<input id="emailAddress" type="email" multiple required>
```

9. 表单元素： 9.4 input

- input

- email

- 该类型的配套属性如下。

- maxlength: 可以输入的最大字符数。
- minlength: 可以输入的最少字符数。
- multiple: 布尔属性，是否允许输入多个以逗号分隔的电子邮箱。
- pattern: 输入必须匹配的正则表达式。
- placeholder: 输入为空时的显示文本。
- readonly: 布尔属性，该输入框是否只读。
- size: 一个非负整数，表示输入框的显示长度为多少个字符。
- spellcheck: 是否对输入内容启用拼写检查，可能的值为true或false。

9. 表单元素： 9.4 input

- input

- email

- 该类型还可以搭配<datalist>元素，提供输入的备选项。

```
<input type="email" size="40" list="defaultEmails">
<datalist id="defaultEmails">
  <option value="jbond007@mi6.defence.gov.uk">
  <option value="jbourne@unknown.net">
  <option value="nfury@shield.org">
  <option value="tony@starkindustries.com">
  <option value="hulk@grrrrrrrrr.arg">
</datalist>
```

9. 表单元素： 9.4 input

- input

- password

- type="password"是一个密码输入框。用户的输入会被遮挡，字符通常显示星号（*）或点（·）。
 - 浏览器对该类型输入框的显示，会有所差异。一种常见的处理方法是，用户每输入一个字符，先在输入框里面显示一秒钟，然后再遮挡该字符。
 - 如果用户输入内容包含换行符（U+000A）和回车符（U+000D），浏览器会自动将这两个字符过滤掉。

```
<input type="password" id="pass" name="password" minlength="8" required>
```

9. 表单元素： 9.4 input

- input

- password

- 该类型的配套属性如下。
 - maxlength: 可以输入的最大字符数。
 - minlength: 可以输入的最少字符数。
 - pattern: 输入必须匹配的正则表达式。
 - placeholder: 输入为空时的显示文本。
 - readonly: 布尔属性，该输入框是否只读。
 - size: 一个非负整数，表示输入框的显示长度为多少个字符。
 - autocomplete: 是否允许自动填充，可能的值有on（允许自动填充）、off（不允许自动填充）、current-password（填入当前网站保存的密码）、new-password（自动生成一个随机密码）。
 - inputmode: 允许用户输入的数据类型，可能的值有none（不使用系统输入法）、text（标准文本输入）、decimal（数字，包含小数）、numeric（数字0-9）等。

9. 表单元素： 9.4 input

- input

- file

- type="file"是一个文件选择框，允许用户选择一个或多个文件，常用于文件上传功能。
- 该类型有以下属性。
 - accept: 允许选择的文件类型，使用逗号分隔，可以使用 MIME 类型（比如image/jpeg），也可以使用后缀名（比如.doc），还可以使用audio/*（任何音频文件）、video/*（任何视频文件）、image/*（任何图像文件）等表示法。
 - capture: 用于捕获图像或视频数据的源，可能的值有user（面向用户的摄像头或麦克风），environment（外接的摄像头或麦克风）。
 - multiple: 布尔属性，是否允许用户选择多个文件。

```
<input type="file" id="avatar" name="avatar" accept="image/png, image/jpeg">
```

9. 表单元素： 9.4 input

- input

- hidden

- type="hidden"是一个不显示在页面的控件，用户无法输入它的值，主要用来向服务器传递一些隐藏信息。比如，CSRF 攻击会伪造表单数据，那么使用这个控件，可以为每个表单生成一个独一无二的隐藏编号，防止伪造表单提交。

```
<input id="prodId" name="prodId" type="hidden" value="xm234jq">
```

- 上面这个控件，页面上是看不见的。用户提交表单的时候，浏览器会将prodId=xm234jq发给服务器。

9. 表单元素： 9.4 input

- input

- number

- type="number"是一个数字输入框，只能输入数字。浏览器通常会在输入框的最右侧，显示一个可以点击的上下箭头，点击向上箭头，数字会递增，点击向下箭头，数字会递减。

```
<input type="number" id="tentacles" name="tentacles" min="10" max="100">
```

- 上面代码指定数字输入框，最小可以输入10，最大可以输入100。
 - 该类型可以接受任何数值，包括小数和整数。可以通过step属性，限定只接受整数。
 - 该类型有以下配套属性。
 - max: 允许输入的最大数值。
 - min: 允许输入的最小数值。
 - placeholder: 用户输入为空时，显示的示例值。
 - readonly: 布尔属性，表示该控件是否为只读。
 - step: 点击向上和向下箭头时，数值每次递减的步长值。如果用户输入的值，不符合步长值的设定，浏览器会自动四舍五入到最近似的值。默认的步长值是1，如果初始的value属性设为1.5，那么点击向上箭头得到2.5，点击向下箭头得到0.5。

9. 表单元素： 9.4 input

- input

- range

- type="range"是一个滑块，用户拖动滑块，选择给定范围之中的一个数值。因为拖动产生的值是不精确的，如果需要精确数值，不建议使用这个控件。常见的例子是调节音量。

```
<input type="range" id="start" name="volume" min="0" max="11">
```

- 上面代码会产生一个最小值为0、最大值为11的滑块区域。用户拖动滑块，选择想要的音量。
 - 该类型的配套属性如下，用法与type="number"一致。
 - max: 允许的最大值，默认为100。
 - min: 允许的最小值，默认为0。
 - step: 步长值，默认为1。
 - value属性的初始值就是滑块的默认位置。如果没有设置value属性，滑块默认就会停在最大值和最小值中间。如果max属性、min属性、value属性都没有设置，那么value属性为50。

9. 表单元素： 9.4 input

- input

- range

- 该类型与<datalist>元素配合使用，可以在滑动区域产生刻度。

```
<input type="range" list="tickmarks">
<datalist id="tickmarks">
  <option value="0" label="0%">
  <option value="10">
  <option value="20">
  <option value="30">
  <option value="40">
  <option value="50" label="50%">
  <option value="60">
  <option value="70">
  <option value="80">
  <option value="90">
  <option value="100" label="100%">
</datalist>
```

- 上面代码会在0~100之间产生11个刻度。其中，0%、50%和100%三个位置会有文字提示，不过浏览器很可能不支持。
 - 注意，浏览器生成的都是水平滑块。如果想要生成垂直滑块，可以使用 CSS 改变滑块区域的方向。

9. 表单元素： 9.4 input

- input

- url

- type="url"是一个只能输入网址的文本框。提交表单之前，浏览器会自动检查网址格式是否正确，如果不正确，就会无法提交。

```
<input type="url" name="url" id="url" placeholder="https://example.com" pattern="https://.*" size="30"
required>
```

- 上面代码的pattern属性指定输入的网址只能使用 HTTPS 协议。
 - 注意，该类型规定，不带有协议的网址是无效的，比如foo.com是无效的，http://foo.com是有效的。

9. 表单元素： 9.4 input

- input

- url

- 该类型的配套属性如下。

- maxlength: 允许的最大字符数。
 - minlength: 允许的最少字符串。
 - pattern: 输入内容必须匹配的正则表达式。
 - placeholder: 输入为空时显示的示例文本。
 - readonly: 布尔属性，表示该控件的内容是否只读。
 - size: 一个非负整数，表示该输入框显示宽度为多少个字符。
 - spellcheck: 是否启动拼写检查，可能的值为true（启用）和false（不启用）。。

9. 表单元素： 9.4 input

- input

- url

- 该类型与<datalist>元素搭配使用，可以形成下拉列表供用户选择。随着用户不断键入，会缩小显示范围，只显示匹配的备选项。

```
<input id="myURL" name="myURL" type="url" list="defaultURLs">
<datalist id="defaultURLs">
  <option value="https://developer.mozilla.org/" label="MDN Web Docs">
  <option value="http://www.google.com/" label="Google">
  <option value="http://www.microsoft.com/" label="Microsoft">
  <option value="https://www.mozilla.org/" label="Mozilla">
  <option value="http://w3.org/" label="W3C">
</datalist>
```

- 上面代码中，<option>的label属性表示文本元素，显示在备选下拉框的右侧，网址显示在左侧。

9. 表单元素： 9.4 input

- input

- tel

- type="tel"是一个只能输入电话号码的输入框。由于全世界的电话号码格式都不相同，因此浏览器没有默认的验证模式，大多数时候需要自定义验证。

```
<input type="tel" id="phone" name="phone" pattern="[0-9]{3}-[0-9]{3}-[0-9]{4}" required>
<small>Format: 123-456-7890</small>
```

- 上面代码定义了一个只能输入10位电话号码的输入框。
- 该类型的配套属性如下。
 - maxlength: 允许的最大字符数。
 - minlength: 允许的最少字符串。
 - pattern: 输入内容必须匹配的正则表达式。
 - placeholder: 输入为空时显示的示例文本。
 - readonly: 布尔属性，表示该控件的内容是否只读。
 - size: 一个非负整数，表示该输入框显示宽度为多少个字符。

9. 表单元素： 9.4 input

- input

- color

- type="color"是一个选择颜色的控件，它的值一律都是#rrggbb格式。

```
<input type="color" id="background" name="background" value="#e66465">
```

- 上面代码在浏览器中，会显示一个#e66465的色块。点击色块，就会出现一个拾色器，供用户选择颜色。
- 如果没有指定value属性的初始值，默认值为#000000（黑色）。

9. 表单元素： 9.4 input

• input

• date

- type="date"是一个只能输入日期的输入框，用户可以输入年月日，但是不能输入时分秒。输入格式是YYYY-MM-DD。

```
<input type="date" id="start" name="start" value="2018-07-22" min="2018-01-01" max="2018-12-31">
```

- 上面代码会显示一个输入框，默认日期是2018年7月22日。用户点击以后，会日期选择器，供用户选择新的日期。
- 该类型有以下配套属性。
 - max: 可以允许的最晚日期，格式为yyyy-MM-dd。
 - min: 可以允许的最早日期，格式为yyyy-MM-dd。
 - step: 步长值，一个数字，以天为单位。

9. 表单元素： 9.4 input

• input

• time

- type="time"是一个只能输入时间的输入框，可以输入时分秒，不能输入年月日。日期格式是24小时制的hh:mm，如果包括秒数，格式则是hh:mm:ss。日期选择器的形式则随浏览器不同而不同。

```
<input type="time" id="appt" name="appt" min="9:00" max="18:00" required>
<small>营业时间上午9点到下午6点</small>
```

- 该类型有以下配套属性。
 - max: 允许的最晚时间。
 - min: 允许的最早时间。
 - readonly: 布尔属性，表示用户是否不可以编辑时间。
 - step: 步长值，单位为秒。

```
<input id="appt" type="time" name="appt" step="2">
```

9. 表单元素： 9.4 input

• input

• month

- type="month"是一个只能输入年份和月份的输入框，格式为YYYY-MM。

```
<input type="month" id="start" name="start" min="2018-03" value="2018-05">
```

- 该类型有以下配套属性。

- max: 允许的最晚时间，格式为yyyy-MM。
- min: 允许的最早时间，格式为yyyy-MM。
- readonly: 布尔属性，表示用户是否不可以编辑时间。
- step: 步长值，单位为月。

9. 表单元素： 9.4 input

• input

• week

- type="week"是一个输入一年中第几周的输入框。格式为yyyy-Www，比如2018-W18表示2018年第18周。

```
<input type="week" name="week" id="camp-week" min="2018-W18" max="2018-W26" required>
```

- 该类型有以下配套属性。

- max: 允许的最晚时间，格式为yyyy-Www。
- min: 允许的最早时间，格式为yyyy-Www。
- readonly: 布尔属性，表示用户是否不可以编辑时间。
- step: 步长值，单位为周。

9. 表单元素： 9.4 input

- input

- datetime-local

- type="datetime-local"是一个时间输入框，让用户输入年月日和时分，格式为yyyy-MM-ddThh:mm。注意，该控件不支持秒。

```
<input type="datetime-local" id="meeting-time" name="meeting-time" value="2018-06-12T19:30" min="2018-06-07T00:00" max="2018-06-14T00:00">
```

- 该类型有以下配套属性。

- max: 允许的最晚时间，格式为yyyy-MM-ddThh:mm。
 - min: 允许的最早时间，格式为yyyy-MM-ddThh:mm。
 - step: 步长值，单位为秒，默认值是60。



示例2-9-4: input元素

9. 表单元素： 9.5 button

- button

- button元素会生成一个可以点击的按钮，没有默认行为，通常需要用type属性或脚本指定按钮的功能。
 - button除支持全局属性外，具有以下特殊属性。
 - autofocus
 - 布尔属性，表示网页加载时，焦点就在这个按钮。网页里面只能有一个元素，具有这个属性。
 - disabled
 - 布尔属性，表示按钮不可用，会导致按钮变灰，不可点击。
 - name
 - 按钮的名称（与value属性配合使用），将以name=value的形式，随表单一起提交到服务器。
 - value
 - 按钮的值（与name属性配合使用），将以name=value的形式，随表单一起提交到服务器。

9. 表单元素： 9.5 button

- button
 - type
 - 按钮的类型，可能的值有三种：submit（点击后将数据提交给服务器），reset（将所有控件的值重置为初始值），button（没有默认行为，由脚本指定按钮的行为）。
 - form
 - 指定按钮关联的<form>表单，值为<form>的id属性。如果省略该属性，默认关联按钮所在父表单。
 - formaction
 - 数据提交到服务器的目标 URL，会覆盖<form>元素的action属性。
 - formenctype
 - 数据提交到服务器的编码方式，会覆盖<form>元素的enctype属性。可能的值有三种：application/x-www-form-urlencoded（默认值），multipart/form-data（只用于文件上传），text/plain。
 - formmethod
 - 数据提交到服务器使用的 HTTP 方法，会覆盖<form>元素的method属性，可能的值为post或get。

9. 表单元素： 9.5 button

- button
 - formnovalidate
 - 布尔属性，数据提交到服务器时关闭本地验证，会覆盖<form>元素的novalidate属性。
 - formtarget
 - 数据提交到服务器后，展示服务器返回数据的窗口，会覆盖<form>元素的target属性。可能的值有_self（当前窗口），_blank（新的空窗口）、_parent（父窗口）、_top（顶层窗口）。



示例2-9-5: button元素

9. 表单元素： 9.5 select

- select
 - select除支持全局属性外，具有以下特殊属性。
 - autofocus
 - 布尔属性，页面加载时是否自动获得焦点。
 - disabled
 - 布尔属性，是否禁用当前控件。
 - form
 - 关联表单的id属性。
 - multiple
 - 布尔属性，是否可以选择多个菜单项。默认情况下，只能选择一项。一旦设置，多数浏览器会显示一个滚动列表框。用户可能需要按住Shift或其他功能键，选中多项。

9. 表单元素： 9.5 select

- select
 - name
 - 控件名。
 - required
 - 布尔属性，是否为必填控件。
 - size
 - 设置了multiple属性时，页面显示时一次可见的行数，其他行需要滚动查看。



示例2-9-6：select元素

9. 表单元素： 9.5 option、optgroup

- option、optgroup
 - option元素用在select、optgroup、datalist里面，表示一个菜单项，参见select的示例。
 - optgroup表示菜单项的分组，通常用在select内部。
 - option除支持全局属性外，具有以下特殊属性。
 - disabled
 - 布尔属性，是否禁用该项。
 - label
 - 该项的说明。如果省略，则等于该项的文本内容。
 - selected
 - 布尔属性，是否为默认值。显然，一组菜单中，只能有一个菜单项设置该属性。
 - value
 - 该项提交到服务器的值。如果省略，则等于该项的文本内容。



示例2-9-6: select元素

9. 表单元素： 9.5 option、optgroup

- option、optgroup
 - optgroup除支持全局属性外，具有以下特殊属性。
 - disabled
 - 布尔设置，是否禁用该组。一旦设置，该组所有的菜单项都不可选。
 - label
 - 菜单项的标题。



示例2-9-7: option、optgroup元素

9. 表单元素： 9.5 datalist

- datalist
 - datalist元素是一个容器元素，用于为指定控件提供一组相关数据，通常用于生成输入提示。它的内部使用option，生成每个菜单项。
 - 仅支持全局属性。



示例2-9-8: datalist元素

9. 表单元素： 9.5 textarea

- textarea
 - textarea是一个块级元素，用来生成多行的文本框。
 - 除支持全局属性外，具有以下特殊属性。
 - autofocus
 - 布尔属性，是否自动获得焦点。
 - cols
 - 文本框的宽度，单位为字符，默认值为20。
 - disabled
 - 布尔属性，是否禁用该控件。
 - form
 - 关联表单的id属性。
 - maxlength
 - 允许输入的最大字符数。如果未指定此值，用户可以输入无限数量的字符。
 - minlength
 - 允许输入的最小字符数。

9. 表单元素： 9.5 textarea

- textarea
 - name
 - 控件的名称。
 - placeholder
 - 输入为空时显示的提示文本。
 - readonly
 - 布尔属性，控件是否为只读。
 - required
 - 布尔属性，控件是否为必填。
 - rows
 - 文本框的高度，单位为行。

9. 表单元素： 9.5 textarea

- textarea
 - spellcheck
 - 是否打开浏览器的拼写检查。可能的值有true（打开），default（由父元素或网页设置决定），false（关闭）。
 - wrap
 - 输入的文本是否自动换行。可能的值有hard（浏览器自动插入换行符CR + LF，使得每行不超过控件的宽度），soft（输入内容超过宽度时自动换行，但不会加入新的换行符，并且浏览器保证所有换行符都是CR + LR，这是默认值），off（关闭自动换行，单行长度超过宽度时，会出现水平滚动条）。



示例2-9-10: textarea元素

9. 表单元素： 9.5 progress

- progress
 - progress元素是一个行内元素，表示任务的完成进度。浏览器通常会将显示为进度条。
 - 除支持全局属性外，具有以下特殊属性。
 - max
 - 进度条的最大值，应该是一个大于0的浮点数。默认值为1。
 - value
 - 进度条的当前值。它必须是0和max属性之间的一个有效浮点数。如果省略了max属性，该值则必须在0和1之间。如果省略了value属性，则进度条会出现滚动，表明正在进行中，无法知道完成的进度。



示例2-9-10: progress元素

10. 多媒体元素： 10.1 video

- video
 - video元素是一个块级元素，用于放置视频。如果浏览器支持加载的视频格式，就会显示一个播放器，否则显示video内部的子元素。
 - 除支持全局属性外，具有以下特殊属性。
 - src
 - 视频文件的网址。
 - controls
 - 播放器是否显示控制栏。该属性是布尔属性，不用赋值，只要写上属性名，就表示打开。如果不想使用浏览器默认的播放器，而想使用自定义播放器，就不要使用该属性。
 - width
 - 视频播放器的宽度，单位像素。
 - height
 - 视频播放器的高度，单位像素。
 - autoplay
 - 视频是否自动播放，该属性为布尔属性。

10. 多媒体元素： 10.1 video

- video
 - video元素是一个块级元素，用于放置视频。如果浏览器支持加载的视频格式，就会显示一个播放器，否则显示video内部的子元素。
 - 除支持全局属性外，具有以下特殊属性。
 - loop
 - 视频是否循环播放，该属性为布尔属性。
 - muted
 - 是否默认静音，该属性为布尔属性。
 - poster
 - 视频播放器的封面图片的 URL。
 - preload
 - 视频播放之前，是否缓冲视频文件。这个属性仅适合没有设置autoplay的情况。它有三个值，分别是none（不缓冲）、metadata（仅仅缓冲视频文件的元数据）、auto（可以缓冲整个文件）。
 - playsinline
 - iPhone 的 Safari 浏览器播放视频时，会自动全屏，该属性可以禁止这种行为。该属性为布尔属性。

10. 多媒体元素： 10.1 video

- video
 - crossorigin
 - 是否采用跨域的方式加载视频。它可以取两个值，分别是anonymous（跨域请求时，不发送用户凭证，主要是 Cookie），use-credentials（跨域时发送用户凭证）。
 - currentTime
 - 指定当前播放位置（双精度浮点数，单位为秒）。如果尚未开始播放，则会从这个属性指定的位置开始播放。
 - duration
 - 该属性只读，指示时间轴上的持续播放时间（总长度），值为双精度浮点数（单位为秒）。如果是流媒体，没有已知的结束时间，属性值为+Infinity。



示例2-10-1: video元素

10. 多媒体元素： 10.1 audio

- audio
 - audio元素是一个块级元素，用于放置音频，用法与video元素基本一致。
 - 除支持全局属性外，具有以下特殊属性。
 - autoplay
 - 是否自动播放，布尔属性。
 - controls
 - 是否显示播放工具栏，布尔属性。如果不设置，浏览器不显示播放界面，通常用于背景音乐。
 - crossorigin
 - 是否使用跨域方式请求。
 - loop
 - 是否循环播放，布尔属性。
 - muted
 - 是否静音，布尔属性。
 - preload
 - 音频文件的缓冲设置。
 - src
 - 音频文件网址。



示例2-10-2: audio元素

10. 多媒体元素： 10.1 audio

- video和audio在浏览器开始请求媒体数据、下载媒体数据、播放媒体数据到播放结束，触发的事件如下所示。

事件	描述	事件	描述
abort	当音频/视频的加载已放弃时	play	当音频/视频已开始或不再暂停时
canplay	当浏览器可以播放音频/视频时	playing	当音频/视频在已因缓冲而暂停或停止后已就绪时
canplaythrough	当浏览器可在不因缓冲而停顿的情况下进行播放时	progress	当浏览器正在下载音频/视频时
durationchange	当音频/视频的时长已更改时	ratechange	当音频/视频的播放速度已更改时
emptied	当目前的播放列表为空时	seeked	当用户已移动/跳跃到音频/视频中的新位置时
ended	当目前的播放列表已结束时	seeking	当用户开始移动/跳跃到音频/视频中的新位置时
error	当在音频/视频加载期间发生错误时	stalled	当浏览器尝试获取媒体数据，但数据不可用时
loadeddata	当浏览器已加载音频/视频的当前帧时	suspend	当浏览器刻意不获取媒体数据时
loadedmetadata	当浏览器已加载音频/视频的元数据时	timeupdate	当目前的播放位置已更改时
loadstart	当浏览器开始查找音频/视频时	volumechange	当音量已更改时
pause	当音频/视频已暂停时		

10. 多媒体元素： 10.1 source

- source

source元素用于picture、video、audio的内部，用于指定一项外部资源。除支持全局属性外，具有以下特殊属性。

- type
 - 指定外部资源的 MIME 类型。
- src
 - 指定源文件，用于<video>和<audio>。
- srcset
 - 指定不同条件下加载的图像文件，用于<picture>。
- media
 - 指定媒体查询表达式，用于<picture>。
- sizes
 - 指定不同设备的显示大小，用于<picture>，必须跟srcset搭配使用。

10. 多媒体元素： 10.1 source

- source

- <source>元素是空元素，这意味着它不仅没有内容，也没有结束标记。也就是说，在 HTML 中永远不会使用 “</source>”。
- 浏览器会通过 <source> 元素列表来查找它所支持的格式。浏览器会使用它能显示的第一个格式。对于每个 <source> 元素：
 - 如果未指定 type 属性，浏览器会从服务器检索媒体类型，并确定是否可以显示。如果媒体无法呈现，浏览器会检查列表中的下一个 <source>。
 - 如果指定了 type 属性，浏览器会立即将其与可显示的媒体类型进行比较。如果不支持该类型，浏览器会跳过服务器查询，直接检查下一个 <source> 元素。
- 如果没有一个 <source> 元素提供可用的源：
 - 如果是 <picture> 元素，浏览器将退回到使用 <picture> 元素的 子元素中指定的图片。
 - 如果是 <audio> 或 <video> 元素，浏览器将恢复显示该元素的开头和结尾标记之间的内容。



示例2-10-3: source元素

10. 多媒体元素： 10.1 track

- track

track元素用于指定视频的字幕，格式是 WebVTT （.vtt文件），放置在元素内部。除支持全局属性外，具有以下特殊属性。

- label
 - 播放器显示的字幕名称，供用户选择。
- kind
 - 字幕的类型，默认是subtitles，表示将原始声音翻译成外国文字，比如英文视频提供中文字幕。另一个常见的值是captions，表示原始声音的文字描述，通常是视频原始使用的语言，比如英文视频提供英文字幕。
- src
 - vtt 字幕文件的网址。
- srclang
 - 字幕的语言，必须是有效的语言代码。
- default
 - 是否默认打开，布尔属性。

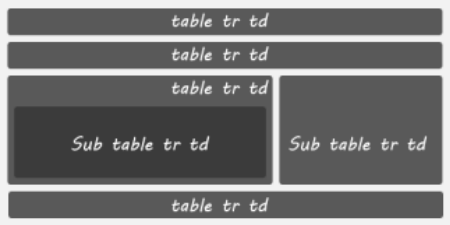


示例2-10-4: track元素

扩展：布局的变化：从HTML4到HTML5

- 使用表格布局

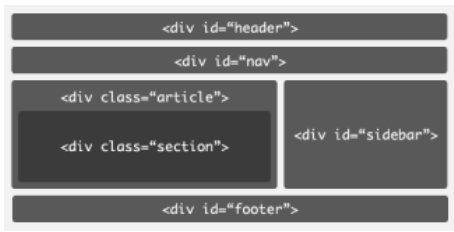
- 传统表格布局方式实际上是利用HTML中的表格元素table具有的无边框特性，当表格元素在显示时，使单元格的边框和间距设置为0，然后将网页中的各个元素按版式划分放入表格的各单元格中，从而实现复杂的排版组合。





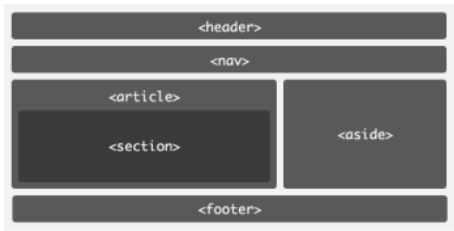
扩展：布局的变化：从HTML4到HTML5

- 使用区块布局
 - 在XHTML 1.0之后，W3C开始大力推行Web标准，开发工程师基本上都是用了DIV+CSS的布局方式，即区块布局。但是，搜索引擎去抓取页面的内容时，它只能猜测某个DIV内的内容是文章内容容器，或者是导航模块的容器，或者是作者介绍的容器等等，整个HTML文档结构定义不清晰，仅仅是在展示和页面解析性能上得到了提升。



扩展：布局的变化：从HTML4到HTML5

- 使用结构元素布局
 - 在XHTML 1.0之后，W3C开始大力推行Web标准，开发工程师基本上都是用了DIV+CSS的布局方式，即区块布局。但是，搜索引擎去抓取页面的内容时，它只能猜测某个DIV内的内容是文章内容容器，或者是导航模块的容器，或者是作者介绍的容器等等，整个HTML文档结构定义不清晰，仅仅是在展示和页面解析性能上得到了提升。





扩展：语义Web

- div元素是Web前端开发中最常用的元素，利用div元素，可以把整个HTML文档分隔为页眉、侧边面板、导航条等等。再辅以少量可靠的CSS，就可以把这些区块转换成带边框的盒子或带阴影的分栏，而且各就各位。
- div加样式的技术简明、强大且非常灵活，但却不够透明。也就是说，在查看别人的源代码时，必须费点功夫才能知道哪个div表示什么，而整个页面又是怎么搭建起来的。
- HTML5为此引用了一组构造页面的新元素，实现对页面区块内容的定义。这些元素可以为它们标注的内容赋予额外的含义。



扩展：语义Web

- 使用语义元素有多种原因：
- 容易修改和维护
 - 解读传统的网页比较困难。要想理解整体布局 and 不同区块的重要程度，需要一遍一遍地看网页的样式表。但通过HTML5的语义元素，使用标记就可以传达出额外的结构化信息。
- 无障碍性
 - 现代Web设计的一个重要主题，就是让任何人都能无障碍访问页面。换句话说，要让使用屏幕阅读器和其他辅助工具的人都能在页面中自由导航。目前，开发无障碍工具的公司在为兼容HTML5而努力工作中，等他们开发完成了，残疾人士就可以有更好的上网体验了。
- 搜索引擎优化
 - 像谷歌这样的搜索引擎，会使用强大的搜索机器人，这些搜索机器人自动在Web中爬行并获取每一个网页。然后扫描网页内容并将它们索引到搜索数据库中。如果谷歌能够更好地理解你的站点，那搜索者的查询就会越容易与你的内容匹配，因而你的网站列在搜索结果中的可能性也就越大。通过HTML5的语义化元素可以使谷歌的搜索引擎更容易地理解你的网页，提高网站的排名。
- 未来的功能
 - 新浏览器和Web编辑工具一定会以这样或者那样的方式利用语义元素。比如，浏览器可以提供一个页面的内容摘要，以便跳转到页面汇总适当的区块。类似地，网页设计工具也可能会包含一些方便构建或编辑导航菜单的功能，而方法就是组织放在<nav>中的内容。

重点回顾

● HTML基础和文档结构

- HTML概述：HTML是构建网页的基础技术，用于创建网页结构、排列内容、展示多媒体元素等。HTML文档由HTML元素组成，最新版本为HTML5。
- HTML元素结构：HTML元素由开始标签、内容和结束标签组成。标签用于标识元素的开始和结束，内容则是元素所包含的信息。
- HTML元素分类：HTML元素根据内容和功能被分为多个类别，包括基础、格式、表单、框架、图像、音频/视频、链接、列表、表格、样式/节、元信息和编程元素。
- HTML文档结构：一个标准的HTML文档结构包括<!DOCTYPE html>声明、<html>根元素、<head>头部区域和<body>主要内容区域。文档类型声明是历史遗留问题，<html>元素包裹所有内容，<head>包含元数据，<body>包含页面可见内容。

● HTML元素的使用和属性

- 块级元素和内联元素：HTML中有块级元素和内联元素之分。块级元素独占一行，内联元素则不会换行。
- 嵌套元素：元素可以嵌套在其他元素中，但必须正确地开启和关闭。
- 空元素：某些元素只有一个标签，如，用于插入图片。
- HTML中的空白：HTML解释器会将连续的空白字符减少为一个空格。
- HTML中的特殊字符：特殊字符如<、>、"、'和&在HTML中有特殊含义，必须使用字符引用表示。
- HTML中的注释：注释用于解释代码，不会显示在页面上。

重点回顾

● HTML属性

- 属性介绍：属性提供元素的额外信息，如class属性用于样式设置。
- 公共属性：HTML元素包含多种属性，如基本属性、语言属性、键盘属性、内容属性和延伸属性。
- 全局属性：全局属性适用于所有HTML元素，如id、class、style等。

● HTML结构元素

- header：表示页面或内容区块的头部，可以包含导航链接、标题等。
- footer：表示页面、文章或章节的尾部，通常包含版权信息或其他相关信息。
- nav：用于放置页面或文档的导航信息，如站点导航或文章导航。
- article：表示页面中一段完整的内容，如一篇文章或论坛帖子。
- section：用于对页面上的内容进行分块，通常包含标题。
- aside：放置与主要内容间接相关的部分，如侧边栏或评论。
- hgroup：将标题及其子标题进行分组。
- h1 ~ h6：六个不同级别的标题元素。
- address：表示文档作者或维护者的联系信息。
- time：表示时间或日期，支持机器可读格式。



重点回顾

- HTML链接元素
 - a: 用于创建超链接，可以链接到其他页面或文件。
- HTML文本元素
 - div: 通用块级容器元素，用于分组内容。
 - p: 表示文本的一个段落。
- span: 通用行级容器，用于对元素进行编组。
 - br、wbr:
用于换行，<wbr>表示可选的断行位置。
- HTML列表元素
 - ol、ul、li: 有序列表、无序列表和列表项。
 - dl、dd、dt: 描述列表、描述术语和描述细节。
- HTML图像元素
 - img: 用于插入图片，具有src、alt、width、height等属性。
- HTML表格元素
 - table、caption、thead、tbody、tfoot、colgroup、col、tr、th、td: 用于创建和定义表格结构。



重点回顾

- HTML表单元素
 - form: 用于声明包含表单元素的区域。
 - fieldset、legend: 用于将一组相关控件组合成一组，并提供控件组的标题。
 - label: 提供控件的文字说明。
 - input: 接收用户输入的多种类型控件。
 - button: 生成可点击的按钮。
 - select、option、optgroup: 生成下拉菜单和菜单项。
 - datalist: 为指定控件提供一组相关数据。
 - textarea: 生成多行文本框。
 - progress: 表示任务的完成进度。
- HTML多媒体元素
 - video: 用于放置视频。
 - audio: 用于放置音频。
 - source: 指定外部资源。
 - track: 指定视频的字幕。



重点回顾

- HTML布局和语义Web
 - 布局的变化：从HTML4到HTML5，布局方式从表格布局、区块布局发展到结构元素布局。
 - 语义Web：HTML5引入的结构元素赋予内容额外的含义，有助于页面的修改、维护、无障碍访问和搜索引擎优化。

信创智能医疗系统研发课程体系

河南中医药大学信息技术学院（智能医疗行业学院）



河南中医药大学信息技术学院（智能医疗行业学院）智能医疗教研室

河南中医药大学医疗健康信息工程技术研究所