**HTML学习**

HTML实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<h1>我的第一个标题</h1>

<p>我的第一个段落。</p>

​

</body>

</html>

**实例解析：**

* <!DOCTYPE html>声明为HTML5文档
* <html>元素是HTML页面的根元素
* <head>元素包含了文档的元数据，如<meta charset=”utf-8”>定义网页编码格式为utf-8
* <title>元素描述了文档的标题
* <body>元素包含了可见的页面内容
* <h1>元素定义了一个大标题
* <p>元素定义了一个段落

**注**：在浏览器的页面上使用键盘上的F12按键开启调试模式，就可以看到组成标签

**什么是HTML?**

HTML是用来描述网页的一种语言

* HTML指的是超文本标记语言
* HTML不是一种编程语言，而是一种标记语言
* 标记语言是一套标记标签
* HTML使用标记标签来描述网页
* HTML文档包含了HTML标签及文本内容
* HTML文档也叫做web页面

**HTML标签**

* 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<b>和</b>
* 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

<标签>内容</标签>

HTML元素

“HTML标签”和”HTML元素”通常都是描述同样的意思

严格来说，一个HTML元素包含了开始标签与结束标签，如下实例：

<p>这是一个段落。</p>

<!DOCTYPE>声明

<!DOCTYPE>声明有助于浏览器中正确显示网页

网络上有很多不同的文件，如果能够正确声明HTML的版本，浏览器就能正确显示网页内容

doctype声明是不区分大小写的，以下方式均可：

<!DOCTYPE html>   
<!DOCTYPE HTML>   
<!doctype html>   
<!Doctype Html>

通用声明

HTML5

<!DOCTYPE html>

HTML4.01

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

XHTML1.0

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"  
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

中文编码

目前在大部分浏览器中，直接输出中文会出现中文乱码的情况，这时候就需要在头部将字符声明为UTF-8或GBK.

HTML实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>页面标题</title>

</head>

<body>

​

<h1>我的第一个标题</h1>

​

<p>我的第一个段落。</p>

​

</body>

</html>

​

**HTML基础-4个实例**

HTML标题

HTML标题是通过<h1>-<h6>标签来定义的

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<h1>这是标题 1</h1>

<h2>这是标题 2</h2>

<h3>这是标题 3</h3>

<h4>这是标题 4</h4>

<h5>这是标题 5</h5>

<h6>这是标题 6</h6>

​

</body>

</html>

​

HTML段落

HTML段落是通过标签<p>来定义的

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<p>这是一个段落。</p>

<p>这是一个段落。</p>

<p>这是一个段落。</p>

​

</body>

</html>

**HTML链接**

HTML链接是通过标签<a>来定义的

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<a href="//www.runoob.com">这是一个链接使用了 href 属性</a>

​

</body>

</html>

提示：在href属性中指定链接的地址

**HTML图像**

HTML图像是通过标签<img>来定义的

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<img src="/images/logo.png" width="258" height="39" />

​

</body>

</html>

**HTML元素**

HTML文档由HTML元素定义

HTML元素

|  |  |  |
| --- | --- | --- |
| 开始标签 | 元素内容 | 结束标签 |
| <p> | 这是一个段落 | </p> |
| <a href=”default.htm”> | 这是一个链接 | </a> |
| <br> | 换行 |  |

**HTML元素语法**

* HTML元素以开始标签起始
* HTML元素以结束标签中止
* 元素的内容是开始标签与结束标签之间的内容
* 某些HTML元素具有空内容
* 空元素在开始标签中进行关闭
* 大多数HTML元素可拥有属性

**嵌套的HTML元素**

HTML文档由嵌套的HTML元素构成

实例：

<!DOCTYPE html>  
<html>  
  
<body>  
<p>这是第一个段落。</p>  
</body>  
  
</html>

**以上实例包含了三个HTML元素**

实例解析：

<p>元素：

<p>这是第一个段落。</p>

这个<p>元素定义了HTML文档中的一个段落

这个元素拥有一个开始标签<p>以及一个结束标签</p>

元素内容：这是一个段落

<body>元素

<body>  
<p>这是第一个段落。</p>  
</body>

<body>元素定义了HTML文档的主体

这个元素拥有一个开始标签<body>以及一个结束标签</body>

元素内容是另一个HTML元素(p元素）

<html>元素：

<html>  
  
<body>  
<p>这是第一个段落。</p>  
</body>  
  
</html>

<html>元素定义了整个HTML文档

这个元素拥有一个开始标签<html>和一个结束标签</html>

元素内容是另一个HTML元素(body元素）

**HTML空元素**

没有内容的HTML元素被称为空元素，空元素是在开始标签中关闭的

<br>就是没有关闭标签的空元素（<br>标签定义换行）

在XHTML,XML以及未来版本的HTML中，所有元素都必须被关闭

在开始标签中添加斜杠，比如<br/>是关闭元素的正确方法，HTML,XHTML和XML都接受这种方式。

即使<br>在所有浏览器中都是有效的，但使用<br/>其实是更长远的保障

**HTML属性**

* HTML元素可以设置属性
* 属性可以在元素中添加附加信息
* 属性一般描述于开始标签
* 属性总是以名称/值对的形式出现，比如：name=“value”

属性实例

HTML链接由<a>标签定义。链接的地址在href属性中指定：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<a href="//www.runoob.com">这是一个链接使用了 href 属性</a>

​

</body>

</html>

HTML属性常用引用属性值

属性值应该始终被包括在引号内

双引号是最常用的，不过使用单引号也没有问题

提示：在某些个别的情况下，比如属性值本身就含有双引号，那么必须使用单引号

**HTML标题**

在HTML文档中，标题很重要

**HTML标题**

标题是通过<h1>-<h6>标签进行定义的

<h1>定义最大的标题

<h6>定义最小的标题

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<h1>这是标题 1</h1>

<h2>这是标题 2</h2>

<h3>这是标题 3</h3>

<h4>这是标题 4</h4>

<h5>这是标题 5</h5>

<h6>这是标题 6</h6>

​

</body>

</html>

**注释**：浏览器会自动的在标题的前后添加空行

**HTML水平线**

<hr>标签在HTML页面中创建水平线

<hr>元素可用于分隔内容

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

<p>hr 标签定义水平线：</p>

<hr />

<p>这是段落。</p>

<hr />

<p>这是段落。</p>

<hr />

<p>这是段落。</p>

</body>

</html>

**<HTML>注释**

可以将注释插入HTML代码中，可以提高可读性。浏览器会忽略注释，也不会显示他们。

注释：开始括号之后（左边的括号）需要紧跟一个叹号，结束括号（右边的括号）不需要，合理地使用注释可以对未来的代码编辑工作产生帮助。

**HTML段落**

HTML可以将文档分割为若干段落。

段落是通过<p>标签定义的

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<p>这是一个段落。</p>

<p>这是一个段落。</p>

<p>这是一个段落。</p>

​

</body>

</html>

**注意**：浏览器会自动的段落的前后添加空行。

**HTML折行**

在不产生一个新段落的情况下进行换行，请使用<br>标签

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<p>这个<br>段落<br>演示了分行的效果</p>

​

</body>

<br/>元素是一个空的HTML元素，由于关闭标签没有任何意义，因此它没有结束标签。

**HTML输出-使用提醒**

无法确定HTML被显示的确切效果。屏幕的大小，以及对窗口的调整都可能导致不同的结果。对于HTML，无法通过在HTML代码中添加额外的空格或换行来改变输出的效果。

当显示页面是，浏览器会移除源代码中多余的空格和空行。所有连续的空格或空行都会被算作一个空格。需要注意的是，HTML代码中的所有连续的空行（换行）也被显示为一个空格

**HTML文本格式化**

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<b>加粗文本</b><br><br>

<i>斜体文本</i><br><br>

<code>电脑自动输出</code><br><br>

这是 <sub> 下标</sub> 和 <sup> 上标</sup>

​

</body>

</html>

**HTML格式化标签**

HTML使用便签<b>与<i>对输出的文本

这些HTML标签被称为格式化标签

**通常标签 <strong> 替换加粗标签 <b> 来使用, <em> 替换 <i>标签使用。**  
然而，这些标签的含义是不同的：  
<b> 与<i> 定义粗体或斜体文本。  
<strong> 或者 <em>意味着你要呈现的文本是重要的，所以要突出显示。现今所有主要浏览器都能渲染各种效果的字体。不过，未来浏览器可能会支持更好的渲染效果。

**HTML文本格式化标签**

|  |  |
| --- | --- |
| 标签 | 描述 |
| <b> | 定义粗体文本 |
| <em> | 定义着重文字 |
| <i> | 定义斜体字 |
| <small> | 定义小号字 |
| <strong> | 定义加重语气 |
| <sub> | 定义下标字 |
| <sup> | 定义上标字 |
| <ins> | 定义插入字 |
| <del> | 定义删除字 |

**HTML“计算机输出”标签**

|  |  |
| --- | --- |
| 标签 | 描述 |
| <code> | 定义计算机代码 |
| <kbd> | 定义键盘码 |
| <samp> | 定义计算机代码样本 |
| <var> | 定义变量 |
| <pre> | 定义预格式文本 |

**HTML引文，引用，及标签定义**

|  |  |
| --- | --- |
| 标签 | 描述 |
| <abbr> | 定义缩写 |
| <address> | 定义地址 |
| <bdo> | 定义文字方向 |
| <blockquote> | 定义长的引用 |
| <q> | 定义短的引用语 |
| <cite> | 定义引用，引证 |
| <dfn> | 定义一个定义项目 |

**HTML链接**

HTML使用超级链接与网络上的另一个文档连接，几乎可以在所有网页中找到链接。点解链接可以从一个页面跳转到另一张页面。

**HTML超链接**

HTML使用标签<a>来设置超文本链接

超链接可以是一个字，一个词，或者一组词也可以是一幅图像，可以点击这些内容来跳转到新的文档或者当前文档中的某个部分

当把鼠标指针移动到网页中的某个链接上时，箭头会变为一只小手

在标签<a>中使用了href属性来描述链接的地址

默认情况下，链接将以以下形式出现在浏览器中：

* 一个未访问过的链接显示为蓝色字体并带有下划线
* 访问过的链接显示为紫色并带有下划线
* 点击链接时，链接显示为红色并带有下划线

注意：如果为这些超链接设置了CSS样式，展示样式会根据CSS的设定而显示

**HTML链接语法**

链接的HTML代码很简单。它类似这样：

<a href="*url*">*链接文本*</a>

href属性描述了链接的目标

实例：

<a href="https://www.runoob.com/">访问菜鸟教程</a>

上面这行代码显示为：访问菜鸟教程

点击这个超链接会把用户带到菜鸟教程的首页

提示：“链接文本”不必一定是文本。图片或其他HTML元素可以成为链接

**HTML链接-target属性**

使用target属性，可以定义被链接的文档在何处显示

下面的这行会在新窗口打开文档

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<a href="https://www.runoob.com/" target="\_blank">访问菜鸟教程!</a>

​

<p>如果你将 target 属性设置为 &quot;\_blank&quot;, 链接将在新窗口打开。</p>

​

</body>

</html>

**HTML链接-id属性**

id属性可用于创建在一个HTML文档书签标记

提示：书签是不以任何特殊的方式显示，在HTML文档中是不显示的，所以对于读者来说是隐藏的

实例：

在HTML文档中插入ID：

<a id="tips">有用的提示部分</a>

在HTML文档中创建一个链接到“有用 提示部分（id=”tips”）”:

<a href="#tips">访问有用的提示部分</a>

或者，从另一个页面创建一个链接到“有用 提示部分（id=”tips”）”：

<a href="https://www.runoob.com/html/html-links.html#tips">  
访问有用的提示部分</a>

**HTML<head>**

HTML<head>元素

<head>元素包含了所有的头部标签元素。在<head>元素中可以插入脚本，样式文件，及各种meta信息

可以添加在头部区域的元素标签为：<title>,<style>,<meta>,<link>,<script>,<noscript>,and<base>

**HTML<tltle>元素**

<title>标签定义了不同文档的标题

<title>在HTML/XHTML文档中是必须的

<title>元素：

* 定义了浏览器工具栏的标题
* 当网页添加到收藏夹时，显示在收藏夹中的标题
* 显示在搜索引擎结果页面的标题

一个简单的HTML文档：

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>文档标题</title>

</head>

<body> 文档内容......

</body>

</html>

**HTML<base>元素**

<base>标签描述了基本的链接地址/链接目标，该标签作为HTML文档中所有的链接标签的默认链接：

<head>

<base href="http://www.runoob.com/images/" target="\_blank">

</head>

**HTML<link>元素**

<link>标签定义了文档与外部资源之间的关系

<link>标签通常用于链接到样式表

<head>

<link rel="stylesheet" type="text/css" href="mystyle.css">

</head>

**HTML<style>元素**

<style>标签定义了HTML文档的样式文件引用地址

在<style>元素中你也可以直接添加样式来渲染HTML文档：

<head>

<style type="text/css">

body {background-color:yellow}

p {color:blue}

</style>

</head>

**HTML<meta>元素**

meta标签描述了一些基本的元数据

<meta>标签提供了元数据。元数据也不显示在页面上，但会被浏览器解析

META元素通常用于指定网页的描述，关键词，文件的最后修改时间，作者和其他元数据。元数据可以使用于浏览器，搜索引擎，或其他Web服务

<meta>一般放置于<head>区域

<meta>标签-使用实例

为搜索引擎定义关键词：

<meta name="keywords" content="HTML, CSS, XML, XHTML, JavaScript">

为网页定义描述内容：

<meta name="description" content="免费 Web & 编程 教程">

定义网页作者：

<meta name="author" content="Runoob">

每30秒钟刷新当前页面：

<meta http-equiv="refresh" content="30">

**HTML<script>元素**

<script>标签用于加载脚本文件，如：JavaScript

HTML head元素

|  |  |
| --- | --- |
| 标签 | 描述 |
| <head> | 定义了文档的信息 |
| <title> | 定义了文档的标题 |
| <base> | 定义了页面链接标签的默认链接地址 |
| <link> | 定义了一个文档和外部资源之间的关系 |
| <meta> | 定义了HTML文档中的元数据 |
| <script> | 定义了客户端的脚本文件 |
| <style> | 定义了HTML文档的样式文件 |

**HTML样式-CSS**

用于渲染HTML元素标签的样式

**如何使用CSS**

CSS是在HTML4开始使用的，是为了更好地渲染HTML元素引入的

可以通过以下方式添加到HTML中：

* 内联样式：在HTML元素中使用“style”属性
* 内部样式表：在HTML文档头部<head>区域使用<style>元素来包含CSS
* 外部引用：使用外部CSS文件（**best!**）

**内联样式**

当特殊的样式需要用到个别元素时，可以使用内联样式，方法是在相关的标签中使用样式属性，可以包含任何CSS属性。

实例：

<p style="color:blue;margin-left:20px;">这是一个段落。</p>

**HTML样式实例-背景颜色**

背景色属性定义一个元素的背景颜色

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body style="background-color:yellow;">

<h2 style="background-color:red;">这是一个标题</h2>

<p style="background-color:green;">这是一个段落。</p>

</body>

</html>

**HTML样式实例-字体，字体颜色，字体大小**

可以使用font-family（字体），color(颜色）,和font-size（字体大小）属性来定义字体的样式

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

<h1 style="font-family:verdana;">一个标题</h1>

<p style="font-family:arial;color:red;font-size:20px;">一个段落。</p>

</body>

</html>

**HTML样式实例-文本对齐方式**

使用texu-align(文字对齐）属性指定文本的水平与垂直对齐方式

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<h1 style="text-align:center;">居中对齐的标题</h1>

<p>这是一个段落。</p>

​

</body>

</html>

**内部样式表**

当单个文件需要特别样式时，可以使用内部样式表，可以在<head>部分通过<style>标签定义内部样式表：

<head>

<style type="text/css">

body {background-color:yellow;}

p {color:blue;}

</style>

</head>

**外部样式表**

当阳市需要被应用到很多页面时，使用外部样式表，可以通过更改一个文件来改变整个站点的外观。

<head>

<link rel="stylesheet" type="text/css" href="mystyle.css">

</head>

**HTML样式标签**

|  |  |
| --- | --- |
| 标签 | 描述 |
| <style> | 定义文本样式 |
| <link> | 定义资源引用地址 |

**HTML图像**

HTML图像-图像标签（<img>）和源属性（Src）

在HTML中，<img>定义图像

<img>是空标签，只包含属性，并且没有闭合标签

要让页面显示图像，需要使用源属性（src），源属性的值是图像的URL地址

语法：

<img src="*url*" alt="*some\_text*">

URL指存储图像的位置。浏览器将图像显示在文档中图像标签出现的地方。

**HTML图像-Alt属性**

用来为图像定义遗传预备队可替换的文本

替换的文本属性的值是自定义的：

<img src="boat.gif" alt="Big Boat">

在浏览器无法载入图像时，替换文本属性告诉读者她们失去的信息。此时，浏览器将显示这个替代性的文本而不是图像。为页面上的图像都加上替换文本属性是个好习惯，这样有助于更好的显示信息，并且对于那些使用纯文本浏览器的人来说是非常有用的。

**HTML图像-设置图像的高度与宽度**

height与width属性用于设置图像的高度与宽度

属性值默认单位为像素：

<img src="pulpit.jpg" alt="Pulpit rock" width="304" height="228">

提示：如果指定了高度宽度，页面加载时就会保留指定的尺寸。如果没有指定大小，加载页面可能会破坏HTML页面的整体布局。

**HTML图像标签**

|  |  |
| --- | --- |
| 标签 | 描述 |
| <img> | 定义图像 |
| <map> | 定义图像地图 |
| <area> | 定义图像地图中的可点击区域 |

**HTML表格**

表格有<table>标签来定义，每个表格都有若干行（由<tr>标签定义）每行被分隔为若干个单元格（由<td>标签定义）。字母td指表格数据。数据单元格可以包含文本，图片，列表，段落，表单，水平线，表格等等。

实例：

<table border="1">

<tr>

<td>row 1, cell 1</td>

<td>row 1, cell 2</td>

</tr>

<tr>

<td>row 2, cell 1</td>

<td>row 2, cell 2</td>

</tr>

</table>

**HTML表格和边框属性**

不顶用边框属性，表格不显示边框，属于边框属性来显示一个带有边框的表格

实例：

<table border="1">

<tr>

<td>Row 1, cell 1</td>

<td>Row 1, cell 2</td>

</tr>

</table>

**HTML表格表头**

使用<th>标签进行定义。大多数会把表头显示为粗体居中的文本

实例：

<table border="1">

<tr>

<th>Header 1</th>

<th>Header 2</th>

</tr>

<tr>

<td>row 1, cell 1</td>

<td>row 1, cell 2</td>

</tr>

<tr>

<td>row 2, cell 1</td>

<td>row 2, cell 2</td>

</tr>

</table>

**HTML表格标签**

|  |  |
| --- | --- |
| 标签 | 描述 |
| <table> | 定义表格 |
| <th> | 定义表格的表头 |
| <tr> | 定义表格的行 |
| <td> | 定义表格单元 |
| <caption> | 定义表格标题 |
| <colgroup> | 定义表格列的组 |
| <col> | 定义用于表格列的属性 |
| <thead> | 定义表格的页眉 |
| <tbody> | 定义表格的主体 |
| <tfoot> | 定义表格的页脚 |

**HTML列表**

支持有序，无序和定义列表

**HTML无序列表**

无序列表是一个项目的列表，项目使用粗体原点进行标记

无序列表使用<ul>标签

<ul>  
<li>Coffee</li>  
<li>Milk</li>  
</ul>

浏览器显示如下：

* Coffee
* Milk

**HTML有序列表**

使用数字进行标记。始于<ol>标签。每个列表项始于<li>标签

<ol>  
<li>Coffee</li>  
<li>Milk</li>  
</ol>

浏览器中显示如下：

1. Coffee
2. Milk

**HTML自定义列表**

项目及其注释的组合。以<dl>标签开始。每个自定义列表项以<dt>开始。每个自定义列表项的定义以<dd>开始

<dl>  
<dt>Coffee</dt>  
<dd>- black hot drink</dd>  
<dt>Milk</dt>  
<dd>- white cold drink</dd>  
</dl>

浏览器显示如下：

Coffee

- black hot drink

Milk

- white cold drink

**HTML列表标签**

|  |  |
| --- | --- |
| 标签 | 描述 |
| <ol> | 定义有序列表 |
| <ul> | 定义无序列表 |
| <li> | 定义列表项 |
| <dl> | 定义列表 |
| <dt> | 自定义列表项目 |
| <dd> | 定义自定列表项的描述 |

**HTML<div>和<span>**

HTML通过<div>和<span>将 元素组合起来

**HTML区块元素**

大多HTML元素被定义为会计元素或内联元素

块级元素在浏览器先试试，通常会以新行来开始（和结束）

实例：<h1>,<p>,<ul>,<table>

**HTML内联元素**

在显示时通常不会以新行开始

实例：<b>,<td>,<a>,<img>

**HTML<div>元素**

是块级元素，可用于组合替他HTML元素的容器

没有特定的含有，浏览器会在其前后显示折行

如果和CSS一起用，可用于对大多内容块设置样式属性

另一个用途是文档布局，使用<table>元素进行文档布局不是表格的正确用法。<table>元素的作用是显示表格化的数据。

**HTML<span>元素**

是内联元素，可用作文本的容器

<span>没有特定的含义

当和CSS一起用时，<span>元素可用于为部分文本设置样式属性

**HTML分组标签**

|  |  |
| --- | --- |
| 标签 | 描述 |
| <div> | 定义了文档的区域，块级 |
| <span> | 用来组合文档中的内行元素，内联元素 |

**HTML布局**

网站布局

可以使用<div>或者<table>元素来创建多列。CSS用于对元素进行定位，或者为页面创建背景以及色彩丰富的外观。

**使用<div>元素**

用于分组HTML元素

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<div id="container" style="width:500px">

​

<div id="header" style="background-color:#FFA500;">

<h1 style="margin-bottom:0;">主要的网页标题</h1></div>

​

<div id="menu" style="background-color:#FFD700;height:200px;width:100px;float:left;">

<b>菜单</b><br>

HTML<br>

CSS<br>

JavaScript</div>

​

<div id="content" style="background-color:#EEEEEE;height:200px;width:400px;float:left;">

内容在这里</div>

​

<div id="footer" style="background-color:#FFA500;clear:both;text-align:center;">

版权 © runoob.com</div>

​

</div>

</body>

</html>

**使用表格**

使用<table>标签是创建布局的一种简单的方式

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<table width="500" border="0">

<tr>

<td colspan="2" style="background-color:#FFA500;">

<h1>主要的网页标题</h1>

</td>

</tr>

​

<tr>

<td style="background-color:#FFD700;width:100px;vertical-align:top;">

<b>菜单</b><br>

HTML<br>

CSS<br>

JavaScript

</td>

<td style="background-color:#eeeeee;height:200px;width:400px;vertical-align:top;">

内容在这里</td>

</tr>

​

<tr>

<td colspan="2" style="background-color:#FFA500;text-align:center;">

版权 © runoob.com</td>

</tr>

</table>

​

</body>

</html>

**HTML布局标签**

|  |  |
| --- | --- |
| 标签 | 描述 |
| <div> | 定义文档区块，块级 |
| <span> | 定义span,组合文档中的行内元素 |

**HTML表单和输入**

用于收集不同类型的用户输入

**HTML表单**

包含表单元素的区域

表单元素允许用户在表单中输入内容，比如：文本域（textarea）,下拉列表，单选框（radio-buttons）.复选框(checkboxes)等等

表单使用表单标签<form>来设置

<form>  
.  
*input 元素*  
.  
</form>

**HTML表单-输入元素**

大多数被送到的表单标签是输入标签（<input>）

输入类型是由类型属性（type）定义的，经常用到的输入类型如下：

* 文本域（Text Fields）

通过<input type=”text”>标签来设定，当用户要在表单中键入字母，数字等内容时，就会用到

<form>  
First name: <input type="text" name="firstname"><br>  
Last name: <input type="text" name="lastname">  
</form>

注意：表单本身并不可见，大多数浏览器中，文本域默认宽度是20个字符

* 密码字段

通过标签<input type=”password”>来定义

<form>  
Password: <input type="password" name="pwd">  
</form>

注意：密码字段字符不会明文显示，而是以星号或圆点代替

* 单选按钮（Radio Buttons）

<input type=”radio”>标签定义了表单单选选框选项

<form>  
<input type="radio" name="sex" value="male">Male<br>  
<input type="radio" name="sex" value="female">Female  
</form>

* 复选框（Checkboxes）

<input type=”checkbox”>定义了复选框，需要从若干给定的选择中选取一个或若干选项

<form>  
<input type="checkbox" name="vehicle" value="Bike">I have a bike<br>  
<input type="checkbox" name="vehicle" value="Car">I have a car   
</form>

* 提交按钮（Submit Button）

<input type=”submit”>定义了提交按钮

当用户单击确认按钮时，表单中的内容会被传送到另一个文件。表单的动作属性定义了目的文件的文件名。由动作属性定义的文件或接收到输入数据进行相关的处理

<form name="input" action="html\_form\_action.php" method="get">  
Username: <input type="text" name="user">  
<input type="submit" value="Submit">  
</form>

**HTML表单标签**

|  |  |
| --- | --- |
| 标签 | 描述 |
| <form> | 定义供用户输入的表单 |
| <input> | 定义输入域 |
| <textarea> | 定义文本域（一个多行的输入控件） |
| <label> | 定义<input>元素的标签，一般为输入标题 |
| <fieldset> | 定义了一组相关的表单元素，并使用外框包含起来 |
| <legend> | 定义<fieldset>元素的标题 |
| <select> | 定义了下拉选项列表 |
| <optgroup> | 定义选项组 |
| <option> | 定义下拉列表中的选项 |
| <button> | 定义一个点击按钮 |
| <datalist> | 指定一个预先定义的输入控件选项列表 |
| <keygen> | 定义了表单的秘钥对生成器字段 |
| <output> | 定义一个计算结果 |

**HTML框架**

通过使用框架，可以在同一个浏览器窗口中显示不止一个页面

**iframe语法：**

<iframe src="URL"></iframe>

该URL指向不同的网页

**Iframe-设置高度和宽度**

height和width属性用来定义iframe标签的高度和宽度

属性默认为像素单位，可以指定其按比例显示

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<iframe src="demo\_iframe.htm" width="200" height="200"></iframe>

​

<p>一些旧的浏览器不支持 iframe。</p>

<p>如果浏览器不支持 iframes 则不会显示。</p>

​

​

</body>

</html>

**Iframe-移除边框**

frameborder属性用于定义iframe表示是否显示边框

设置输出值为“0”移除iframe的边框

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<iframe src="demo\_iframe.htm" width="200" height="200" frameborder="0">

<p>您的浏览器不支持 iframe 标签。</p>

</iframe>

​

</body>

</html>

使用iframe来显示目标链接页面

iframe可以显示目标链接的页面

目标链接的属性必须使用iframe属性

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<iframe src="demo\_iframe.htm" name="iframe\_a"></iframe>

<p><a href="//www.runoob.com" target="iframe\_a">RUNOOB.COM</a></p>

​

<p><b>注意：</b> 因为 a 标签的 target 属性是名为 iframe\_a 的 iframe 框架，所以在点击链接时页面会显示在 iframe框架中。</p>

​

</body>

</html>

**HTML iframe标签**

|  |  |
| --- | --- |
| 标签 | 说明 |
| <iframe> | 定义一个内联的iframe |

**HTML颜色**

HTML颜色由红色，绿色和蓝色混合而成

**颜色值**

由一个十六进制符号来定义，符号由红色，绿色和蓝色的值组成（RGB）

每种颜色的最小值是0（十六进制：#00），最大值是255（十六进制：#FF）

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<p style="background-color:#FFFF00">

通过十六进制设置背景颜色

</p>

​

<p style="background-color:rgb(255,255,0)">

通过 rbg 值设置背景颜色

</p>

​

<p style="background-color:yellow">

通过颜色名设置背景颜色

</p>

​

</body>

</html>

**HTML颜色名**

目前所有浏览器都支持以下颜色名

141个颜色名称是在HTML和CSS颜色规范定义的（17标准颜色，再加124）.

提示：17标准颜色：黑色，蓝色，水，紫红色，灰色，绿色，石灰，栗色，海军，橄榄，橙，紫，红，白，银，蓝绿色，黄色。

**HTML颜色值**

由十六进制来表示颜色值

每个颜色的最低值为0（十六进制为00），最高值为255（十六进制为FF）

十六进制的写法为#号后跟三个或者六个十六进制字符

三位数表示法为：#RGB，转换为6位数表示为：#RRGGBB

**HTML脚本**

JavaScript使HTML页面具有更强的动态和交互性

**HTML<script>标签**

<script>标签用于定义客户端脚本

<script>元素既可包含脚本语句，也可通过src属性指向外部脚本文件

JavaScript最常用于图片操作，表单验证以及内容动态更新

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<script>

document.write("Hello World!")

</script>

​

</body>

</html>

**HTML<noscript>标签**

<noscript>标签提供无法使用脚本时的替代内容，比如在浏览器禁用脚本时，或浏览器不支持客户端脚本时。

<noscript>元素可包含普通的HTML页面的body元素中能够找到的所有元素

只有在浏览器不支持脚本或者禁用脚本时，才会显示<noscript>元素中的内容

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<script>

document.write("Hello World!")

</script>

<noscript>抱歉，你的浏览器不支持 JavaScript!</noscript>

​

<p>不支持 JavaScript 的浏览器会使用 &lt;noscript&gt; 元素中定义的内容（文本）来替代。</p>

</body>

</html>

**JavaScript体验**

JavaScript可以直接在HTML输出：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

<p>

JavaScript 能够直接写入 HTML 输出流中：

</p>

<script>

document.write("<h1>这是一个标题</h1>");

document.write("<p>这是一个段落。</p>");

</script>

<p>

您只能在 HTML 输出流中使用 <strong>document.write</strong>。

如果您在文档已加载后使用它（比如在函数中），会覆盖整个文档。

</p>

</body>

</html>

**JavaScript事件响应：**

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<h1>我的第一个 JavaScript </h1>

​

<p id="demo">

JavaScript 可以触发事件，就像按钮点击。</p>

​

<script>

function myFunction()

{

document.getElementById("demo").innerHTML="Hello JavaScript!";

}

</script>

​

<button type="button" onclick="myFunction()">点我</button>

​

</body>

</html>

窗体顶端

**JavaScript处理HTML样式：**

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

<h1>我的第一段 JavaScript</h1>

<p id="demo">

JavaScript 能改变 HTML 元素的样式。

</p>

<script>

function myFunction()

{

x=document.getElementById("demo") // 找到元素

x.style.color="#ff0000"; // 改变样式

}

</script>

<button type="button" onclick="myFunction()">点击这里</button>

</body>

</html>

**HTML脚本标签**

|  |  |
| --- | --- |
| 标签 | 描述 |
| <script> | 定义了客户端脚本 |
| <noscript> | 定义了不支持脚本浏览器输出的文本 |

**HTML字符实体**

HTML中的预留字符必须被替换为字符实体

一些在键盘找不到的字符也可以使用字符实体来替换

**HTML实体**

在HTML中，某些字符是预留的

在HTML中不能使用小于号（<）和大于号（>）,因为浏览器会误认为他们是标签

如果希望正确的显示预留字符，必须在HTML源代码中使用字符实体。比如：

&*entity\_name*;

或

&#*entity\_number*;

如需显示小于号，必须这样写：**&lt;** 或**&#60;** 或 **&#060;**

**提示：**使用实体名而不是数字的好处是便于记忆，但并不是浏览器支持所有试题名称

**不间断空格**

HTML中的常用字符实体是不间断空格

浏览器总会截断HTML页面中的空格，如果在文本中写10个空格，在显示该页面前，浏览器会删除它们中的9个，如需在页面中增加空格的数量，需要使用&nbsp;字符实体

**结合音标符**

发音符号是加到字母上的一个“glyph”

变音符号可以出现在字母的上面和下面，或者字母里面或者两个子母件

变音符号可以与字母，数字符号的组合来使用

**HTML字符实体**

实体名称对大小写敏感

|  |  |  |  |
| --- | --- | --- | --- |
| 显示结果 | 描述 | 实体名称 | 实体编号 |
|  | 空格 | &nbsp; | &#160; |
| < | 小于号 | &lt; | &#60; |
| > | 大于号 | &gt; | &#62; |
| & | 和号 | &amp; | &#38; |
| “ | 引号 | &quot; | &#34 |
| ‘ | 撇号 | &apos; | &#39 |
| ￠ | 分 | &cent; | &#162; |
| £ | 镑 | &pound; | &#163; |
| ¥ | 人命币/日元 | &yen; | &#165; |
| € | 欧元 | &euro; | &#8364; |
| § | 小节 | &sect; | &#167; |
| © | 版权 | &copy; | &#169; |
| ® | 注册商标 | &reg; | &#174; |
| ™ | 商标 | &trade | &#8482; |
| × | 乘号 | &times | &#215; |
| ÷ | 除号 | &divide; | &#247; |

虽然html不区分大小写，但实体字符对大小写敏感

**HTML统一资源定位器**

URL是一个网页地址

可以由字母组成，或互联网协议地址

**URL-统一资源定位器**

当点击HTML页面中的某个链接时，对应的<a>标签指向万维网的一个地址

说明：

* Scheme:定义因特网服务的类型，最常见的是http
* Host:定义域主机
* Domain:定义因特网域名
* :port:定义主机上的端口号
* Path:定义服务器上的路径
* Filename:定义文档/资源名称

**常见的URL Scheme**

|  |  |  |
| --- | --- | --- |
| Scheme | 访问 | 用于... |
| http | 超文本传输协议 | 以http://开头的普通网页，不加密 |
| https | 安全超文本传输协议 | 安全网页，加密所有信息交换 |
| ftp | 文件传输协议 | 用于将文件下载或上传至网站 |
| file |  | 自己计算机上的文件 |

**URL字符编码**

只能使用ASCII字符集

URL编码使用“%”其后跟随两位的十六进制来替换非ASCII字符

URL不能包含空格，通话擦红使用+来替换空格

**HTML速查列表**

HTML基本文档

<!DOCTYPE html>

<html>

<head>

<title>文档标题</title>

</head>

<body> 可见文本...

</body>

</html>

**基本标签**

<h1>最大的标题</h1>

<h2> . . . </h2>

<h3> . . . </h3>

<h4> . . . </h4>

<h5> . . . </h5>

<h6>最小的标题</h6>

<p>这是一个段落。</p>

<br> （换行）

<hr> （水平线）

<!-- 这是注释 -->

**文本格式化**

<b>粗体文本</b>

<code>计算机代码</code>

<em>强调文本</em>

<i>斜体文本</i>

<kbd>键盘输入</kbd>

<pre>预格式化文本</pre>

<small>更小的文本</small>

<strong>重要的文本</strong>

<abbr> （缩写）

<address> （联系信息）

<bdo> （文字方向）

<blockquote> （从另一个源引用的部分）

<cite> （工作的名称）

<del> （删除的文本）

<ins> （插入的文本）

<sub> （下标文本）

<sup> （上标文本）

**链接**

普通的链接：<a href="http://www.example.com/">链接文本</a>

图像链接： <a href="http://www.example.com/"><img src="URL" alt="替换文本"></a>

邮件链接： <a href="mailto:webmaster@example.com">发送e-mail</a>

书签：

<a id="tips">提示部分</a>

<a href="#tips">跳到提示部分</a>

图片

<img src="URL" alt="替换文本" height="42" width="42">

**样式/区块**

<style type="text/css">

h1 {color:red;}

p {color:blue;}

</style>

<div>文档中的块级元素</div>

<span>文档中的内联元素</span>

**无序列表**

<ul>

<li>项目</li>

<li>项目</li>

</ul>

**有序列表**

<ol>

<li>第一项</li>

<li>第二项</li>

</ol>

**定义列表**

<dl>

<dt>项目 1</dt>

<dd>描述项目 1</dd>

<dt>项目 2</dt>

<dd>描述项目 2</dd>

</dl>

**表格**

<table border="1">

<tr>

<th>表格标题</th>

<th>表格标题</th>

</tr>

<tr>

<td>表格数据</td>

<td>表格数据</td>

</tr>

</table>

**框架**

<iframe src="demo\_iframe.htm"></iframe>

**表单**

<form action="demo\_form.php" method="post/get">

<input type="text" name="email" size="40" maxlength="50">

<input type="password">

<input type="checkbox" checked="checked">

<input type="radio" checked="checked">

<input type="submit" value="Send">

<input type="reset">

<input type="hidden">

<select>

<option>苹果</option>

<option selected="selected">香蕉</option>

<option>樱桃</option>

</select>

<textarea name="comment" rows="60" cols="20"></textarea>

</form>

**实体**

&lt; 等同于 <

&gt; 等同于 >

&#169; 等同于 ©

**HTML-XHTML**

以XML格式编写的HTML

**什么是XHTML**

* XHTML指的是可扩展超文本标记语言
* XHTML与HTML4.01几乎相同
* XHTML是更严格更纯洁的HTML版本
* XHTML是以XML应用的方式定义的HTML

XML是一种必须整齐标记且格式良好的标记语言

**与HTML的区别：**

* 文档结果
* XHTMLDOCTYPE是强制性的
* <html>中的XML namespace属性也是强制性
* <html>,<head>,<title>以及<body>也是强制性的
* 元素语法
* XHTML元素必须正确嵌套
* XHTML元素必须始终关闭
* XHTML元素必须小写
* XHTML文档必须有一个根元素
* 属性语法
* XHTML属性必须使用小写
* XHTML属性值必须用引号包围
* XHTML属性最小化也是禁止的

<!DOCTYPE...>是强制性的

XHTML文档必须进行XHTML文档类型声明

<html>,<head>,<title>,<body>元素必须存在，必须使用<html>中的xmlns属性为文档规定xml命名空间

实例：

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

<meta charset="utf-8">

<title>文档标题</title>

</head>

<body>

文档内容

</body>

</html>

**XHTML元素必须合理嵌套**

在HTML中，一些元素可以不互相嵌套，like:

<b><i>粗体和斜体文本</b></i>

在XHTML中，所有元素必须互相合理的嵌套，like:

<b><i>粗体和斜体文本</i></b>

**XHTML元素必须有关闭标签**

错误实例：

<p>这是一个段落

<p>这是另外一个段落

正确实例：

<p>这是一个段落</p>

<p>这是另外一个段落</p>

**空元素必须包含关闭标签**

错误实例：

分行:<br>

水平线: <hr>

图片: <img src="happy.gif" alt="Happy face">

正确实例：

分行:<br />

水平线: <hr />

图片: <img src="happy.gif" alt="Happy face" />

**XHTML元素必须是小写**

错误实例：

<BODY>

<P>这是一个段落</P>

</BODY>

正确实例：

<body>

<p>这是一个段落</p>

</body>

**属性名称必须是小写**

错误实例：

<table WIDTH="100%">

正确实例：

<table width="100%">

**属性值必须有引号**

错误实例：

<table width=100%>

正确实例：

<table width="100%">

**不允许属性简写**

错误实例：

<input checked>

<input readonly>

<input disabled>

<option selected>

正确实例：

<input checked="checked">

<input readonly="readonly">

<input disabled="disabled">

<option selected="selected">

**如何将HTML转换为XHTML**

1. 添加一个XHTML<!DOCTYPE>到网页中
2. 添加xmlns属性添加到每个页面的html元素中
3. 改变所有的元素为小写
4. 关闭所有的空元素
5. 修改所有的属性名称为小写
6. 所有属性值添加引号

**HTML5学习**

**新特性：**

* 用于绘画的canvas元素
* 用于媒介回放的video和audio元素
* 对本地离线存储的更好地支持
* 新的特殊内容元素，比如article,footer,header,nav,section
* 新的表单控件，比如calendar,date,time,email,url,search

**HTML<!DOCTYPE>**

<!doctype>声明必须位于HTML5文档中的第一行：

<!DOCTYPE html>

最小的HTML5文档：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>文档标题</title>

</head>

<body>

文档内容......

</body>

</html>

注意：对于中文网页需要使用<meta charset=”utf-8”>声明编码，否则会出现乱码

**HTML5的改进**

* 新元素
* 新属性
* 完全支持CSS3
* Video和Audio
* 2D/3D制图
* 本地存储
* 本地SQL数据
* Web应用

**HTML5多媒体**

使用HTML5可以简单地在网页中播放视频和音频

* HTML5<video>
* HTML5<audio>

**HTML5应用**

* 本地数据存储
* 访问本地文件
* 本地SQL数据
* 缓存引用
* JavaScript工作者
* XHTMLHttpRequest 2

**HTML5图形**

* 使用<canvas>元素
* 使用内联SVG
* 使用CSS3 2D转换，CSS3 3D转换

**HTML5使用CSS3**

* 新选择器
* 新属性
* 动画
* 2D/3D转换
* 圆角
* 阴影效果
* 可下载的字体

**语义元素**

**HTML5添加了很多语义元素如下：**

|  |  |
| --- | --- |
| 标签 | 描述 |
| <article> | 定义页面独立的内容区域 |
| <aside> | 定义页面的侧边栏内容 |
| <bdi> | 允许设置一段文本，使其脱离其父元素的文本方向设置 |
| <command> | 定义命令按钮，比如安选按钮，复选框或按钮 |
| <details> | 用于描述文档或文档某个部分的细节 |
| <dialog> | 定义对话框，比如提示框 |
| <summary> | 标签包含details元素的标题 |
| <figure> | 规定独立的流内容（图像，图表，照片，代码等等） |
| <figcaption> | 定义<figure>元素的标题 |
| <footer> | 定义section或document的页脚 |
| <header> | 定义了文档的头部区域 |
| <mark> | 定义带有记号的文本 |
| <meter> | 定义度量衡。仅用于已知对打和最小值的度量 |
| <nav> | 定义导航链接的部分 |
| <progress> | 定义任何类型的任务的进度 |
| <ruby> | 定义ruby注释（中文注音或字符） |
| <rt> | 定义字符（中文注音或字符）的解释或发音 |
| <rp> | 在ruby注释中使用，定义不支持ruby元素的浏览器所显示的内容 |
| <session> | 定义文档中的节（section,区段） |
| <time> | 定义日期或时间 |
| <wrb> | 规定在文本中的何处适合添加换行符 |

**已移除元素**

* <acronym>
* <applet>
* <basefont>
* <big>
* <center>
* <dir>
* <font>
* <frame>
* <frameset>
* <noframes>
* <strike>

**HTML5 Canvas**

<canvas>标签定义图形，比如图表和其他图像，必须使用脚本来绘制图形

**创建一个画布**

一个画布在网页中是一个矩形框，通过<canvas>元素来绘制

**注意：**默认情况下<canvas>元素没有边框和内容

实例：

<canvas id="myCanvas" width="200" height="100"></canvas>

注意：标签通常需要指定一个id属性，width和height属性定义的画布的大小

提示：可以在HTML页面中使用多个<canvas>元素

使用style属性来添加边框

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<canvas id="myCanvas" width="200" height="100" style="border:1px solid #000000;">

您的浏览器不支持 HTML5 canvas 标签。

</canvas>

​

</body>

</html>

**使用JavaScript来绘制图像**

canvas元素本身是没有绘图能力的。所有的绘制工作必须在JavaScript内部完成

实例：

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<canvas id="myCanvas" width="200" height="100" style="border:1px solid #c3c3c3;">

您的浏览器不支持 HTML5 canvas 标签。

</canvas>

​

<script>

​

var c=document.getElementById("myCanvas");

var ctx=c.getContext("2d");

ctx.fillStyle="#FF0000";

ctx.fillRect(0,0,150,75);

​

</script>

​

</body>

</html>

窗体顶端

实例解析：

首先找到<canvas>元素

var c=document.getElementById("myCanvas");

然后，常见context对象

var ctx=c.getContext("2d");

**Canvas - 路径**

在Canvas上画线，使用以下两种方法

* moveTo(x,y)定义线条开始坐标
* lineTo（x,y）定义线条结束坐标

绘制线条必须使用“ink”方法，就像stroke()

实例：

定义开始坐标（0，0）和结束坐标（200,100）,然后使用stroke()方法来绘制线条

JavaSript:

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<canvas id="myCanvas" width="200" height="100" style="border:1px solid #d3d3d3;">

您的浏览器不支持 HTML5 canvas 标签。</canvas>

​

<script>

​

var c=document.getElementById("myCanvas");

var ctx=c.getContext("2d");

ctx.moveTo(0,0);

ctx.lineTo(200,100);

ctx.stroke();

​

</script>

​

</body>

</html>

在canvas中绘制圆形，使用以下方法：

arc(x,y,r,start,stop)

实例：使用arc()方法绘制一个圆：

JavaScript:

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<canvas id="myCanvas" width="200" height="100" style="border:1px solid #d3d3d3;">

您的浏览器不支持 HTML5 canvas 标签。</canvas>

​

<script>

​

var c=document.getElementById("myCanvas");

var ctx=c.getContext("2d");

ctx.beginPath();

ctx.arc(95,50,40,0,2\*Math.PI);

ctx.stroke();

​

</script>

​

</body>

</html>

**Canvas - 文本**

使用canvas绘制文本，重要属性和方法：

* Font - 定义字体
* fillText(text,x,y） - 在canvas上绘制实心的文本
* strokeText(text,x,y) - 在canvas上绘制空心的文本

使用fillText():

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<canvas id="myCanvas" width="200" height="100" style="border:1px solid #d3d3d3;">

您的浏览器不支持 HTML5 canvas 标签。</canvas>

​

<script>

​

var c=document.getElementById("myCanvas");

var ctx=c.getContext("2d");

ctx.font="30px Arial";

ctx.fillText("Hello World",10,50);

​

</script>

​

</body>

</html>

使用stroke Text():

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<canvas id="myCanvas" width="200" height="100" style="border:1px solid #d3d3d3;">

您的浏览器不支持 HTML5 canvas 标签。</canvas>

​

<script>

​

var c=document.getElementById("myCanvas");

var ctx=c.getContext("2d");

ctx.font="30px Arial";

ctx.strokeText("Hello World",10,50);

​

</script>

​

</body>

</html>

**Canvas - 渐变**

两种方式设置：

* createLinearGradient(x,y,x1,y1) - 创建线条渐变
* createRadiaGradient(x,y,r,x1,y1,r1) - 创建一个径向/圆渐变

addColorStop()方法指定颜色停止，参数使用坐标描述，可以是0到1

设置fillStyle或strokeStyle的值为渐变，然后绘制形状

使用createLinearGradient():

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<canvas id="myCanvas" width="200" height="100" style="border:1px solid #d3d3d3;">

您的浏览器不支持 HTML5 canvas 标签。</canvas>

​

<script>

​

var c=document.getElementById("myCanvas");

var ctx=c.getContext("2d");

​

// Create gradient

var grd=ctx.createLinearGradient(0,0,200,0);

grd.addColorStop(0,"red");

grd.addColorStop(1,"white");

​

// Fill with gradient

ctx.fillStyle=grd;

ctx.fillRect(10,10,150,80);

​

</script>

​

</body>

</html>

使用createRadialGradient():

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<canvas id="myCanvas" width="200" height="100" style="border:1px solid #d3d3d3;">

您的浏览器不支持 HTML5 canvas 标签。</canvas>

​

<script>

​

var c=document.getElementById("myCanvas");

var ctx=c.getContext("2d");

​

// Create gradient

var grd=ctx.createRadialGradient(75,50,5,90,60,100);

grd.addColorStop(0,"red");

grd.addColorStop(1,"white");

​

// Fill with gradient

ctx.fillStyle=grd;

ctx.fillRect(10,10,150,80);

​

</script>

​

</body>

</html>

**Canvas - 图像**

把一幅图像放置到画布上，使用以下方法：

* drawImage(image,x,y)

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<p>Image to use:</p>

<img id="scream" src="img\_the\_scream.jpg" alt="The Scream" width="220" height="277"><p>Canvas:</p>

<canvas id="myCanvas" width="250" height="300" style="border:1px solid #d3d3d3;">

您的浏览器不支持 HTML5 canvas 标签。</canvas>

​

<script>

​

var c=document.getElementById("myCanvas");

var ctx=c.getContext("2d");

var img=document.getElementById("scream");

​

img.onload = function() {

ctx.drawImage(img,10,10);

}

</script>

​

</body>

</html>

**HTML<canvas>标签**

|  |  |
| --- | --- |
| TAG | 描述 |
| <canvas> | HTML5的canvas元素使用JavaScript在网页上绘制图像 |

**HTML5内联SVG**

**什么是SVG**

* 指可伸缩矢量图形
* 用于定义用于网络的基于矢量的图形
* 使用XML格式定义图形
* 图像在放大或改变尺寸的情况下其图形质量不会有损失
* 万维网联盟的标准

**将SVG直接嵌入HTML页面**

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<svg xmlns="http://www.w3.org/2000/svg" version="1.1" height="190">

<polygon points="100,10 40,180 190,60 10,60 160,180"

style="fill:lime;stroke:purple;stroke-width:5;fill-rule:evenodd;"/>

</svg>

</body>

</html>

**Canvas与SVG的比较**

|  |  |
| --- | --- |
| Canvas | SVG |
| * 依赖分辨率 * 不支持事件处理器 * 弱的文本渲染能力 * 能够以.png或.jpg格式保存结果图像 * 最适合图像密集型的游戏，其中许多对象会被频繁重绘 | * 不依赖分辨率 * 支持事件处理器 * 最适合带有大型渲染区域的应用程序 * 复杂度高会减慢渲染速度 * 不适合游戏应用 |

**HTML5 MathML**

可以在文档中使用MathML元素，对应的标签是<math>...</math>

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

<math xmlns="http://www.w3.org/1998/Math/MathML">

<mrow>

<msup><mi>a</mi><mn>2</mn></msup>

<mo>+</mo>

<msup><mi>b</mi><mn>2</mn></msup>

<mo>=</mo>

<msup><mi>c</mi><mn>2</mn></msup>

</mrow>

</math>

</body>

</html>

​

以下实例添加了一些运算符：

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

<math xmlns="http://www.w3.org/1998/Math/MathML">

<mrow>

<mrow>

<msup>

<mi>x</mi>

<mn>2</mn>

</msup>

<mo>+</mo>

<mrow>

<mn>4</mn>

<mo>⁢</mo>

<mi>x</mi>

</mrow>

<mo>+</mo>

<mn>4</mn>

</mrow>

<mo>=</mo>

<mn>0</mn>

</mrow>

</math>

</body>

</html>

​

以下实例是一个2\*2矩阵：

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

<math xmlns="http://www.w3.org/1998/Math/MathML">

<mrow>

<mi>A</mi>

<mo>=</mo>

<mfenced open="[" close="]">

<mtable>

<mtr>

<mtd><mi>x</mi></mtd>

<mtd><mi>y</mi></mtd>

</mtr>

<mtr>

<mtd><mi>z</mi></mtd>

<mtd><mi>w</mi></mtd>

</mtr>

</mtable>

</mfenced>

</mrow>

</math>

</body>

</html>

​

窗体顶端

**HTML新的Input类型**

* Color
* Date
* Datetime
* Datetime-local
* Email
* Month
* Number
* Range
* Search
* Tel
* Time
* Url
* Week

**注意：**不是所有浏览器都支持新的input类型

* **Color:**主要用于选取颜色

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<form action="demo-form.php">

选择你喜欢的颜色: <input type="color" name="favcolor"><br>

<input type="submit">

</form>

​

</body>

</html>

* **Date：**允许从一个日期选择器选择一个日期

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<form action="demo-form.php">

生日: <input type="date" name="bday">

<input type="submit">

</form>

​

</body>

</html>

* **Datetime：**允许选择一个日期

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<form action="demo-form.php">

生日 (日期和时间): <input type="datetime" name="bdaytime">

<input type="submit">

</form>

​

</body>

</html>

* **Datetime-local：**允许选择一个日期和时间（无时区）

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<form action="demo-form.php">

生日 (日期和时间): <input type="datetime-local" name="bdaytime">

<input type="submit">

</form>

​

</body>

</html>

* **Email：**用于应该包含e-mail地址的输入域

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<form action="demo-form.php">

E-mail: <input type="email" name="usremail">

<input type="submit">

</form>

​

<p><b>注意:</b> Internet Explorer 9 及更早 IE 版本不支持 type="email" 。</p>

​

</body>

</html>

* **Month：**允许选择一个月份

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<form action="demo-form.php">

生日 ( 月和年 ): <input type="month" name="bdaymonth">

<input type="submit">

</form>

​

</body>

</html>

* **Number：**用于应该包含数值的输入域，还能设定所接受数字的限定

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<form action="demo-form.php">

数量 ( 1 到 5 之间): <input type="number" name="quantity" min="1" max="5">

<input type="submit">

</form>

​

<p><b>注意:</b>Internet Explorer 9 及更早 IE 版本不支持 type="number" 。</p>

​

</body>

</html>

**使用下面的属性来规定对数字类型的限定：**

|  |  |
| --- | --- |
| 属性 | 描述 |
| disabled | 规定输入字段是禁用的 |
| max | 规定允许的最大值 |
| maxlength | 规定输入字段的最大字符长度 |
| min | 规定允许的最小值 |
| pattern | 规定用于验证输入字段的模式 |
| readonly | 规定输入字段的值无法修改 |
| required | 规定输入字段的值是必须的 |
| size | 规定输入字段中的可见字符数 |
| step | 规定字段的合法数字间隔 |
| value | 规定输入字字段的默认值 |

* **Range:**用于应该包含一定范围内数字值的输入域，显示为滑动条

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<form action="demo-form.php" method="get">

Points: <input type="range" name="points" min="1" max="10">

<input type="submit">

</form>

​

<p><b>注意:</b> Internet Explorer 9 及更早 IE 版本不支持 type="range"。</p>

​

</body>

</html>

**可以使用下面的属性来规定对数字类型的限定：**

* Max:规定允许的最大值
* Min：规定允许的最小值
* Step：规定合法的数字间隔
* Value：规定默认值
* **Search：**用于搜索域

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<form action="demo-form.php">

Search Google: <input type="search" name="googlesearch"><br>

<input type="submit">

</form>

​

</body>

</html>

* **Tel:**定义输入电话号码字段

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<form action="demo-form.php">

电话号码: <input type="tel" name="usrtel"><br>

<input type="submit">

</form>

​

</body>

</html>

* **Time:**允许选择一个时间

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<form action="demo-form.php">

选择时间: <input type="time" name="usr\_time">

<input type="submit">

</form>

​

</body>

</html>

* **Url：**用于应该包含URL地址的输入域，在提交表单时，自动验证url的值

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<form action="demo-form.php">

添加你的主页: <input type="url" name="homepage"><br>

<input type="submit">

</form>

​

<p><b>注意:</b> Internet Explorer 9及更早 IE 版本不支持 type="url" 。</p>

​

</body>

</html>

* **Week：**允许选择周和年

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

​

<body>

​

<form action="demo-form.php">

选择周: <input type="week" name="year\_week">

<input type="submit">

</form>

​

</body>

</html>

**HTML5<input>标签**

|  |  |
| --- | --- |
| 标签 | 描述 |
| <input> | 描述input输入器 |

**HTML5表单元素**

由以下新的表单元素:

* <datalist>
* <keygen>
* <output>
* **<datalist>元素：**规定输入域的选项列表

使用<input>元素的列表属性与<datalist>元素绑定：

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

<form action="demo-form.php" method="get">

<input list="browsers" name="browser">

<datalist id="browsers">

<option value="Internet Explorer">

<option value="Firefox">

<option value="Chrome">

<option value="Opera">

<option value="Safari">

</datalist>

<input type="submit">

</form>

​

<p><strong>注意:</strong> Internet Explorer 9（更早 IE 版本），Safari 不支持 datalist 标签。</p>

​

</body>

</html>

* **<keygen>元素：**提供一种验证用户的可靠方法。用于规定用于表单的秘钥对生成器字段。当提交表单时，会生成两个键，一个是私钥，一个公钥。私钥存储于客户端，公钥被发送到服务器。

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<form action="demo\_keygen.php" method="get">

用户名: <input type="text" name="usr\_name">

加密: <keygen name="security">

<input type="submit">

</form>

​

<p><strong>注意:</strong> Internet Explorer 不支持 keygen 标签。</p>

​

</body>

</html>

* **<output>元素：**用于不同类型的输出

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<form oninput="x.value=parseInt(a.value)+parseInt(b.value)">0

<input type="range" id="a" value="50">100

+<input type="number" id="b" value="50">

=<output name="x" for="a b"></output>

</form>

​

<p><strong>注意:</strong> Internet Explorer 不支持 output 标签。</p>

​

</body>

</html>

**HTML5新表单元素**

|  |  |
| --- | --- |
| 标签 | 描述 |
| <datalist> | <input>标签定义选项标签配合input元素使用，来定义input可能的值 |
| <keygen> | 用于规定用于表单的密钥对生成器字段 |
| <output> | 定义不同类型的输出 |

**HTML5新的表单属性**

**<form>新属性：**

* Autocomplete
* Novalidate

**<input>新属性：**

* Autocomplete
* Autofocus
* Form
* Formaction
* Formenctype
* Formmethod
* Formnovalidate
* Formtarget
* height与width
* List
* min与max
* Multiple
* Pattern
* Placeholder
* Step
* **<form>/<input> autocomplete属性：**应该用于自动完成功能

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<form action="demo-form.php" 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>

​

<p>填写并提交表单，然后重新刷新页面查看如何自动填充内容。</p>

<p>注意 form 的 autocomplete 属性为 "on"（开），但是 e-mail 自动为“off”（关）。</p>

​

</body>

</html>

* **<form> novalidate属性：**规定在提交表单时不应该验证form或input域

实例:

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<form action="demo-form.php" novalidate>

E-mail: <input type="email" name="user\_email">

<input type="submit">

</form>

​

<p><strong>注意：</strong>在 Safari 和 Internet Explorer 9 及之前的版本中不支持 novalidate 属性。</p>

​

</body>

</html>

* **<input> autofocus属性：**规定在页面加载时，域自动获得焦点

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<form action="demo-form.php">

First name: <input type="text" name="fname" autofocus><br>

Last name: <input type="text" name="lname"><br>

<input type="submit">

</form>

<p><strong>注意：</strong> Internet Explorer 9及更早 IE 版本不支持 input 标签的 autofocus 属性。</p>

​

</body>

</html>

* **<input> form属性：**规定输入域所属的一个或多个表单

实例:

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<form action="demo-form.php" id="form1">

First name: <input type="text" name="fname"><br>

<input type="submit" value="提交">

</form>

​

<p> "Last name" 字段没有在 form 表单之内，但它也是 form 表单的一部分。</p>

​

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

​

<p><b>注意:</b> IE 不支持 form 属性</p>

​

</body>

</html>

* **<input> formaction属性：**用于描述表单提交的URL地址

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<form action="demo-form.php">

First name: <input type="text" name="fname"><br>

Last name: <input type="text" name="lname"><br>

<input type="submit" value="提交"><br>

<input type="submit" formaction="demo-admin.php" value="提交">

</form>

<p><strong>注意：</strong> Internet Explorer 9及更早 IE 版本不支持 input 标签的 formaction 属性。</p>

​

</body>

</html>

* **<input> formenctype属性：**表单提交到服务器的数据编码

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<form action="demo-post-enctype.php" method="post">

First name: <input type="text" name="fname"><br>

<input type="submit" value="提交">

<input type="submit" formenctype="multipart/form-data" value="以 Multipart/form-data 提交">

</form>

​

<p><strong>注意：</strong> Internet Explorer 9及更早 IE 版本不支持 input 标签的 formenctype 属性。</p>

​

</body>

</html>

* **<input> formmethod属性：**定义了表单提交的方式

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<form action="demo-form.php" method="get">

First name: <input type="text" name="fname"><br>

Last name: <input type="text" name="lname"><br>

<input type="submit" value="提交">

<input type="submit" formmethod="post" formaction="demo-post.php" value="使用 POST 提交">

</form>

​

<p><strong>注意：</strong> Internet Explorer 9及更早 IE 版本不支持 input 标签的 formmethod 属性。</p>

​

</body>

</html>

* **<input> formnovalidate属性：**<input>元素在表单提交时无需被验证

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<form action="demo-form.php">

E-mail: <input type="email" name="userid"><br>

<input type="submit" value="提交"><br>

<input type="submit" formnovalidate="formnovalidate" value="不验证提交">

</form>

​

<p><strong>注意：</strong> Internet Explorer 9及更早 IE 版本,或 Safari 不支持 input 标签的 formnovalidate 属性。</p>

​

</body>

</html>

* **<input> formtarget属性：**指定一个名称或一个关键字来指明表单提交数据接收后的展示

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<form action="demo-form.php">

First name: <input type="text" name="fname"><br>

Last name: <input type="text" name="lname"><br>

<input type="submit" value="正常提交">

<input type="submit" formtarget="\_blank" value="提交到一个新的页面上">

</form>

​

<p><strong>注意：</strong> Internet Explorer 9及更早 IE 版本不支持 input 标签的 formtarget 属性。</p>

​

</body>

</html>

* **<input> height和width属性：**规定用于image类型的<input>标签的图像高度和宽度

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<form action="demo-form.php">

First name: <input type="text" name="fname"><br>

Last name: <input type="text" name="lname"><br>

<input type="image" src="img\_submit.gif" alt="Submit" width="48" height="48">

</form>

​

​

</body>

</html>

* **<input> list属性：**规定输入域的datalist,datalist是输入域的选项列表

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

<form action="demo-form.php" method="get">

<input list="browsers" name="browser">

<datalist id="browsers">

<option value="Internet Explorer">

<option value="Firefox">

<option value="Chrome">

<option value="Opera">

<option value="Safari">

</datalist>

<input type="submit">

</form>

​

<p><strong>注意:</strong> Internet Explorer 9（更早 IE 版本），Safari 不支持 datalist 标签。</p>

​

</body>

</html>

* **<input> min和max属性：**用于为包含数字或日期的input类型规定约束

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<form action="demo-form.php">

​

输入 1980-01-01 之前的日期:

<input type="date" name="bday" max="1979-12-31"><br>

​

输入 2000-01-01 之后的日期:

<input type="date" name="bday" min="2000-01-02"><br>

​

数量 (在1和5之间):

<input type="number" name="quantity" min="1" max="5"><br>

​

<input type="submit">

</form>

​

<p><strong>注意:</strong> Internet Explorer 9及更早 IE 版本，Firefox 不支持 input 标签的 max 和 min 属性。</p>

<p><strong>注意:</strong>

在 Internet Explorer 10中 max 和 min 属性不支持输入日期和时间，IE 10 不支持这些输入类型。</p>

​

</body>

</html>

* **<input> multiple属性：**规定<input>元素可选择多个值

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<form action="demo-form.php">

选择图片: <input type="file" name="img" multiple>

<input type="submit">

</form>

​

<p>尝试选取一张或者多种图片。</p>

​

<p><strong>注意:</strong> Internet Explorer 9及更早 IE 版本不支持 input 标签的 multiple 属性。</p>

​

</body>

</html>

* **<input> pattern属性：**描述了一个正则表达式用于验证<input>元素的值

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<form action="demo-form.php">

Country code: <input type="text" name="country\_code" pattern="[A-Za-z]{3}" title="Three letter country code">

<input type="submit">

</form>

​

<p><strong>注意:</strong> Internet Explorer 9及更早 IE 版本，或 Safari 不支持 input 标签的 pattern 属性。</p>

​

</body>

</html>

* **<input> placeholder属性：**提供一种提示，描述输入域所期待的值

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<form action="demo-form.php">

<input type="text" name="fname" placeholder="First name"><br>

<input type="text" name="lname" placeholder="Last name"><br>

<input type="submit" value="提交">

</form>

​

<p><strong>注意：</strong> Internet Explorer 9及更早 IE 版本不支持 input 标签的 placeholder 属性。</p>

​

</body>

</html>

* **<inpur> required属性：**规定必须在提交之前填写输入域（不能为空）

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<form action="demo-form.php">

Username: <input type="text" name="usrname" required>

<input type="submit">

</form>

​

<p><strong>注意：</strong> Internet Explorer 9及更早 IE 版本，或 Safari 不支持 input 标签的 required 属性。</p>

​

</body>

</html>

* **<input> step属性：**为输入域规定合法的数字间隔

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<form action="demo-form.php">

<input type="number" name="points" step="3">

<input type="submit">

</form>

​

<p><strong>注意：</strong> Internet Explorer 9及更早 IE 版本，或 Firefox 不支持 input 标签的 step 属性。</p>

​

</body>

</html>

**HTML5<input>标签**

|  |  |
| --- | --- |
| 标签 | 描述 |
| <form> | 定义一个form表单 |
| <input> | 定义一个input域 |

**HTML5中新的语义元素**

* <header>
* <nav>
* <section>
* <article>
* <aside>
* <figcaption>
* <figure>
* <footer>
* **<section>元素：**定义文档中的节

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<section>

<h1>WWF</h1>

<p>The World Wide Fund for Nature (WWF) is an international organization working on issues regarding the conservation, research and restoration of the environment, formerly named the World Wildlife Fund. WWF was founded in 1961.</p>

</section>

​

<section>

<h1>WWF's Panda symbol</h1>

<p>The Panda has become the symbol of WWF. The well-known panda logo of WWF originated from a panda named Chi Chi that was transferred from the Beijing Zoo to the London Zoo in the same year of the establishment of WWF.</p>

</section>

​

</body>

</html>

* **<article>元素：**定义独立的内容

使用实例：

* Forum post
* Blog post
* News story
* Comment

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<article>

<h1>Internet Explorer 9</h1>

<p> Windows Internet Explorer 9(缩写为 IE9 )在2011年3月14日21:00 发布。</p>

</article>

​

</body>

</html>

* **<nav>元素：**定义导航链接的部分

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<nav>

<a href="/html/">HTML</a> |

<a href="/css/">CSS</a> |

<a href="/js/">JavaScript</a> |

<a href="/jquery/">jQuery</a>

</nav>

​

</body>

</html>

* **<aside>元素：**定义页面主区域内容之外的内容

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<p>My family and I visited The Epcot center this summer.</p>

​

<aside>

<h4>Epcot Center</h4>

<p>The Epcot Center is a theme park in Disney World, Florida.</p>

</aside>

​

</body>

</html>

* **<header>元素：**描述了文档的头部区域，主要用于定义内容的介绍展示区域

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<article>

<header>

<h1>Internet Explorer 9</h1>

<p><time pubdate datetime="2011-03-15"></time></p>

</header>

<p> Windows Internet Explorer 9(缩写为 IE9 )是在2011年3月14日21:00发布的</p>

</article>

​

</body>

</html>

* **<footer>元素：**描述文档底部区域

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<footer>

<p>Posted by: Hege Refsnes</p>

<p><time pubdate datetime="2012-03-01"></time></p>

</footer>

​

</body>

</html>

* **<figure>和<figcaption>元素：**规定对独立的流内容，应该与主内容相关

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<p>The Pulpit Rock is a massive cliff 604 metres (1982 feet) above Lysefjorden, opposite the Kjerag plateau, in Forsand, Ryfylke, Norway. The top of the cliff is approximately 25 by 25 metres (82 by 82 feet) square and almost flat, and is a famous tourist attraction in Norway.</p>

​

<figure>

<img src="img\_pulpit.jpg" alt="The Pulpit Rock" width="304" height="228">

<figcaption>Fig.1 - A view of the pulpit rock in Norway.</figcaption>

</figure>

​

</body>

</html>

**HTML Web存储**

**localStorage和sessionStorage**

* localStorage:用于长久保存整个网站的数据，保存的数据没有过期时间，直到手动去除
* sessionStorage：用于临时保存同一窗口的数据，在关闭窗口或标签页以后将会删除数据

在使用web存储前，应检查浏览器是否支持localStorage和sessionStorage：

if(typeof(Storage)!=="undefined")

{

// 是的! 支持 localStorage sessionStorage 对象!

// 一些代码.....

} else {

// 抱歉! 不支持 web 存储。

}

**localStorage对象**

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<div id="result"></div>

<script>

if(typeof(Storage)!=="undefined")

{

localStorage.sitename="菜鸟教程";

document.getElementById("result").innerHTML="网站名：" + localStorage.sitename;

}

else

{

document.getElementById("result").innerHTML="对不起，您的浏览器不支持 web 存储。";

}

</script>

​

</body>

</html>

实例解析：

* 使用key=”sitename”和value=“菜鸟教程”创建一个localStorage键/值对
* 检索键值为“sitename”的值然后将数据插入id=”result”的元素中

以上实例也可以这么写：

// 存储

localStorage.sitename = "菜鸟教程";

// 查找

document.getElementById("result").innerHTML = localStorage.sitename;

移除localStorage中的“sitename”

localStorage.removeItem("sitename");

不管是localStorage，还是sessionStorage，可使用的API都相同，常用的有以下几个：

* 保存数据：localStorage.setItem(key,value);
* 读取数据：localStorage.getItem(key);
* 删除单个数据：localStorage.removeItem(key);
* 删除所有数据：localStorage.clear();
* 得到某个索引的key：localStorage.key(index);

**提示：**键/值对通常以字符串存储，可以按自己的需要转换该格式

下面实例展示用户点击按钮的次数，代码中的字符串值转换为数字类型：

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

<script>

function clickCounter()

{

if(typeof(Storage)!=="undefined")

{

if (localStorage.clickcount)

{

localStorage.clickcount=Number(localStorage.clickcount)+1;

}

else

{

localStorage.clickcount=1;

}

document.getElementById("result").innerHTML=" 你已经点击了按钮 " + localStorage.clickcount + " 次 ";

}

else

{

document.getElementById("result").innerHTML="对不起，您的浏览器不支持 web 存储。";

}

}

</script>

</head>

<body>

<p><button onclick="clickCounter()" type="button">点我！</button></p>

<div id="result"></div>

<p>点击该按钮查看计数器的增加。</p>

<p>关闭浏览器选项卡(或窗口),重新打开此页面,计数器将继续计数(不是重置)。</p>

</body>

</html>

**sessionStorage对象**

针对一个session进行数据存储，当用户关闭浏览器后，数据会删除

以下是如何创建并访问一个sessionStorage

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

<script>

function clickCounter()

{

if(typeof(Storage)!=="undefined")

{

if (sessionStorage.clickcount)

{

sessionStorage.clickcount=Number(sessionStorage.clickcount)+1;

}

else

{

sessionStorage.clickcount=1;

}

document.getElementById("result").innerHTML="在这个会话中你已经点击了该按钮 " + sessionStorage.clickcount + " 次 ";

}

else

{

document.getElementById("result").innerHTML="抱歉，您的浏览器不支持 web 存储";

}

}

</script>

</head>

<body>

<p><button onclick="clickCounter()" type="button">点我！</button></p>

<div id="result"></div>

<p>点击该按钮查看计数器的增加。</p>

<p>关闭浏览器选项卡(或窗口),重新打开此页面,计数器将重置。</p>

</body>

</html>

**Web Storage开发一个简单的网站列表程序**

实现以下功能：

* 可以输入网站名，网址，以网站名作为key存入localStorage
* 根据网站名，查找网址
* 列出当前已保存的所有网站

以下代码用于保存与查找数据：

save()和find()方法：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>HTML5本地存储之Web Storage篇</title>

</head>

<body>

<div style="border: 2px dashed #ccc;width:320px;text-align:center;">

<label for="sitename">网站名(key)：</label>

<input type="text" id="sitename" name="sitename" class="text"/>

<br/>

<label for="siteurl">网 址(value)：</label>

<input type="text" id="siteurl" name="siteurl"/>

<br/>

<input type="button" onclick="save()" value="新增记录"/>

<hr/>

<label for="search\_phone">输入网站名：</label>

<input type="text" id="search\_site" name="search\_site"/>

<input type="button" onclick="find()" value="查找网站"/>

<p id="find\_result"><br/></p>

</div>

<br/>

<div id="list">

</div>

<script>

// 载入所有存储在localStorage的数据

loadAll();

//保存数据

function save(){

var siteurl = document.getElementById("siteurl").value;

var sitename = document.getElementById("sitename").value;

localStorage.setItem(sitename, siteurl);

alert("添加成功");

}

//查找数据

function find(){

var search\_site = document.getElementById("search\_site").value;

var siteurl = localStorage.getItem(search\_site);

var find\_result = document.getElementById("find\_result");

find\_result.innerHTML = search\_site + "的网址是：" + siteurl;

}

//将所有存储在localStorage中的对象提取出来，并展现到界面上

function loadAll(){

var list = document.getElementById("list");

if(localStorage.length>0){

var result = "<table border='1'>";

result += "<tr><td>key</td><td>value</td></tr>";

for(var i=0;i<localStorage.length;i++){

var sitename = localStorage.key(i);

var siteurl = localStorage.getItem(sitename);

result += "<tr><td>"+sitename+"</td><td>"+siteurl+"</td></tr>";

}

result += "</table>";

list.innerHTML = result;

}else{

list.innerHTML = "数据为空……";

}

}

</script>

</body>

</html>

使用JSON.stringify来存储对象数据，可以将对象转换为字符串

var site = new Object;

...

var str = JSON.stringify(site); // 将对象转换为字符串

使用JSON.parse方法将字符串转换为JSON对象：

var site = JSON.parse(str);

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>HTML5本地存储之Web Storage篇</title>

</head>

<body>

<div style="border: 2px dashed #ccc;width:320px;text-align:center;">

<label for="keyname">别名(key):</label>

<input type="text" id="keyname" name="keyname" class="text"/>

<br/>

<label for="sitename">网站名：</label>

<input type="text" id="sitename" name="sitename" class="text"/>

<br/>

<label for="siteurl">网 址：</label>

<input type="text" id="siteurl" name="siteurl"/>

<br/>

<input type="button" onclick="save()" value="新增记录"/>

<hr/>

<label for="search\_phone">输入别名(key)：</label>

<input type="text" id="search\_site" name="search\_site"/>

<input type="button" onclick="find()" value="查找网站"/>

<p id="find\_result"><br/></p>

</div>

<br/>

<div id="list">

</div>

<script>

//保存数据

function save(){

var site = new Object;

site.keyname = document.getElementById("keyname").value;

site.sitename = document.getElementById("sitename").value;

site.siteurl = document.getElementById("siteurl").value;

var str = JSON.stringify(site); // 将对象转换为字符串

localStorage.setItem(site.keyname,str);

alert("保存成功");

}

//查找数据

function find(){

var search\_site = document.getElementById("search\_site").value;

var str = localStorage.getItem(search\_site);

var find\_result = document.getElementById("find\_result");

var site = JSON.parse(str);

find\_result.innerHTML = search\_site + "的网站名是：" + site.sitename + "，网址是：" + site.siteurl;

}

//将所有存储在localStorage中的对象提取出来，并展现到界面上

// 确保存储的 keyname 对应的值为转换对象，否则JSON.parse会报错

function loadAll(){

var list = document.getElementById("list");

if(localStorage.length>0){

var result = "<table border='1'>";

result += "<tr><td>别名</td><td>网站名</td><td>网址</td></tr>";

for(var i=0;i<localStorage.length;i++){

var keyname = localStorage.key(i);

var str = localStorage.getItem(keyname);

var site = JSON.parse(str);

result += "<tr><td>"+site.keyname+"</td><td>"+site.sitename+"</td><td>"+site.siteurl+"</td></tr>";

}

result += "</table>";

list.innerHTML = result;

}else{

list.innerHTML = "数据为空...";

}

}

</script>

</body>

</html>

**视频格式：**

|  |  |  |
| --- | --- | --- |
| 格式 | 文件 | 描述 |
| AVI | .avi | AVI格式不是所有的Windows计算机都能播放 |
| WMV | .wmv | 如果未安装额外的组件，无法播放。一些后期的windows media电影在所有非windows计算机上都无法播放，因为没有合适的播放器 |
| MPEG | .mpg  .mpeg | 跨平台，得到了所有最流行的浏览器支持 |
| QuickTime | .mov | 不能在没有安装额外的组件的Windows计算机上播放 |
| RealVideo | .rm  .ram | 允许低带宽条件下的视频流，由于低带宽优先，质量常会降低 |
| Flash | .swf  .flv | Shockwave格式需要额外的组件来播放，该嘴贱会预装到Firefox或IE之类的浏览器上 |
| Mpeg-4 | .mp4 | Mpeg-4是一种只针对因特网的新格式。越来越多的视频发布者转到MP4，将它作为Flash播放器和HTML5的因特网共享格式 |

**声音格式：**

|  |  |  |
| --- | --- | --- |
| 格式 | 文件 | 描述 |
| MIDI | .mid  .midi | 一种针对电子音乐设备的格式。MIDI文件不含有声音，但包含可被电子产品播放的数字音乐指令  MIDI格式只包含指令，所有及其小巧 |
| RealAudio | .rm  .ram | 由Real Media针对因特网开发的。该格式支持视频，运行低带宽条件下的音频流。由于是低带宽优先的，质量常会降低 |
| Wave | .wav | 所有运行的Windows的计算机和所有网络浏览器（除了Google Chrome）都支持 |
| WMA | .wma | 质量优于MP3，兼容大多数播放器，除了iPod。可作为连续的数据流来传输，对于网络电台或在线音乐很实用 |
| MP3 | .mp3  .mpga | 实际是MPEG文件的的声音部分 |

**HTML插件**

扩展HTML浏览器的功能

**HTML助手**

辅助程序可用于播放视频和音频。使用<object>标签来加载。优势是能够允许用户来控制部分或全部播放设置

插件可以通过<object>标签或者<embed>标签添加在页面中

大多数辅助应用程序允许对音量设置和播放功能（比如后退，暂停，停止和播放）的手工控制

提示：可以使用<vedio>和<audio>标签赖先生音频和视频

**<object>元素**

定义了在HTML文档中嵌入的对象

该标签用于插入对象

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<object width="400" height="50" data="bookmark.swf"></object>

</body>

</html>

<object>元素同样可用于包含HTML文件：

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<object width="100%" height="500px" data="snippet.html"></object>

</body>

</html>

或者插入一张图片：

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<object data="logo.png"></object>

</body>

</html>

**<embed>元素**

代表一个HTML Embed对象

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<embed width="400" height="50" src="bookmark.swf">

​

</body>

</html>

**注意：**<embed>元素没有关闭标签，不能使用替代文本

<embed>元素同样可用于包含HTML文件：

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<embed width="100%" height="500px" src="snippet.html">

​

</body>

</html>

或者插入一张图片：

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<embed src="logo.png">

​

</body>

</html>

**HTML音频**

使用插件:<object>或者<embed>标签添加在页面上

这些标签定义资源的容器，根据类型，会由浏览器显示，也会由外部插件显示

**使用<embed>元素**

定义外部内容的容器

下面的代码能够显示嵌入网页中的MP3文件：

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<embed height="50" width="100" src="horse.mp3">

<p>如果你无法听到该音频，那么可能你的电脑或浏览器不支持该文件格式。</p>

<p>或者你没有打开扬声器。</p>

​

</body>

</html>

**问题：**

* <embed>标签在HTML 4中是无效的，页面无法通过HTML 4验证
* 不同的浏览器对音频格式的支持也不同
* 如果浏览器不支持该文件格式，没有插件的话就无法播放该音频
* 如果用户的计算机未安装插件，无法播放音频
* 如果把该文件转换为其他格式，任然无法在所有浏览器中播放

**使用<object>元素**

<object tag>标签也可以定义外部内容的容器

下面的代码片段能够显示嵌入网页中的MP3文件：

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<object height="50" width="100" data="horse.mp3"></object>

<p>如果你无法听到该音频，那么可能你的电脑或浏览器不支持该文件格式。</p>

<p>或者你没有打开扬声器。</p>

​

​

</body>

</html>

**问题：**

* 不同的浏览器对音频格式的支持也不同
* 如果浏览器不支持该文件格式，没有插件的话就无法播放该音频
* 如果用户的计算机未安装插件，无法播放音频
* 如果把该文件转换为其他格式，仍然无法在所有浏览器中播放

**使用HTML5<audio>元素**

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<audio controls>

<source src="horse.mp3" type="audio/mpeg">

<source src="horse.ogg" type="audio/ogg">

您的浏览器不支持该音频格式。

</audio>

​

</body>

</html>

**问题：**

* <audio>标签在HTML 4中是无效的。无法通过HTML 4验证
* 必须把音频文件转换为不同的格式
* <audio>元素在老式浏览器中不起作用

**最好的HTML解决方法**

下面的例子使用了两个不同的音频格式。HTML5<audio>元素会尝试mp3或ogg来播放音频。

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<audio controls>

<source src="horse.mp3" type="audio/mpeg">

<source src="horse.ogg" type="audio/ogg">

<embed height="50" width="100" src="horse.mp3">

</audio>

​

</body>

</html>

**问题：**

* 必须把音频转换为不同的格式
* <embed>元素无法回退来显示错误信息

**使用超链接**

如果网页包含指向媒体文件的超链接，大多数浏览器会使用“辅助应用程序”来播放文件。

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<h2>Linking To a Song</h2>

​

<p><a href="horse.mp3">点击此处播放</a></p>

​

</body>

</html>

**HTML多媒体标签**

|  |  |
| --- | --- |
| 标签 | 描述 |
| <embed> | 定义内嵌对象 |
| <object> | 定义内嵌对象 |
| <param> | 定义对象的参数 |
| <audio> | 定义了声音内容 |
| <video> | 定义一个视频或者影片 |
| <source> | 定义了media元素的多媒体资源 |
| <track> | 规定media元素的字幕文件或其他包含文本的文件 |

**HTTP方法：GET对比POST**

两种HTTP请求方法：GET和POST

GET:从指定的资源请求数据

POST:向指定的资源提交要被处理的数据

**GET方法**

注意：查询字符串是在GET请求的URL中发送的

/test/demo\_form.php**?name1=value1&name2=value2**

**有关GET请求的其他一些注释：**

* GET请求可被缓存
* GET请求保留在浏览器历史记录中
* GET请求可被收藏为书签
* GET请求不应在处理敏感数据时使用
* GET请求有长度限制
* GET请求只应当用于取回数据

**POST方法**

注意：查询字符串是在POST请求的HTTP消息主体中发出的

POST /test/demo\_form.php HTTP/1.1  
Host: runoob.com  
**name1=value1&name2=value2**

**有关POST请求的其他一些注释：**

* POST请求不会被缓存
* POST请求不会保留在浏览器历史记录中
* POST不能被收藏为书签
* POST请求对数据长度没有要求

**比较GET和POST**

|  |  |  |
| --- | --- | --- |
| GET | | POST |
| 后退按钮/刷新 | 无害 | 数据会被重新提交（浏览器应该告知用户数据会被重新提交） |
| 书签 | 可收藏为书签 | 不可收藏为书签 |
| 缓存 | 能被缓存 | 不能缓存 |
| 编码类型 | Application/x-www-form-urlencoded | Application-x-www-form-urlencoded or multipart/form-data.为二进制数据使用多重编码 |
| 历史 | 参数保留在浏览器历史中 | 参数不会保存在浏览器历史中 |
| 对数据长度的限制 | 是的。当发送数据时，GET方法向URL添加数据，URL的长度是受限制的 | 无限制 |
| 对数据类型的限制 | 只允许ASCII字符 | 没有限制，也允许二进制数据 |
| 安全性 | 与POST相比，GET的安全性较差，因为所发送的数据是URL的一部分。在发送密码或其他敏感信息时绝不要使用GET | POST比GET更安全，因为参数不会被保存在浏览器历史或web服务器日志中 |
| 可见性 | 数据在URL中对所有人都是可见的 | 数据不会显示在URL中 |

其他HTTP请求方法

|  |  |
| --- | --- |
| 方法 | 描述 |
| HEAD | 于GET相同，但只返回HTTP报头，不返回文档主体 |
| PUT | 上传指定的URI表示 |
| DELETE | 删除指定资源 |
| OPTIONS | 返回服务器支持的HTTP方法 |
| CONNECT | 把请求连接转换到透明的TCP/IP通道 |

**HTML<!DOCTYPE>声明**

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>文档标题</title>

</head>

​

<body>

文档内容......

</body>

​

</html>

**定义及使用说明：**

<!DOCTYPE>声明位于文档中的最前面的位置，处于<html>标签之前

<!DOCTYPE>声明不是一个HTML标签，是用来告知Web浏览器页面使用了哪种HTML版本

提示：总是给HTML文档添加<!DOCTYPE>声明，确保浏览器能够预先知道文档类型

**提示和注释**

注释：<!DOCTYPE>标签没有结束标签

提示：<!DOCTYPE>声明不区分大小写

提示：使用W3C验证，检查是否编写了一个带有正确DTD的合法的HTML/XHTML文档

常见的DOCTYPE声明

HTML5:<!DOCTYPE html>

**HTML<a>标签**

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<a href="//www.runoob.com">访问菜鸟教程!</a>

​

</body>

</html>

**标签定义及使用说明**

<a>标签定义超链接，用于从一个页面链接到另一个页面

<a>元素最重要的属性是href属性，指定链接的目标

在所有浏览器中，链接的默认外观如下：

* 未被访问的链接带有下划线且是蓝色的
* 已被访问的链接带有下划线而且是紫色的
* 活动链接带有下划线而且是红色的

**提示和注释**

提示：如果没有使用href属性，则不能使用hreflang,rel,target以及type属性

提示：通常在当前浏览器窗口显示被链接页面，除非规定了其他target

提示：请使用CSS来改变链接的样式

**属性**

|  |  |  |
| --- | --- | --- |
| 属性 | 值 | 描述 |
| charset | char\_encoding | HTML5不支持。规定目标URL的字符编码 |
| coords | coordinates | HTML5不支持。规定链接的坐标 |
| download | filename | 指定下载链接 |
| href | URL | 规定链接的目标URL |
| hreflang | language\_code | 规定目标URL的基准语言，尽在href属性存在时使用 |
| media | media\_query | 规定目标URL的媒介类型，默认值：all,尽在href属性存在时使用 |
| name | section\_name | HTML5不支持。规定锚的名称 |
| rel | alternate  author  bookmark  help  license  next  nofollow  noreferrer  prefetch  prev  search  tag | 规定当前文档与目标URL之间的关系，仅在href属性存在时使用 |
| rev | text | HTML5不支持。规定目标URL与当前文档之间的关系 |
| shape | default  rect  circle  poly | HTML5不支持。规定链接的形状 |
| target | \_blank  \_parent  \_self  \_top  framename | 规定在何处打开目标URL仅在href属性存在时使用 |
| type | MIME\_type | 规定目标URL的MIME类型，仅在href属性存在时使用  注：MIME=Multipurpose Internet Mail Extensions |

**全局属性**

<a>标签支持HTML的全局属性

**事件属性**

<a>标签支持HTML的事件属性

**HTML<abbr>标签**

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<p>The <abbr title="World Health Organization">WHO</abbr> was founded in 1948.</p>

​

</body>

</html>

**标签定义及使用说明**

<abbr>标签用来表示一个缩写词或者首字母缩略词

通过对缩写词语进行标记，能够为浏览器，拼写检查程序，翻译系统以及搜索引擎分度器提供有用的信息

**提示和注释**

提示：在某些浏览器中，当把鼠标移至带有<abbr>标签的缩写词/首字母缩略词上时，<abbr>标签的title属性可被用来展示缩写词/首字母缩略词的完整版本

**全局属性**

<abbr>标签支持HTML的全局属性

**事件属性**

<abbr>标签支持HTML的事件属性

**HTML<address>标签**

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<address>

Written by <a href="mailto:webmaster@example.com">Jon Doe</a>.<br>

Visit us at:<br>

Example.com<br>

Box 564, Disneyland<br>

USA

</address>

​

</body>

</html>

**标签定义及使用说明**

<address>标签定义文档作者/所有者的联系信息

如果<address>元素位于<body>元素内部，则表示该文档作者/所有者的联系信息

如果<address>元素位于<article>元素内部，则表示该文章作者/所有者的联系信息

<address>元素的文本通常呈现为斜体，大多数浏览器会在该元素的前后添加换行

**提示和注释**

提示：不应该使用<address>标签来描述邮政地址，除非这些信息是联系信息的组成部分

提示：<address>元素通常被包含在<footer>元素的其他信息中

**全局属性**

<address>标签支持HTML的全局属性

**事件属性**

<address>标签支持HTML的事件属性

**HTML<area>标签**

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<p>点击太阳或其他行星，注意变化：</p>

​

<img src="planets.gif" width="145" height="126" alt="Planets" usemap="#planetmap">

​

<map name="planetmap">

<area shape="rect" coords="0,0,82,126" alt="Sun" href="sun.htm">

<area shape="circle" coords="90,58,3" alt="Mercury" href="mercur.htm">

<area shape="circle" coords="124,58,8" alt="Venus" href="venus.htm">

</map>

​

</body>

</html>

​

**标签定义及使用说明**

<area>标签定义图像映射内部的区域（图像映射指的是带有可点击区域的图像）

<area>元素始终嵌套在<map>标签内部

注释：<img>标签中的usemap属性与<map>元素中的name相关联，以创建图像与映射之间的关系

**属性**

|  |  |  |
| --- | --- | --- |
| 属性 | 值 | 描述 |
| alt | text | 规定区域的替代文本，如果使用href属性，则该属性时必需的 |
| coords | coordinates | 规定区域的坐标 |
| href | URL | 规定区域的目标URL |
| hreflang | language\_code | 规定目标URL的语言 |
| media | media query | 规定目标URL是为何种媒介/设备优化的，默认：all |
| nohref | value | HTML5不支持，规定没有相关链接的区域 |
| rel | alternate  author  bookmark  help  license  next  nofollow  noreferrer  prefetch  prev  search  tag | 规定当前文档与目标URL之间的关系 |
| shape | default  rect  ciecle  poly | 规定区域的形状 |
| target | \_blank  \_parent  \_self  framename | 规定在何处打开目标URL |
| type | MIME\_type | 规定目标URL的MIME类型  注：MIME=Multipurpose Internet Mail Extensions |

**全局属性**

<area>标签支持HTML的全局属性

**事件属性**

<area>标签支持HTML的事件属性

**HTML<article>标签**

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<article>

<h1>Internet Explorer 9</h1>

<p> Windows Internet Explorer 9(缩写为 IE9 )在2011年3月14日21:00 发布。</p>

</article>

​

</body>

</html>

**标签定义及使用说明**

<article>标签定义独立的内容

<article>标签定义的内容本身必须是由意义的且必须是独立于文档的其余部分

<article>的潜在来源：

* 论坛帖子
* 博客文章
* 新闻故事
* 评论

**全局属性**

<article>标签支持HTM的全局属性

**事件属性**

<article>标签支持HTML的事件属性

**HTML<aside>标签**

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<p>My family and I visited The Epcot center this summer.</p>

​

<aside>

<h4>Epcot Center</h4>

<p>The Epcot Center is a theme park in Disney World, Florida.</p>

</aside>

​

</body>

</html>

**标签定义及使用说明**

<aside>标签定义<article>标签外的内容

aside的内容应该与附近的内容相关

**提示和注释**

提示：<aside>的内容可用作文章的侧栏

**全局属性**

<aside>标签支持HTML的全局属性

**事件属性**

<aside>标签支持HTML的事件属性

**HTML<audio>标签**

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<audio controls>

<source src="horse.ogg" >

<source src="horse.mp3" >

您的浏览器不支持 audio 元素。

</audio>

​

</body>

</html>

**标签定义及使用说明**

<audio>标签定义声音，比如音乐或其他音频流

目前，<audio>元素支持的3中文件格式：MP3,Wav,Ogg

|  |  |  |  |
| --- | --- | --- | --- |
| 浏览器 | MP3 | Wav | Ogg |
| Internet Explorer | YES | NO | NO |
| Chrome | YES | YES | YES |
| Firefox | YES | YES | YES |
| Safari | YES | YES | NO |
| Opera | YES | YES | YES |

**提示和注释**

提示：可以在<audio>和</audio>之间放置文本内容，这些文本信息将会被显示在那些不支持<audio>标签的浏览器中

**属性**

|  |  |  |
| --- | --- | --- |
| 属性 | 值 | 描述 |
| autoplay | autoplay | 如果出现该属性，则音频在就绪后马上播放 |
| controls | controls | 如果出现该属性，则向用户显示音频控件（比如播放/暂停按钮） |
| loop | loop | 如果出现该属性，则每当音频结束时重新开始播放 |
| muted | muted | 如果出现该属性，则音频输出为静音 |
| preload | auto  metadata  none | 规定当网页加载时，音频是否默认被加载以及如何被加载 |
| src | URL | 规定音频文件的URL |

**全局属性**

<audio>标签支持HTML的全局属性

**事件属性**

<audio>标签支持HTML的事件属性

**HTML<b>标签**

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<p>这是一个普通的文本- <b>这是一个加粗文本</b>。</p>

​

</body>

</html>

**标签定义及使用说明**

<b>标签定义粗体的文本

**提示和注释**

注释：根据HTML 5的规范，<b>标签应该作为最后的选择，只有在没有其他标记比较适合时才使用它。

HTML 5规范声明：标题应该用<h1>-<h6>标签表示，被强调的文本应该用<em>标签标示，重要的文本应该用<strong>标签表示，被标记的或者高亮显示的文本应该用<mark>标签表示

提示：可以使用CSS的“font-weight”属性设置粗体文本

**全局属性**

<b>标签支持HTML的全局属性

**事件属性**

<b>标签支持HTML的事件属性

**HTML<base>标签**

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

<base href="//www.runoob.com//images/" target="\_blank">

</head>

<body>

​

<p><img src="logo.png" > - 注意这里我们设置了图片的相对地址。能正常显示是因为我们在 head 部分设置了 base 标签，该标签指定了页面上所有链接的默认 URL，所以该图片的访问地址为 "http://www.runoob.com/images/logo.png"</p>

​

<p><a href="//www.runoob.com/">runoob.com</a> - 注意这个链接会在新窗口打开，即便它没有 target="\_blank" 属性。因为在 base 标签里我们已经设置了 target 属性的值为 "\_blank"。</p>

​

</body>

</html>

**标签定义及使用说明**

<base>标签为页面上的所有的相对链接规定默认URL或默认目标

在一个文档中，最多使用一个<base>。<base>标签必须位于<head>元素内部

**提示和注释**

提示：把<base>标签排在<head>元素中第一个元素的位置，这样head区域中其他元素就可以使用<base>元素中的信息了

注释：如果使用了<base>标签，则必须具备href属性或者target属性或者两个属性都具备。

**属性**

|  |  |  |
| --- | --- | --- |
| 属性 | 值 | 描述 |
| href | URL | 规定页面中所有相对链接的基准URL |
| target | \_blank  \_parent  \_self  \_top  framename | 规定页面中所有的超链接和表单在何处打开，该属性会被每个链接中的target属性覆盖 |

**全局属性**

<base>标签支持HTML的全局属性

**事件属性**

<base>标签不支持任何的事件属性

**HTML<bdi>标签**

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<p>在下面的示例中,显示着每个用户比赛的分数。将用户名从周围的文本方向设置中隔离出来。</p>

​

<ul>

<li>用户 <bdi>hrefs</bdi>: 60 分</li>

<li>用户 <bdi>jdoe</bdi>: 80 分</li>

<li>用户 <bdi>إيان</bdi>: 90 分</li>

</ul>

​

<p><b>注意：</b> 目前只有 Firefox 和 Chrome 浏览器支持 bdi 标签。</p>

​

</body>

</html>

**标签定义及使用说明**

bdi指的是bidi隔离

<bdi>标签允许设置一段文本，使其脱离其父元素的文本方向设置

在发布用户评论或其他无法完全控制的内容时，该标签很有用

**全局属性**

<bdi>标签支持HTML的全局属性

**事件属性**

<bdi>标签支持HTML的事件属性

**HTML<bdo>标签**

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<p>该段落文字从左到右显示。</p>

<p><bdo dir="rtl">该段落文字从右到左显示。</bdo></p>

​

</body>

</html>

**标签定义及使用说明**

bdo指的是bidi覆盖

<bdo>标签用来覆盖默认的文本方向

**属性**

|  |  |  |
| --- | --- | --- |
| 属性 | 值 | 描述 |
| dir | ltr  rtl | 必需。规定<bdo>元素内的文本方向 |

**全局属性**

<bdo>标签支持HTML的全局属性

**事件属性**

<bdo>标签支持HTML的事件属性

**HTML<blockquote>标签**

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<h1>About WWF</h1>

<p>Here is a quote from WWF's website:</p>

​

<blockquote cite="http://www.worldwildlife.org/who/index.html">

For 50 years, WWF has been protecting the future of nature. The world•s leading conservation organization, WWF works in 100 countries and is supported by 1.2 million members in the United States and close to 5 million globally.

</blockquote>

​

</body>

</html>

**标签定义及使用说明**

<blockquote>标签定义摘自另一个源的块引用

浏览器通常会对<blockquote>元素进行缩进

**提示和注释**

提示：如果标记是不需要段落分隔的短引用，请使用<q>

**属性**

|  |  |  |
| --- | --- | --- |
| 属性 | 值 | 描述 |
| cite | URL | 规定引用的来源 |

**全局属性**

<blockquote>标签支持HTML的全局属性

**事件属性**

<blockquote>标签支持HTML的事件属性

**HTML<body>标签**

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>文档标题</title>

</head>

​

<body>

文档内容......

</body>

​

</html>

**标签定义及使用说明**

<body>标签定义文档的主体

<body>元素包含文档的所有内容（比如文本，超链接，图像，表格和列表等等）

**全局属性**

<body>标签支持HTML的全局属性

**事件属性**

<body>标签支持HTML的事件属性

**HTML<br>标签**

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<p>

使用 br 元素<br>在文本中<br>换行。

</p>

​

</body>

</html>

**标签定义及使用说明**

<br>标签插入一个简单的换行符

<br>标签是一个空标签，意味着他没有结束标签

**提示和注释**

提示：在写地址信息或者写诗词时<br>标签非常有用

注释：请使用<br>标签来输入空行，不是分割段落

**全局属性**

<br>标签支持HTML的全局属性

**事件属性**

<br>标签支持HTML的事件属性

**HTML<button>标签**

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<button type="button" onclick="alert('你好，世界!')">点我!</button>

</body>

</html>

**标签定义及说明**

<button>标签定义一个按钮

在<button>元素内部，可以放置内容，比如文本或图像。这是该元素与使用<input>元素创建的按钮的不同之处

提示：请始终为<button>元素规定type属性，不同的浏览器对<button>元素的type属性使用不同的默认值

**提示和注释**

注释：如果在HTML表单中使用<button>元素，不同的浏览器可能会提交不同的按钮值。请使用<input>在HTML表单中创建按钮

**属性**

|  |  |  |
| --- | --- | --- |
| 属性 | 值 | 描述 |
| autofocus | autofocus | 规定当页面加载时按钮应当自动的获得焦点 |
| disabled | disabled | 规定应该禁用该按钮 |
| form | form\_id | 规定按钮属于一个或多个表单 |
| formaction | URL | 规定当提交表单时向何处发送表单数据，覆盖form元素的action属性，该属性与type=”submit”配合使用 |
| formenctype | application/x-www-form-urlencoded multipart/form-data  text/plain | 规定在向服务器发送表单数据之前如何对其进行编码，覆盖form元素的enctype属性，该属性与type=”submit”配合使用 |
| formmethod | get  post | 规定用于发送表单数据的HTTP方法，覆盖form元素的method属性，该属性与type=”submit”配合使用 |
| formnovalidate | formnovalidate | 如果使用该属性，则提交表单时不进行验证，覆盖form元素的novalidate属性，该属性与type=”submit”配合使用 |
| formtarget | \_blank  \_self  \_parent  \_top  framename | 规定在何处打开action URL,覆盖form元素的target属性，该属性与type=”submit”配合使用 |
| name | name | 规定按钮的名称 |
| type | button  reset  submit | 规定按钮的类型 |
| value | text | 规定按钮的初始值，可由脚本进行修改 |

**全局属性**

<button>标签支持HTML的全局属性

**事件属性**

<button>标签支持HTML的事件属性

**HTML<canvas>标签**

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<canvas id="myCanvas">你的浏览器不支持 HTML5 canvas 标签。</canvas>

​

<script>

var c=document.getElementById('myCanvas');

var ctx=c.getContext('2d');

ctx.fillStyle='#FF0000';

ctx.fillRect(0,0,80,100);

</script>

​

</body>

</html>

**标签定义及时说明**

<canvas>标签通过脚本来绘制图形（比如图表和其他图像）

<canvas>标签只是图形容器，必须使用脚本来绘制图形

**提示和注释**：

注释：<canvas>元素中的任何文本将会被显示在不支持<canvas>的浏览器中

**属性**

|  |  |  |
| --- | --- | --- |
| 属性 | 值 | 描述 |
| height | pixels | 规定画布的高度 |
| width | pixels | 规定画布的宽度 |

**全局属性**

<canvas>标签支持HTML的全局属性

**事件属性**

<canvas>标签支持HTML的事件属性

**HTML<caption>标签**

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<table border="1">

<caption>Monthly savings</caption>

<tr>

<th>Month</th>

<th>Savings</th>

</tr>

<tr>

<td>January</td>

<td>$100</td>

</tr>

<tr>

<td>February</td>

<td>$50</td>

</tr>

</table>

​

</body>

</html>

**标签定义及使用说明**

<caption>标签定义表格的问题

<caption>标签必须直接放置到<table>标签之后

只能对每个表格定义一个标题

提示：通常这个标题会被居中于表格之上，然而,CSS属性”text-align”和”caption-side”能用来设置标题的对齐方式和显示位置

**全局属性**

<caption>标签支持HTML的全局属性

**事件属性**

<caption>标签支持HTML的事件属性

**HTML<cite>标签**

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<img src="img\_the\_scream.jpg" width="220" height="277" alt="The Scream">

<p><cite>The Scream</cite> by Edward Munch. Painted in 1893.</p>

​

</body>

</html>

**标签定义及使用说明**

<cite>标签定义作品（比如书籍，歌曲，电影，电视节目，绘画，雕塑等等）的标题

注释：人民不属于作品的标题

**全局属性**

<cite>标签支持HTML的全局属性

**事件属性**

<cite>标签支持HTML的事件属性

**HTML<code>标签**

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<em>强调文本</em><br>

<strong>加粗文本</strong><br>

<dfn>定义项目</dfn><br>

<code>一段电脑代码</code><br>

<samp>计算机样本</samp><br>

<kbd>键盘输入</kbd><br>

<var>变量</var>

​

</body>

</html>

**标签定义及使用说明**

<code>标签是一个短语标签，用来定义计算机代码文本

提示：如果只是为了达到某种视觉效果而使用这个标签的话，建议使用CSS，这样可能会取得更丰富的效果

**所有短语标签：**

|  |  |
| --- | --- |
| 标签 | 描述 |
| <em> | 呈现为被强调的文本 |
| <strong> | 定义重要的文本 |
| <dfn> | 定义一个定义项目 |
| <code> | 定义计算机代码文本 |
| <samp> | 定义样本文本 |
| <kbd> | 定义键盘文本，它表示文本是从键盘上键入的，经常用在与计算机相关的文档或手册中 |
| <var> | 定义变量，可以将此标签与<pre>及<code>标签配合使用 |

**全局属性**

<code>标签支持HTML的全局属性

**事件属性**

<code>标签支持HTML的事件属性

**HTML<col>标签**

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<table border="1">

<colgroup>

<col span="2" style="background-color:red">

<col style="background-color:yellow">

</colgroup>

<tr>

<th>ISBN</th>

<th>Title</th>

<th>Price</th>

</tr>

<tr>

<td>3476896</td>

<td>My first HTML</td>

<td>$53</td>

</tr>

<tr>

<td>5869207</td>

<td>My first CSS</td>

<td>$49</td>

</tr>

</table>

​

</body>

</html>

**标签定义及使用说明**

<col>标签规定了<colgroup>元素内部的每一列的列属性

通过使用<col>标签，可以向整个列应用样式，而不需要重复为每个单元格或每一行设置样式

**属性**

|  |  |  |
| --- | --- | --- |
| 属性 | 值 | 描述 |
| span | number | 规定<col>元素应该横跨的列数 |

**全局属性**

<col>标签支持HTML的全局属性

**事件属性**

<col>标签支持HTML的事件属性

**HTML<colgroup>标签**

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<table border="1">

<colgroup>

<col span="2" style="background-color:red">

<col style="background-color:yellow">

</colgroup>

<tr>

<th>ISBN</th>

<th>Title</th>

<th>Price</th>

</tr>

<tr>

<td>3476896</td>

<td>My first HTML</td>

<td>$53</td>

</tr>

<tr>

<td>5869207</td>

<td>My first CSS</td>

<td>$49</td>

</tr>

</table>

​

</body>

</html>

**标签定义及使用说明**

<colgroup>标签用于对表格中的列进行组合，以便对其进行格式化

通过使用<colgroup>标签，可以向整个列应用样式，不需要为每个单元格或每一行设置样式

**注释**：只能在,table>元素之内，在任何一个<caption>元素之后，在任何一个<thead>,<tbody>,<tfoot>,<tr>元素之前使用<colgrpup>标签

**提示**：如果想对<colgroup>中的某列定义不同的属性，在<colgroup>标签内使用<col>标签

**属性**

|  |  |  |
| --- | --- | --- |
| 属性 | 值 | 描述 |
| span | number | 规定列组应该横跨的列数 |

**全局属性**

<colgroup>标签支持HTML的全局属性

**事件属性**

<colgroup>标签支持HTML的事件属性

**HTML<command>标签**

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

<script>

function save()

{

alert("Some javaScript....");

}

</script>

</head>

<body>

​

<menu>

<command type="command" label="Save" onclick="save()">Save</command>

</menu>

​

<p><b>注意：</b>只有 IE 9 支持 command 标签。</p>

​

</body>

</html>

**标签定义及使用说明**

<command>标签可以定义用户可能调用的命令（比如单选按钮，复选框或按钮）

当使用<menu>元素时，command元素将作为菜单或者工具栏的一部分显示出来，但是，用command规定键盘快捷键时，command元素能被放置在页面的任何位置，但元素不可见

**属性**

|  |  |  |
| --- | --- | --- |
| 属性 | 值 | 描述 |
| checked | checked | 规定当页面加载时，command是否被选中，仅用于radio或checkbox类型 |
| disabled | disabled | 规定command是否可用 |
| icon | URL | 规定作为command来显示的图像的URL |
| label | text | 必需，规定command的名字，对用户可见 |
| radiogroup | groupname | 规定可进行切换且将切换的command所属的组名，仅在类型为radio时使用 |
| type | checkbox  command  radio | 定义command的类型，默认是”command” |

**全局属性**

<command>标签支持HTML的全局属性

**事件属性**

<command>标签支持HTML的事件属性

**HTML<datalist>标签**

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

<form action="demo-form.php" method="get">

<input list="browsers" name="browser">

<datalist id="browsers">

<option value="Internet Explorer">

<option value="Firefox">

<option value="Chrome">

<option value="Opera">

<option value="Safari">

</datalist>

<input type="submit">

</form>

​

<p><strong>注意:</strong> Internet Explorer 9（更早 IE 版本），Safari 不支持 datalist 标签。</p>

​

</body>

</html>

**标签定义及使用说明**

<datalist>标签规定了<input>元素肯的选项列表

<datalist>标签被用来在为<input>元素提供“自动完成”的特性，用户能看到一个下拉列表，里边的选项是预先定义好的，将作为用户的输入数据

请使用<input>元素的list属性来绑定<datalist>元素

**全局属性**

<datalist>标签支持HTML的全局属性

**事件属性**

<datalist>标签支持HTML的事件属性

**HTML<dd>标签**

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<dl>

<dt>Coffee</dt>

<dd>Black hot drink</dd>

<dt>Milk</dt>

<dd>White cold drink</dd>

</dl>

​

</body>

</html>

**标签定义及使用说明**

<dd>标签被用来对一个描述列表中的项目/名字进行描述

<dd>标签与<dl>（定义一个描述列表）和<dt>（定义项目/名字）一起使用

在<dd>标签内，能放置段落，换行，图片，链接，列表等等

**全局属性**

<dd>标签支持HTML的全局属性

**事件属性**

<dd>标签支持HTML的事件属性

**HTML<del>标签**

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<p>My favorite color is <del>blue</del> <ins>red</ins>!</p>

​

</body>

</html>

**标签定义及使用说明**

<del>标签定义文档中已删除的文本

**提示和注释**

提示：可以看看<ins>标签如何标记先插入的文本

提示：<del>和<ins>一起使用，描述文档中的更新和修正，浏览器通常会在已删除文本上添加一条删除线，在新插入文本下添加一条下划线

**属性**

|  |  |  |
| --- | --- | --- |
| 属性 | 值 | 描述 |
| cite | URL | 规定一个解释了文本被删除的原因的文档是URL |
| datetime | YYYY-MM-DDThh:mm:ssTZD | 规定文本被删除的日期和时间 |

**全局属性**

<del>标签支持HTML的全局属性

**事件属性**

<del>标签支持HTML的事件属性

**HTML<details>标签**

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<details>

<summary>Copyright 1999-2011.</summary>

<p> - by Refsnes Data. All Rights Reserved.</p>

<p>All content and graphics on this web site are the property of the company Refsnes Data.</p>

</details>

​

<p><b>注意：</b>目前，只有 Chrome 和 Safari 6 支持 details 标签。</p>

​

</body>

</html>

**标签定义及使用说明**

<details>标签规定了用户可见的或者隐藏的需求的补充细节

<details>标签用来供用户开启关闭的交互式控件，任何形式的内容都能被放在<details>标签里面

<details>元素的内容对用户是不可见的，除非设置了open属性

**提示和注释**

提示：与<summary>标签配合使用可以为details定义标题，标题是可见的，用户点击标题时，会显示出details

**属性**

|  |  |  |
| --- | --- | --- |
| 属性 | 值 | 描述 |
| open | open | 规定details是否可见 |

**全局属性**

<details>标签支持HTML的全局属性

**事件属性**

<details>标签支持HTML的事件属性

**HTML<dfn>标签**

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<em>强调文本</em><br>

<strong>加粗文本</strong><br>

<dfn>定义项目</dfn><br>

<code>一段电脑代码</code><br>

<samp>计算机样本</samp><br>

<kbd>键盘输入</kbd><br>

<var>变量</var>

​

</body>

</html>

**标签定义及使用说明**

<dfn>标签是一个短语标签，用来定义一个定义项目

**所有短语标签：**

|  |  |
| --- | --- |
| 标签 | 描述 |
| <em> | 呈现为被强调的文本 |
| <strong> | 定义重要的文本 |
| <dfn> | 定义一个定义项目 |
| <code> | 定义计算机代码文本 |
| <samp> | 定义样本文本 |
| <kbd> | 定义键盘文本，它表示文本是从键盘上键入的，经常用在与计算机相关的文档或手册中 |
| <var> | 定义变量，可以将此标签与<pre>及<code>标签配合使用 |

**全局属性**

<dfn>标签支持HTML的全局属性

**事件属性**

<dfn>标签支持HTML的事件属性

**HTML<dialog>标签**

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<p><b>注意：</b>目前只有 Chrome 和 Safari 6 支持 dialog 标签。</p>

​

​

<table border="1">

<tr>

<th>January <dialog open>This is an open dialog window</dialog></th>

<th>February</th>

<th>March</th>

</tr>

<tr>

<td>31</td>

<td>28</td>

<td>31</td>

</tr>

</table>

​

</body>

</html>

**标签定义及使用说明**

<dialog>标签定义一个对话框，确认框或窗口

**属性**

|  |  |  |
| --- | --- | --- |
| 属性 | 值 | 描述 |
| open | open | 规定dialog元素时有效的，用户可以与他进行交互 |

**全局属性**

<dialog>标签支持HTML的全局属性

**事件属性**

<dialog>标签支持HTML的事件属性

**HTML<div>标签**

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<p>这是一些文本。</p>

​

<div style="color:#0000FF">

<h3>这是一个在 div 元素中的标题。</h3>

<p>这是一个在 div 元素中的文本。</p>

</div>

​

<p>这是一些文本。</p>

​

</body>

</html>

**标签定义及使用说明**

<div>标签定义HTML文档中的一个分隔区块或者一个区域部分

<div>标签常用语组合块级元素，以便通过CSS来对这些元素进行格式化

**提示和注释**

提示：<div>元素经常与CSS一起使用，用来布局网页

注释：默认情况下，浏览器通常会在<div>元素前后放置一个换行符，然而，可以通过CSS改变这种情况

**全局属性**

<div>标签支持HTML的全局属性

**事件属性**

<div>标签支持HTML的事件属性

**HTML<dl>标签**

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<dl>

<dt>Coffee</dt>

<dd>Black hot drink</dd>

<dt>Milk</dt>

<dd>White cold drink</dd>

</dl>

​

</body>

</html>

**标签定义及使用说明**

<dl>标签定义一个描述列表

<dl>标签与<dt>（定义项目/名字）和<dd>（描述每一个项目/名字）一起使用

**全局属性**

<dl>标签支持HTML的全局属性

**事件属性**

<dl>标签支持HTML的事件属性

**HTML<dt>标签**

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<dl>

<dt>Coffee</dt>

<dd>Black hot drink</dd>

<dt>Milk</dt>

<dd>White cold drink</dd>

</dl>

​

</body>

</html>

**标签定义及使用说明**

<dt>标签定义一个描述列表的项目/名字

<dt>标签与<dl>（定义一个描述列表）和<dd>（描述每一个项目/名字）一起使用

**全局属性**

<dt>标签支持HTML的全局属性

**事件属性**

<dt>标签支持HTML的事件属性

**HTML<em>标签**

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<em>强调文本</em><br>

<strong>加粗文本</strong><br>

<dfn>定义项目</dfn><br>

<code>一段电脑代码</code><br>

<samp>计算机样本</samp><br>

<kbd>键盘输入</kbd><br>

<var>变量</var>

​

</body>

</html>

**标签定义及使用说明**

<em>标签是一个短语标签，用来呈现为被强调的文本

**所有短语标签：**

|  |  |
| --- | --- |
| 标签 | 描述 |
| <em> | 呈现为被强调的文本 |
| <strong> | 定义重要的文本 |
| <dfn> | 定义一个定义项目 |
| <code> | 定义计算机代码文本 |
| <samp> | 定义样本文本 |
| <kbd> | 定义键盘文本，它表示文本是从键盘上键入的，经常用在与计算机相关的文档或手册中 |
| <var> | 定义变量，可以将此标签与<pre>及<code>标签配合使用 |

**全局属性**

<em>标签支持HTML的全局属性

**事件属性**

<em>标签支持HTML的事件属性

# HTML <embed> 标签

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<embed src="helloworld.swf" tppabs="http://www.runoob.com//try/demo\_source/helloworld.swf">

​

</body>

</html>

## 标签定义及使用说明

<embed> 标签定义了一个容器，用来嵌入外部应用或者互动程序（插件）。

属性

|  |  |  |
| --- | --- | --- |
| 属性 | 值 | 描述 |
| height | pixels | 规定嵌入内容的高度 |
| src | URL | 规定被嵌入内容的URL |
| type | MIME\_type | 规定嵌入内容的MIME类型  注：MIME=Multipurpose Internet Mail Extensions |
| width | pixels | 规定嵌入内容的宽度 |

## 全局属性

<embed> 标签支持 [HTML的全局属性](https://www.runoob.com/tags/ref-standardattributes.html)。

## 事件属性

<embed> 标签支持 [HTML的事件属性](https://www.runoob.com/tags/ref-eventattributes.html)。

# HTML <fieldset> 标签

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<form>

<fieldset>

<legend>Personalia:</legend>

Name: <input type="text"><br>

Email: <input type="text"><br>

Date of birth: <input type="text">

</fieldset>

</form>

​

</body>

</html>

## 标签定义及使用说明

<fieldset> 标签可以将表单内的相关元素分组。

<fieldset> 标签会在相关表单元素周围绘制边框。

## 提示和注释

**提示：**[<legend>](https://www.runoob.com/tags/tag-legend.html) 标签为 <fieldset> 元素定义标题。

属性

|  |  |  |
| --- | --- | --- |
| 属性 | 值 | 描述 |
| disabled | disabled | 规定该组合中的相关表单元素应该被禁用 |
| form | form\_id | 规定fieldset所属的一个或多个表单 |
| name | text | 规定fieldset的名称 |

## 全局属性

<fieldset> 标签支持 [HTML 的全局属性](https://www.runoob.com/tags/ref-standardattributes.html)。

## 事件属性

<fieldset> 标签支持 [HTML 的事件属性](https://www.runoob.com/tags/ref-eventattributes.html)。

# HTML <figcaption> 标签

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<p>The Pulpit Rock is a massive cliff 604 metres (1982 feet) above Lysefjorden, opposite the Kjerag plateau, in Forsand, Ryfylke, Norway. The top of the cliff is approximately 25 by 25 metres (82 by 82 feet) square and almost flat, and is a famous tourist attraction in Norway.</p>

​

<figure>

<img src="img\_pulpit.jpg" alt="The Pulpit Rock" width="304" height="228">

<figcaption>Fig.1 - A view of the pulpit rock in Norway.</figcaption>

</figure>

​

</body>

</html>

## 标签定义及使用说明

<figcaption> 标签为 [<figure>](https://www.runoob.com/tags/tag-figure.html) 元素定义标题。

<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。

## 全局属性

<figcaption> 标签支持 [HTML 的全局属性](https://www.runoob.com/tags/ref-standardattributes.html)。

## 事件属性

<figcaption> 标签支持 [HTML 的事件属性](https://www.runoob.com/tags/ref-eventattributes.html)。

# **HTML <figure> 标签**

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<p>The Pulpit Rock is a massive cliff 604 metres (1982 feet) above Lysefjorden, opposite the Kjerag plateau, in Forsand, Ryfylke, Norway. The top of the cliff is approximately 25 by 25 metres (82 by 82 feet) square and almost flat, and is a famous tourist attraction in Norway.</p>

​

<figure>

<img src="img\_pulpit.jpg" tppabs="http://w3schools.com/tags/img\_pulpit.jpg" alt="The Pulpit Rock" width="304" height="228">

</figure>

​

</body>

</html>

## 标签定义及使用说明

<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，同时元素的位置相对于主内容是独立的。如果被删除，则不应对文档流产生影响。

## 提示和注释

**提示：**[<figcaption>](https://www.runoob.com/tags/tag-figcaption.html) 元素被用来为 <figure> 元素定义标题。

## 全局属性

<figure> 标签支持 [HTML 的全局属性](https://www.runoob.com/tags/ref-standardattributes.html)。

## 事件属性

<figure> 标签支持 [HTML 的事件属性](https://www.runoob.com/tags/ref-eventattributes.html)。

# **HTML <footer> 标签**

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<footer>

<p>Posted by: Hege Refsnes</p>

<p><time pubdate datetime="2012-03-01"></time></p>

</footer>

​

</body>

</html>

## 标签定义及使用说明

<footer> 标签定义文档或者文档的一部分区域的页脚。

<footer> 元素应该包含它所包含的元素的信息。

在典型情况下，该元素会包含文档创作者的姓名、文档的版权信息、使用条款的链接、联系信息等等。

在一个文档中，您可以定义多个 <footer> 元素。

## 提示和注释

**提示：**假如您使用 <footer> 元素来插入联系信息，应该在 <footer> 元素内使用 [<address>](https://www.runoob.com/tags/tag-address.html) 标签。

## 全局属性

<footer> 标签支持 [HTML 的全局属性](https://www.runoob.com/tags/ref-standardattributes.html)。

## 事件属性

<footer> 标签支持 [HTML 的事件属性](https://www.runoob.com/tags/ref-eventattributes.html)。

# **HTML <form> 标签**

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<form action="demo-form.php">

First name: <input type="text" name="FirstName" value="Mickey"><br>

Last name: <input type="text" name="LastName" value="Mouse"><br>

<input type="submit" value="提交">

</form>

​

<p>点击"提交"按钮，表单数据将被发送到服务器上的“demo-form.php”。</p>

​

</body>

</html>

## 标签定义及使用说明

<form> 标签用于创建供用户输入的 HTML 表单。

<form> 元素包含一个或多个如下的表单元素：

* [<input>](https://www.runoob.com/tags/tag-input.html)
* [<textarea>](https://www.runoob.com/tags/tag-textarea.html)
* [<button>](https://www.runoob.com/tags/tag-button.html)
* [<select>](https://www.runoob.com/tags/tag-select.html)
* [<option>](https://www.runoob.com/tags/tag-option.html)
* [<optgroup>](https://www.runoob.com/tags/tag-optgroup.html)
* [<fieldset>](https://www.runoob.com/tags/tag-fieldset.html)
* [<label>](https://www.runoob.com/tags/tag-label.html)

属性

|  |  |  |
| --- | --- | --- |
| 属性 | 值 | 描述 |
| Accept-charset | character\_set | 规定服务器可处理的表单数据字符集 |
| action | URL | 规定当提交表单时向何处发送表单数据 |
| autocomplete | on  off | 规定是否启用表单的自动完成功能 |
| enctype | application/x-www-form-urlencoded multipart/form-data  text/plain | 规定在向服务器发送表单数据之前如何对其进行编码（适用于method=“post”的情况） |
| method | get  post | 规定用于发送表单数据的HTTP方法 |
| name | text | 规定表单的名称 |
| novalidate | novalidate | 如果使用该属性，则提交表单时不进行验证 |
| target | \_blank  \_self  \_parent  \_top | 规定在何处打开action URL |

## 全局属性

<form> 标签支持 [HTML 的全局属性](https://www.runoob.com/tags/ref-standardattributes.html)。

## 事件属性

<form> 标签支持 [HTML 的事件属性](https://www.runoob.com/tags/ref-eventattributes.html)。

# **HTML <head> 标签**

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>文档标题</title>

</head>

​

<body>

文档内容......

</body>

​

</html>

## 标签定义及使用说明

<head> 元素是所有头部元素的容器。

<head> 元素必须包含文档的标题（title），可以包含脚本、样式、meta 信息 以及其他更多的信息。

以下列出的元素能被用在 <head> 元素内部：

* [<title>](https://www.runoob.com/tags/tag-title.html) （在头部中，这个元素是必需的）
* [<style>](https://www.runoob.com/tags/tag-style.html)
* [<base>](https://www.runoob.com/tags/tag-base.html)
* [<link>](https://www.runoob.com/tags/tag-link.html)
* [<meta>](https://www.runoob.com/tags/tag-meta.html)
* [<script>](https://www.runoob.com/tags/tag-script.html)
* [<noscript>](https://www.runoob.com/tags/tag-noscript.html)

## 全局属性

<head> 标签支持 [HTML 的全局属性](https://www.runoob.com/tags/ref-standardattributes.html)。

# **HTML <header> 标签**

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<article>

<header>

<h1>Internet Explorer 9</h1>

<p><time pubdate datetime="2011-03-15"></time></p>

</header>

<p> Windows Internet Explorer 9(缩写为 IE9 )是在2011年3月14日21:00发布的</p>

</article>

​

</body>

</html>

## 标签定义及使用说明

<header> 标签定义文档或者文档的一部分区域的页眉。

<header> 元素应该作为介绍内容或者导航链接栏的容器。

在一个文档中，您可以定义多个 <header> 元素。

**注释：**<header> 标签不能被放在 <footer>、<address> 或者另一个 <header> 元素内部。

## 全局属性

<header> 标签支持 [HTML 的全局属性](https://www.runoob.com/tags/ref-standardattributes.html)。

## 事件属性

<header> 标签支持 [HTML 的事件属性](https://www.runoob.com/tags/ref-eventattributes.html)。

# **HTML <hgroup> 标签**

实例：

<!DOCTYPE html>

<html>

<body>

​

<hgroup>

<h1>Welcome to my WWF</h1>

<h2>For a living planet</h2>

</hgroup>

​

<p>The rest of the content...</p>

​

</body>

</html>

## 标签定义及使用说明

<hgroup> 标签被用来对标题元素进行分组。

当标题有多个层级（副标题）时，<hgroup> 元素被用来对一系列 [<h1>](https://www.runoob.com/tags/tag-hn.html) - [<h6>](https://www.runoob.com/tags/tag-hn.html) 元素进行分组。

## 全局属性

<hgroup> 标签支持 [HTML 的全局属性](https://www.runoob.com/tags/ref-standardattributes.html)。

## 事件属性

<hgroup> 标签支持 [HTML 的事件属性](https://www.runoob.com/tags/ref-eventattributes.html)。

# **HTML <h1> - <h6> 标签**

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<h1>这是标题 1</h1>

<h2>这是标题 2</h2>

<h3>这是标题 3</h3>

<h4>这是标题 4</h4>

<h5>这是标题 5</h5>

<h6>这是标题 6</h6>

​

</body>

</html>

​

## 标签定义及使用说明

<h1> - <h6> 标签被用来定义 HTML 标题。

<h1> 定义重要等级最高的标题。<h6> 定义重要等级最低的标题。

## 全局属性

<h1> - <h6> 标签支持 [HTML 的全局属性](https://www.runoob.com/tags/ref-standardattributes.html)。

## 事件属性

<h1> - <h6> 标签支持 [HTML 的事件属性](https://www.runoob.com/tags/ref-eventattributes.html)。

# **HTML <hr> 标签**

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<h1>HTML</h1>

<p>HTML 是用于描述 web 页面的一种语言。</p>

​

<hr>

​

<h1>CSS</h1>

<p>CSS 定义如何显示 HTML 元素。</p>

​

</body>

</html>

## 标签定义及使用说明

<hr> 标签定义 HTML 页面中的主题变化（比如话题的转移），并显示为一条水平线。

<hr> 元素被用来分隔 HTML 页面中的内容（或者定义一个变化）。

## 全局属性

<hr> 标签支持 [HTML 的全局属性](https://www.runoob.com/tags/ref-standardattributes.html)。

## 事件属性

<hr> 标签支持 [HTML 的事件属性](https://www.runoob.com/tags/ref-eventattributes.html)。

**HTML<html>标签**

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>文档标题</title>

</head>

​

<body>

文档内容......

</body>

​

</html>

**标签定义及使用说明：**

<html>标签告知浏览器这是一个HTML文档

<html>标签是HTML文档中最外层的元素

<html>标签是所有其他HTML元素（除了<!DOCTYPE>标签）的容器

**属性**

|  |  |  |
| --- | --- | --- |
| 属性 | 值 | 描述 |
| manifest | URL | 定义一个URL，在这个URL上描述了文档的缓存信息 |
| xmlns | http://www.w3.org/1999/xhtml | HTML不支持，只有XHTML支持。规定XML的namespace属性（如果需要内容符合XHTML,则使用这个属性） |

**全局属性**

<html>标签支持HTML的全局属性

# **HTML <i> 标签**

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<p>He named his car <i>The lightning</i>, because it was very fast.</p>

​

</body>

</html>

## 标签定义及使用说明

<i> 定义与文本中其余部分不同的部分，并把这部分文本呈现为斜体文本。

<i> 标签被用来表示科技术语、其他语种的成语俗语、想法、宇宙飞船的名字等等。

在没有其他适当语义的元素可以使用时，请使用 <i> 元素。其他语义的元素如下：

* [<em>](https://www.runoob.com/tags/tag-em.html) （被强调的文本）
* [<strong>](https://www.runoob.com/tags/tag-strong.html) （重要的文本）
* [<mark>](https://www.runoob.com/tags/tag-mark.html) （被标记的/高亮显示的文本）
* [<cite>](https://www.runoob.com/tags/tag-cite.html) （作品的标题）
* [<dfn>](https://www.runoob.com/tags/tag-dfn.html) （一个定义项目）

## 全局属性

<i> 标签支持 [HTML 的全局属性](https://www.runoob.com/tags/ref-standardattributes.html)。

## 事件属性

<i> 标签支持 [HTML 的事件属性](https://www.runoob.com/tags/ref-eventattributes.html)。

# **HTML <iframe> 标签**

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<iframe src="//www.runoob.com">

<p>您的浏览器不支持 iframe 标签。</p>

</iframe>

​

</body>

</html>

## 标签定义及使用说明

<iframe> 标签规定一个内联框架。

一个内联框架被用来在当前 HTML 文档中嵌入另一个文档。

## 提示和注释

**提示：**您可以把需要的文本放置在 <iframe> 和 </iframe> 之间，这样就可以应对不支持 <iframe> 的浏览器。

**提示：**使用 CSS 为 <iframe> （包括滚动条）定义样式。

属性

|  |  |  |
| --- | --- | --- |
| 属性 | 值 | 描述 |
| height | pixels | 规定<iframe>的高度 |
| name | name | 规定<iframe>的名称 |
| sandbox | “”  Allow\_forms  Allow\_same-origin  Allow-scripts  Allow-top-navigation | 对<iframe>的内容定义一系列额外的限制 |
| seamless | seamless | 规定<iframe>看起来像是父文档中的一部分 |
| src | URL | 规定在<iframe>中显示的文档的URL |
| srcdoc | HTML\_code | 规定页面中的HTML内容显示在<iframe>中 |
| width | pixels | 规定<iframe>的宽度 |

## 全局属性

<iframe> 标签支持 [HTML 的全局属性](https://www.runoob.com/tags/ref-standardattributes.html)。

## 事件属性

<iframe> 标签支持 [HTML 的事件属性](https://www.runoob.com/tags/ref-eventattributes.html)。

# **HTML <img> 标签**

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<img src="smiley-2.gif" alt="Smiley face" width="42" height="42">

​

</body>

</html>

## 标签定义及使用说明

<img> 标签定义 HTML 页面中的图像。

<img> 标签有两个必需的属性：src 和 alt。

**注释：**从技术上讲，图像并不会插入 HTML 页面中，而是链接到 HTML 页面上。<img> 标签的作用是为被引用的图像创建占位符。

**提示：**通过在 <a> 标签中嵌套 <img> 标签，给图像添加到另一个文档的链接。

|  |  |  |
| --- | --- | --- |
| 属性 | 值 | 描述 |
| alt | text | 规定图像的替代文本 |
| crossrigin | anonymous  use-credentials | 设置图像的跨域属性 |
| height | pixels | 规定图像的高度 |
| ismap | ismap | 将图像规定为服务器端图像映射 |
| src | URL | 规定显示图像的URL |
| usemap | #mapname | 将图像定义为客户端图像映射 |
| width | pixels | 规定图像的宽度 |

## 全局属性

<img> 标签支持 [HTML 的全局属性](https://www.runoob.com/tags/ref-standardattributes.html)。

## 事件属性

<img> 标签支持 [HTML 的事件属性](https://www.runoob.com/tags/ref-eventattributes.html)。

# **HTML <input> 标签**

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<form action="demo-form.php">

First name: <input type="text" name="FirstName" value="Mickey"><br>

Last name: <input type="text" name="LastName" value="Mouse"><br>

<input type="submit" value="提交">

</form>

​

<p>点击"提交"按钮，表单数据将被发送到服务器上的“demo-form.php”。</p>

​

</body>

</html>

## 标签定义及使用说明

<input> 标签规定了用户可以在其中输入数据的输入字段。

<input> 元素在 <form> 元素中使用，用来声明允许用户输入数据的 input 控件。

输入字段可通过多种方式改变，取决于 type 属性。

## 提示和注释

**注意:**<input> 元素是空的,它只包含标签属性。

**提示:** 你可以使用 [<label>](https://www.runoob.com/tags/tag-label.html) 元素来定义 <input> 元素的标注。

**属性**

|  |  |  |
| --- | --- | --- |
| 属性 | 值 | 描述 |
| accept | audio/\*video/\*image/\*MIME\_type | 规定通过文件上传来提交的文件的类型（只针对type=”file”） |
| alt | text | 定义图像输入的替代文本（只针对type=”image”） |
| autocomplete | on off | autocomplete属性规定<input>元素输入字段是否应该启用自动完成功能 |
| autofocus | autofocus | 属性规定当页面加载时<input>元素应该自动获得焦点 |
| checked | checked | checked属性规定在页面加载时应该被预先选定的<input>元素（只针对type=”checkbox”或者type=”radio”） |
| disabled | disabled | disabled属性规定应该禁用的<input>元素 |
| form | form\_id | form属性规定<input>元素所属的一个或多个表单 |
| formaction | URL | 属性规定当表单提交时处理输入控件的文件的URL（只针对type=”submit”和type=”image”） |
| formenctype | application/x-www-form-urlencoded multipart/form-data text/plain | 属性规定当表单数据提交到服务器时如何编码（只适合type=”submit”和type=”image”） |
| formmethod | get post | 定义发送表单数据到action URL的HTTP方法（只适合type=”submit”和type=”image”） |
| formnovalidate | formnovalidate | formnivalidate属性覆盖<form>元素的novalidate属性 |
| formtarget | \_blank  \_self  \_parent  \_top  framename | 规定表示提交表单后在哪里显示接受到响应的名称或关键词（只适合type=”submit”和type=”image”） |
| height | pixels | 规定<input>元素的高度（只针对type=”image”） |
| list | datalist\_id | 属性引用<datalist>元素，其中包含<input>元素的预定义选项 |
| max | number date | 属性规定<input>元素的最大值 |
| maxlength | number | 属性规定<input>元素中允许的最大字符数 |
| min | number date | 属性规定<input>元素的最小值 |
| multiple | multiple | 属性规定允许用户输入到<input>元素的多个值 |
| name | text | name属性规定<input>元素的名称 |
| pattern | regexp | pattern属性规定用于验证<input>元素的值的正则表达式 |
| placeholder | text | placeholder属性规定可描述输入<input>字段预期值的简短的提示信息 |
| readonly | readonly | readonly属性规定输入字段是只读的 |
| required | required | 属性规定必需在提交表单之前填写输入字段 |
| size | number | size属性规定字符数计的<input>元素的可见宽度 |
| src | URL | src属性规定显示为提交按钮的图像的URL（只针对type=”image”） |
| step | number | step属性规定<input>元素的合法数字间隔 |
| type | button  checkbox  color  date  datetime  datetime-local  email  file  hidden  image  month  number  password  radio  range  reset  search  submit  tel  text  time  url  week | type属性规定要显示的<input>元素的类型 |
| value | text | 指定<input>元素value的值 |
| width | pixels | width属性规定<input>元素的宽度（只针对type=”image”） |

## 全局属性

<input> 标签支持全局属性，查看完整属性表 [HTML全局属性](https://www.runoob.com/tags/ref-standardattributes.html)。

## 事件属性

<input> 标签支持所有 [HTML事件属性](https://www.runoob.com/tags/ref-eventattributes.html)。

# **HTML <ins> 标签**

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<p>My favorite color is <del>blue</del> <ins>red</ins>!</p>

​

</body>

</html>

## 标签定义及使用说明

<ins> 标签定义已经被插入文档中的文本。

## 提示和注释

**提示：**您也可以看看标记已删除文本的 [<del>](https://www.runoob.com/tags/tag-del.html) 标签。

**提示：**<del> 和 <ins> 一起使用，描述文档中的更新和修正。浏览器通常会在已删除文本上添加一条删除线，在新插入文本下添加一条下划线。

**属性**

|  |  |  |
| --- | --- | --- |
| 属性 | 值 | 描述 |
| cite | URL | 规定一个文档的URL,该文档解释了文本被插入的原因 |
| datetime | YYYY-MM-DDThh:mm:ssTZD | 规定文本被插入的日期和时间 |

## 全局属性

<ins> 标签支持 [HTML 的全局属性](https://www.runoob.com/tags/ref-standardattributes.html)。

## 事件属性

<ins> 标签支持 [HTML 的事件属性](https://www.runoob.com/tags/ref-eventattributes.html)。

# **HTML <kbd> 标签**

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<em>强调文本</em><br>

<strong>加粗文本</strong><br>

<dfn>定义项目</dfn><br>

<code>一段电脑代码</code><br>

<samp>计算机样本</samp><br>

<kbd>键盘输入</kbd><br>

<var>变量</var>

​

</body>

</html>

## 标签定义及使用说明

<kbd> 标签定义键盘文本。

|  |  |
| --- | --- |
| 标签 | 描述 |
| <em> | 呈现为被强调的文本 |
| <strong> | 定义重要的文本 |
| <dfn> | 定义一个定义项目 |
| <code> | 定义计算机代码文本 |
| <samp> | 定义样本文本 |
| <kbd> | 定义键盘文本，表示文本是从键盘上键入的，经常用在与计算机相关的文档或手册中 |
| <var> | 定义变量，可以将此标签与<pre>及<code>标签配合使用 |

## 全局属性

<kbd> 标签支持全局属性，查看完整属性表 [HTML全局属性](https://www.runoob.com/tags/ref-standardattributes.html)。

## 事件属性

<kbd> 标签支持所有 [HTML事件属性](https://www.runoob.com/tags/ref-eventattributes.html)。

# **HTML <label> 标签**

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<p>点击其中一个文本标签选中选项：</p>

​

<form action="demo\_form.php">

<label for="male">Male</label>

<input type="radio" name="sex" id="male" value="male"><br>

<label for="female">Female</label>

<input type="radio" name="sex" id="female" value="female"><br><br>

<input type="submit" value="提交">

</form>

​

</body>

</html>

## 标签定义及使用说明

<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。

## 提示和注释

**提示:**"for" 属性可把 label 绑定到另外一个元素。请把 "for" 属性的值设置为相关元素的 id 属性的值。

属性

|  |  |  |
| --- | --- | --- |
| 属性 | 值 | 描述 |
| for | Element\_id | 规定label与哪个表单元素绑定 |
| form | Form\_id | 规定label字段所属的一个或多个表单 |

## 全局属性

<label> 标签支持全局属性，查看完整属性表 [HTML全局属性](https://www.runoob.com/tags/ref-standardattributes.html)。

## 事件属性

<label> 标签支持所有 [HTML事件属性](https://www.runoob.com/tags/ref-eventattributes.html)。

# **HTML <legend> 标签**

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<form>

<fieldset>

<legend>Personalia:</legend>

Name: <input type="text"><br>

Email: <input type="text"><br>

Date of birth: <input type="text">

</fieldset>

</form>

​

</body>

</html>

## 标签定义及使用说明

The <legend> 元素为 [<fieldset>](https://www.runoob.com/tags/tag-fieldset.html)元素定义标题。

## 全局属性

<legend> 标签支持全局属性，查看完整属性表 [HTML全局属性](https://www.runoob.com/tags/ref-standardattributes.html)。

## 事件属性

<legend> 标签支持所有 [HTML事件属性](https://www.runoob.com/tags/ref-eventattributes.html)。

# **HTML <li> 标签**

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<p>有序列表:</p>

<ol>

<li>Coffee</li>

<li>Tea</li>

<li>Milk</li>

</ol>

​

<p>无序列表:</p>

<ul>

<li>Coffee</li>

<li>Tea</li>

<li>Milk</li>

</ul>

​

</body>

</html>

## 标签定义及使用说明

<li> 标签定义列表项目。

<li> 标签可用在有序列表（[<ol>](https://www.runoob.com/tags/tag-ol.html)）、无序列表（[<ul>](https://www.runoob.com/tags/tag-ul.html)）和菜单列表（[<menu>](https://www.runoob.com/tags/tag-menu.html)）中。

## 提示和注释

**提示:**请使用 CSS 来定义列表和列表项目的类型。

|  |  |  |
| --- | --- | --- |
| 属性 | 值 | 描述 |
| value | number | 不赞成使用，请使用样式取代他，规定列表项目的数字 |

## 全局属性

<li> 标签支持全局属性，查看完整属性表 [HTML 全局属性](https://www.runoob.com/tags/ref-standardattributes.html)。

## 事件属性

<li> 标签支持所有 [HTML 事件属性](https://www.runoob.com/tags/ref-eventattributes.html)。

# **HTML <link> 标签**

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

<link rel="stylesheet" type="text/css" href="styles.css">

</head>

​

<body>

<h1>我是通过样式文件 styles.css 渲染后显示的。</h1>

<p>我也是。</p>

</body>

​

</html>

## 标签定义及使用说明

<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。

**注意：** link 元素是空元素，它仅包含属性。

**注意：** 此元素只能存在于 head 部分，不过它可出现任何次数。

**属性**

|  |  |  |
| --- | --- | --- |
| 属性 | 值 | 描述 |
| href | URL | 定义被链接文档的位置 |
| hreflang | language\_code | 定义被链接文档中文本的语言 |
| media | media\_query | 规定被链接文档将显示在什么设备上 |
| rel | alternate  archives  author  bookmark  external  first  help  icon  last  license  next  nofollow  noreferrer  pingback  prefetch  prev  search  sidebar  stylesheet  tag  up | 必需，定义当前文档与被链接文档之间的关系 |
| sizes | HeightxWidth  any | 定义了链接属性大小，只对属性rel=”icon”起作用 |
| type | MIME\_type | 规定被链接文档的MIME类型 |

## 全局属性

<link> 标签支持全局属性，查看完整属性表 [HTML 全局属性](https://www.runoob.com/tags/ref-standardattributes.html)。

## 事件属性

<link> 标签支持所有 [HTML 事件属性](https://www.runoob.com/tags/ref-eventattributes.html)。

# **HTML <map> 标签**

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<p>点击太阳或其他行星，注意变化：</p>

​

<img src="planets.gif" width="145" height="126" alt="Planets" usemap="#planetmap">

​

<map name="planetmap">

<area shape="rect" coords="0,0,82,126" alt="Sun" href="sun.htm">

<area shape="circle" coords="90,58,3" alt="Mercury" href="mercur.htm">

<area shape="circle" coords="124,58,8" alt="Venus" href="venus.htm">

</map>

​

</body>

</html>

## 标签定义及使用说明

<map> 标签用于客户端图像映射。图像映射指带有可点击区域的一幅图像。

<img>中的 usemap 属性可引用 <map> 中的 id 或 name 属性（取决于浏览器），所以我们应同时向 <map> 添加 id 和 name 属性。

area 元素永远嵌套在 map 元素内部。area 元素可定义图像映射中的区域。

**属性**

|  |  |  |
| --- | --- | --- |
| 属性 | 值 | 描述 |
| name | mapname | 必需，为image-map规定的名称 |

## 全局属性

<map> 标签支持全局属性，查看完整属性表 [HTML全局属性](https://www.runoob.com/tags/ref-standardattributes.html)。

## 事件属性

<map> 标签支持所有 [HTML事件属性](https://www.runoob.com/tags/ref-eventattributes.html)。

# **HTML5 <mark> 标签**

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<p>Do not forget to buy <mark>milk</mark> today.</p>

​

</body>

</html>

## 标签定义及使用说明

<mark> 标签定义带有记号的文本。

请在需要突出显示文本时使用 <mark> 标签。

## 全局属性

<mark> 标签支持全局属性，查看完整属性表 [HTML全局属性](https://www.runoob.com/tags/ref-standardattributes.html)。

## 事件属性

<mark> 标签支持所有 [HTML事件属性](https://www.runoob.com/tags/ref-eventattributes.html)。

# **HTML <menu> 标签**

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<menu type="toolbar">

<li>

<menu label="File">

<button type="button" onclick="file\_new()">New...</button>

<button type="button" onclick="file\_open()">Open...</button>

<button type="button" onclick="file\_save()">Save</button>

</menu>

</li>

<li>

<menu label="Edit">

<button type="button" onclick="edit\_cut()">Cut</button>

<button type="button" onclick="edit\_copy()">Copy</button>

<button type="button" onclick="edit\_paste()">Paste</button>

</menu>

</li>

</menu>

​

<p><b>注意：</b>目前主流浏览器并不支持 menu 标签。</p>

​

</body>

</html>

## 标签定义及使用说明

<menu> 标签定义了一个命令列表或菜单。

<menu> 标签通常用于文本菜单，工具条和命令列表选项。

## 提示和注释

**提示:** 使用 CSS 来定义菜单列表样式。

**属性**

|  |  |  |
| --- | --- | --- |
| 属性 | 值 | 描述 |
| label | text | 描述菜单项的标记 |
| type | context  toolbar  list | 描述显示菜单类型。默认为“list” |

## 全局属性

<menu> 标签支持全局属性，查看完整属性表 [HTML全局属性](https://www.runoob.com/tags/ref-standardattributes.html)。

## 事件属性

<menu> 标签支持所有 [HTML事件属性](https://www.runoob.com/tags/ref-eventattributes.html)。

# **HTML <meta> 标签**

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

<meta name="description" content="免费在线教程">

<meta name="keywords" content="HTML,CSS,XML,JavaScript">

<meta name="author" content="runoob">

</head>

<body>

​

<p>所有 meta 标签显示在 head 部分...</p>

​

</body>

</html>

## 标签定义及使用说明

元数据（Metadata）是数据的数据信息。

<meta> 标签提供了 HTML 文档的元数据。元数据不会显示在客户端，但是会被浏览器解析。

META元素通常用于指定网页的描述，关键词，文件的最后修改时间，作者及其他元数据。

元数据可以被使用浏览器（如何显示内容或重新加载页面），搜索引擎（关键词），或其他 Web 服务调用。

## 提示和注释

**注意：**<meta> 标签通常位于 <head> 区域内。

**注意：** 元数据通常以 名称/值 对出现。

**注意：** 如果没有提供 name 属性，那么名称/值对中的名称会采用 http-equiv 属性的值。

## 实例

**实例 1 - 定义文档关键词，用于搜索引擎：**

<meta name="keywords" content="HTML, CSS, XML, XHTML, JavaScript">

**实例 2 - 定义web页面描述：**

<meta name="description" content="Free Web tutorials on HTML and CSS">

**实例 3 - 定义页面作者：**

<meta name="author" content="Hege Refsnes">

**实例 4 - 每30秒刷新页面：**

<meta http-equiv="refresh" content="30">

**属性**

|  |  |  |
| --- | --- | --- |
| 属性 | 值 | 描述 |
| charset | character\_set | 定义文档的字符编码 |
| content | text | 定义与http-equiv或name属性相关的元信息 |
| http-equiv | content-type  default-style  refresh | 把content属性关联到HTTP头部 |
| name | application-name  author  description  generator  keywords | 把content属性关联到一个名称 |

## 事件属性

<p> 标签支持所有 [HTML事件属性](https://www.runoob.com/tags/ref-eventattributes.html)。

# **HTML <meter> 标签**

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<p>展示给定的数据范围：</p>

<meter value="2" min="0" max="10">2 out of 10</meter><br>

<meter value="0.6">60%</meter>

​

<p><strong>注意：</strong> IE 浏览器不支持 meter 标签。</p>

​

</body>

</html>

## 标签定义及使用说明

<meter> 标签定义度量衡。仅用于已知最大和最小值的度量。

比如：磁盘使用情况，查询结果的相关性等。

**注意：** <meter> 不能作为一个进度条来使用， 进度条 [<progress>](https://www.runoob.com/tags/tag-progress.html) 标签。

属性

|  |  |  |
| --- | --- | --- |
| 属性 | 值 | 描述 |
| form | form-id | 规定<meter>元素所属的一个或多个表单 |
| high | number | 规定被界定为高的值的范围 |
| low | number | 规定被界定为低的值的范围 |
| max | number | 规定范围的最大值 |
| min | number | 规定范围的最小值 |
| optimum | number | 规定度量的最优值 |
| value | number | 必需，规定度量的当前值 |

## 全局属性

<meter> 标签支持全局属性，查看完整属性表 [HTML全局属性](https://www.runoob.com/tags/ref-standardattributes.html)。

## 事件属性

<meter> 标签支持所有 [HTML事件属性](https://www.runoob.com/tags/ref-eventattributes.html)。

# **HTML <nav> 标签**

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<nav>

<a href="/html/">HTML</a> |

<a href="/css/">CSS</a> |

<a href="/js/">JavaScript</a> |

<a href="/jquery/">jQuery</a>

</nav>

​

</body>

</html>

## 标签定义及使用说明

<nav> 标签定义导航链接的部分。

并不是所有的 HTML 文档都要使用到 <nav> 元素。<nav> 元素只是作为标注一个导航链接的区域。

在不同设备上（手机或者PC）可以制定导航链接是否显示，以适应不同屏幕的需求。

## 全局属性

<nav> 标签支持全局属性，查看完整属性表 [HTML全局属性](https://www.runoob.com/tags/ref-standardattributes.html)。

## 事件属性

<nav> 标签支持所有 [HTML事件属性](https://www.runoob.com/tags/ref-eventattributes.html)。

# **HTML <noscript> 标签**

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

<script>

document.write("Hello World!")

</script>

<noscript>抱歉，你的浏览器不支持 JavaScript!</noscript>

<p>不支持 JavaScript 的浏览器会使用 &lt;noscript&gt; 元素中定义的内容（文本）来替代。</p>

</body>

</html>

## 标签定义及使用说明

noscript 元素用来定义在脚本未被执行时的替代内容（文本）。

此标签可被用于可识别 <noscript> 标签但无法支持其中的脚本的浏览器。

## 提示和注释

**提示：**如果浏览器支持脚本，那么它不会显示出 noscript 元素中的文本。

**注释：**无法识别 <script> 标签的浏览器会把标签的内容显示到页面上。为了避免浏览器这样做，您应当在注释标签中隐藏脚本。老式的（无法识别 <script> 标签的）浏览器会忽略注释，这样就不会把标签的内容写到页面上，而新式的浏览器则懂得执行这些脚本，即使它们被包围在注释标签中！

<script>  
<!--  
function displayMsg()  
{  
alert("Hello World!")  
}  
//-->  
</script>

## 全局属性

<noscript> 标签支持全局属性，查看完整属性表 [HTML全局属性](https://www.runoob.com/tags/ref-standardattributes.html)。

# **HTML <object> 标签**

实例:

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<object width="400" height="400" data="helloworld.swf" >

</object>

</body>

</html>

## 标签定义及使用说明

定义一个嵌入的对象。请使用此元素向您的 XHTML 页面添加多媒体。此元素允许您规定插入 HTML 文档中的对象的数据和参数，以及可用来显示和操作数据的代码。

<object> 标签用于包含对象，比如图像、音频、视频、Java applets、ActiveX、PDF 以及 Flash。

object 的初衷是取代 img 和 applet 元素。不过由于漏洞以及缺乏浏览器支持，这一点并未实现。

浏览器的对象支持有赖于对象类型。不幸的是，主流浏览器都使用不同的代码来加载相同的对象类型。

而幸运的是，object 对象提供了解决方案。如果未显示 object 元素，就会执行位于 <object> 和 </object> 之间的代码。通过这种方式，我们能够嵌套多个 object 元素（每个对应一个浏览器）。

属性

|  |  |  |
| --- | --- | --- |
| 属性 | 值 | 描述 |
| data | URL | 规定对象使用的资源的URL |
| form | form\_id | 规定对象所属的一个或多个表单 |
| height | pixels | 规定对象的高度 |
| name | name | 对对象规定名称 |
| type | MIME\_type | 规定data属性中规定的数据的MIME类型 |
| usemap | #mapname | 规定与对象一同使用的客户端图像映射的名称 |
| width | pixels | 规定对象的宽度 |

## 全局属性

<object> 标签支持全局属性，查看完整属性表 [HTML全局属性](https://www.runoob.com/tags/ref-standardattributes.html)。

## 事件属性

<object> 标签支持所有 [HTML事件属性](https://www.runoob.com/tags/ref-eventattributes.html)。

# **HTML <ol> 标签**

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

<ol>

<li>Coffee</li>

<li>Tea</li>

<li>Milk</li>

</ol>

<ol start="50">

<li>Coffee</li>

<li>Tea</li>

<li>Milk</li>

</ol>

</body>

</html>

## 标签定义及使用说明

<ol> 标签定义了一个有序列表. 列表排序以数字来显示。

使用[<li>](https://www.runoob.com/tags/tag-li.html) 标签来定义列表选项。

## 提示和注释

**提示：** 如果需要无序列表，请使用 [<ul>](https://www.runoob.com/tags/tag-ul.html) 标签。

**提示：**使用 CSS 来定义列表样式。

属性

|  |  |  |
| --- | --- | --- |
| 属性 | 值 | 描述 |
| reversed | reversed | 指定列表倒序（9,8,7...） |

## 全局属性

<ol> 标签支持全局属性，查看完整属性表 [HTML全局属性](https://www.runoob.com/tags/ref-standardattributes.html)。

## 事件属性

<ol> 标签支持所有 [HTML事件属性](https://www.runoob.com/tags/ref-eventattributes.html)。

# HTML <optgroup> 标签

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<select>

<optgroup label="Swedish Cars">

<option value="volvo">Volvo</option>

<option value="saab">Saab</option>

</optgroup>

<optgroup label="German Cars">

<option value="mercedes">Mercedes</option>

<option value="audi">Audi</option>

</optgroup>

</select>

</body>

</html>

## 标签定义及使用说明

<optgroup> 标签经常用于把相关的选项组合在一起。

如果你有很多的选项组合, 你可以使用<optgroup> 标签能够很简单的将相关选项组合在一起。

属性

|  |  |  |
| --- | --- | --- |
| 属性 | 值 | 描述 |
| disabled | disabled | 规定禁用该选项组 |
| label | text | 为选项组规定描述 |

## 全局属性

<optgroup> 标签支持全局属性，查看完整属性表 [HTML全局属性](https://www.runoob.com/tags/ref-standardattributes.html)。

## 事件属性

<optgroup> 标签支持所有 [HTML事件属性](https://www.runoob.com/tags/ref-eventattributes.html)。

# **HTML <option> 标签**

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<select>

<option value="volvo">Volvo</option>

<option value="saab">Saab</option>

<option value="opel">Opel</option>

<option value="audi">Audi</option>

</select>

</body>

</html>

## 标签定义及使用说明

The <option> 标签定义下拉列表中的一个选项（一个条目）。

<option> 标签中的内容作为 [<select>](https://www.runoob.com/tags/tag-select.html) 或者[<datalist>](https://www.runoob.com/tags/tag-datalist.html) 一个元素使用。

## 提示和注释

注释：<option> 标签可以在不带有任何属性的情况下使用，但是您通常需要使用 value 属性，此属性会指示出被送往服务器的内容。

注释：请与 select 元素配合使用此标签，否则这个标签是没有意义的。

提示：如果列表选项很多，可以使用 <optgroup> 标签对相关选项进行组合。

属性

|  |  |  |
| --- | --- | --- |
| 属性 | 值 | 描述 |
| disabled | disabled | 规定此选项应在首次加载时被禁用 |
| label | text | 定义当使用<optgroup>时所使用的标注 |
| selected | selected | 规定选项（在首次显示在列表中时）表现为选中状态 |
| value | text | 定义送往服务器的选项值 |

## 全局属性

<option> 标签支持全局属性，查看完整属性表 [HTML全局属性](https://www.runoob.com/tags/ref-standardattributes.html)。

## 事件属性

<option> 标签支持所有 [HTML事件属性](https://www.runoob.com/tags/ref-eventattributes.html)。

# **HTML <output> 标签**

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<form oninput="x.value=parseInt(a.value)+parseInt(b.value)">0

<input type="range" id="a" value="50">100

+<input type="number" id="b" value="50">

=<output name="x" for="a b"></output>

</form>

​

<p><strong>注意:</strong> Internet Explorer 不支持 output 标签。</p>

​

</body>

</html>

## 标签定义及使用说明

<output> 标签作为计算结果输出显示(比如执行脚本的输出)。

属性

|  |  |  |
| --- | --- | --- |
| 属性 | 值 | 描述 |
| for | element\_id | 描述计算机中使用的元素与计算结果之间的关系 |
| form | form\_id | 定义输入字段所属的一个或多个表单 |
| name | name | 定义对象的唯一名称（表单提交时使用） |

## 全局属性

<output> 标签支持全局属性，查看完整属性表 [HTML全局属性](https://www.runoob.com/tags/ref-standardattributes.html)。

## 事件属性

<output> 标签支持所有 [HTML事件属性](https://www.runoob.com/tags/ref-eventattributes.html)。

# **HTML <p> 标签**

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<p>这是一个段落。</p>

<p>这是一个段落。</p>

<p>这是一个段落。</p>

​

</body>

</html>

## 标签定义及使用说明

<p> 标签定义段落。

<p>元素会自动在其前后创建一些空白。浏览器会自动添加这些空间，您也可以在样式表中规定。

## 全局属性

<p> 标签支持全局属性，查看完整属性表 [HTML全局属性](https://www.runoob.com/tags/ref-standardattributes.html)。

## 事件属性

<p> 标签支持所有 [HTML事件属性](https://www.runoob.com/tags/ref-eventattributes.html)。

# **HTML <param> 标签**

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<p><b>注意：</b> Internet Explorer 8和早期版本,不支持 .wav 文件。尝试将文件重命名为　"horse.mp3" 来检验该示例在 IE 版本中的使用。</p>

​

<object data="horse.wav">

<param name="autoplay" value="true">

</object>

​

</body>

</html>

## 标签定义及使用说明

<param>元素允许您为插入 XHTML 文档的对象规定 run-time 设置，也就是说，此标签可为包含它的 [<object>](https://www.runoob.com/tags/tag-object.html" \o "HTML <object> 标签) 或者 [<applet>](https://www.runoob.com/tags/tag-applet.html" \o "HTML <applet> 标签) 标签提供参数。

属性

|  |  |  |
| --- | --- | --- |
| 属性 | 值 | 描述 |
| name | name | 定义参数的名称（用在脚本中） |
| value | value | 描述参数值 |

## 全局属性

<param> 标签支持全局属性，查看完整属性表 [HTML全局属性](https://www.runoob.com/tags/ref-standardattributes.html)。

## 事件属性

<param> 标签支持所有 [HTML事件属性](https://www.runoob.com/tags/ref-eventattributes.html)。

# **HTML <pre> 标签**

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<pre>

此例演示如何使用 pre 标签

对空行和 空格

进行控制

</pre>

​

</body>

</html>

## 标签定义及使用说明

<pre> 标签可定义预格式化的文本。

被包围在 <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。

## 提示和注释

**提示:** <pre> 标签的一个常见应用就是用来表示计算机的源代码。

## 全局属性

<pre> 标签支持全局属性，查看完整属性表 [HTML全局属性](https://www.runoob.com/tags/ref-standardattributes.html)。

## 事件属性

<pre> 标签支持所有 [HTML事件属性](https://www.runoob.com/tags/ref-eventattributes.html)。

# **HTML5 <progress> 标签**

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

下载进度：

<progress value="22" max="100">

</progress>

​

<p><strong>注意：</strong> IE 9 或者更早版本的 IE 浏览器不支持 progress 标签。</p>

​

</body>

</html>

## 标签定义及使用说明

<progress> 标签定义运行中的任务进度（进程）。

## 提示和注释

**提示：**请将 <progress> 标签与 JavaScript 一起使用来显示任务的进度。

**注释：**<progress> 标签不适合用来表示度量衡（例如，磁盘空间使用情况或相关的查询结果）。表示度量衡，请使用 [<meter>](https://www.runoob.com/tags/tag-meter.html) 标签代替。

属性

|  |  |  |
| --- | --- | --- |
| 属性 | 值 | 描述 |
| max | number | 规定需要完成的值 |
| value | number | 规定进程的当前值 |

## 全局属性

<progress> 标签支持 [HTML 的全局属性](https://www.runoob.com/tags/ref-standardattributes.html)。

## 事件属性

<progress> 标签支持 [HTML 的事件属性](https://www.runoob.com/tags/ref-eventattributes.html)。

# **HTML <q> 标签**

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<p>WWF's goal is to:

<q>Build a future where people live in harmony with nature.</q>

We hope they succeed.</p>

​

</body>

</html>

## 标签定义及使用说明

<q> 标签定义一个短的引用。

浏览器经常会在这种引用的周围插入引号。

## 提示和注释

**提示：**请使用 [<blockquote>](https://www.runoob.com/tags/tag-blockquote.html) 来标记摘自另一个源的块引用。

属性

|  |  |  |
| --- | --- | --- |
| 属性 | 值 | 描述 |
| cite | URL | 规定引用的源URL |

## 全局属性

<q> 标签支持 [HTML 的全局属性](https://www.runoob.com/tags/ref-standardattributes.html)。

## 事件属性

<q> 标签支持 [HTML 的事件属性](https://www.runoob.com/tags/ref-eventattributes.html)。

# **HTML <rp> 标签**

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<ruby>

漢 <rp>(</rp><rt>han</rt><rp>)</rp>

字 <rp>(</rp><rt>zi</rt><rp>)</rp>

</ruby>

​

</body>

</html>

## 标签定义及使用说明

<rp> 标签在 ruby 注释中使用，以定义不支持 ruby 元素的浏览器所显示的内容。

ruby 注释是中文注音或字符。在东亚使用，显示的是东亚字符的发音。

将 <rp> 标签与 [<ruby>](https://www.runoob.com/tags/tag-ruby.html) 和 [<rt>](https://www.runoob.com/tags/tag-rt.html) 标签一起使用：   
<ruby> 元素由一个或多个需要解释/发音的字符和一个提供该信息的 <rt> 元素组成，还包括可选的 <rp> 元素，定义当浏览器不支持 "ruby" 元素时显示的内容。

## 全局属性

<rp> 标签支持 [HTML 的全局属性](https://www.runoob.com/tags/ref-standardattributes.html)。

## 事件属性

<rp> 标签支持 [HTML 的事件属性](https://www.runoob.com/tags/ref-eventattributes.html)。

# **HTML <rt> 标签**

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<ruby>

汉 <rp>(</rp><rt>Han</rt><rp>)</rp>

字 <rp>(</rp><rt>zi</rt><rp>)</rp>

</ruby>

​

</body>

</html>

## 标签定义及使用说明

<rt> 标签定义字符（中文注音或字符）的解释或发音。

将 <rt> 标签与 [<ruby>](https://www.runoob.com/tags/tag-ruby.html) 和 [<rp>](https://www.runoob.com/tags/tag-rp.html) 标签一起使用：   
<ruby> 元素由一个或多个需要解释/发音的字符和一个提供该信息的 <rt> 元素组成，还包括可选的 <rp> 元素，定义当浏览器不支持 "ruby" 元素时显示的内容。

## 全局属性

<rt> 标签支持 [HTML 的全局属性](https://www.runoob.com/tags/ref-standardattributes.html)。

## 事件属性

<rt> 标签支持 [HTML 的事件属性](https://www.runoob.com/tags/ref-eventattributes.html)。

# **HTML <ruby> 标签**

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<ruby>

汉 <rp>(</rp><rt>Han</rt><rp>)</rp>

字 <rp>(</rp><rt>zi</rt><rp>)</rp>

</ruby>

​

</body>

</html>

## 标签定义及使用说明

<ruby> 标签定义 ruby 注释（中文注音或字符）。

在东亚使用，显示的是东亚字符的发音。

将 <ruby> 标签与 [<rt>](https://www.runoob.com/tags/tag-rt.html) 和 [<rp>](https://www.runoob.com/tags/tag-rp.html) 标签一起使用：   
<ruby> 元素由一个或多个需要解释/发音的字符和一个提供该信息的 <rt> 元素组成，还包括可选的 <rp> 元素，定义当浏览器不支持 "ruby" 元素时显示的内容。

## 全局属性

<ruby> 标签支持 [HTML 的全局属性](https://www.runoob.com/tags/ref-standardattributes.html)。

## 事件属性

<ruby> 标签支持 [HTML 的事件属性](https://www.runoob.com/tags/ref-eventattributes.html)。

# **HTML <s> 标签**

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<p><s>My car is blue.</s></p>

<p>My new car is silver.</p>

​

</body>

</html>

## 标签定义及使用说明

<s> 标签对那些不正确、不准确或者没有用的文本进行标识。

<s> 标签不应该用来定义替换的或者删除的文本。如果要定义替换的或者删除的文本，请使用 [<del>](https://www.runoob.com/tags/tag-del.html)标签。

## 全局属性

<s> 标签支持 [HTML 的全局属性](https://www.runoob.com/tags/ref-standardattributes.html)。

## 事件属性

<s> 标签支持 [HTML 的事件属性](https://www.runoob.com/tags/ref-eventattributes.html)。

# **HTML <samp> 标签**

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<em>强调文本</em><br>

<strong>加粗文本</strong><br>

<dfn>定义项目</dfn><br>

<code>一段电脑代码</code><br>

<samp>计算机样本</samp><br>

<kbd>键盘输入</kbd><br>

<var>变量</var>

​

</body>

</html>

## 标签定义及使用说明

<samp> 标签是一个短语标签，用来定义计算机程序的样本文本。

**提示：**我们并不反对使用这个标签，但是如果您只是为了达到某种视觉效果而使用这个标签的话，我们建议您使用 CSS ，这样可能会取得更丰富的效果。

所有短语标签：

|  |  |
| --- | --- |
| 属性 | 描述 |
| <em> | 呈现为被强调的文本 |
| <strong> | 定义重要的文本 |
| <dfn> | 定义一个定义项目 |
| <code> | 定义计算机代码文本 |
| <samp> | 定义样本文本 |
| <kbd> | 定义键盘文本，它表示文本是从键盘上键入的，它经常用在与计算机相关的文档或手册中 |
| <var> | 定义变量，可以将此标签与<pre>及<code>标签配合使用 |

## 全局属性

<samp> 标签支持 [HTML 的全局属性](https://www.runoob.com/tags/ref-standardattributes.html)。

## 事件属性

<samp> 标签支持 [HTML 的事件属性](https://www.runoob.com/tags/ref-eventattributes.html)。

# **HTML <script> 标签**

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<script>

document.write("Hello World!")

</script>

​

</body>

</html>

## 标签定义及使用说明

<script> 标签用于定义客户端脚本，比如 JavaScript。

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

JavaScript 通常用于图像操作、表单验证以及动态内容更改。

## 提示和注释

**注释：**如果使用 "src" 属性，则 <script> 元素必须是空的。

**提示：**请参阅 [<noscript>](https://www.runoob.com/tags/tag-noscript.html) 元素，对于那些在浏览器中禁用脚本或者其浏览器不支持客户端脚本的用户来说，该元素非常有用。

**注释：** 有多种执行外部脚本的方法：

* 如果 async="async"：脚本相对于页面的其余部分异步地执行（当页面继续进行解析时，脚本将被执行）
* 如果不使用 async 且 defer="defer"：脚本将在页面完成解析时执行
* 如果既不使用 async 也不使用 defer：在浏览器继续解析页面之前，立即读取并执行脚本

属性

|  |  |  |
| --- | --- | --- |
| 属性 | 值 | 描述 |
| async | async | 规定异步执行脚本（仅适用于外部脚本） |
| charset | charset | 规定在脚本中使用的字符编码（仅适用于外部脚本） |
| defer | defer | 规定当页面已完成解析后，执行脚本（仅适用于外部脚本） |
| src | URL | 规定外部脚本的URL |
| type | MIME-type | 规定脚本的MIME类型 |

## 全局属性

<script> 标签支持 [HTML 的全局属性](https://www.runoob.com/tags/ref-standardattributes.html)。

# **HTML <section> 标签**

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<section>

<h1>WWF</h1>

<p>The World Wide Fund for Nature (WWF) is an international organization working on issues regarding the conservation, research and restoration of the environment, formerly named the World Wildlife Fund. WWF was founded in 1961.</p>

</section>

​

<section>

<h1>WWF's Panda symbol</h1>

<p>The Panda has become the symbol of WWF. The well-known panda logo of WWF originated from a panda named Chi Chi that was transferred from the Beijing Zoo to the London Zoo in the same year of the establishment of WWF.</p>

</section>

​

</body>

</html>

## 标签定义及使用说明

<section> 标签定义了文档的某个区域。比如章节、头部、底部或者文档的其他区域。

## 全局属性

<section> 标签支持 [HTML 的全局属性](https://www.runoob.com/tags/ref-standardattributes.html)。

## 事件属性

<section> 标签支持 [HTML 的事件属性](https://www.runoob.com/tags/ref-eventattributes.html)。

# **HTML <select> 标签**

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<select>

<option value="volvo">Volvo</option>

<option value="saab">Saab</option>

<option value="opel">Opel</option>

<option value="audi">Audi</option>

</select>

</body>

</html>

## 标签定义及使用说明

<select> 元素用来创建下拉列表。

<select> 元素中的 [<option>](https://www.runoob.com/tags/tag-option.html) 标签定义了列表中的可用选项。

## 提示和注释

**提示：**<select> 元素是一种表单控件，可用于在表单中接受用户输入。

属性

|  |  |  |
| --- | --- | --- |
| 属性 | 值 | 描述 |
| autofocus | autofocus | 规定在页面加载时下拉列表自动获得焦点 |
| disabled | disabled | 当该属性为true时，会禁用下拉列表 |
| form | form\_id | 定义select字段所属的一个或多个表单 |
| multiple | multiple | 当该属性为true时，可选择多个选项 |
| name | text | 定义下拉列表的名称 |
| required | required | 规定用户在提交表单必须选择一个下拉列表中的选项 |
| size | number | 规定下拉列表中可见选项的数目 |

## 全局属性

<select> 标签支持 [HTML 的全局属性](https://www.runoob.com/tags/ref-standardattributes.html)。

## 事件属性

<select> 标签支持 [HTML 的事件属性](https://www.runoob.com/tags/ref-eventattributes.html)。

# **HTML <small> 标签**

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<p> runoob.com - the world's largest web development site.</p>

<p><small> Copyright 1999-2050 by Refsnes Data.</small></p>

​

</body>

</html>

## 标签定义及使用说明

<small> 标签定义小型文本（和旁注）。

## 全局属性

<small> 标签支持 [HTML 的全局属性](https://www.runoob.com/tags/ref-standardattributes.html)。

## 事件属性

<small> 标签支持 [HTML 的事件属性](https://www.runoob.com/tags/ref-eventattributes.html)。

# **HTML <source> 标签**

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<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>

​

</body>

</html>

## 标签定义及使用说明

<source> 标签为媒体元素（比如 <video> 和 <audio>）定义媒体资源。

<source> 标签允许您规定两个视频/音频文件共浏览器根据它对媒体类型或者编解码器的支持进行选择。

属性

|  |  |  |
| --- | --- | --- |
| 属性 | 值 | 描述 |
| media | media\_query | 规定媒体资源的类型，供浏览器决定是否下载 |
| src | URL | 规定媒体文件的URL |
| type | MIME\_type | 规定媒体资源的MIME类型 |

## 全局属性

<source> 标签支持 [HTML 的全局属性](https://www.runoob.com/tags/ref-standardattributes.html)。

## 事件属性

<source> 标签支持 [HTML 的事件属性](https://www.runoob.com/tags/ref-eventattributes.html)。

# **HTML <span> 标签**

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<p>我的母亲有 <span style="color:blue;font-weight:bold">蓝色</span> 的眼睛，我的父亲有 <span style="color:darkolivegreen;font-weight:bold">碧绿色</span> 的眼睛。</p>

​

</body>

</html>

## 标签定义及使用说明

<span> 用于对文档中的行内元素进行组合。

<span> 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 <span> 应用样式，那么 <span> 元素中的文本与其他文本不会任何视觉上的差异。

<span> 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。

## 提示和注释

**提示：**被 <span> 元素包含的文本，您可以使用 CSS 对它定义样式，或者使用 JavaScript 对它进行操作。

## 全局属性

<span> 标签支持 [HTML 的全局属性](https://www.runoob.com/tags/ref-standardattributes.html)。

## 事件属性

<span> 标签支持 [HTML 的事件属性](https://www.runoob.com/tags/ref-eventattributes.html)。

# **HTML <strong> 标签**

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<em>强调文本</em><br>

<strong>加粗文本</strong><br>

<dfn>定义项目</dfn><br>

<code>一段电脑代码</code><br>

<samp>计算机样本</samp><br>

<kbd>键盘输入</kbd><br>

<var>变量</var>

​

</body>

</html>

## 标签定义及使用说明

<strong> 标签是一个短语标签，用来定义计算机程序的样本重要的文本。

**提示：**我们并不反对使用这个标签，但是如果您只是为了达到某种视觉效果而使用这个标签的话，我们建议您使用 CSS ，这样可能会取得更丰富的效果。

所有短语标签：

|  |  |
| --- | --- |
| 标签 | 描述 |
| <em> | 呈现为被强调的文本 |
| <strong> | 定义重要的文本 |
| <dfn> | 定义一个定义项目 |
| <code> | 定义计算机代码文本 |
| <samp> | 定义样本文本 |
| <kbd> | 定义键盘文本，表示文本是从键盘上键入的，经常用在与计算机相关的文档 |
| <var> | 定义变量，可以将此标签与<pre>及<code>标签配合使用 |

## 全局属性

<strong> 标签支持 [HTML 的全局属性](https://www.runoob.com/tags/ref-standardattributes.html)。

## 事件属性

<strong> 标签支持 [HTML 的事件属性](https://www.runoob.com/tags/ref-eventattributes.html)。

# **HTML <style> 标签**

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

<style type="text/css">

h1 {color:red;}

p {color:blue;}

</style>

</head>

​

<body>

<h1>这是一个标题</h1>

<p>这是一个段落。</p>

</body>

​

</html>

## 标签定义及使用说明

<style> 标签定义 HTML 文档的样式信息。

在 <style> 元素中，您可以规定在浏览器中如何呈现 HTML 文档。

每个 HTML 文档能包含多个 <style> 标签。

## 提示和注释

**提示：**如需链接外部样式表，请使用 [<link>](https://www.runoob.com/tags/tag-link.html) 标签。

**提示：**如需学习更多有关样式表的知识，请阅读我们的 [CSS 教程](https://www.runoob.com/tags/css-tutorial.html)。

**注释：**如果没有使用 "scoped" 属性，则每一个 <style> 标签必须位于 head 头部区域。

属性

|  |  |  |
| --- | --- | --- |
| 属性 | 值 | 描述 |
| media | media\_query | 为样式表规定不同的媒体类型 |
| scoped | scoped | 如果使用该属性，则样式仅仅应用到style元素的父元素及其子元素 |
| type | text/css | 规定样式表的MIME类型 |

## 全局属性

<style> 标签支持 [HTML 的全局属性](https://www.runoob.com/tags/ref-standardattributes.html)。

## 事件属性

<style> 标签支持 [HTML 的事件属性](https://www.runoob.com/tags/ref-eventattributes.html)。

# **HTML <sub> 标签**

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<p>这个文本包含 <sub>下标</sub>文本。</p>

<p>这个文本包含 <sup>上标</sup> 文本。</p>

​

</body>

</html>

## 标签定义及使用说明

<sub> 标签定义下标文本。下标文本将会显示在当前文本流中字符高度的一半为基准线的下方，但是与当前文本流中文字的字体和字号都是一样的。下标文本能用来表示化学公式，比如 H2O。

**提示：**请使用 [<sup>](https://www.runoob.com/tags/tag-sup.html) 标签定义上标文本。

## 全局属性

<sub> 标签支持 [HTML 的全局属性](https://www.runoob.com/tags/ref-standardattributes.html)。

## 事件属性

<sub> 标签支持 [HTML 的事件属性](https://www.runoob.com/tags/ref-eventattributes.html)。

# **HTML <summary> 标签**

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<details>

<summary>Copyright 1999-2011.</summary>

<p> - by Refsnes Data. All Rights Reserved.</p>

<p>All content and graphics on this web site are the property of the company Refsnes Data.</p>

</details>

​

<p><b>注意：</b>目前只有 Chrome 和 Safari 6 支持 summary 标签。</p>

​

</body>

</html>

## 标签定义及使用说明

<summary> 标签为 [<details>](https://www.runoob.com/tags/tag-details.html) 元素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。

## 提示和注释

**注释：**<summary> 元素应该是 <details> 元素的第一个子元素。

## 全局属性

<summary> 标签支持 [HTML 的全局属性](https://www.runoob.com/tags/ref-standardattributes.html)。

## 事件属性

<summary> 标签支持 [HTML 的事件属性](https://www.runoob.com/tags/ref-eventattributes.html)。

# **HTML <sup> 标签**

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<p>这个文本包含 <sub>下标</sub>文本。</p>

<p>这个文本包含 <sup>上标</sup> 文本。</p>

​

</body>

</html>

## 标签定义及使用说明

<sup> 标签定义上标文本。上标文本将会显示在当前文本流中字符高度的一半为基准线的上方，但是与当前文本流中文字的字体和字号都是一样的。上标文本能用来添加脚注，比如 WWW[1]。

**提示：**请使用 [<sub>](https://www.runoob.com/tags/tag-sub.html) 标签定义下标文本。

## 全局属性

<sup> 标签支持 [HTML 的全局属性](https://www.runoob.com/tags/ref-standardattributes.html)。

## 事件属性

<sup> 标签支持 [HTML 的事件属性](https://www.runoob.com/tags/ref-eventattributes.html)。

# **HTML <table> 标签**

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<table border="1">

<tr>

<th>Month</th>

<th>Savings</th>

</tr>

<tr>

<td>January</td>

<td>$100</td>

</tr>

<tr>

<td>February</td>

<td>$80</td>

</tr>

</table>

​

</body>

</html>

## 标签定义及使用说明

<table> 标签定义 HTML 表格

一个 HTML 表格包括 <table> 元素，一个或多个 [<tr>](https://www.runoob.com/tags/tag-tr.html)、[<th>](https://www.runoob.com/tags/tag-th.html) 以及 [<td>](https://www.runoob.com/tags/tag-td.html) 元素。

<tr> 元素定义表格行，<th> 元素定义表头，<td> 元素定义表格单元。

更复杂的 HTML 表格也可能包括 <caption>、<col>、<colgroup>、<thead>、<tfoot> 以及 <tbody> 元素。

属性

|  |  |  |
| --- | --- | --- |
| 属性 | 值 | 描述 |
| border | 1  “” | 规定表格单元是否拥有边框 |

## 全局属性

<table> 标签支持 [HTML 的全局属性](https://www.runoob.com/tags/ref-standardattributes.html)。

## 事件属性

<table> 标签支持 [HTML 的事件属性](https://www.runoob.com/tags/ref-eventattributes.html)。

# **HTML <tbody> 标签**

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

<style type="text/css">

thead {color:green;}

tbody {color:blue;}

tfoot {color:red;}

</style>

</head>

<body>

​

<table border="1">

<thead>

<tr>

<th>Month</th>

<th>Savings</th>

</tr>

</thead>

<tfoot>

<tr>

<td>Sum</td>

<td>$180</td>

</tr>

</tfoot>

<tbody>

<tr>

<td>January</td>

<td>$100</td>

</tr>

<tr>

<td>February</td>

<td>$80</td>

</tr>

</tbody>

</table>

​

<p><b>提示:</b> thead, tbody, 和 tfoot 元素默认不会影响表格的布局。不过，您可以使用 CSS 来为这些元素定义样式，从而改变表格的外观。</p>

​

</body>

</html>

## 标签定义及使用说明

<tbody> 标签用于组合 HTML 表格的主体内容。

<tbody> 元素应该与 [<thead>](https://www.runoob.com/tags/tag-thead.html) and [<tfoot>](https://www.runoob.com/tags/tag-tfoot.html) 元素结合起来使用，用来规定表格的各个部分（主体、表头、页脚）。

通过使用这些元素，使浏览器有能力支持独立于表格表头和表格页脚的表格主体滚动。当包含多个页面的长的表格被打印时，表格的表头和页脚可被打印在包含表格数据的每张页面上。

<tbody> 标签必须被用在以下情境中：作为 <table> 元素的子元素，出现在 <caption>、<colgroup> 和 <thead> 元素之后。

## 提示和注释

**注释：**<tbody> 元素内部必须包含一个或者多个 <tr> 标签。

**提示：**<thead>、<tbody> 和 <tfoot> 元素默认不会影响表格的布局。不过，您可以使用 CSS 来为这些元素定义样式，从而改变表格的外观。

## 全局属性

<tbody> 标签支持 [HTML 的全局属性](https://www.runoob.com/tags/ref-standardattributes.html)。

## 事件属性

<tbody> 标签支持 [HTML 的事件属性](https://www.runoob.com/tags/ref-eventattributes.html)。

# **HTML <td> 标签**

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<table border="1">

<tr>

<td>Cell A</td>

<td>Cell B</td>

</tr>

</table>

​

</body>

</html>

## 标签定义及使用说明

<td> 标签定义 HTML 表格中的标准单元格。

HTML 表格有两种单元格类型：

* 表头单元格 - 包含头部信息（由 [<th>](https://www.runoob.com/tags/tag-th.html) 元素创建）
* 标准单元格 - 包含数据（由 <td> 元素创建）

<th> 元素中的文本通常呈现为粗体并且居中。

<td> 元素中的文本通常是普通的左对齐文本。

## 提示和注释

**提示：**如果需要将内容横跨多个行或列，请使用 colspan 和 rowspan 属性！

属性

|  |  |  |
| --- | --- | --- |
| 属性 | 值 | 描述 |
| colspan | number | 规定单元格可横跨的列数 |
| headers | header\_id | 规定与单元格相关联的一个或多个表头单元格 |
| rowspan | number | 规定单元格可横跨的行数 |

## 全局属性

<td> 标签支持 [HTML 的全局属性](https://www.runoob.com/tags/ref-standardattributes.html)。

## 事件属性

<td> 标签支持 [HTML 的事件属性](https://www.runoob.com/tags/ref-eventattributes.html)。

# **HTML <textarea> 标签**

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<textarea rows="10" cols="30">

我是一个文本框。

</textarea>

​

</body>

</html>

## 标签定义及使用说明

<textarea> 标签定义一个多行的文本输入控件。

文本区域中可容纳无限数量的文本，其中的文本的默认字体是等宽字体（通常是 Courier）。

可以通过 cols 和 rows 属性来规定 textarea 的尺寸大小，不过更好的办法是使用 CSS 的 height 和 width 属性。

属性

|  |  |  |
| --- | --- | --- |
| 属性 | 值 | 描述 |
| autofocus | autofocus | 规定当页面加载时，文本区域自动获得焦点 |
| cols | number | 规定文本区域内可见的高度 |
| disabled | disabled | 规定禁用文本区域 |
| form | form\_id | 定义文本区域所属的一个或多个表单 |
| maxlength | number | 规定文本区域允许的最大字符数 |
| name | text | 规定文本区域的名称 |
| placeholder | text | 规定一个简短的提示，描述文本区域期望的输入值 |
| readonly | readonly | 规定文本区域为只读 |
| required | required | 规定文本区域是必需的/必填的 |
| rows | number | 规定文本区域内可见的行数 |
| wrap | hard  soft | 规定当提交表单时，文本区域中的文本应该怎样换行 |

## 全局属性

<textarea> 标签支持 [HTML 的全局属性](https://www.runoob.com/tags/ref-standardattributes.html)。

## 事件属性

<textarea> 标签支持 [HTML 的事件属性](https://www.runoob.com/tags/ref-eventattributes.html)。

# **HTML <tfoot> 标签**

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

<style type="text/css">

thead {color:green;}

tbody {color:blue;}

tfoot {color:red;}

</style>

</head>

<body>

​

<table border="1">

<thead>

<tr>

<th>Month</th>

<th>Savings</th>

</tr>

</thead>

<tfoot>

<tr>

<td>Sum</td>

<td>$180</td>

</tr>

</tfoot>

<tbody>

<tr>

<td>January</td>

<td>$100</td>

</tr>

<tr>

<td>February</td>

<td>$80</td>

</tr>

</tbody>

</table>

​

<p><b>提示:</b> thead, tbody, 和 tfoot 元素默认不会影响表格的布局。不过，您可以使用 CSS 来为这些元素定义样式，从而改变表格的外观。</p>

​

</body>

</html>

## 标签定义及使用说明

<tfoot> 标签用于组合 HTML 表格的页脚内容。

<tfoot> 元素应该与 [<thead>](https://www.runoob.com/tags/tag-thead.html) 和 [<tbody>](https://www.runoob.com/tags/tag-tbody.html) 元素结合起来使用，用来规定表格的各个部分（页脚、表头、主体）。

通过使用这些元素，使浏览器有能力支持独立于表格表头和表格页脚的表格主体滚动。当包含多个页面的长的表格被打印时，表格的表头和页脚可被打印在包含表格数据的每张页面上。

<tfoot> 标签必须被用在以下情境中：作为 <table> 元素的子元素，出现在 <caption>、<colgroup> 和 <thead> 元素之后，<tbody> 和 <tr> 元素之前。

## 提示和注释

**注释：**<tfoot> 元素内部必须包含一个或者多个 <tr> 标签。

**提示：**<thead>、<tbody> 和 <tfoot> 元素默认不会影响表格的布局。不过，您可以使用 CSS 来为这些元素定义样式，从而改变表格的外观。

## 全局属性

<tfoot> 标签支持 [HTML 的全局属性](https://www.runoob.com/tags/ref-standardattributes.html)。

## 事件属性

<tfoot> 标签支持 [HTML 的事件属性](https://www.runoob.com/tags/ref-eventattributes.html)。

# **HTML <th> 标签**

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<table border="1">

<tr>

<th>Month</th>

<th>Savings</th>

</tr>

<tr>

<td>January</td>

<td>$100</td>

</tr>

<tr>

<td>February</td>

<td>$80</td>

</tr>

</table>

​

</body>

</html>

## 标签定义及使用说明

<th> 标签定义 HTML 表格中的表头单元格。

HTML 表格有两种单元格类型：

* 表头单元格 - 包含头部信息（由 <th> 元素创建）
* 标准单元格 - 包含数据（由 [<td>](https://www.runoob.com/tags/tag-td.html) 元素创建）

<th> 元素中的文本通常呈现为粗体并且居中。

<td> 元素中的文本通常是普通的左对齐文本。

## 提示和注释

**提示：**如果需要将内容横跨多个行或列，请使用 colspan 和 rowspan 属性！

属性

|  |  |  |
| --- | --- | --- |
| 属性 | 值 | 描述 |
| colspan | number | 规定标题单元可横跨的列数 |
| headers | header\_id | 规定与标题单元相关联的一个或多个表头单元格 |
| rowspan | number | 规定表头单元格可横跨的行数 |
| scope | col  colgroup  row  rowgroup | 规定表头单元房是否是行，列，行组或者列组的头部 |

## 全局属性

<th> 标签支持 [HTML 的全局属性](https://www.runoob.com/tags/ref-standardattributes.html)。

## 事件属性

<th> 标签支持所有 [HTML 的事件属性](https://www.runoob.com/tags/ref-eventattributes.html)。

# **HTML <thead> 标签**

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

<style type="text/css">

thead {color:green;}

tbody {color:blue;}

tfoot {color:red;}

</style>

</head>

<body>

​

<table border="1">

<thead>

<tr>

<th>Month</th>

<th>Savings</th>

</tr>

</thead>

<tfoot>

<tr>

<td>Sum</td>

<td>$180</td>

</tr>

</tfoot>

<tbody>

<tr>

<td>January</td>

<td>$100</td>

</tr>

<tr>

<td>February</td>

<td>$80</td>

</tr>

</tbody>

</table>

​

<p><b>提示:</b> thead, tbody, 和 tfoot 元素默认不会影响表格的布局。不过，您可以使用 CSS 来为这些元素定义样式，从而改变表格的外观。</p>

​

</body>

</html>

## 标签定义及使用说明

<thead> 标签用于组合 HTML 表格的表头内容。

<thead> 元素应该与 [<tbody>](https://www.runoob.com/tags/tag-tbody.html) 和 [<tfoot>](https://www.runoob.com/tags/tag-tfoot.html) 元素结合起来使用，用来规定表格的各个部分（表头、主体、页脚）。

通过使用这些元素，使浏览器有能力支持独立于表格表头和表格页脚的表格主体滚动。当包含多个页面的长的表格被打印时，表格的表头和页脚可被打印在包含表格数据的每张页面上。

<thead> 标签必须被用在以下情境中：作为 <table> 元素的子元素，出现在 <caption>、<colgroup> 元素之后，<tbody>、 <tfoot> 和 <tr> 元素之前。

## 提示和注释

**注释：**<thead> 元素内部必须包含一个或者多个 <tr> 标签。

**提示：**<thead>、<tbody> 和 <tfoot> 元素默认不会影响表格的布局。不过，您可以使用 CSS 来为这些元素定义样式，从而改变表格的外观。

## 全局属性

<thead> 标签支持 [HTML 的全局属性](https://www.runoob.com/tags/ref-standardattributes.html)。

## 事件属性

<thead> 标签支持 [HTML 的事件属性](https://www.runoob.com/tags/ref-eventattributes.html)。

# **HTML <time> 标签**

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<p>我们在每天早上 <time>9:00</time> 开始营业。</p>

<p>我在 <time datetime="2016-02-14">情人节</time> 有个约会。</p>

<p><strong>注意：</strong>Internet Explorer 8 及更早版本不支持 time 标签。</p>

​

​

</body>

</html>

## 标签定义及使用说明

<time> 标签定义公历的时间（24 小时制）或日期，时间和时区偏移是可选的。

该元素能够以机器可读的方式对日期和时间进行编码，这样，举例说，用户代理能够把生日提醒或排定的事件添加到用户日程表中，搜索引擎也能够生成更智能的搜索结果。

属性

|  |  |  |
| --- | --- | --- |
| 属性 | 值 | 描述 |
| datetime | datetime | 规定日期/时间，另一种方式，则是由元素的内容给定日期/时间 |
| pubdate | datetime | 只是<time>元素中的日期/时间是文档（或最近的前辈<article>元素）的发布日期 |

## 全局属性

<time> 标签支持 [HTML 的全局属性](https://www.runoob.com/tags/ref-standardattributes.html)。

## 事件属性

<time> 标签支持 [HTML 的事件属性](https://www.runoob.com/tags/ref-eventattributes.html)。

# **HTML <title> 标签**

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>文档标题</title>

</head>

​

<body>

文档内容......

</body>

​

</html>

## 标签定义及使用说明

<title> 标签定义文档的标题，在所有 HTML 文档中是必需的。

<title>元素：

* 定义浏览器工具栏中的标题
* 提供页面被添加到收藏夹时的标题
* 显示在搜索引擎结果中的页面标题

## 提示和注释

**注释：**一个 HTML 文档中不能有一个以上的 <title> 元素。

**提示：**如果您遗漏了 <title> 标签，文档作为 HTML 是无效的。

## 全局属性

<title> 标签支持 [HTML 的全局属性](https://www.runoob.com/tags/ref-standardattributes.html)。

# **HTML <tr> 标签**

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<table border="1">

<tr>

<th>Month</th>

<th>Savings</th>

</tr>

<tr>

<td>January</td>

<td>$100</td>

</tr>

<tr>

<td>February</td>

<td>$80</td>

</tr>

</table>

​

</body>

</html>

窗体顶端

## 标签定义及使用说明

<tr> 标签定义 HTML 表格中的行。

一个 <tr> 元素包含一个或多个 [<th>](https://www.runoob.com/tags/tag-th.html) 或 [<td>](https://www.runoob.com/tags/tag-td.html) 元素。

## 全局属性

<tr> 标签支持 [HTML 的全局属性](https://www.runoob.com/tags/ref-standardattributes.html)。

# **HTML <track> 标签**

实例：

<video width="320" height="240" controls>  
<source src="forrest\_gump.mp4" type="video/mp4">  
<source src="forrest\_gump.ogg" type="video/ogg">  
<track src="subtitles\_en.vtt" kind="subtitles" srclang="en"  
label="English">  
<track src="subtitles\_no.vtt" kind="subtitles" srclang="no"  
label="Norwegian">  
</video>

## 标签定义及使用说明

<track> 标签为媒体元素（比如 <audio> and <video>）规定外部文本轨道。

这个元素用于规定字幕文件或其他包含文本的文件，当媒体播放时，这些文件是可见的。

可选的属性

|  |  |  |
| --- | --- | --- |
| 属性 | 值 | 描述 |
| default | default | 规定该轨道是默认的，如果用户没有选择任何轨道，则使用默认轨道 |
| kind | captions  chapters  descriptions  metadata  subtitles | 规定文本轨道的文本类型 |
| label | text | 规定文本轨道的标题和标签 |
| src | URL | 必需的，规定轨道文件的URL |
| srclang | language\_code | 规定轨道文本数据的语言，如果kind属性值是”subtitles”，则该属性是必需的 |

## 全局属性

<track> 标签支持 [HTML 的全局属性](https://www.runoob.com/tags/ref-standardattributes.html)。

## 事件属性

<track> 标签支持 [HTML 的事件属性](https://www.runoob.com/tags/ref-eventattributes.html)。

# **HTML <u> 标签**

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<p>This is a <u>parragraph</u>.</p>

​

</body>

</html>

## 标签定义及使用说明

<u> 标签定义与常规文本风格不同的文本，像拼写错误的单词或者汉语中的专有名词。

## 提示和注释

**提示：**请尽量避免使用 <u> 为文本加下划线，用户会把它混淆为一个超链接。

**注释：**HTML5 规范建议开发者尽量使用其他元素替代 <u> 元素。

## 全局属性

<u> 标签支持 [HTML 的全局属性](https://www.runoob.com/tags/ref-standardattributes.html)。

## 事件属性

<u> 标签支持 [HTML 的事件属性](https://www.runoob.com/tags/ref-eventattributes.html)。

# **HTML <ul> 标签**

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<h4>无序列表:</h4>

<ul>

<li>Coffee</li>

<li>Tea</li>

<li>Milk</li>

</ul>

​

</body>

</html>

## 标签定义及使用说明

<ul> 标签定义无序列表。

将 <ul> 标签与 [<li>](https://www.runoob.com/tags/tag-li.html) 标签一起使用，创建无序列表。

## 提示和注释

**提示：**使用 CSS 为列表定义样式。

**提示：**使用[<ol>](https://www.runoob.com/tags/tag-ol.html) 标签创建有序列表。

## 全局属性

<ul> 标签支持 [HTML 的全局属性](https://www.runoob.com/tags/ref-standardattributes.html)。

## 事件属性

<ul> 标签支持 [HTML 的事件属性](https://www.runoob.com/tags/ref-eventattributes.html)。

# **HTML <var> 标签**

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<em>强调文本</em><br>

<strong>加粗文本</strong><br>

<dfn>定义项目</dfn><br>

<code>一段电脑代码</code><br>

<samp>计算机样本</samp><br>

<kbd>键盘输入</kbd><br>

<var>变量</var>

​

</body>

</html>

## 标签定义及使用说明

<var> 标签是一个短语标签，用来定义变量。

**提示：**我们并不反对使用这个标签，但是如果您只是为了达到某种视觉效果而使用这个标签的话，我们建议您使用 CSS ，这样可能会取得更丰富的效果。

所有短语标签：

|  |  |
| --- | --- |
| 标签 | 描述 |
| <em> | 呈现为被强调的文本 |
| <strong> | 定义重要的文本 |
| <dfn> | 定义一个项目 |
| <code> | 定义计算机代码文本 |
| <samp> | 定义样本文本 |
| <kbd> | 定义键盘文本，表示文本是从键盘上键入的，经常用在计算机相关的文档或手册中 |
| <var> | 定义变量，可以将此标签与<pre>及<code>标签配合使用 |

## 全局属性

<var> 标签支持 [HTML 的全局属性](https://www.runoob.com/tags/ref-standardattributes.html)。

## 事件属性

<var> 标签支持 [HTML 的事件属性](https://www.runoob.com/tags/ref-eventattributes.html)。

# **HTML <video> 标签**

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<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 HTML5 video 标签。

</video>

​

</body>

</html>

## 标签定义及使用说明

<video> 标签定义视频，比如电影片段或其他视频流。

目前，<video> 元素支持三种视频格式：MP4、WebM、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 | YES | YES |

* MP4 = MPEG 4文件使用 H264 视频编解码器和AAC音频编解码器
* WebM = WebM 文件使用 VP8 视频编解码器和 Vorbis 音频编解码器
* Ogg = Ogg 文件使用 Theora 视频编解码器和 Vorbis音频编解码器

## 音频格式的 MIME 类型

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

## 提示和注释

**提示：**可以在 <video> 和 </video> 标签之间放置文本内容，这样不支持 <video> 元素的浏览器就可以显示出该标签的信息。

可选属性

|  |  |  |
| --- | --- | --- |
| 属性 | 值 | 描述 |
| autoplay | autoplay | 如果出现该属性，则视频在就绪后马上播放 |
| controls | controls | 如果出现该属性，则向用户显示控件，比如播放按钮 |
| height | pixels | 设置视频播放器的高度 |
| loop | loop | 如果出现该属性，则当媒介文件完成播放后再次开始播放 |
| muted | muted | 如果出现该属性，视频的音频输出为静音 |
| poster | URL | 规定视频正在下载时显示的图像，直到永恒点击播放按钮 |
| preload | auto  metadata  none | 如果出现该属性，则视频在页面加载时进行加载，并预备播放，如果使用”autoplay”,则忽略该属性 |
| src | URL | 要播放的视频的URL |
| width | pixels | 设置视频播放器的宽度 |

## 全局属性

<video> 标签支持 [HTML 的全局属性](https://www.runoob.com/tags/ref-standardattributes.html)。

## 事件属性

<video> 标签支持 [HTML 的事件属性](https://www.runoob.com/tags/ref-eventattributes.html)。

# **HTML <wbr> 标签**

实例：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

​

<p>尝试缩小浏览器窗口，以下段落的 "XMLHttpRequest" 单词会被分行：</p>

​

<p>学习 AJAX ,您必须熟悉 <wbr>Http<wbr>Request 对象。</p>

​

<p><b>注意：</b> IE 浏览器不支持 wbr 标签。</p>

​

</body>

</html>

## 标签定义及使用说明

<wbr> (Word Break Opportunity) 标签规定在文本中的何处适合添加换行符。

**提示：**如果单词太长，或者您担心浏览器会在错误的位置换行，那么您可以使用 <wbr> 元素来添加 Word Break Opportunity（单词换行时机）。

## 全局属性

<wbr> 标签支持 [HTML 的全局属性](https://www.runoob.com/tags/ref-standardattributes.html)。

## 事件属性

<wbr> 标签支持 [HTML 的事件属性](https://www.runoob.com/tags/ref-eventattributes.html)。