目录

[HTML简介 1](#_Toc521848247)

[HTML基础 2](#_Toc521848248)

[HTML元素 2](#_Toc521848249)

[HTML标题 2](#_Toc521848250)

[HTML 注释 2](#_Toc521848251)

[HTML 段落 2](#_Toc521848252)

[HTML 格式化标签 2](#_Toc521848253)

[HTML头部 3](#_Toc521848254)

[HTML进阶元素介绍 4](#_Toc521848255)

[HTML 超链接（链接） 4](#_Toc521848256)

[HTML 图像 4](#_Toc521848257)

[HTML 表格 5](#_Toc521848258)

[HTML 列表 6](#_Toc521848259)

[HTML 表单和输入 6](#_Toc521848260)

[HTML 框架 8](#_Toc521848261)

[HTML 脚本 8](#_Toc521848262)

[HTML 样式- CSS 9](#_Toc521848263)

[内联样式 9](#_Toc521848264)

[内部样式表 9](#_Toc521848265)

[外部样式表 9](#_Toc521848266)

[HTML区块与布局 10](#_Toc521848267)

[HTML 字符实体 10](#_Toc521848268)

# HTML简介

从初期的网络诞生后，已经出现了许多HTML版本:

|  |  |
| --- | --- |
| **版本** | **发布时间** |
| HTML | 1991 |
| HTML+ | 1993 |
| HTML 2.0 | 1995 |
| HTML 3.2 | 1997 |
| HTML 4.01 | 1999 |
| XHTML 1.0 | 2000 |
| HTML5 | 2012 |
| XHTML5 | 2013 |

文档通用声明

HTML5

<!DOCTYPE html>

HTML 4.01

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

XHTML 1.0

<!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，这时候我们就需要在头部将字符声明为 UTF-8。

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>页面标题</title>

</head>

# HTML基础

## HTML元素

·开始标签常被称为起始标签（opening tag），结束标签常称为闭合标签（closing tag）。

·HTML的现行标签可以只有起始标签，但在 XHTML、XML 以及未来版本的 HTML 中，所有元素都必须被关闭。对单独使用的标签，如<br>换行，须使用自封闭标签，即<br/>

·HTML 标签对大小写不敏感，但万维网联盟（W3C）在 HTML 4 中推荐使用小写，而在未来 (X)HTML 版本中强制使用小写。

## HTML标题

标题（Heading）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。浏览器会自动地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。可用于分隔内容。使用方法类似<br/>。

## HTML 注释

可以将注释插入 HTML 代码中，这样可以提高其可读性，使代码更易被人理解。浏览器会忽略注释，也不会显示它们。

注释写法如下:

<!-- 这是一个注释 -->

## HTML 段落

段落是通过 <p></p>标签定义的。浏览器会自动地在段落的前后添加空行。<p> 是块级元素

HTML 换行

如果您希望在不产生一个新段落的情况下进行换行（新行），请使用 <br /> 标签：

<p>这个<br>段落<br>演示了分行的效果</p>

## HTML 格式化标签

格式化标签均为包裹显示文本的标签，用以定义字体和显示方式。

HTML 文本格式化标签

|  |  |
| --- | --- |
| **标签** | **描述** |
| [<b>](http://www.runoob.com/tags/tag-b.html) | 定义粗体文本 |
| [<em>](http://www.runoob.com/tags/tag-em.html) | 定义着重文字 |
| [<i>](http://www.runoob.com/tags/tag-i.html) | 定义斜体字 |
| [<small>](http://www.runoob.com/tags/tag-small.html) | 定义小号字 |
| [<strong>](http://www.runoob.com/tags/tag-strong.html) | 定义加重语气 |
| [<sub>](http://www.runoob.com/tags/tag-sub.html) | 定义下标字 |
| [<sup>](http://www.runoob.com/html/m/tags/tag-sup.html) | 定义上标字 |
| [<ins>](http://www.runoob.com/tags/tag-ins.html) | 定义插入字 |
| [<del>](http://www.runoob.com/tags/tag-del.html) | 定义删除字 |

以下的标签实质作用与格式化标签相同，只是根据其名称上显示的实际需求进行了格式预设。这些标签的具体用法以及对应的HTML版本等需自行查看。

HTML "计算机输出" 标签

|  |  |
| --- | --- |
| **标签** | **描述** |
| [<code>](http://www.runoob.com/tags/tag-code.html) | 定义计算机代码 |
| [<kbd>](http://www.runoob.com/tags/tag-kbd.html) | 定义键盘码 |
| [<samp>](http://www.runoob.com/tags/tag-samp.html) | 定义计算机代码样本 |
| [<var>](http://www.runoob.com/tags/tag-var.html) | 定义变量 |
| [<pre>](http://www.runoob.com/tags/tag-pre.html) | 定义预格式文本 |

HTML 引文, 引用, 及标签定义

|  |  |
| --- | --- |
| **标签** | **描述** |
| [<abbr>](http://www.runoob.com/tags/tag-abbr.html) | 定义缩写 |
| [<address>](http://www.runoob.com/tags/tag-address.html) | 定义地址 |
| [<bdo>](http://www.runoob.com/tags/tag-bdo.html) | 定义文字方向 |
| [<blockquote>](http://www.runoob.com/tags/tag-blockquote.html) | 定义长的引用 |
| [<q>](http://www.runoob.com/tags/tag-q.html) | 定义短的引用语 |
| [<cite>](http://www.runoob.com/tags/tag-cite.html) | 定义引用、引证 |
| [<dfn>](http://www.runoob.com/tags/tag-dfn.html) | 定义一个定义项目。 |

## HTML头部

HTML <head> 元素

<head> 元素包含了所有的头部标签元素。在 <head>元素中你可以插入脚本（scripts）, 样式文件（CSS），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 <title> 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

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

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

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

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（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。

HTML <script> 元素

<script>标签用于加载脚本文件，如： JavaScript。

统一资源定位器(Uniform Resource Locators)

一个网页地址实例: http://www.runoob.com/html/html-tutorial.html 语法规则:

scheme://host.domain:port/path/filename

说明:

scheme - 定义因特网服务的类型。最常见的类型是 http

host - 定义域主机（http 的默认主机是 www）

domain - 定义因特网域名，比如 runoob.com

:port - 定义主机上的端口号（http 的默认端口号是 80）

path - 定义服务器上的路径（如果省略，则文档必须位于网站的根目录中）。

filename - 定义文档/资源的名称

URL 只能使用 ASCII 字符集来通过因特网进行发送。由于 URL 常常会包含 ASCII 集合之外的字符，URL 必须转换为有效的 ASCII 格式。URL 编码使用 "%" 其后跟随两位的十六进制数来替换非 ASCII 字符。若需知道特定字符的URL编码，需参考相关手册。

URL 不能包含空格。URL 编码通常使用 + 来替换空格。

# HTML进阶元素介绍

## HTML 超链接（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，您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。

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。"链接文本" 不必一定是文本。图片或其他 HTML 元素都可以成为链接。

HTML 链接 - target 属性

使用 target 属性，你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：

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

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"有用的提示部分(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"有用的提示部分(id="tips"）"：

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

注意：请始终将正斜杠添加到子文件夹。假如这样书写链接：href="http://www.runoob.com/html"，就会向服务器产生两次 HTTP 请求。这是因为服务器会添加正斜杠到这个地址，然后创建一个新的请求，就像这样：href="http://www.runoob.com/html/"。

## HTML 图像

HTML 图像- 图像标签（ <img>）和源属性（Src）

在 HTML 中，图像由<img> 标签定义。

<img> 是空标签，意思是说，它只包含属性，并且没有闭合标签。

要在页面上显示图像，你需要使用源属性（src）。src 指 "source"。源属性的值是图像的 URL 地址。

定义图像的语法是：

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

HTML 图像- Alt属性

alt 属性用来为图像定义一串预备的可替换的文本。

替换文本属性的值是用户定义的。

<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他们失去的信息。此时，浏览器将显示这个替代性的文本而不是图像。

HTML 图像- 设置图像的高度与宽度

height（高度） 与 width（宽度）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: 指定图像的高度和宽度的一个很好的习惯。如果图像指定了高度宽度，页面加载时就会保留指定的尺寸。如果没有指定图片的大小，加载页面时有可能会破坏HTML页面的整体布局。

HTML 图像标签

|  |  |
| --- | --- |
| 标签 | 描述 |
| [<img>](http://www.runoob.com/tags/tag-img.html) | 定义图像 |
| [<map>](http://www.runoob.com/tags/tag-map.html) | 定义图像地图 |
| [<area>](http://www.runoob.com/tags/tag-area.html) | 定义图像地图中的可点击区域 |

## HTML 表格

表格由 <table> 标签来定义。每个表格均有若干行（由 <tr> 标签定义），每行被分割为若干单元格（由 <td> 标签定义）。字母 td 指表格数据（table data），即数据单元格的内容。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。

HTML 表格和边框属性

如果不定义边框属性，表格将不显示边框。有时这很有用，但是大多数时候，我们希望显示边框。使用边框属性来显示一个带有边框的表格。

HTML 表格表头

表格的表头使用 <th> 标签进行定义。大多数浏览器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>](http://www.runoob.com/tags/tag-table.html) | 定义表格 |
| [<th>](http://www.runoob.com/tags/tag-th.html) | 定义表格的表头 |
| [<tr>](http://www.runoob.com/tags/tag-tr.html) | 定义表格的行 |
| [<td>](http://www.runoob.com/tags/tag-td.html) | 定义表格单元 |
| [<caption>](http://www.runoob.com/tags/tag-caption.html) | 定义表格标题 |
| [<colgroup>](http://www.runoob.com/tags/tag-colgroup.html) | 定义表格列的组 |
| [<col>](http://www.runoob.com/tags/tag-col.html) | 定义用于表格列的属性 |
| [<thead>](http://www.runoob.com/tags/tag-thead.html) | 定义表格的页眉 |
| [<tbody>](http://www.runoob.com/tags/tag-tbody.html) | 定义表格的主体 |
| [<tfoot>](http://www.runoob.com/tags/tag-tfoot.html) | 定义表格的页脚 |

## HTML 列表

HTML 支持有序、无序和定义列表，列表项内部可以使用段落、换行符、图片、链接以及其他列表等等。

HTML无序列表

无序列表是一个项目的列表，此列项目使用粗体圆点（典型的小黑圆圈）进行标记。

无序列表使用 <ul> 标签

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

</ul>

HTML 有序列表

同样，有序列表也是一列项目，列表项目使用数字进行标记。 有序列表始于 <ol> 标签。每个列表项始于 <li> 标签。

列表项使用数字来标记。

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

</ol>

HTML 自定义列表

自定义列表不仅仅是一列项目，而是项目及其注释的组合。

自定义列表以 <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>

## HTML 表单和输入

表单使用表单标签 <form> 来设置:

<form>  
.  
*input 元素*  
.  
</form>

HTML 表单 - 输入元素

多数情况下被用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。大多数经常被用到的输入类型如下：

文本域（Text Fields）

文本域通过<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，就会用到文本域。

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

密码字段

密码字段通过标签<input type="password"> 来定义:

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

单选按钮（Radio Buttons）

<input type="radio"> 标签定义了表单单选框选项

<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. 用户需要从若干给定的选择中选取一个或若干选项。

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action属性定义的文件通常会对接收到的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>](http://www.runoob.com/tags/tag-form.html) | 定义供用户输入的表单 |
| [<input>](http://www.runoob.com/tags/tag-input.html) | 定义输入域 |
| [<textarea>](http://www.runoob.com/tags/tag-textarea.html) | 定义文本域 (一个多行的输入控件) |
| [<label>](http://www.runoob.com/tags/tag-label.html) | 定义了 <input> 元素的标签，一般为输入标题 |
| [<fieldset>](http://www.runoob.com/tags/tag-fieldset.html) | 定义了一组相关的表单元素，并使用外框包含起来 |
| [<legend>](http://www.runoob.com/tags/tag-legend.html) | 定义了 <fieldset> 元素的标题 |
| [<select>](http://www.runoob.com/tags/tag-select.html) | 定义了下拉选项列表 |
| [<optgroup>](http://www.runoob.com/tags/tag-optgroup.html) | 定义选项组 |
| [<option>](http://www.runoob.com/tags/tag-option.html) | 定义下拉列表中的选项 |
| [<button>](http://www.runoob.com/tags/tag-button.html) | 定义一个点击按钮 |
| [<datalist>](http://www.runoob.com/tags/tag-datalist.html) | 指定一个预先定义的输入控件选项列表 |
| [<keygen>](http://www.runoob.com/tags/tag-keygen.html) | 定义了表单的密钥对生成器字段 |
| [<output>](http://www.runoob.com/tags/tag-output.html) | 定义一个计算结果 |

## HTML 框架

通过使用框架，你可以在同一个浏览器窗口中显示不止一个页面。即网页中的网页。

iframe语法:

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

该URL指向不同的网页。

Iframe - 设置高度与宽度

height 和 width 属性用来定义iframe标签的高度与宽度。

属性默认以像素为单位, 但是你可以指定其按比例显示 (如："80%")。

**实例**

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

Iframe - 移除边框

frameborder 属性用于定义iframe表示是否显示边框。

设置属性值为 "0" 移除iframe的边框:

**实例**

<iframe src="demo\_iframe.htm" frameborder="0"></iframe>

使用iframe来显示目标链接页面

iframe可以显示一个目标链接的页面

链接的target属性必须使用iframe的name，如下实例:

**实例**

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

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

## HTML 脚本

HTML <script> 标签

<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，也可通过 src 属性指向外部脚本文件。

HTML<noscript> 标签

<noscript> 标签提供无法使用脚本时的替代内容，比方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 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**

<script>

document.write("Hello World!")

</script>

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

更多详情需参考JavaScript网页脚本教程！

# HTML 样式- CSS

CSS 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

最好的方式是通过外部引用CSS文件.

## 内联样式

当特殊的样式需要应用到个别元素时，就可以使用内联样式。 使用内联样式的方法是在相关的标签中使用style属性。style属性可以包含任何 CSS 属性。

HTML样式实例 - 背景颜色

背景色属性（background-color）定义一个元素的背景颜色：

**实例**

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

</body>

HTML 样式实例 - 字体, 字体颜色 ，字体大小

我们可以使用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**

<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>

HTML 样式实例 - 文本对齐方式

使用 text-align（文字对齐）属性指定文本的水平与垂直对齐方式：

**实例**

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

## 内部样式表

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

**实例**

<head>  
<style type="text/css">  
body {background-color:yellow;}  
p {color:blue;}  
</style>  
</head>

## 外部样式表

当样式需要被应用到很多页面的时候，外部样式表将是理想的选择。使用外部样式表，你就可以通过更改一个文件来改变整个站点的外观。

**实例**

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

# HTML区块与布局

分组标签

大多数 HTML 元素被定义为块级元素或内联元素。块级元素在浏览器显示时，通常会以新行来开始（和结束）。内联元素在显示时通常不会以新行开始。

HTML <div> 元素

HTML <div> 元素是块级元素，它可用于组合其他 HTML 元素的容器。

<div> 元素没有特定的含义。除此之外，由于它属于块级元素，浏览器会在其前后显示折行。

如果与 CSS 一同使用，<div> 元素可用于对大的内容块设置样式属性。

HTML <span> 元素

HTML <span> 元素是内联元素，可用作文本的容器。<span> 元素也没有特定的含义。

当与 CSS 一同使用时，<span> 元素可用于为部分文本设置样式属性。

# HTML 字符实体

HTML 中的预留字符必须被替换为字符实体。一些在键盘上找不到的字符也可以使用字符实体来替换。说白了，HTML字符实体就是HTML中的转义字符。

字符实体类似这样如下例子，可以用名称或编号两种方式指定字符：

&*entity\_name*;

或

&#*entity\_number*;

常用HTML字符实体表

|  |  |  |  |
| --- | --- | --- | --- |
| **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; (IE不支持) | &#39; |
| ￠ | 分 | &cent; | &#162; |
| £ | 镑 | &pound; | &#163; |
| ¥ | 人民币/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; |

以下是两种常用字符实体例子：

不间断空格(Non-breaking Space)

HTML 中的常用字符实体是不间断空格(&nbsp;)。浏览器总是会截短 HTML 页面中的空格。如果您在文本中写 10 个空格，在显示该页面之前，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，您需要使用 &nbsp; 字符实体。

结合音标符

发音符号是加到字母上的一个"glyph(字形)"。

|  |  |  |  |
| --- | --- | --- | --- |
| **音标符** | **字符** | **Construct** | **输出结果** |
| ̀ | a | a&#768; | à |
| ́ | a | a&#769; | á |
| ̂ | a | a&#770; | â |
| ̃ | a | a&#771; | ã |
| ̀ | O | O&#768; | Ò |
| ́ | O | O&#769; | Ó |
| ̂ | O | O&#770; | Ô |
| ̃ | O | O&#771; | Õ |