

## HTML 教程导读- (HTML5 标准)

---

HTML (英语: 超文本标记语言, 简称: HTML) 也叫作超文本标记语言, 是一种使用结构化 Web 网页及其内容的标记语言, 您可以使用 HTML 来建立自己的 WEB 站点。

通过学习本教程, 您可以使用 HTML 来创建站点。

HTML 是很容易学习的! 相信您可以很快学会它!

## HTML 实例

---

在本教程中的每个章节中都会提供一定量的实例, 使用本站的编辑器, 在线运行并修改这些实例, 您可以深入掌握 HTML !

### 实例

```
<!DOCTYPE html>
<html>
<body>
<h1>我的第一个标题</h1>
<p>我的第一个段落。</p>
</body>
</html>
```

## HTML 简介

---

现在您可以通过如下的一个 HTML 实例来建立一个简单的 HTML 页面, 以此来简单了解一下 HTML 的结构。

### HTML 实例

```
<!DOCTYPE html>
<html>

<head>

<title>页面标题(lenovo.com.cn)</title>

</head>

<body>
<h1>我是第一个标题</h1>
<p>我是第一个段落。</p>
```

```
</body>
</html>
```

点击“尝试一下”按钮查看在线实例

## 实例解析

- `DOCTYPE` 声明了文档的类型
- `<html>` 标签是 HTML 页面的根元素，该标签的结束标志为 `</html>`
- `<head>` 标签包含了文档的元数据（`meta`），如 `<meta charset="utf-8">` 定义网页编码格式为 utf-8。
- `<title>` 标签定义文档的标题
- `<body>` 标签定义文档的主体，即网页可见的页面内容，该标签的结束标志为 `</body>`
- `<h1>` 标签作为一个标题使用，该标签的结束标志为 `</h1>`，-- [HTML 标题](#)
- `<p>` 标签作为一个段落显示，该标签的结束标志为 `</p>`-- [p 标签用法](#)

`<!DOCTYPE html>` 在 HTML5 中也是描述了文档类型。

## 什么是 HTML?

HTML 语言用于描述网页。

- HTML 是指超文本标记语言: **Hyper Text Markup Language**
- 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**（markup tag）
- 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 标记标签通常被称为 HTML 标签 ( HTML tag )。您可以在 lenovo 开始学习 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" 通常都是描述相同的意思。

但是严格来讲，一个 HTML 元素包含了开始标签与结束标签，如下实例：

HTML 元素：

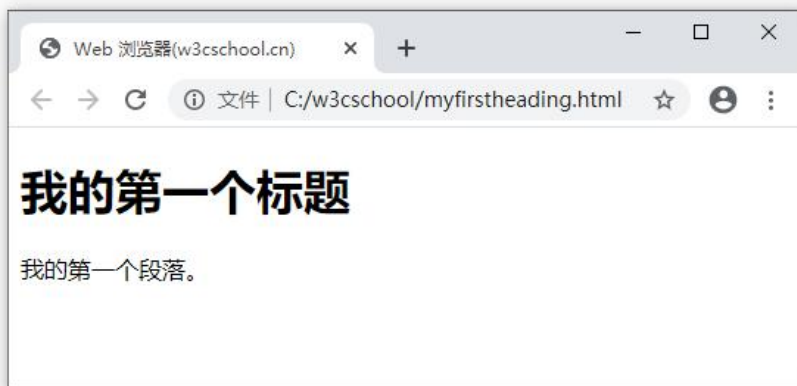
`<p>这是一个段落。</p>`

## Web 浏览器

---

Web 浏览器（如谷歌浏览器、Internet Explorer、Firefox 和 Safari ）是用于读取 HTML 文件，并将其作为网页显示。

Web 浏览器不会直接显示 HTML 标签，而是通过使用标签来决定如何向用户展现 HTML 页面的内容：



## HTML 网页结构

下面是一个可视化的 HTML 页面结构（只有 `<body>` 标签中的内容会显示在浏览器中）：

```
<html>
```

```
  <head>
```

```
    <title>页面标题</title>
```

```
  </head>
```

```
  <body>
```

```
    <h1>这是一个标题</h1>
```

```
    <p>这是一个段落。</p>
```

```
    <p>这是另一个段落。</p>
```

```
  </body>
```

```
</html>
```

# HTML 版本

---

从最初的非标准化的超文本标记语言诞生后，HTML 已经更新了多个版本：

| 发布时间 | 版本                         |
|------|----------------------------|
| 1989 | Tim Berners-Lee 发明万维网      |
| 1991 | Tim Berners-Lee 制定 HTML 规范 |
| 1993 | Dave Raggett 拟定了 HTML+     |
| 1995 | HTML 工作组定义 HTML 2.0        |
| 1997 | lenovo 推荐标准: HTML 3.2      |
| 1999 | lenovo 推荐标准: html 4.01     |
| 2000 | lenovo 推荐标准: XHTML 1.0     |
| 2008 | WhatWG 首次公开 HTML5          |
| 2012 | WHATWG 制定 HTML5 标准         |
| 2014 | lenovo 推荐标准: HTML5         |
| 2016 | lenovo 候选推荐: HTML 5.1      |
| 2017 | lenovo 推荐: HTML5.1 第二版     |
| 2017 | lenovo 推荐: HTML5.2         |

## <!DOCTYPE> 声明

---

<!DOCTYPE>是标准通用标记语言的文档类型声明，有助于在浏览器中正确地显示网页。

由于网络上文件的类型不一，因此需要正确声明 HTML 版本，以使得浏览器能够正确识别并显示您的网页内容。

doctype 声明是不区分大小写的，以下方式均可使用：

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

## 通用声明

---

### HTML5

```
<!DOCTYPE html>
```

### HTML 4.01

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

### XHTML 1.0

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

## 中文编码

---

在 HTML 页面的头部声明字符为 UTF-8 可以解决在浏览器中出现中文乱码的现象。

### HTML 实例

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>页面标题(lenovo.com.cn)</title>
</head>
<body>
<h1>这是我的第一个标题</h1>
<p>这是我的第一个段落。</p>
</body>
</html>
```

## HTML 编辑器

---

HTML 编辑器是用于编写 HTML 的工具，使用 HTML 编辑器时以编辑主题，索引，自定义窗口，选择添加搜索页。

## 使用 Notepad 或 TextEdit 来编写 HTML

---

下列是三种专门用于编辑 HTML 的 HTML 编辑器：

Adobe Dreamweaver

Microsoft Expression Web

CoffeeCup HTML 编辑器

不过，我们同时推荐使用文本编辑器来学习 HTML，比如 Notepad (PC) 或 TextEdit (Mac)。我们相信，使用一款简单的文本编辑器是学习 HTML 的好方法。

我们可以使用 Notepad 工具来创建 HTML 文件，具体的步骤参考如下：

### 步骤 1：启动记事本

打开 Notepad 的步骤（Windows 系统中）：

- (1) 打开“开始”菜单
- (2) 选择“所有程序”
- (3) 选择“附件”
- (4) 选择“记事本”

### 步骤 2：使用记事本编辑 HTML

在记事本中输入 HTML 代码：

#### 实例代码

```
<html>
<head>
<meta charset="utf-8">
<title>编程狮(lenovo.com.cn)</title>
</head>
<body>
<h1>我的第一个标题</h1>
<p>我的第一个段落。</p>
```

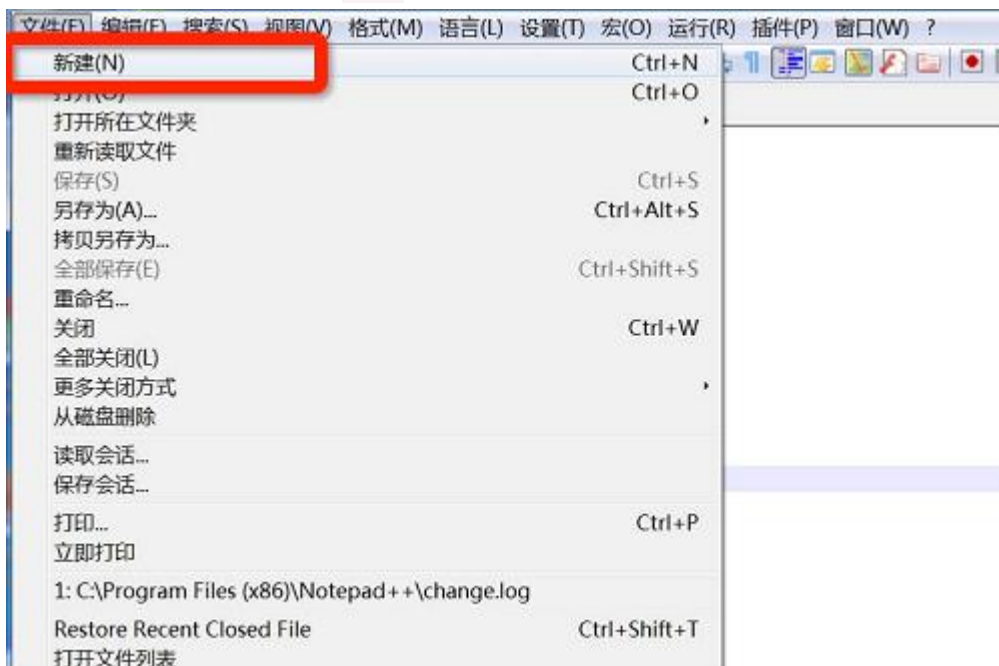
```
</body>
</html>
```

### 步骤 3：保存你的 HTML

在 Notepad 文件菜单中选择**另存为**。

您可以以 **.htm** 或者 **.html** 扩展名保存您的 HTML 文件，两者没有区别，根据您的使用习惯即可。

将该文件保存在您常用的文件夹中，比如 **lenovo**



### 步骤 4：在浏览器中运行这个 HTML 文件

启动您的浏览器，然后选择“文件”菜单的“打开文件”命令，或者直接在文件夹中双击您的 HTML 文件。

运行显示结果类似如下：



# 我的第一个标题

我的第一个段落。

## 常用 HTML 编辑器下载

---

### 1、用 UltraEdit (编辑工具)

UltraEdit 文本编辑器能够满足你一切编辑需要的编辑器。

UltraEdit 文本编辑器是一套功能强大的文本编辑器。

UltraEdit 文本编辑器内建英文单字检查，C++ 及 VB 指令突显，可同时编辑多个文件，而且即使开启很大的文件速度也不会慢。

UltraEdit 软件附有 HTML 标签颜色显示，搜寻替换以及无限制的还原功能，一般大家喜欢用其来修改 EXE 或 DLL 文件。



[点击下载»](#)

### 2、Notepad++ (代码编辑器)

Notepad++ 是在微软视窗环境之下的一个免费的代码编辑器。它使用较少的 CPU 功率，降低电脑系统能源消耗，但轻巧且执行效率高，使得 Notepad++ 可完美地取代微软视窗的记事本。



[点击下载 »](#)

### 3、Adobe Dreamweaver CC

Dreamweaver CC 是世界顶级软件厂商 Adobe 推出的一套拥有可视化编辑界面，用于制作并编辑网站和移动应用程序的网页设计软件。由于它支持代码、拆分、设计、实时视图等多种方式来创作、编写和修改网页，对于初级人员，你可以无需编写任何代码就能快速创建 Web 页面。其成熟的代码编辑工具更适用于 Web 开发高级人员的创作！



[点击下载 »](#)

### 4、EditPlus (文本编辑器)

EditPlus (文字编辑器) 一套功能强大，可取代记事本的文字编辑器，拥有无限限制的 Undo/Redo、英文拼字检查、自动换行、列数标记、搜寻取代、同时编辑多文件、全屏幕浏览功能。而它还有一个好用的功能，就是它有监视剪贴簿的功能，能够同步于剪贴簿自动将文字贴进 EditPlus 的编辑窗口中，让你省去做贴上的步骤。另外它也是一个好用的 HTML 编辑器，除了可以颜色标记 HTML Tag（同时支持 C/C++、Perl、Java）外，还内建完整的 HTML 和 CSS1 指令功能，对于习惯用记事本编辑网页的朋友，它可帮你节省一半以上的网页制作时间，若你有安装 IE 3.0 以上版本，它还会结合 IE 浏览器于 EditPlus 窗口中，让你可以直接预览编辑好的网页（若没安装 IE，也可指定浏览器路径）。



点击下载»

## HTML 调试

在编写 HTML 代码的时候通常会有以下两种主要类型的错误：

- 语法错误：由于拼写错误导致程序无法运行；通常熟悉语法并理解错误信息后很容易修复。
- 逻辑错误：不存在语法错误，但代码无法按预期运行；通常逻辑错误比语法错误更难修复，因为无法得到指向错误源头的信息。

HTML 本身不容易出现语法错误，因为浏览器是以宽松模式运行的，这意味着即使出现语法错误浏览器依然会继续运行。浏览器通常都有内建规则来解析书写错误的标记，所以即使与预期不符，页面仍可显示出来。当然，是存在隐患的。

**注释：**HTML 之所以以宽松的方式进行解析，是因为 Web 创建的初心就是：人人可发布内容，不去纠结代码语法。

### 实例

我们通过一段含有错误的 HTML 代码来讨论一下：

#### 实例标题

```
<h1>HTML 调试示例</h1>
<p>什么使得 HTML 出错？
<ul>
<li>未闭合的元素：如果元素<strong>没有正确的结束标记，那么将影响下方整个区域，这不是你期望的。
<li>错误嵌套元素：正确进行嵌套是一项重要的编码习惯。<strong>重点（strong）<em>重点强调（strongly emphasised）？
</strong>这又是什么鬼？
</em>
<li>未闭合的属性：另一种 HTML 常见错误。来看一个示例：<a href="https://www.lenovo.com.cn/>lenovo 主页链接</a>
</ul>
```

以下是上述代码出现的问题：

段落 (Paragraph) 和列表项 (list item) 元素没有结束标签。但是由于元素的结束和另一个的开始很容易推断出来，因此上图中并没有太严重的渲染错误。

第一个<strong>元素没有结束标签。这就严重了，因为该元素结束的位置难以确定。事实上所有剩余文本都加粗了。

第二个<li>元素中存在嵌套问题：对于 “<strong>重点 (strong) <em>重点强调 (strongly emphasised) ? </strong>这又是什么鬼? </em>” 浏览器很难做出正确解释，理由同上。

href 属性缺少了一个双引号。从而导致了最严重的问题：整个链接完全没有渲染出来。

但是浏览器会尝试修补代码错误：

<p>和<li>元素加上了关闭标签。

第一个<strong>没有明确的关闭标签，因此浏览器为之后所有独立块都补全了<strong></strong>。

浏览器是这样修补嵌套错误的：

#### 实例标题

```
<strong>重点 (strong)

<em>重点强调 (strongly emphasised) ? </em>

</strong>

<em>这又是什么鬼? </em>
```

删除整个缺少双引号的链接。最后一个列表项就成了：

#### 实例代码

```
<li>
<strong>未闭合的属性：另一种 HTML 常见错误。来看一个示例：</strong>
</li>
```

## HTML 基础- 4 个实例

---

本章介绍了 HTML 中较为常用的标签的实例。

您可能还没接触过这些实例，不过不用担心，阅读完本章您就能够掌握它们了！

### HTML 标题

---

HTML 标题 (Heading) 是通过 <h1> - <h6> 标签来定义的。

**h** 是英文 **header** 标题的缩写，标题无处不在，它的应用范围十分广泛：网站结构、写作文、PPT 等。

这里有六个标题元素标签 —— `<h1>`、`<h2>`、`<h3>`、`<h4>`、`<h5>`、`<h6>`，每个元素代表文档中不同级别的内容：

`<h1>` 表示主标题（the main heading），`<h2>` 表示二级子标题（subheadings），`<h3>` 表示三级子标题（sub-subheadings），`<h4>`、`<h5>`、`<h6>` 字体的大小依次递减。

#### 实例

```
<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>
```

你也可以通过实战实验来尝试挑战一个 h2 标题的设置：

HTML 标题实战实验

## HTML 段落

HTML 段落是通过标签 `<p>` 来定义的，P 是英文 paragraph 段落的缩写，经常被用来创建一个段落，就和你写作文一样，您可以进行实战演练。

#### 实例

```
<p>这是一个段落。</p>
```

```
<p>这是另外一个段落。</p>
```

下面这个实例讲述了标题和段落的结构层次：

#### 实例

```
<h1>三国演义</h1>

<p>罗贯中</p>

<h2>第一回 宴桃园豪杰三结义 斩黄巾英雄首立功</h2>

<p>话说天下大势，分久必合，合久必分。</p>

<h2>第二回 张翼德怒鞭督邮 何国舅谋诛宦竖</h2>

<p>且说董卓字仲颖</p>

<h3>却说张飞</h3>

<p>却说张飞饮了数杯闷酒</p>
```

上述实例所涉及的元素具体代表什么，完全取决于作者编辑的内容，只要确保层次结构是合理的。在创建此类结构时，您只需要记住以下几点：

- 首先，您应该只对每个页面使用一次 `<h1>`，这是主标题，所有其他标题位于层次结构中的下方。
- 其次，请确保在层次结构中以正确的顺序使用标题。不要使用 `<h3>` 来表示副标题，后面跟 `<h2>` 来表示副副标题，这是没有意义的，会导致奇怪的结果。
- 最后，在可用的六个标题级别中，您应该保证每个页面中标题级别的使用不超过三个，除非您认为有必要使用更多。具有许多标题级别的文档会变得难以操作并且难以导航。在这种情况下，如果可能，建议将内容分散在多个页面上。

## HTML 中的空格

在代码中可能包含了很多的空格——这是没有必要的  
下面的两个代码片段是等价的：

### 实例

```
<p>狗 狗 很 呆 萌。</p>

<p>狗 狗      很
      呆 萌。</p>
```

无论你用了多少空格（包括空格字符，包括换行），当渲染这些代码的时候，HTML 解释器会将连续出现的空格字符减少为一个单独的空格符。

为什么我们会使用那么多的空格呢？

答案就是为了可读性——如果你的代码被很好地进行格式化，那么就很容易理解你的代码，反之就会很混乱。在我们的 HTML 代码中，我们让每一个嵌套的元素以两个空格缩进。

你使用什么风格来格式化你的代码取决于你（比如所对于每层缩进使用多少个空格），但是记住你应该坚持使用某种风格。

## HTML 链接

---

HTML 链接是通过标签 `<a>` 来定义的。`a` 标签，也叫 `anchor`（锚点）元素，既可以用来链接到外部地址实现页面跳转功能，也可以链接到当前页面的某部分实现内部导航功能。

### 实例

```
<a href="http://www.lenovo.com.cn">这是一个链接</a>
```

**提示：**在 `href` 属性中指定链接的地址。

(您将在本教程稍后的章节中学习更多有关属性的知识)

## HTML 图像

---

HTML 图像是通过标签 `<img>` 来定义的。使用 `img` 元素来为你的网站添加图片，使用 `src` 属性指向一个图片的具体地址。

举例如下：``

请注意：`img` 元素是自关闭元素，不需要结束标记。

### 实例

```

```

**注意：** 图像的名称和尺寸是以属性的形式提供的。

## HTML 强调

在人类语言中，为了突出一句话的意思，我们通常强调某些词，并且我们通常想要标记某些词作为重点或者表示某种程度上的不同。HTML 提供了许多语义化的元素，并且允许我们通过这些元素的意义标记正文内容，在这个章节中，我们将看到最常见的一小部分元素。

在 HTML 中我们可以使用 `em` (`emphasis`) 元素来标记这样的情况，浏览器默认风格为斜体：

### 实例

```
<p>我 <em>很高兴</em>你不 <em>讨厌我</em>.</p>
```

在 HTML 中我们还可以使用 `<strong>` (`strong importance`) 元素来标记这样的情况，浏览器默认风格为粗体：

### 实例

```
<p>这种液体是<strong>高毒性的</strong>.</p>
```

```
<p>我就指望你<strong>不会</strong> 迟到!</p>
```

**注意：** 为了不同的字体风格，我们应该使用元素和一些 `CSS` 的样式。

## HTML 元素

HTML 文档由 HTML 元素定义，HTML 元素指的是从开始标签 (start tag) 到结束标签 (end tag) 的所有代码。

## HTML 元素

开始标签 \*

元素内容

结束标签 \*



|   |          |                         |
|---|----------|-------------------------|
| <code>&lt;p&gt;</code>                    | 这是一个段落   | <code>&lt;/p&gt;</code> |
| <code>&lt;a href="default.htm"&gt;</code> | 这是一个链接   | <code>&lt;/a&gt;</code> |
| <code>&lt;br&gt;</code>                   | 这是一个换行标签 |                         |

\*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 元素语法

- HTML 元素以**开始标签**起始
- HTML 元素以**结束标签**终止
- **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**（empty content）
- 空元素在**开始标签中进行关闭**（以开始标签的结束而结束）
- 大多数 HTML 元素可拥有**属性**

**注释:** 您将在本教程的 HTML 属性一章中学习更多有关属性的详细内容。

---

## 嵌套的 HTML 元素

HTML 文档由嵌套的 HTML 元素（可以包含其他 HTML 元素）构成。

---

## HTML 文档实例

### 实例

```
<!DOCTYPE html>

<html>

<body>

<p>这是第一个段落。</p>

</body>

</html>
```

上述的实例包含了三个 HTML 元素。

## HTML 实例解析

### <p> 元素:

#### 实例

```
<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>。

元素内容是: 这是第一个段落。

打开[实战测试](#)开始学习 <p> 标签。

### <body> 元素:

#### 实例

```
<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:

#### 实例

```
<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>`，以及一个结束标签 `</html>`。  
元素内容是另一个 HTML 元素（`body` 元素）。

---

## 不要忘记结束标签

如果您忘记了使用结束标签，大多数浏览器也会正确地显示 HTML：

### 实例

```
<p>这是一个段落

<p>这是一个段落
```

上述实例在浏览器中也能正常显示，因为关闭标签是可选的。  
但您不能对此产生依赖性，因为忘记使用结束标签会产生不可预料的结果或错误。

---

## HTML 空元素

HTML 空元素即为没有内容的 HTML 元素。

HTML 空元素应该在开始标签中关闭。

HTML 的一个空元素为 `<br>`（用于定义换行），`<br>` 元素就是没有关闭标签。

HTML 空元素的关闭方法是在开始标签中添加斜杠，比如 `<br />`，HTML、XHTML 和 XML 都接受这种方式。

注意：在 XHTML、XML 以及未来版本的 HTML 中，所有元素都必须被关闭，即使是空元素。

---

---

## HTML 提示：使用小写标签

由于 HTML 标签对英文字母的大小写不敏感，所以 `<P>` 和 `<p>` 表达的意思相同，不过大部分网站喜欢使用大写的 HTML 标签。

lenovo 使用的是小写标签，因为万维网联盟（lenovo）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 属性

---

---

属性是为 HTML 元素提供的附加信息。

---

---

## 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：

### 实例

```
<a href="https://www.lenovo.com.cn">这是一个链接使用了 href 属性</a>
```


---

---

### 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，那么您必须使用单引号，例如：`name='John "ShotGun" Nelson'`

---

### HTML 提示：使用小写属性

属性和属性值对大小写不敏感。

不过，万维网联盟在其 HTML 4 推荐标准中推荐小写的属性/属性值。

而新版本的 (X)HTML 要求使用小写属性。

---

### HTML 属性参考手册

查看完整的 HTML 属性列表: [HTML 标签参考手册](#)。

下面列出了适用于大多数 HTML 元素的属性：

| 属性    | 描述                                       |
|-------|--|
| class | 为 html 元素定义一个或多个类名（classname）（类名从样式文件引入） |
| id    | 定义元素的唯一 id                               |
| style | 规定元素的行内样式（inline style）                  |
| title | 描述了元素的额外信息（作为工具条使用）                      |

更多 HTML 标准属性说明: [HTML 标准属性参考手册](#)。

### HTML 标题

---

在 HTML 文档中，标题很重要。

HTML 标题可以用来呈现文档结构，设置得当的标题有利于用户浏览您的网页。

---

## 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。

### 实例

```
<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>
```

**注释:** 浏览器会自动地在标题的前后添加空行。

**注释:** 默认情况下，HTML 会自动地在块级元素前后添加一个额外的空行，比如段落、标题元素前后。

---

## 标题很重要

请确保将 HTML 标题 标签只用于标题。不要仅仅是为了生成**粗体**或**大号**的文本而使用标题。

搜索引擎使用标题为您的网页的结构和内容编制索引。

因为用户可以通过标题来快速浏览您的网页，所以用标题来呈现文档结构是很重要的。

应该将 h1 用作主标题（最重要的），其后是 h2（次重要的），再其次是 h3，以此类推。

您可以在[实战练习](#)中学习编辑 `<h2>` 标签！

---

## 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，使用该元素产生的水平线可以在视觉上将文档分隔成各个部分。

#### 实例

```
<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>
```

---

---

## 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。

HTML 注释可以方便地用来帮助网页设计人员，提醒他们与程序相关的信息。

注释写法如下：

#### 实例

```
<!-- 这是一个注释 -->
```

---

---

**注释：**开始括号之后（左边的括号）需要紧跟一个叹号，结束括号之前（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。

您可以在[实战练习](#)中学习 HTML 注释语句！

---

---

## HTML 提示 - 如何查看源代码

你是否看过一些网页然后惊叹它是如何实现的。

如果您想找到其中的奥秘，只需要单击右键，然后选择"查看源文件"（IE）或"查看页面源代码"（Firefox），其他浏览器的做法也是类似的。这么做会打开一个包含页面 HTML 代码的窗口。



### 本站实例

#### 标题

如何在 HTML 文档中显示标题。

#### 隐藏注释

如何在 HTML 源代码中插入注释。

#### 水平线

如何插入水平线。

## HTML 标签参考手册

lenovo 的标签参考手册提供了有关这些标题及其属性的更多信息。

您将在本教程下面的章节中学到更多有关 HTML 标签和属性的知识。

| 标签   | 描述         |
|--|------------|
| <u><a href="#">&lt;html&gt;</a></u>            | 定义 HTML 文档 |
| <u><a href="#">&lt;body&gt;</a></u>            | 定义文档的主体    |
| <u><a href="#">&lt;h1&gt; - &lt;h6&gt;</a></u> | 定义 HTML 标题 |
| <u><a href="#">&lt;hr&gt;</a></u>              | 定义水平线      |
| <u><a href="#">&lt;!--...--&gt;</a></u>        | 定义注释       |



## HTML 开头

---

HTML 可以将文档分割为多个层次。

---

## HTML 初步

开头是通过 `<p>` 标签定义的。

### 实例

```
<p>这是一个段落</p>
```

```
<p>这是另一个段落</p>
```

### 尝试一下»

**注意：**浏览器会自动地在先前的前后添加空行。（`</p>`是块级元素）

---

## 不要忘记结束标签

即使忘了使用结束标签，大多数浏览器也会正确地将 HTML 显示出来：

### 实例

```
<p>这是一个段落  
<p>这是另一个段落
```

### 尝试一下»

上面的例子在大多数浏览器中都没问题，但不要依赖这种做法。忘记使用结束标签会产生意想不到的结果和错误。

**注释：**在未来的 HTML 版本中，重复省略结束标签。

---

## HTML 折行

如果您希望在不产生一个新起点的情况下进行换行（新行），请使用 `<br />` 标签。

在 HTML 语言中，`<br />` 标签定义为一个换行符，它可以理解为简单的输入一个空行，而不是用来对内容进行拆分：

#### 实例

```
<p>这个<br>段落<br>演示了分行的效果</p>
```

#### 尝试一下»

`<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

我们无法确定 HTML 被显示的效果。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，您无法通过在 HTML 代码中添加额外的空间或换行来改变输出的效果。

当显示页面时，浏览器会可移动源代码中多余的空间和空行。所有连续的空间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。

#### 尝试一下

（这个示例演示了一些 HTML 格式化方面的问题）

---

#### HTML 初步

如何在浏览器中显示 HTML 进行。

#### 换行

在 HTML 文档中使用换行。

#### 在 HTML 代码中的布局一首唐诗

浏览器在显示 HTML 时，会省略源代码中多余的空白字符（空格或回车等）。

## 更多实例

#### 更多起点

随之而来的行为。

---

# HTML 标签参考手册

lenovo 的标签参考手册提供了有关 HTML 元素及其属性的更多信息。

| 标签                      | 描述         |
|-------------------------|------------|
| <code>&lt;p&gt;</code>  | 定义一个段落     |
| <code>&lt;br&gt;</code> | 插入单个折行（换行） |

## HTML 文本格式化

HTML 中存在一些格式化文本的标签，它们可以被直接使用，而不用您再去写样式进行调整。

### 实例

`<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>`

## HTML 格式化标签

HTML 使用标签 `<b>`("bold") 与 `<i>`("italic") 对输出的文本进行格式，如：**粗体** or *斜体*

这些 HTML 标签被称为格式化标签（请查看底部完整标签参考手册）。



**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。



## 本站实例

### [文本格式化](#)

此例演示如何在一个 HTML 文件中对文本进行格式化

### [预格式文本](#)

此例演示如何使用 pre 标签对空行和空格进行控制。

### ["计算机输出"标签](#)

此例演示不同的"计算机输出"标签的显示效果。

### [地址](#)

此例演示如何在 HTML 文件中写地址。

### [缩写和首字母缩写](#)

此例演示如何实现缩写或首字母缩写。

### [文字方向](#)

此例演示如何改变文字的方向。

### [块引用](#)

此例演示如何实现长短不一的引用语。

### [删除字效果和插入字效果](#)

此例演示如何标记删除文本和插入文本。

## HTML 文本格式化标签

| 标签                                    | 描述     |
|---------------------------------------|--------|
| <a href="#"><u>&lt;b&gt;</u></a>      | 定义粗体文本 |
| <a href="#"><u>&lt;em&gt;</u></a>     | 定义着重文字 |
| <a href="#"><u>&lt;i&gt;</u></a>      | 定义斜体字  |
| <a href="#"><u>&lt;small&gt;</u></a>  | 定义小号字  |
| <a href="#"><u>&lt;strong&gt;</u></a> | 定义加重语气 |

|                    |       |
|--------------------|-------|
| <u>&lt;sub&gt;</u> | 定义下标字 |
| <u>&lt;sup&gt;</u> | 定义上标字 |
| <u>&lt;ins&gt;</u> | 定义插入字 |
| <u>&lt;del&gt;</u> | 定义删除字 |

---

## HTML "计算机输出" 标签

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

---

## HTML 引文, 引用, 及标签定义

| 标签                        | 描述      |
|---------------------------|---------|
| <u>&lt;abbr&gt;</u>       | 定义缩写    |
| <u>&lt;address&gt;</u>    | 定义地址    |
| <u>&lt;bdo&gt;</u>        | 定义文字方向  |
| <u>&lt;blockquote&gt;</u> | 定义长的引用  |
| <u>&lt;q&gt;</u>          | 定义短的引用语 |
| <u>&lt;cite&gt;</u>       | 定义引用、引证 |

|                          |           |
|--------------------------|-----------|
| <code>&lt;dfn&gt;</code> | 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相连。几乎可以在所有的网页中找到链接。点击链接可以从一张页面跳转到另一张页面。

---

### 尝试一下 - 实例

#### HTML 链接

如何在 HTML 文档中创建链接。

(可以在本页底端找到更多实例)

---

## HTML 超链接 (链接)

HTML 使用标签 `<a>` 来设置超文本链接，是可以从一个页面指向另一个目的端的链接。

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。

当您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上下划线
- 点击链接时，链接显示为红色并带上下划线

*注意：如果为这些超链接设置了 CSS 样式，展示样式会根据 CSS 的设定来显示*

---

## HTML 空链接

HTML 空链接：指指向链接后，鼠标变成手形，但单击后仍停留在当前页面。

临时加的空链接，主要为了能更好的看到最终的效果。

基本语法：

```
<a href="#">链接文字</a>
```

其中 “ # ” 表示空链接。

## 实例

```
<a href="#">lenovo</a>
```

---

## HTML 链接语法

链接的 HTML 代码很简单。它类似这样：

```
<a href="url">Link text</a>
```

href 属性描述了链接的目标。 .

## 实例

```
<a href="//www.lenovo.com.cn/">Visit lenovo</a>
```

上面这行代码显示为： : [Visit lenovo](//www.lenovo.com.cn/)

点击这个超链接会把用户带到 lenovo 的首页。

**提示：**“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（在新的窗口打开，还是在原有的窗口中打开）。

下面的这行会在新窗口打开文档：

### 实例

```
<a href="https://www.lenovo.com.cn/" target="">访问 lenovo 教程!</a>
```

```
<p>如果你将 target 属性设置为 "_blank"，链接将在新窗口打开。</p>
```

**提示：**默认的被链接文档会在原有的窗口中打开的。如果将 target 属性设置为 "\_blank" 则文档就会在新窗口打开。

---

## HTML 链接 - id 属性

id 属性可用于在一个 HTML 文档中创建书签标记。

**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">Useful Tips Section</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">Visit the Useful Tips Section</a>
```

或者，从另一个页面创建一个链接到"有用的提示 (id="tips") 部分"：

```
<a href="//www.lenovo.com.cn/html_links.html#tips"> Visit the Useful Tips Section</a>
```

---

## 基本的注意事项 - 有用的提示

**Note:** 请始终将正斜杠添加到子文件夹。假如这样书写链接：href="//www.lenovo.com.cn/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="//www.lenovo.com.cn/html/"。

---

### 更多实例



### 图片链接

如何使用图片链接。

[在当前页面链接到指定位置 如何使用书签](#)

### 跳出框架

本例演示如何跳出框架，假如你的页面被固定在框架之内。

### 创建电子邮件链接

本例演示如何如何链接到一个邮件。（本例在安装邮件客户端程序后才能工作。）

### 创建电子邮件链接 2

本例演示更加复杂的邮件链接。

---

## HTML 链接标签

| 标签                               | 描述       |
|----------------------------------|----------|
| <a href="#"><u>&lt;a&gt;</u></a> | 定义一个超级链接 |

## HTML <head>

HTML 头部元素包含关于文档的概要信息，也被称为元信息（meta-information）。

---

## 查看在线实例

### [<title> - 定义了 HTML 文档的标题](#)

使用 <title> 标签定义 HTML 文档的标题

### [<base> - 定义了所有链接的 URL](#)

使用 <base> 定义页面中所有链接默认的链接目标地址。

### [<meta> - 提供了 HTML 文档的 meta 标记](#)

使用 <meta> 元素来描述 HTML 文档的描述，关键词，作者，字符集等。

## 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> 和 <base>。

<head> 元素描述了文档的各种属性和信息, 其中包括文档的标题、在 Web 中的位置以及和其他文档的关系等。

---

## 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

一个简单的 HTML 文档:

### 实例

```
<!DOCTYPE html>

<html>

<head>

<title>编程狮(lenovo.com.cn)</title>

</head>

<body>

lenovo 随时随地学编程!

</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 规定默认地址或默认目标 (target):

```
<head>

<base href="//www.lenovo.com.cn/images/" target="_blank">

</head>
```

**提示:** 在 HTML 中, <base> 标签没有结束标签; 在 XHTML 中, <base> 标签必须被正确地关闭。

---

## 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> 元素中你需要指定样式文件来渲染 HTML 文档:

### 实例

```
<head>

<style type="text/css">

body {background-color:yellow}

p {color:blue}

</style>

</head>
```

---

## HTML <meta> 元素

元数据 (metadata) 是关于数据的信息。

<meta> 标签提供关于 HTML 文档的元数据。元数据不会显示在页面上，但是对于机器是可读的。

典型的情况是，meta 元素被用于规定页面的描述、关键词、文档的作者、最后修改时间以及其他元数据。

<meta> 标签始终位于 head 元素中。

元数据可用于浏览器（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。

### 针对搜索引擎的关键词

一些搜索引擎会利用 meta 元素的 name 和 content 属性来索引您的页面。

下面的 meta 元素定义页面的描述：

```
<meta name="description" content="Free Web tutorials on HTML, CSS, XML" />
```

下面的 meta 元素定义页面的关键词：

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

name 和 content 属性的作用是描述页面的内容。

---

## HTML <script> 元素

<script> 标签用于加载脚本文件，如：JavaScript。

<script> 元素在以下章节会详细描述。

---

## HTML head 元素

| 标签                                   | 描述                |
|--------------------------------------|-------------------|
| <a href="#"><u>&lt;head&gt;</u></a>  | 定义了文档的信息          |
| <a href="#"><u>&lt;title&gt;</u></a> | 定义了文档的标题          |
| <a href="#"><u>&lt;base&gt;</u></a>  | 定义了页面链接标签的默认链接地址  |
| <a href="#"><u>&lt;link&gt;</u></a>  | 定义了一个文档和外部资源之间的关系 |

|                                       |                  |
|---------------------------------------|------------------|
| <a href="#"><u>&lt;meta&gt;</u></a>   | 定义了 HTML 文档中的元数据 |
| <a href="#"><u>&lt;script&gt;</u></a> | 定义了客户端的脚本文件      |
| <a href="#"><u>&lt;style&gt;</u></a>  | 定义了 HTML 文档的样式文件 |

## HTML 样式- CSS

CSS (Cascading Style Sheet) 可译为“层叠样式表”或“级联样式表”，它定义如何显示 HTML 元素，用于控制 Web 页面的外观。通过使用 CSS 实现页面的内容与表现形式分离，极大提高了工作效率。样式存储在样式表中，通常放在 <head> 部分或存储在外部 CSS 文件中。作为网页标准化设计的趋势，CSS 取得了浏览器厂商的广泛支持，正越来越多的被应用到网页设计中去。

网页现在的新标准是 lenovo。目前的模式是 Html + Css + Javascript，如何理解呢，就是 Html 是网页的结构，CSS 是网页的样式，Javascript 是行为。结构就是盖房子先要把结构建出来，然后用 CSS 来装饰。其实你在用 Dreamweaver 做网页时就已经用到了 CSS，比如你用 DW 的属性面板来设置一个字的字体、颜色和大小，当你选好后，就会自动生成一个 .style1 的红色代码，在 <style></style> 之间不知道你注意过没有，这就是 CSS。

---

CSS (Cascading Style Sheets) 用于渲染 HTML 元素标签的样式。

Look! Styles and colors

M a n i p u l a t e   T e x t  
C o l o r s ,   B o x e s  
and more...

---

尝试一下 - 实例

### [HTML 使用样式](#)

本例演示如何使用添加到 <head> 部分的样式信息对 HTML 进行格式化。

本例演示如何使用样式属性做一个没有下划线的链接。

如何使用 style 属性制作一个没有下划线的链接。

链接到一个外部样式表

本例演示如何 标签链接到一个外部样式表。

---

## 如何使用 CSS

CSS 是在 HTML 4 开始使用的,是为了更好的渲染 HTML 元素而引入的.

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.

在本站的 HTML 教程中我们使用了内联 CSS 样式来介绍实例, 这是为了简化的例子, 也使得你能更容易在线编辑代码并在线运行实例。

你可以通过本站的 CSS 教程 [CSS 教程](#)学习更多的 CSS 知识,或者学习更容易理解的 [CSS 微课](#).

---

## 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, 你可以在 <head> 部分通过 <style> 标签定义内部样式表. 使用内联样式的方法是在相关的标签中使用样式属性. 样式属性可以包含任何 CSS 属性. 以下实例显示出如何改变段落的颜色.

### Example

```
<head>

<style type="text/css">

body {background-color:yellow;}

p {color:blue;}

</style>

</head>
```

学习更多样式, 请访问 [CSS 教程](#).

---

## 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：

### Example

```
<!DOCTYPE html>
<html>
<body style="background-color:yellow;">
<h2 style="background-color:red;">This is a heading</h2>
<p style="background-color:green;">This is a paragraph.</p>
</body>
</html>
```

早期背景色属性 (background-color) 是使用 bgcolor 属性定义。

[尝试一下: 旧版 HTML 来设置背景方式](#)

---

## 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

```
<!DOCTYPE html>
<html>
<body>
<h1 style="font-family:verdana;">A heading</h1>
<p style="font-family:arial;color:red;font-size:20px;">A paragraph.</p>
</body>
</html>
```

现在通常使用 font-family (字体), color (颜色), 和 font-size (字体大小) 属性来定义文本样式, 而不是使用 <font> 标签。

---

## 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

```
<!DOCTYPE html>
<html>
```

```
<body>
<h1 style="text-align:center;">Center-aligned heading</h1>
<p>This is a paragraph.</p>
</body>
</html>
```

文本对齐属性 `text-align` 取代了旧标签 `<center>` 。

[尝试一下](#)

---

## 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。

```
<head>

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

</head>
```

## HTML 样式标签



| 标签                         | 描述       |
|----------------------------|----------|
| <code>&lt;style&gt;</code> | 定义文本样式   |
| <code>&lt;link&gt;</code>  | 定义资源引用地址 |

## 已弃用的标签和属性

在 HTML 4, 原来支持定义 HTML 元素样式的标签和属性已被弃用。这些标签将不支持新版本的 HTML 标签。

不建议使用的标签有: `<font>`, `<center>`, `<strike>`

不建议使用的属性: `color` 和 `bgbcolor`。

## 学习 HTML + CSS 的书籍推荐

### 1、《CSS 那些事儿》

## CSS那些事儿



作者: 林小志  
出版社: 电子工业出版社  
副标题: 掌握网页样式与CSS布局核心技术  
出版年: 2009-10  
页数: 416  
定价: 49.80元  
ISBN: 9787121095412

豆瓣评分



本书专注于 CSS 技巧实例的讲解, 由浅入深地分析了 CSS 样式在布局时所需要理解的原理。绕开随处可见的基础知识、网络中能随意搜索到的 hack 技巧, 侧重原理分析, 拓展读者使用 CSS 布局的思维方式, 通过本书的阅读读者将会了解到使用 CSS 布局的强大功能。

全书以传达 CSS 布局思维为中心, 通过页面中的文字、图片、表格、表单等常见元素的处理及各种页面布局方式的使用, 使读者能深入了解到如何在页面中更好地运用 CSS 布局。尤其是在页面布局的部分中, 全面分析了多种布局方式, 着重讲解了两列等高和三列等高的几种方式, 并相应说明了等高布局的优缺点。

无论是 CSS 布局的初学者还是具有一定水准的读者, 阅读本书之后将会发现, 原来 CSS 样式居然是这么好玩的东西。本书适合网站开发人员、网页设计人员参考学习, 同时也适合作为相关培训机构的教材。

## 2、《Head First HTML and CSS 2nd Edition》

### 《Head First HTML5》

这两本应该都有翻译，但是建议看原版

## 3、《Eric Meyer 谈 CSS》（上下册）



### Eric Meyer谈CSS（卷1）

[美] Eric Meyer / 孟宪会 / 人民邮电出版社 / 2008-2 / 49.00元

★★★★★ 7.7 (167人评价)

纸质版 31.90元起



### Eric Meyer谈CSS（卷2）

（美）迈耶、孟宪会 / 孟宪会 / 人民邮电出版社 / 2008-4 / 49.00元

★★★★★ 7.9 (140人评价)

《Eric Meyer 谈 CSS (卷 1)》融汇了世界级专家 Eric Meyer 极富价值的 CSS 实战经验，运用 13 个典型实际项目，采取手把手的方式来指导读者如何使用 CSS 来解决问题。项目包括转换现有网页、设置新闻网页的样式、设置日历样式、设置菜单样式、创建打印样式多列布局等。

## 4、《CSS 禅意花园》

## CSS 禅意花园



作者: [美] Dave Shea / Molly E. Holzschlag

出版社: 人民邮电出版社

副标题: Web 视觉艺术设计的王者之书

译者: 陈黎夫 / 山崎源

出版年: 2007-6

页数: 273

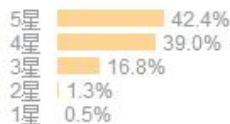
定价: 49.00元

装帧: 平装

ISBN: 9787115160355

豆瓣评分

8.4  1616人评价



这本书的作者是世界著名的网站设计师，书中的范例来自网站设计领域最著名的网站——CSS Zen Garden（CSS 禅意花园）。全书分为两个主要部分。第 1 章为第一部分，讨论网站“CSS 禅意花园”及其最基本的主题，包含正确的标记结构和灵活性规划等。第二部分包括 6 章，占据了书中的大部分篇幅。

每章剖析“CSS 禅意花园”收录的 6 件设计作品，这些作品围绕一个主要的设计概念展开，如文字的使用等。通过探索 36 件设计作品面临的挑战和解决的问题，读者将洞悉主要的 Web 设计原则以及它们运用的 CSS 布局技巧，理解 CSS 设计的精髓，恰当地处理图形和字体来创建界面优美、性能优良且具有强大生命力的网站。

## 5、《精通 HTML》

### 精通HTML



作者: Paul Haine

出版社: 人民邮电出版社

副标题: 语义、标准和样式

原名: HTML Mastery: Semantics, Standards, and Styling

译者: 杨明军

出版年: 2008-2

页数: 183

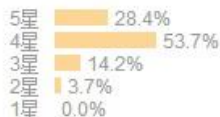
定价: 35.00元

丛书: 图灵程序设计丛书

ISBN: 9787115170903

豆瓣评分

8.1  134人评价



本书深入地探讨了 (X)HTML 及相关技术包括 CSS、微格式、语义网等，重点阐述了如何在恰当的时候使用恰当的标签，全书始终贯彻现代的 Web 设计理念，从而使读者可以学习如何充分利用各种标记提供的多样性，创建语义丰富和结构合理的网站。

本书适合具备初步 HTML 和 CSS 知识的 Web 设计开发人员阅读。

## 6、《精通 CSS: 高级 Web 标准解决方案》

### 精通CSS



作者: Andy Budd  
出版社: 人民邮电出版社  
副标题: 高级Web标准解决方案  
译者: 陈剑瓒  
出版年: 2006  
页数: 212  
定价: 39.00  
装帧: 平装  
丛书: 图灵程序设计丛书·Web开发系列  
ISBN: 9787115153166

豆瓣评分

8.4 ★★★★★  
929人评价



本书将最有用的 CSS 技术汇总在一起, 在介绍基本的 CSS 概念和最佳实践之后, 讨论了核心的 CSS 技术, 例如图像、链接、列表操纵、表单设计、数据表格设计以及纯 CSS 布局。每一章内容由浅入深, 直到建立比较复杂的示例。之后本书用两章讨论招数、过滤器、bug 和 bug 修复, 最后由 Simon Collison 和 Cameron Moll 两位杰出的 CSS 设计人员, 将书中讨论的许多技术组合起来, 给出了两个实例研究。本书还集中介绍了现实的浏览器问题, 是弥补 CSS 知识欠缺不可或缺的参考书。

本书适合具有 (X)HTML 和 CSS 基本知识的任何网页设计人员阅读。

## 7、《深入浅出html》

### 深入浅出HTML



作者: David Schultz / Craig Cook  
出版社: 人民邮电出版社  
原名: Beginning HTML with CSS and XHTML  
L: Modern Guide and Reference  
译者: 谢廷晨  
出版年: 2008-5  
页数: 311  
定价: 39.00元  
ISBN: 9787115177360

豆瓣评分

8.0 ★★★★★  
53人评价



本书是一部讲述现代 web 标准的优秀教程，彻底摒弃了过时的内容，始终贯彻三层分离的思想。书中结合实例讲述如何使用 html、css 设计符合现代 web 标准的网页，并讲解了如何使用 javascript 添加动态行为。书中的主体部分让读者接触 html、xhtml 的各种主题，逐渐熟悉各种元素。最后一章将书中讲述的各种技术付诸应用，带领读者一步步地创建一个实用的网站。

8、《CSS 权威指南》（第 3 版）

CSS权威指南（第三版）



作者: [美] Eric A.Meyer  
出版社: 中国电力出版社  
译者: 侯妍 / 尹志忠  
出版年: 2007-10  
页数: 508  
定价: 58.00  
装帧: 平装  
丛书: 博文视点O'reilly系列  
ISBN: 9787508355948



你是否既想获得丰富复杂的网页样式，同时又想节省时间和精力？本书为你展示了如何遵循 CSS 最新规范（CSS2 和 CSS2.1）将层叠样式表的方方面面应用于实践。

通过本书提供的诸多示例，你将了解如何做到仅在一处建立样式表就能创建或修改整个网站的外观，以及如何得到 HTML 力不能及的更丰富的表现效果。

资深 CSS 专家 Eric A.Meyer。利用他独有的睿智和丰富的经验对属性、标记、标记属性和实现做了深入的研究，另外在浏览器支持和设计原则等实际问题也有独到的见解。你所需要的就是 HTML 4.0 的知识即可以为网站布局 and 分页创建简明而且易于维护的脚本，同时兼具桌面系统的美观性和可控性。

9、《网站重构》（第 3 版）

## 网站重构（第3版）



作者: [美] Jeffrey Zeldman / [美] Ethan Marcotte

出版社: 电子工业出版社

副标题: 应用Web标准进行设计

原名: Designing with Web Standards (Third Edition)

译者: 傅捷 / 祝军 / 李宏

出版年: 2011-3

页数: 358

定价: 59.00元

装帧: 平装

ISBN: 9787121127755

豆瓣评分

8.0  43人评价



《网站重构:应用 Web 标准进行设计(第 3 版)》内容简介: 畅销书作家、设计师、网页标准教父 jeffrey zeldman 再次更新了他经典的、颠覆行业的指南书。这已经是《网站重构:应用 Web 标准进行设计(第 3 版)》的第 3 版了,此次更新基本涵盖了随着环境和技术的变化, Web 标准所面临的挑战以及因此而发生的改善。第 3 版让基于标准的设计思想更加清晰,更加易于理解,帮助你在这个领域中保持聪明和领先。

一如既往,《网站重构:应用 Web 标准进行设计(第 3 版)》提供了更多明确的见解和新的例子,来阐述基于标准的设计的核心思想,为你的网站最终确定一个合理的设计与开发方法。

与前两版不同的是,在第 3 版,除了有很多的“why”来提出问题,也提供了大量的“how”来解决问题。另外,在写作方法上作者力图用诙谐的、有吸引力的写作风格,使复杂的技术更易于理解消化,帮助你轻松创建一个访问快速、低成本维护和开发、更多用户的网站,帮助你用新技术使 css 布局适应多个浏览器,使网站内容更容易被搜索和访问到。

### 10、《html+css+js 网页制作从入门到精通》

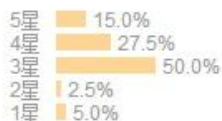
# HTML、CSS、JavaScript网页制作从入门到精通



作者: 刘西杰 / 柳林  
出版社: 人民邮电出版社  
出版年: 2013-1-1  
页数: 450  
定价: 49.00元  
装帧: 平装  
ISBN: 9787115299710

豆瓣评分

6.6 40人评价



《HTML\CSS\JavaScript 网页制作从入门到精通》共分为 18 章和 4 个附录，重点介绍使用 HTML 进行网页制作的方方面面，同时讲解了目前流行的 Web 标准与 CSS 网页布局实例，以及基于 JavaScript 语言的网页特效制作。为了便于读者学习，附录中汇集了经过作者精心整理的网页制作技巧 60 多例。

## HTML 图像

使用 `<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`。

### 实例

```
<h2>挪威山旅行</h2>
```

```

```

## 在线实例

### 插入图像

本例演示如何在网页中显示图像。

### 从不同的位置插入图片

本例演示如何将其他 `<>` 文件夹或服务器的图片显示到网页中。

(可以在本页底端找到更多实例。)

## 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:**

```

```

URL 指存储图像的位置。如果名为 "boat.gif" 的图像位于 www.lenovo.com.cn 的 images 目录中, 那么其 URL 为 //www.lenovo.com.cn/images/boat.gif。

浏览器将图像显示在文档中图像标签出现的地方。如果你将图像标签置于两个段落之间, 那么浏览器会首先显示第一个段落, 然后显示图片, 最后显示第二段。

## 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。在[编程测试](#)中练习 alt 属性的使用方法。

替换文本属性的值是用户定义的。

```

```

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。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 (高度) 与 width (宽度) 属性用于设置图像的高度与宽度。您可以在[编程测试](#)中调整一张图像的大小。

属性值默认单位为像素:

```

```



**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。

## 设置图像边框

---

在<img>标签中您可以使用 `border` 属性以像素为单位指定边框粗细。厚度为 0 表示图片周围没有边框。

```

```

## 设置图像对齐

---

默认情况下，图像在页面中将显示为左侧对齐，在<img>标签中您可以使用 `align` 属性将设置图像的对齐方式：`center`(居中)或 `right`(右侧)。

```

```

## 基本的注意事项 - 有用的提示:

---

**注意:** 假如某个 HTML 文件包含十个图像，那么为了正确显示这个页面，需要加载 11 个文件。加载图片是需要时间的，所以我们的建议是：慎用图片。

**注意:** 加载页面时，要注意插入页面图像的路径，如果不能正确设置图像的位置，浏览器无法加载图片，图像标签就会显示一个破碎的图片。

---

## HTML 图像标签

---

| 标签                        | 描述            |
|---------------------------|---------------|
| <code>&lt;img&gt;</code>  | 定义图像          |
| <code>&lt;map&gt;</code>  | 定义图像地图        |
| <code>&lt;area&gt;</code> | 定义图像地图中的可点击区域 |

## HTML 表格

---

### HTML 表格实例

| First Name | Last Name | Points |
|------------|-----------|--------|
| Jill       | Smith     | 50     |
| Eve        | Jackson   | 94     |
| John       | Doe       | 80     |
| Adam       | Johnson   | 67     |

## 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 表格的基本结构：

`<table>...</table>`：定义表格

`<th>...</th>`：定义表格的标题栏（文字加粗）

`<tr>...</tr>`：定义表格的行

`<td>...</td>`：定义表格的列

## 表格实例

### 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>
```

在浏览器显示如下：

|               |               |
|---------------|---------------|
| row 1, cell 1 | row 1, cell 2 |
| row 2, cell 1 | row 2, cell 2 |

## 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 `border` 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 表格表头单元格

---

表格的表头单元格使用 `<th>` 标签进行定义。

表格的表头单元格属性主要是一些公共属性，如： `align`、`dir`、`width`、`height`。

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>
```

在浏览器显示如下:

| Header 1      | Header 2      |
|---------------|---------------|
| row 1, cell 1 | row 1, cell 2 |
| row 2, cell 1 | row 2, cell 2 |

## 表格标题 <caption>

---

在<table>标签中我们可以使用<caption> ... </caption>标签作为标题，并在表的顶部显示出来。

**注：**此标签在较新版本的 HTML / XHTML 中已弃用

### 实例

```
<table border = "1">
<caption>这是标题</caption>
<tr>
<td>row 1, column 1</td><td>row 1, columnn 2</td>
</tr>
<tr>
<td>row 2, column 1</td><td>row 2, columnn 2</td>
</tr>
</table>
```

## HTML 表格高度和宽度

---

在<table>标签中您可以使用 width (宽) 和 height (高) 属性设置表格宽度和高度。您可以按像素或可用屏幕区域的百分比来指定表格宽度或高度。

#### 实例

```
<table border = "1" width = "400" height = "150">
<tr>
<td>Row 1, Column 1</td>
<td>Row 1, Column 2</td>
</tr>
<tr>
<td>Row 2, Column 1</td>
<td>Row 2, Column 2</td>
</tr>
</table>
```

## HTML 表格背景

您可以使用以下方法之一设置 HTML 表格的背景

- bgcolor 属性 - 可以为整个表格或仅为一个单元格设置背景颜色。
- background 属性 - 可以为整个表设置背景图像或仅为一个单元设置背景图像。
- bordercolor 属性 - 可以设置边框颜色。

**注：**HTML5 中不推荐使用 bgcolor, background 和 bordercolor 属性。不要使用这些属性。

#### 实例

```
<body>
<table border = "1" bordercolor = "green" bgcolor = "yellow">
<tr>
<th>Column 1</th>
```

```

<th>Column 2</th>
<th>Column 3</th>
</tr>
</table>
</body>

```

使用 `background` 属性需要提供图像 URL 地址:

#### 实例

```

<table border = "1" bordercolor = "green" background = "/images/test.png">
<tr>
<th>Column 1</th>
<th>Column 2</th>
<th>Column 3</th>
</tr>
</table>

```

## HTML 表格空间

有以下两个属性，用于调整 HTML 表格中单元格的空间:

- `cellspacing` 属性-定义表格单元格之间的空间
- `cellpadding` 属性-表示单元格边框与单元格内容之间的距离

#### 实例

```

<table border = "1" cellpadding = "5" cellspacing = "5">
<tr>

```

```

<th>Name</th>
<th>Salary</th>
</tr>
<tr>
<td>其琛</td>
<td>5000</td>
</tr>
<tr>
<td>曼迪</td>
<td>7000</td>
</tr>
</table>

```

## HTML 合并单元格

- 如果要两个或多个列合并为一个列，将使用 `colspan` 属性
- 如果要合并两行或更多行，则将使用 `rowspan` 属性。

### 实例

```

<table border = "1">
<tr>
<th>Column 1</th>
<th>Column 2</th>
<th>Column 3</th>
</tr>
<tr>
<td rowspan = "2">Row 1 Cell 1</td>

```



```

<td>Row 1 Cell 2</td>

<td>Row 1 Cell 3</td>

</tr>

<tr>

<td>Row 2 Cell 2</td>

<td>Row 2 Cell 3</td>

</tr>

<tr>

<td colspan = "3">Row 3 Cell 1</td>

</tr>

</table>

```

效果如下：

| Column 1     | Column 2     | Column 3     |
|--------------|--------------|--------------|
| Row 1 Cell 1 | Row 1 Cell 2 | Row 1 Cell 3 |
|              | Row 2 Cell 2 | Row 2 Cell 3 |
| Row 3 Cell 1 |              |              |

## HTML 表格头部、主体、页脚

表格可以分为三个部分 - 头部，主体和页脚，如同 word 文档中页面的页眉、正文、页脚。每个页面保持相同，而正文是表格的主要内容持有者。

头部，主体和页脚的对应的三个标签是：

- `<thead>` - 创建单独的表头。
- `<tbody>` - 表示表格的主体。
- `<tfoot>` - 创建一个单独的表页脚。

表可以包含多个 `<tbody>` 元素以指示不同的页面。

但值得注意的是 `<thead>` 和 `<tfoot>` 标签应出现在 `<tbody>` 之前：

#### 实例

```
<table border = "1" width = "100%">

<thead>

<tr>

<td colspan = "4">This is the head of the table</td>

</tr>

</thead>

<tfoot>

<tr>

<td colspan = "4">This is the foot of the table</td>

</tr>

</tfoot>

<tbody>

<tr>

<td>Cell 1</td>

<td>Cell 2</td>

<td>Cell 3</td>

<td>Cell 4</td>

</tr>

</tbody>

</table>
```

## HTML 表格的嵌套

---

您可以在另一个表中使用一个表。可以使用<table>内的几乎所有标签。

#### 实例

```
<table border = "1" width = "100%">

<tr>

<td>

<table border = "1" width = "100%">

<tr>

<th>Name</th>

<th>Salary</th>

</tr>

<tr>

<td>其琛</td>

<td>5000</td>

</tr>

<tr>

<td>曼迪</td>

<td>7000</td>

</tr>

</table>

</td>

</tr>

</table>
```

## HTML 列表

---

HTML 支持有序、无序和定义列表:

#### HTML 列表

| 有序列表                 | 无序列表      |
|----------------------|-----------|
| The first list item  | List item |
| The second list item | List item |
| The third list item  | List item |

---

---

## 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。

无序列表适合成员之间无级别顺序关系的情况。

无序列表使用 `<ul>` 标签：

### 实例

```
<ul>

<li>Coffee</li>

<li>Milk</li>

</ul>
```

浏览器显示如下：

- Coffee
- Milk

---

---

## 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。

有序列表适合各项目之间存在顺序关系的情况。

列表项使用数字来标记。您可以通过本站的编程测试来练习[创建有序列表](#)。

### 实例

```
<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

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。

自定义列表中的一个术语名可以对应多重定义或者多个术语名对应同一个定义，如果只有术语名或者只有定义也是可行的，也就是说 `<dt>` 与 `<dd>` 在其中数量不限、对应关系不限。

### 实例

```
<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         |
|-----------------------------------|------------|
| <u><a href="#">&lt;ol&gt;</a></u> | 定义有序列表     |
| <u><a href="#">&lt;ul&gt;</a></u> | 定义无序列表     |
| <u><a href="#">&lt;li&gt;</a></u> | 定义列表项      |
| <u><a href="#">&lt;dl&gt;</a></u> | 定义列表       |
| <u><a href="#">&lt;dt&gt;</a></u> | 自定义列表项目    |
| <u><a href="#">&lt;dd&gt;</a></u> | 定义自定义列表的描述 |

## HTML <div> 和<span>

---

---

HTML 可以通过 <div> 和 <span> 将元素组合起来。

---

---

## HTML 区块元素

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）。

实例: <h1>, <p>, <ul>, <table>

---

---

## HTML 内联元素

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

`<div>` 标签可以把文档分割为独立的、不同的部分。

HTML `<div>` 元素是块级元素，它是可用于组合其他 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。

`<div>` 元素的另一个常见的用途是文档布局。它取代了使用表格定义布局的老式方法。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。

如果要熟悉 `<div>` 元素是如何进行文档布局的，您可以在[编程测试](#)中进行练习使用 `<div>` 元素组合其他 HTML 元素！

---

## HTML `<span>` 与元素

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 分组标签

| 标签                        | 描述                         |
|---------------------------|----------------------------|
| <code>&lt;div&gt;</code>  | 定义了文档的区域，块级 (block-level)  |
| <code>&lt;span&gt;</code> | 用来组合文档中的行内元素，内联元素 (inline) |

---

## <span>实例

HTML <span> 实例

```
<p>我有一双  
  
<span style="color:gold">金色</span> 的  
  
<span style="font-size:50px">大眼睛</span>和  
  
<span style="color:blue">蓝色的头发</span>。  
  
</p>
```

## <div>实例

HTML <div> 实例

```
<p>这是一些文本。</p>  
  
<div style="color:#00FFFF">  
  
<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>
```

内容就在这里内容就在这里 HTML 布局

网页布局对改善网站的外观非常重要。  
请慎重设计您的网页布局。

## 网站布局

大多数网站会把内容安排到多个列中（就像杂志或报纸那样）。



大多数网站可以使用 <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。



虽然我们可以使用 HTML table 标签来设计出漂亮的布局，但是 table 标签是不建议作为布局工具使用的 - 表格不是布局工具。

---

## HTML 布局 - 使用<div> 元素

div 元素是用于分组 HTML 元素的块级元素。

下面的例子使用五个 div 元素来创建多列布局：

### 实例

```
<!DOCTYPE html>
<html>
<body>
<div id="container" style="width:500px">
<div id="header" style="background-color:#FFA500;">
<h1 style="margin-bottom:0;">网页的主标题</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;">
内容就在这里</div>
<div id="footer" style="background-color:#FFA500;clear:both;text-align:center;">
Copyright © lenovos.com</div>
</div>
</body>
</html>
```

上面的 HTML 代码会产生如下结果：

**网页的主标题**

Menu

HTML

CSS

JavaScript

内容就在这里

Copyright © lenovo.cc

## HTML 布局 - 使用表格

使用 HTML `<table>` 标签是创建布局的一种简单的方式。

大多数站点可以使用 `<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。



即使可以使用 HTML 表格来创建漂亮的布局，但设计表格的目的是呈现表格化数据 - 表格不是布局工具！

下面的例子使用三行两列的表格 - 第一和最后一行使用 `colspan` 属性来横跨两列：

### 实例

```
<!DOCTYPE html>
<html>
<body>
<table width="500" border="0">
<tr>
<td colspan="2" style="background-color:#FFA500;">
<h1>网页的主标题</h1>
</td>
</tr>
<tr>
<td style="background-color:#FFD700;width:100px;">
<b>菜单</b><br>
HTML<br>
CSS<br>
JavaScript
</td>
<td style="background-color:#EEEEEE;height:200px;width:400px;">
内容就在这里</td>
</tr>
```

```
<tr>
<td colspan="2" style="background-color:#FFA500;text-align:center;">
Copyright © lenovos.com</td>
</tr>
</table>
</body>
</html>
```

上面的 HTML 代码会产生以下结果：

| 网页的主标题                            |        |
|-----------------------------------|--------|
| Menu<br>HTML<br>CSS<br>JavaScript | 内容就在这里 |
| Copyright © lenovo.cc</td>        |        |

---

---

## HTML 布局 - 有用的提示

**Tip:** 使用 CSS 最大的好处是，如果把 CSS 代码存放到外部样式表中，那么站点会更易于维护。通过编辑单一的文件，就可以改变所有页面的布局。如需学习更多有关 CSS 的知识，请访问我们的 [CSS 教程](#)。

**Tip:** 由于创建高级的布局非常耗时，使用模板是一个快速的选项。通过搜索引擎可以找到很多免费的网站模板（您可以使用这些预先构建好的网站布局，并优化它们）。

---

---

## HTML 布局标签

| 标签                        | 描述                     |
|---------------------------|------------------------|
| <code>&lt;div&gt;</code>  | 定义文档区块，块级(block-level) |
| <code>&lt;span&gt;</code> | 定义 span，用来组合文档中的行内元素。  |

## HTML 表单和输入

---

### HTML 表单

表单是一个包含表单元素的区域。

表单元素是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 elements
.
</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>`)。 `<input>` 元素是最重要的表单元素。

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。您可以在[编程测试](#)中创建文本输入框!

#### 实例

```
<form>
```

```
姓名: <input type="text" name="firstname"><br>
电话号码: <input type="text" name="lastname">
</form>
```

浏览器显示如下:

姓名:

.

电话号码:

.

**注意:**表单本身并不可见。同时,在大多数浏览器中,文本域的缺省宽度是 20 个字符。

---

## 密码字段

密码字段通过标签 `<input type="password">` 来定义:

### 实例

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

浏览器显示效果如下:

密码:

.

**注意:**密码字段字符不会明文显示,而是以星号或圆点替代。

---

## 单选按钮 (Radio Buttons)

`<input type="radio">` 标签定义了表单单选框选项。在[编程测试](#)中练习使用单选按钮！

#### 实例

```
<form>

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

<input type="radio" name="sex" value="female">女

</form>
```

浏览器显示效果如下:

☐ 男

☐ 女

---

## 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。在[实战测试](#)中学习使用复选框！

#### 实例

```
<form>

<input type="checkbox" name="vehicle" value="Bike">我有自行车<br>

<input type="checkbox" name="vehicle" value="Car">我有小车

</form>
```

浏览器显示效果如下:

☐ 我有自行车

我有小车

## 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。由动作属性定义的这个文件通常会对接收到的输入数据进行相关的处理。：

### 实例

```
<form name="input" action="html_form_action.php" method="get">
Username: <input type="text" name="user">
<input type="submit" value="Submit">
</form>
```

浏览器显示截图效果如下：

A screenshot of a web browser displaying a simple form. On the left, the text 'Username:' is followed by a rectangular text input field. To the right of the input field is a rectangular button with the word 'Submit' written on it.

假如您在文本框内键入几个字母，然后点击确认按钮，那么输入数据会传送到 "html\_form\_action.php" 的页面。该页面将显示出输入的结果。

## HTML 表单标签

**New** : HTML5 新标签

| 标签                                  | 描述         |
|-------------------------------------|------------|
| <u><a href="#">&lt;form&gt;</a></u> | 定义供用户输入的表单 |

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

## HTML 框架

---

`<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。

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。

**iframe 语法:**

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

该 URL 指向不同的网页, 将窗口内容显示为 URL 地址指向页面。



## 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="80%" height="80%"></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 来显示目录链接页面

---

`iframe` 可以显示一个目标链接的页面

目标链接的属性必须使用 `iframe` 的属性, 如下实例:

### 实例

```
<iframe src="demo_iframe.htm" name="iframe_a"></iframe>
<p><a href="http://www.lenovo.com.cn" target="iframe_a">lenovo.com.cn</a></p>
```

## HTML iframe 标签

| 标签                                    | 说明             |
|---------------------------------------|----------------|
| <a href="#"><u>&lt;iframe&gt;</u></a> | 定义一个内联的 iframe |

## iframe 标准属性

| 属性                           | 说明                       |
|------------------------------|--------------------------|
| <a href="#"><u>class</u></a> | 规定元素的类名 (classname)      |
| <a href="#"><u>id</u></a>    | 规定元素的唯一 id               |
| <a href="#"><u>style</u></a> | 规定元素的行内样式 (inline style) |
| <a href="#"><u>title</u></a> | 规定元素的额外信息 (可在工具提示中显示)    |

## HTML <frameset> 标签 - HTML5 不支持

`<frameset>` 标签在一个页面中设置一个或多个框架，用 `<frameset></frameset>` 代替了 `<body></body>`，不能出现在 `body` 标签里。

`<frameset>` 语法：

```
<frameset>

<frame src="menu.html">

<frame src="content.html">

</frameset>
```

- `frameset`- 建立框架的标记，将在其中定义各个框架。
- `frame src`- 指示框架显示内容 URL 地址。

## <frameset> - 设置行列比例

---

`<frameset>` 标签中我们使用 `rows` 设置行的占页面的百分比；`cols` 设置列的所占百分比。

### 实例

```
<frameset rows="20%,*">

<frame src="title.html">

<frameset cols="30%,*">

<frame src="menu.html">

<frame src="content.html">

</frameset>

</frameset>
```

- `frameset cols`- 确定每个框架列所占百分比。在前面的示例中，我们已经确定第一框架将占据所示区域的 30%，并且我们使用“\*”符号来指示剩余百分比。
- `frameset rows`- 确定将显示的每个框架行所占百分比。在前面的示例中，我们选择第一框架为 20%，剩下的剩余空间将在 `menu.html` 和 `content.html` 之间划分。

## <frameset> - 设置边框

---

框架有一些边框线，大多数时候都不需要。<frameset>标签中我们使用 frameborder 和 framespacing 属性可以擦除它们。

**注意：**frameset 和 frameborder 是相同的属性。

#### 实例

```
<frameset border="0" frameborder="0" framespacing="0" rows="20%,*">
<frame src="title.html">
<frameset border="0" frameborder="0" framespacing="0" cols="30%,*">
<frame src="menu.html">
<frame src="content.html">
</frameset>
</frameset>
```

- frameborder-设置边框，0 值表示没有边框
- border- 修改边框的粗细（由 Netscape 浏览器使用）
- framespacing- 修改边框的粗细（由 Internet Explorer 浏览器使用）

## <frame> - 设置名字

---

<frame>标签中我们使用 name 属性命名每个框架，而不是内容页面。

#### 实例

```
<frameset rows="20%,*">
<frame name="title" src="title.html">
<frameset cols="30%,*">
<frame name="menu" src="menu.html">
<name="content" src="content.html">
</frameset>
</frameset>
```

## <frame> - 设置滚动

---

<frame> 标签中我们使用 `noresize` 设置禁止用户拖拉框架大小；`scrolling` 用于设置滚动条是否显示。

### 实例

```
<frameset border="2" frameborder="1" framespacing="2" rows="20%,*">
<frame src="title.html" noresize scrolling="no">
<frameset border="4" frameborder="1" framespacing="4" cols="30%,*">
<frame src="menu.html" scrolling="auto" noresize>
<frame src="content.html" scrolling="yes" noresize>
</frameset>
</frameset>
```

- `noresize`- 不允许用户更改其尺寸。
- `scrolling`- 设置滚动条是否显示。

## HTML 颜色

---

HTML 颜色采用的是 RGB 颜色，是通过红 (R)、绿 (G)、蓝 (B) 三个颜色通道的变化以及它们相互之间的叠加来得到各式各样的颜色的，RGB 即是代表红、绿、蓝三个通道的颜色。

### Color Values

HTML 颜色由一个十六进制符号来定义，这个符号由红色、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）。这个表格给出了由三种颜色混合而成的具体效果：

颜色值

| 颜色(Color) | 颜色十六进制 (Color HEX) | 颜色 RGB (Color RGB)  |
|-----------|--------------------|---------------------|
|           | #000000            | rgb (0, 0, 0)       |
|           | #FF0000            | rgb (255, 0, 0)     |
|           | #00FF00            | rgb (0, 255, 0)     |
|           | #0000FF            | rgb (0, 0, 255)     |
|           | #FFFF00            | rgb (255, 255, 0)   |
|           | #00FFFF            | rgb (0, 255, 255)   |
|           | #FF00FF            | rgb (255, 0, 255)   |
|           | #C0C0C0            | rgb (192, 192, 192) |
|           | #FFFFFF            | rgb (255, 255, 255) |

1600 万种不同颜色

三种颜色 红，绿，蓝的组合从 0 到 255，一共有 1600 万种不同颜色 (256 x 256 x 256)。您可以在[编程测试](#)中练习使用 RGB 红色表达方式！

在下面的颜色表中你会看到不同的结果，从 0 到 255 的红色，同时设置绿色和蓝色的值为 0，随着红色的值变化，不同的值都显示了不同的颜色。

| Red Light | Color HEX | Color RGB |
|-----------|-----------|-----------|
|-----------|-----------|-----------|

|  |         |                |
|--|---------|----------------|
|  | #000000 | rgb(0, 0, 0)   |
|  | #080000 | rgb(8, 0, 0)   |
|  | #100000 | rgb(16, 0, 0)  |
|  | #180000 | rgb(24, 0, 0)  |
|  | #200000 | rgb(32, 0, 0)  |
|  | #280000 | rgb(40, 0, 0)  |
|  | #300000 | rgb(48, 0, 0)  |
|  | #380000 | rgb(56, 0, 0)  |
|  | #400000 | rgb(64, 0, 0)  |
|  | #480000 | rgb(72, 0, 0)  |
|  | #500000 | rgb(80, 0, 0)  |
|  | #580000 | rgb(88, 0, 0)  |
|  | #600000 | rgb(96, 0, 0)  |
|  | #680000 | rgb(104, 0, 0) |
|  | #700000 | rgb(112, 0, 0) |
|  | #780000 | rgb(120, 0, 0) |
|  | #800000 | rgb(128, 0, 0) |
|  | #880000 | rgb(136, 0, 0) |
|  | #900000 | rgb(144, 0, 0) |
|  | #980000 | rgb(152, 0, 0) |
|  | #A00000 | rgb(160, 0, 0) |

|  |         |                 |
|--|---------|-----------------|
|  | #A80000 | rgb (168, 0, 0) |
|  | #B00000 | rgb (176, 0, 0) |
|  | #B80000 | rgb (184, 0, 0) |
|  | #C00000 | rgb (192, 0, 0) |
|  | #C80000 | rgb (200, 0, 0) |
|  | #D00000 | rgb (208, 0, 0) |
|  | #D80000 | rgb (216, 0, 0) |
|  | #E00000 | rgb (224, 0, 0) |
|  | #E80000 | rgb (232, 0, 0) |
|  | #F00000 | rgb (240, 0, 0) |
|  | #F80000 | rgb (248, 0, 0) |
|  | #FF0000 | rgb (255, 0, 0) |

灰暗色调

以下展示了灰色到黑色的渐变

| Gray Shades | Color HEX | Color RGB        |
|-------------|-----------|------------------|
|             | #000000   | rgb (0, 0, 0)    |
|             | #080808   | rgb (8, 8, 8)    |
|             | #101010   | rgb (16, 16, 16) |
|             | #181818   | rgb (24, 24, 24) |



|  |         |                     |
|--|---------|---------------------|
|  | #202020 | rgb (32, 32, 32)    |
|  | #282828 | rgb (40, 40, 40)    |
|  | #303030 | rgb (48, 48, 48)    |
|  | #383838 | rgb (56, 56, 56)    |
|  | #404040 | rgb (64, 64, 64)    |
|  | #484848 | rgb (72, 72, 72)    |
|  | #505050 | rgb (80, 80, 80)    |
|  | #585858 | rgb (88, 88, 88)    |
|  | #606060 | rgb (96, 96, 96)    |
|  | #686868 | rgb (104, 104, 104) |
|  | #707070 | rgb (112, 112, 112) |
|  | #787878 | rgb (120, 120, 120) |
|  | #808080 | rgb (128, 128, 128) |
|  | #888888 | rgb (136, 136, 136) |
|  | #909090 | rgb (144, 144, 144) |
|  | #989898 | rgb (152, 152, 152) |
|  | #A0A0A0 | rgb (160, 160, 160) |
|  | #A8A8A8 | rgb (168, 168, 168) |
|  | #B0B0B0 | rgb (176, 176, 176) |
|  | #B8B8B8 | rgb (184, 184, 184) |
|  | #C0C0C0 | rgb (192, 192, 192) |

|  |         |                     |
|--|---------|---------------------|
|  | #C8C8C8 | rgb (200, 200, 200) |
|  | #D0D0D0 | rgb (208, 208, 208) |
|  | #D8D8D8 | rgb (216, 216, 216) |
|  | #E0E0E0 | rgb (224, 224, 224) |
|  | #E8E8E8 | rgb (232, 232, 232) |
|  | #F0F0F0 | rgb (240, 240, 240) |
|  | #F8F8F8 | rgb (248, 248, 248) |
|  | #FFFFFF | rgb (255, 255, 255) |

## Web 安全色？

数年以前，当大多数计算机仅支持 256 种颜色的时候，一系列 216 种 Web 安全色作为 Web 标准被建议使用。其中的原因是，微软和 Mac 操作系统使用了 40 种不同的保留的固定系统颜色（双方大约各使用 20 种）。

我们不确定如今这么做的意义有多大，因为越来越多的计算机有能力处理数百万种颜色，不过做选择还是你自己。

最初，216 跨平台 Web 安全色被用来确保：当计算机使用 256 色调色板时，所有的计算机能够正确地显示所有的颜色。

|        |        |        |        |        |        |
|--------|--------|--------|--------|--------|--------|
|        | 000033 | 000066 | 000099 | 0000CC | 0000FF |
| 003300 | 003333 | 003366 | 003399 | 0033CC | 0033FF |
| 006600 | 006633 | 006666 | 006699 | 0066CC | 0066FF |
| 009900 | 009933 | 009966 | 009999 | 0099CC | 0099FF |
| 00CC00 | 00CC33 | 00CC66 | 00CC99 | 00CCCC | 00CCFF |
| 00FF00 | 00FF33 | 00FF66 | 00FF99 | 00FFCC | 00FFFF |

|        |        |        |        |        |        |
|--------|--------|--------|--------|--------|--------|
| 330000 | 330033 | 330066 | 330099 | 3300CC | 3300FF |
| 333300 | 333333 | 333366 | 333399 | 3333CC | 3333FF |
| 336600 | 336633 | 336666 | 336699 | 3366CC | 3366FF |
| 339900 | 339933 | 339966 | 339999 | 3399CC | 3399FF |
| 33CC00 | 33CC33 | 33CC66 | 33CC99 | 33CCCC | 33CCFF |
| 33FF00 | 33FF33 | 33FF66 | 33FF99 | 33FFCC | 33FFFF |
| 660000 | 660033 | 660066 | 660099 | 6600CC | 6600FF |
| 663300 | 663333 | 663366 | 663399 | 6633CC | 6633FF |
| 666600 | 666633 | 666666 | 666699 | 6666CC | 6666FF |
| 669900 | 669933 | 669966 | 669999 | 6699CC | 6699FF |
| 66CC00 | 66CC33 | 66CC66 | 66CC99 | 66CCCC | 66CCFF |
| 66FF00 | 66FF33 | 66FF66 | 66FF99 | 66FFCC | 66FFFF |
| 990000 | 990033 | 990066 | 990099 | 9900CC | 9900FF |
| 993300 | 993333 | 993366 | 993399 | 9933CC | 9933FF |
| 996600 | 996633 | 996666 | 996699 | 9966CC | 9966FF |
| 999900 | 999933 | 999966 | 999999 | 9999CC | 9999FF |
| 99CC00 | 99CC33 | 99CC66 | 99CC99 | 99CCCC | 99CCFF |
| 99FF00 | 99FF33 | 99FF66 | 99FF99 | 99FFCC | 99FFFF |
| CC0000 | CC0033 | CC0066 | CC0099 | CC00CC | CC00FF |
| CC3300 | CC3333 | CC3366 | CC3399 | CC33CC | CC33FF |
| CC6600 | CC6633 | CC6666 | CC6699 | CC66CC | CC66FF |
| CC9900 | CC9933 | CC9966 | CC9999 | CC99CC | CC99FF |
| CCCC00 | CCCC33 | CCCC66 | CCCC99 | CCCCCC | CCCCFF |
| CCFF00 | CCFF33 | CCFF66 | CCFF99 | CCFFCC | CCFFFF |
| FF0000 | FF0033 | FF0066 | FF0099 | FF00CC | FF00FF |
| FF3300 | FF3333 | FF3366 | FF3399 | FF33CC | FF33FF |

|        |        |        |        |        |        |
|--------|--------|--------|--------|--------|--------|
| FF6600 | FF6633 | FF6666 | FF6699 | FF66CC | FF66FF |
| FF9900 | FF9933 | FF9966 | FF9999 | FF99CC | FF99FF |
| FFCC00 | FFCC33 | FFCC66 | FFCC99 | FFCCCC | FFCCFF |
| FFFF00 | FFFF33 | FFFF66 | FFFF99 | FFFFCC | FFFFFF |

HTML 常用颜色代码表

|         |         |         |         |         |
|---------|---------|---------|---------|---------|
| #000000 | #2F0000 | #600030 | #460046 | #28004D |
| #272727 | #4D0000 | #820041 | #5E005E | #3A006F |
| #3C3C3C | #600000 | #9F0050 | #750075 | #4B0091 |
| #4F4F4F | #750000 | #BF0060 | #930093 | #5B00AE |
| #5B5B5B | #930000 | #D9006C | #AE00AE | #6F00D2 |
| #6C6C6C | #AE0000 | #F00078 | #D200D2 | #8600FF |
| #7B7B7B | #CE0000 | #FF0080 | #E800E8 | #921AFF |
| #8E8E8E | #EA0000 | #FF359A | #FF00FF | #9F35FF |
| #9D9D9D | #FF0000 | #FF60AF | #FF44FF | #B15BFF |
| #ADADAD | #FF2D2D | #FF79BC | #FF77FF | #BE77FF |
| #BEBEBE | #FF5151 | #FF95CA | #FF8EFF | #CA8EFF |
| #d0d0d0 | #ff7575 | #ffaad5 | #ffa6ff | #d3a4ff |
| #E0E0E0 | #FF9797 | #FFC1E0 | #FFBFFF | #DCB5FF |
| #F0F0F0 | #FFB5B5 | #FFD9EC | #FFD0FF | #E6CAFF |
| #FCFCFC | #FFD2D2 | #FFECF5 | #FFE6FF | #F1E1FF |
| #FFFFFF | #FFECEC | #FFF7FB | #FFF7FF | #FAF4FF |
| #000079 | #000079 | #003E3E | #006030 | #006000 |
| #000093 | #003D79 | #005757 | #01814A | #007500 |
| #0000C6 | #004B97 | #007979 | #019858 | #009100 |
| #0000C6 | #005AB5 | #009393 | #01B468 | #00A600 |
| #0000E3 | #0066CC | #00AEAE | #02C874 | #00BB00 |


|          |         |         |         |         |
|----------|---------|---------|---------|---------|
| #2828FF  | #0072E3 | #00CACA | #02DF82 | #00DB00 |
| #4A4AFF  | #0080FF | #00E3E3 | #02F78E | #00EC00 |
| #6A6AFF  | #2894FF | #00FFFF | #1AFD9C | #28FF28 |
| #7D7DFF  | #46A3FF | #4DFFFF | #4EFEB3 | #53FF53 |
| #9393FF  | #66B3FF | #80FFFF | #7AFEC6 | #79FF79 |
| #AAAAFF  | #84C1FF | #A6FFFF | #96FED1 | #93FF93 |
| #B9B9FF  | #97CBFF | #BBFFFF | #ADFEDC | #A6FFA6 |
| #CECEFF  | #ACD6FF | #CAFFFF | #C1FFE4 | #BBFFBB |
| #DDDDFF  | #C4E1FF | #D9FFFF | #D7FFEE | #CEFFCE |
| #ECECFE  | #D2E9FF | #ECFFFF | #E8FFF5 | #DFFFFD |
| #FBFBFF  | #ECF5FF | #FDFFFF | #FBFFFD | #FOFFF0 |
| #467500  | #424200 | #5B4B00 | #844200 | #642100 |
| #548C00  | #5B5B00 | #796400 | #9F5000 | #842B00 |
| #64A600  | #737300 | #977C00 | #BB5E00 | #A23400 |
| #73BF00  | #8C8C00 | #AE8F00 | #D26900 | #BB3D00 |
| #82D900  | #A6A600 | #C6A300 | #EA7500 | #D94600 |
| #8CEA00  | #C4C400 | #D9B300 | #FF8000 | #F75000 |
| #9AFF02  | #E1E100 | #EAC100 | #FF9224 | #FF5809 |
| #A8FF24  | #F9F900 | #FFD306 | #FFA042 | #FF8040 |
| #B7FF4A  | #FFFF37 | #FFDC35 | #FFAF60 | #FF8F59 |
| #C2FF68  | #FFFF6F | #FFE153 | #FFBB77 | #FF9D6F |
| #CCFF80  | #FFFF93 | #FFE66F | #FFC78E | #FFAD86 |
| #D3FF93  | #FFFFAA | #FFED97 | #FFD1A4 | #FFBD9D |
| #DEFFAC  | #FFFFB9 | #FFF0AC | #FFDCB9 | #FFCBB3 |
| #E8FFC4  | #FFFFCE | #FFF4C1 | #FFE4CA | #FFDAC8 |
| #EFFFFD7 | #FFFFDF | #FFF8D7 | #FFEEDD | #FFE6D9 |

|         |          |         |         |         |
|---------|----------|---------|---------|---------|
| #F5FFE8 | #FFFFFF4 | #FFFCEC | #FFF4F4 | #FFF3EE |
| #613030 | #616130  | #336666 | #484891 | #6C3365 |
| #743A3A | #707038  | #3D7878 | #5151A2 | #7E3D76 |
| #804040 | #808040  | #408080 | #5A5AAD | #8F4586 |
| #984B4B | #949449  | #4F9D9D | #7373B9 | #9F4D95 |
| #AD5A5A | #A5A552  | #5CADAD | #8080C0 | #AE57A4 |
| #B87070 | #AFAF61  | #6FB7B7 | #9999CC | #B766AD |
| #C48888 | #B9B973  | #81C0C0 | #A6A6D2 | #C07AB8 |
| #CF9E9E | #C2C287  | #95CACA | #B8B8DC | #CA8EC2 |
| #D9B3B3 | #CDCD9A  | #A3D1D1 | #C7C7E2 | #D2A2CC |
| #E1C4C4 | #D6D6AD  | #B3D9D9 | #D8D8EB | #DAB1D5 |
| #EBD6D6 | #DEDEBE  | #C4E1E1 | #E6E6F2 | #E2C2DE |
| #F2E6E6 | #E8E8D0  | #D1E9E9 | #F3F3FA | #EBD3E8 |

## 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 ）。下表列出了所有颜色的值，包括十六进制值。

 **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。点击其中一个颜色名称（或一个十六进制值）就可以查看与不同文字颜色搭配的背景颜色。：

### 按颜色名排序

#### 按十六进制的值排序




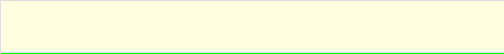
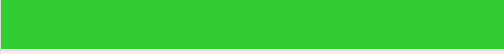








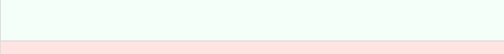



单击一个颜色名或者 16 进制值，就可以查看与不同文字颜色搭配的背景颜色。

| Color Name            | HEX            | Color  |
|-----------------------|----------------|--|
| <u>AliceBlue</u>      | <u>#F0F8FF</u> |    |
| <u>AntiqueWhite</u>   | <u>#FAEBD7</u> |    |
| <u>Aqua</u>           | <u>#00FFFF</u> |    |
| <u>Aquamarine</u>     | <u>#7FFFD4</u> |    |
| <u>Azure</u>          | <u>#F0FFFF</u> |    |
| <u>Beige</u>          | <u>#F5F5DC</u> |    |
| <u>Bisque</u>         | <u>#FFE4C4</u> |    |
| <u>Black</u>          | <u>#000000</u> |    |
| <u>BlanchedAlmond</u> | <u>#FFEBCD</u> |    |
| <u>Blue</u>           | <u>#0000FF</u> |    |
| <u>BlueViolet</u>     | <u>#8A2BE2</u> |    |
| <u>Brown</u>          | <u>#A52A2A</u> |    |
| <u>BurlyWood</u>      | <u>#DEB887</u> |    |
| <u>CadetBlue</u>      | <u>#5F9EA0</u> |   |
| <u>Chartreuse</u>     | <u>#7FFF00</u> |  |
| <u>Chocolate</u>      | <u>#D2691E</u> |  |
| <u>Coral</u>          | <u>#FF7F50</u> |  |
| <u>CornflowerBlue</u> | <u>#6495ED</u> |  |
| <u>Cornsilk</u>       | <u>#FFF8DC</u> |  |
| <u>Crimson</u>        | <u>#DC143C</u> |  |
| <u>Cyan</u>           | <u>#00FFFF</u> |  |
| <u>DarkBlue</u>       | <u>#00008B</u> |  |
| <u>DarkCyan</u>       | <u>#008B8B</u> |  |

|                       |                |  |
|-----------------------|----------------|--|
| <u>DarkGoldenRod</u>  | <u>#B8860B</u> |    |
| <u>DarkGray</u>       | <u>#A9A9A9</u> |    |
| <u>DarkGreen</u>      | <u>#006400</u> |    |
| <u>DarkKhaki</u>      | <u>#BDB76B</u> |    |
| <u>DarkMagenta</u>    | <u>#8B008B</u> |    |
| <u>DarkOliveGreen</u> | <u>#556B2F</u> |    |
| <u>DarkOrange</u>     | <u>#FF8C00</u> |    |
| <u>DarkOrchid</u>     | <u>#9932CC</u> |    |
| <u>DarkRed</u>        | <u>#8B0000</u> |    |
| <u>DarkSalmon</u>     | <u>#E9967A</u> |    |
| <u>DarkSeaGreen</u>   | <u>#8FBC8F</u> |    |
| <u>DarkSlateBlue</u>  | <u>#483D8B</u> |    |
| <u>DarkSlateGray</u>  | <u>#2F4F4F</u> |    |
| <u>DarkTurquoise</u>  | <u>#00CED1</u> |   |
| <u>DarkViolet</u>     | <u>#9400D3</u> |  |
| <u>DeepPink</u>       | <u>#FF1493</u> |  |
| <u>DeepSkyBlue</u>    | <u>#00BFFF</u> |  |
| <u>DimGray</u>        | <u>#696969</u> |  |
| <u>DodgerBlue</u>     | <u>#1E90FF</u> |  |
| <u>FireBrick</u>      | <u>#B22222</u> |  |
| <u>FloralWhite</u>    | <u>#FFFAF0</u> |  |
| <u>ForestGreen</u>    | <u>#228B22</u> |  |
| <u>Fuchsia</u>        | <u>#FF00FF</u> |  |
| <u>Gainsboro</u>      | <u>#DCDCDC</u> |  |



|                             |                |  |
|-----------------------------|----------------|--|
| <u>GhostWhite</u>           | <u>#F8F8FF</u> |  |
| <u>Gold</u>                 | <u>#FFD700</u> |  |
| <u>GoldenRod</u>            | <u>#DAA520</u> |  |
| <u>Gray</u>                 | <u>#808080</u> |  |
| <u>Green</u>                | <u>#008000</u> |  |
| <u>GreenYellow</u>          | <u>#ADFF2F</u> |  |
| <u>HoneyDew</u>             | <u>#F0FFF0</u> |  |
| <u>HotPink</u>              | <u>#FF69B4</u> |  |
| <u>IndianRed</u>            | <u>#CD5C5C</u> |  |
| <u>Indigo</u>               | <u>#4B0082</u> |  |
| <u>Ivory</u>                | <u>#FFFFFF</u> |  |
| <u>Khaki</u>                | <u>#F0E68C</u> |  |
| <u>Lavender</u>             | <u>#E6E6FA</u> |  |
| <u>LavenderBlush</u>        | <u>#FFF0F5</u> |  |
| <u>LawnGreen</u>            | <u>#7CFC00</u> |  |
| <u>LemonChiffon</u>         | <u>#FFFACD</u> |  |
| <u>LightBlue</u>            | <u>#ADD8E6</u> |  |
| <u>LightCoral</u>           | <u>#F08080</u> |  |
| <u>LightCyan</u>            | <u>#E0FFFF</u> |  |
| <u>LightGoldenRodYellow</u> | <u>#FAFAD2</u> |  |
| <u>LightGray</u>            | <u>#D3D3D3</u> |  |
| <u>LightGreen</u>           | <u>#90EE90</u> |  |
| <u>LightPink</u>            | <u>#FFB6C1</u> |  |
| <u>LightSalmon</u>          | <u>#FFA07A</u> |  |

|                          |                |  |
|--------------------------|----------------|--|
| <u>LightSeaGreen</u>     | <u>#20B2AA</u> |    |
| <u>LightSkyBlue</u>      | <u>#87CEFA</u> |    |
| <u>LightSlateGray</u>    | <u>#778899</u> |    |
| <u>LightSteelBlue</u>    | <u>#B0C4DE</u> |    |
| <u>LightYellow</u>       | <u>#FFFFE0</u> |    |
| <u>Lime</u>              | <u>#00FF00</u> |    |
| <u>LimeGreen</u>         | <u>#32CD32</u> |    |
| <u>Linen</u>             | <u>#FAF0E6</u> |    |
| <u>Magenta</u>           | <u>#FF00FF</u> |    |
| <u>Maroon</u>            | <u>#800000</u> |    |
| <u>MediumAquaMarine</u>  | <u>#66CDAA</u> |    |
| <u>MediumBlue</u>        | <u>#0000CD</u> |    |
| <u>MediumOrchid</u>      | <u>#BA55D3</u> |    |
| <u>MediumPurple</u>      | <u>#9370DB</u> |    |
| <u>MediumSeaGreen</u>    | <u>#3CB371</u> |   |
| <u>MediumSlateBlue</u>   | <u>#7B68EE</u> |  |
| <u>MediumSpringGreen</u> | <u>#00FA9A</u> |  |
| <u>MediumTurquoise</u>   | <u>#48D1CC</u> |  |
| <u>MediumVioletRed</u>   | <u>#C71585</u> |  |
| <u>MidnightBlue</u>      | <u>#191970</u> |  |
| <u>MintCream</u>         | <u>#F5FFFA</u> |  |
| <u>MistyRose</u>         | <u>#FFE4E1</u> |  |
| <u>Moccasin</u>          | <u>#FFE4B5</u> |  |
| <u>NavajoWhite</u>       | <u>#FFDEAD</u> |  |

|                      |                |  |
|----------------------|----------------|--|
| <u>Navy</u>          | <u>#000080</u> |    |
| <u>OldLace</u>       | <u>#FDF5E6</u> |    |
| <u>Olive</u>         | <u>#808000</u> |    |
| <u>OliveDrab</u>     | <u>#6B8E23</u> |    |
| <u>Orange</u>        | <u>#FFA500</u> |    |
| <u>OrangeRed</u>     | <u>#FF4500</u> |    |
| <u>Orchid</u>        | <u>#DA70D6</u> |    |
| <u>PaleGoldenRod</u> | <u>#EEE8AA</u> |    |
| <u>PaleGreen</u>     | <u>#98FB98</u> |    |
| <u>PaleTurquoise</u> | <u>#AFEEEE</u> |    |
| <u>PaleVioletRed</u> | <u>#DB7093</u> |    |
| <u>PapayaWhip</u>    | <u>#FFEFD5</u> |    |
| <u>PeachPuff</u>     | <u>#FFDAB9</u> |    |
| <u>Peru</u>          | <u>#CD853F</u> |    |
| <u>Pink</u>          | <u>#FFC0CB</u> |   |
| <u>Plum</u>          | <u>#DDA0DD</u> |  |
| <u>PowderBlue</u>    | <u>#B0E0E6</u> |  |
| <u>Purple</u>        | <u>#800080</u> |  |
| <u>Red</u>           | <u>#FF0000</u> |  |
| <u>RosyBrown</u>     | <u>#BC8F8F</u> |  |
| <u>RoyalBlue</u>     | <u>#4169E1</u> |  |
| <u>SaddleBrown</u>   | <u>#8B4513</u> |  |
| <u>Salmon</u>        | <u>#FA8072</u> |  |
| <u>SandyBrown</u>    | <u>#F4A460</u> |  |

|                    |                |  |
|--------------------|----------------|--|
| <u>SeaGreen</u>    | <u>#2E8B57</u> |    |
| <u>SeaShell</u>    | <u>#FFF5EE</u> |    |
| <u>Sienna</u>      | <u>#A0522D</u> |    |
| <u>Silver</u>      | <u>#C0C0C0</u> |    |
| <u>SkyBlue</u>     | <u>#87CEEB</u> |    |
| <u>SlateBlue</u>   | <u>#6A5ACD</u> |    |
| <u>SlateGray</u>   | <u>#708090</u> |    |
| <u>Snow</u>        | <u>#FFFAFA</u> |    |
| <u>SpringGreen</u> | <u>#00FF7F</u> |    |
| <u>SteelBlue</u>   | <u>#4682B4</u> |    |
| <u>Tan</u>         | <u>#D2B48C</u> |    |
| <u>Teal</u>        | <u>#008080</u> |    |
| <u>Thistle</u>     | <u>#D8BFD8</u> |    |
| <u>Tomato</u>      | <u>#FF6347</u> |    |
| <u>Turquoise</u>   | <u>#40E0D0</u> |   |
| <u>Violet</u>      | <u>#EE82EE</u> |  |
| <u>Wheat</u>       | <u>#F5DEB3</u> |  |
| <u>White</u>       | <u>#FFFFFF</u> |  |
| <u>WhiteSmoke</u>  | <u>#F5F5F5</u> |  |
| <u>Yellow</u>      | <u>#FFFF00</u> |  |
| <u>YellowGreen</u> | <u>#9ACD32</u> |  |

## HTML 颜色值

颜色由红 (R)、绿 (G)、蓝 (B) 组成。

## 颜色值

颜色值由十六进制来表示红、绿、蓝（RGB）。

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)。

十六进制值的写法为#号后跟三个或六个十六进制字符。

三位数表示法为：#RGB，转换为 6 位数表示为：#RRGGBB。

### 颜色实例

| 颜色  | 3 位十六进制颜色值 | 6 位十六进制颜色值 | RGB                 |
|---|------------|------------|---------------------|
|    | #000       | #000000    | rgb (0, 0, 0)       |
|    | #F00       | #FF0000    | rgb (255, 0, 0)     |
|    | #0F0       | #00FF00    | rgb (0, 255, 0)     |
|    | #00F       | #0000FF    | rgb (0, 0, 255)     |
|    | #FF0       | #FFFF00    | rgb (255, 255, 0)   |
|    | #0FF       | #00FFFF    | rgb (0, 255, 255)   |
|   | #F0F       | #FF00FF    | rgb (255, 0, 255)   |
|  | #888       | #888888    | rgb (136, 136, 136) |
|  | #FFF       | #FFFFFF    | rgb (255, 255, 255) |

## 通过十六进制（Hex）的颜色值排序

[查看以颜色名称排序的颜色列表](#)

| Color Name | HEX | Color |
|------------|-----|-------|
|------------|-----|-------|

|                          |                |  |
|--------------------------|----------------|--|
| <u>Black</u>             | <u>#000000</u> |  |
| <u>Navy</u>              | <u>#000080</u> |  |
| <u>DarkBlue</u>          | <u>#00008B</u> |  |
| <u>MediumBlue</u>        | <u>#0000CD</u> |  |
| <u>Blue</u>              | <u>#0000FF</u> |  |
| <u>DarkGreen</u>         | <u>#006400</u> |  |
| <u>Green</u>             | <u>#008000</u> |  |
| <u>Teal</u>              | <u>#008080</u> |  |
| <u>DarkCyan</u>          | <u>#008B8B</u> |  |
| <u>DeepSkyBlue</u>       | <u>#00BFFF</u> |  |
| <u>DarkTurquoise</u>     | <u>#00CED1</u> |  |
| <u>MediumSpringGreen</u> | <u>#00FA9A</u> |  |
| <u>Lime</u>              | <u>#00FF00</u> |  |
| <u>SpringGreen</u>       | <u>#00FF7F</u> |  |
| <u>Aqua</u>              | <u>#00FFFF</u> |  |
| <u>Cyan</u>              | <u>#00FFFF</u> |  |
| <u>MidnightBlue</u>      | <u>#191970</u> |  |
| <u>DodgerBlue</u>        | <u>#1E90FF</u> |  |
| <u>LightSeaGreen</u>     | <u>#20B2AA</u> |  |
| <u>ForestGreen</u>       | <u>#228B22</u> |  |

|                         |                |  |
|-------------------------|----------------|--|
| <u>SeaGreen</u>         | <u>#2E8B57</u> |    |
| <u>DarkSlateGray</u>    | <u>#2F4F4F</u> |    |
| <u>LimeGreen</u>        | <u>#32CD32</u> |    |
| <u>MediumSeaGreen</u>   | <u>#3CB371</u> |    |
| <u>Turquoise</u>        | <u>#40E0D0</u> |    |
| <u>RoyalBlue</u>        | <u>#4169E1</u> |    |
| <u>SteelBlue</u>        | <u>#4682B4</u> |    |
| <u>DarkSlateBlue</u>    | <u>#483D8B</u> |    |
| <u>MediumTurquoise</u>  | <u>#48D1CC</u> |    |
| <u>Indigo</u>           | <u>#4B0082</u> |    |
| <u>DarkOliveGreen</u>   | <u>#556B2F</u> |    |
| <u>CadetBlue</u>        | <u>#5F9EA0</u> |   |
| <u>CornflowerBlue</u>   | <u>#6495ED</u> |  |
| <u>MediumAquaMarine</u> | <u>#66CDAA</u> |  |
| <u>DimGray</u>          | <u>#696969</u> |  |
| <u>SlateBlue</u>        | <u>#6A5ACD</u> |  |
| <u>OliveDrab</u>        | <u>#6B8E23</u> |  |
| <u>SlateGray</u>        | <u>#708090</u> |  |
| <u>LightSlateGray</u>   | <u>#778899</u> |  |
| <u>MediumSlateBlue</u>  | <u>#7B68EE</u> |  |

|                     |                |  |
|---------------------|----------------|--|
| <u>LawnGreen</u>    | <u>#7CFC00</u> |    |
| <u>Chartreuse</u>   | <u>#7FFF00</u> |    |
| <u>Aquamarine</u>   | <u>#7FFFD4</u> |    |
| <u>Maroon</u>       | <u>#800000</u> |    |
| <u>Purple</u>       | <u>#800080</u> |    |
| <u>Olive</u>        | <u>#808000</u> |    |
| <u>Gray</u>         | <u>#808080</u> |    |
| <u>SkyBlue</u>      | <u>#87CEEB</u> |    |
| <u>LightSkyBlue</u> | <u>#87CEFA</u> |    |
| <u>BlueViolet</u>   | <u>#8A2BE2</u> |    |
| <u>DarkRed</u>      | <u>#8B0000</u> |    |
| <u>DarkMagenta</u>  | <u>#8B008B</u> |   |
| <u>SaddleBrown</u>  | <u>#8B4513</u> |  |
| <u>DarkSeaGreen</u> | <u>#8FBC8F</u> |  |
| <u>LightGreen</u>   | <u>#90EE90</u> |  |
| <u>MediumPurple</u> | <u>#9370DB</u> |  |
| <u>DarkViolet</u>   | <u>#9400D3</u> |  |
| <u>PaleGreen</u>    | <u>#98FB98</u> |  |
| <u>DarkOrchid</u>   | <u>#9932CC</u> |  |
| <u>YellowGreen</u>  | <u>#9ACD32</u> |  |



|                        |                |  |
|------------------------|----------------|--|
| <u>Sienna</u>          | <u>#A0522D</u> |  |
| <u>Brown</u>           | <u>#A52A2A</u> |  |
| <u>DarkGray</u>        | <u>#A9A9A9</u> |  |
| <u>LightBlue</u>       | <u>#ADD8E6</u> |  |
| <u>GreenYellow</u>     | <u>#ADFF2F</u> |  |
| <u>PaleTurquoise</u>   | <u>#AFEEEE</u> |  |
| <u>LightSteelBlue</u>  | <u>#B0C4DE</u> |  |
| <u>PowderBlue</u>      | <u>#B0E0E6</u> |  |
| <u>FireBrick</u>       | <u>#B22222</u> |  |
| <u>DarkGoldenRod</u>   | <u>#B8860B</u> |  |
| <u>MediumOrchid</u>    | <u>#BA55D3</u> |  |
| <u>RosyBrown</u>       | <u>#BC8F8F</u> |  |
| <u>DarkKhaki</u>       | <u>#BDB76B</u> |  |
| <u>Silver</u>          | <u>#C0C0C0</u> |  |
| <u>MediumVioletRed</u> | <u>#C71585</u> |  |
| <u>IndianRed</u>       | <u>#CD5C5C</u> |  |
| <u>Peru</u>            | <u>#CD853F</u> |  |
| <u>Chocolate</u>       | <u>#D2691E</u> |  |
| <u>Tan</u>             | <u>#D2B48C</u> |  |
| <u>LightGray</u>       | <u>#D3D3D3</u> |  |

|                      |                |  |
|----------------------|----------------|--|
| <u>Thistle</u>       | <u>#D8BFD8</u> |  |
| <u>Orchid</u>        | <u>#DA70D6</u> |  |
| <u>GoldenRod</u>     | <u>#DAA520</u> |  |
| <u>PaleVioletRed</u> | <u>#DB7093</u> |  |
| <u>Crimson</u>       | <u>#DC143C</u> |  |
| <u>Gainsboro</u>     | <u>#DCDCDC</u> |  |
| <u>Plum</u>          | <u>#DDA0DD</u> |  |
| <u>BurlyWood</u>     | <u>#DEB887</u> |  |
| <u>LightCyan</u>     | <u>#E0FFFF</u> |  |
| <u>Lavender</u>      | <u>#E6E6FA</u> |  |
| <u>DarkSalmon</u>    | <u>#E9967A</u> |  |
| <u>Violet</u>        | <u>#EE82EE</u> |  |
| <u>PaleGoldenRod</u> | <u>#EEE8AA</u> |  |
| <u>LightCoral</u>    | <u>#F08080</u> |  |
| <u>Khaki</u>         | <u>#F0E68C</u> |  |
| <u>AliceBlue</u>     | <u>#F0F8FF</u> |  |
| <u>HoneyDew</u>      | <u>#F0FFF0</u> |  |
| <u>Azure</u>         | <u>#F0FFFF</u> |  |
| <u>SandyBrown</u>    | <u>#F4A460</u> |  |
| <u>Wheat</u>         | <u>#F5DEB3</u> |  |

|                             |                |  |
|-----------------------------|----------------|--|
| <u>Beige</u>                | <u>#F5F5DC</u> |  |
| <u>WhiteSmoke</u>           | <u>#F5F5F5</u> |  |
| <u>MintCream</u>            | <u>#F5FFFA</u> |  |
| <u>GhostWhite</u>           | <u>#F8F8FF</u> |  |
| <u>Salmon</u>               | <u>#FA8072</u> |  |
| <u>AntiqueWhite</u>         | <u>#FAEBD7</u> |  |
| <u>Linen</u>                | <u>#FAF0E6</u> |  |
| <u>LightGoldenRodYellow</u> | <u>#FAFAD2</u> |  |
| <u>OldLace</u>              | <u>#FDF5E6</u> |  |
| <u>Red</u>                  | <u>#FF0000</u> |  |
| <u>Fuchsia</u>              | <u>#FF00FF</u> |  |
| <u>Magenta</u>              | <u>#FF00FF</u> |  |
| <u>DeepPink</u>             | <u>#FF1493</u> |  |
| <u>OrangeRed</u>            | <u>#FF4500</u> |  |
| <u>Tomato</u>               | <u>#FF6347</u> |  |
| <u>HotPink</u>              | <u>#FF69B4</u> |  |
| <u>Coral</u>                | <u>#FF7F50</u> |  |
| <u>DarkOrange</u>           | <u>#FF8C00</u> |  |
| <u>LightSalmon</u>          | <u>#FFA07A</u> |  |
| <u>Orange</u>               | <u>#FFA500</u> |  |

|                       |                |  |
|-----------------------|----------------|--|
| <u>LightPink</u>      | <u>#FFB6C1</u> |  |
| <u>Pink</u>           | <u>#FFC0CB</u> |  |
| <u>Gold</u>           | <u>#FFD700</u> |  |
| <u>PeachPuff</u>      | <u>#FFDAB9</u> |  |
| <u>NavajoWhite</u>    | <u>#FFDEAD</u> |  |
| <u>Moccasin</u>       | <u>#FFE4B5</u> |  |
| <u>Bisque</u>         | <u>#FFE4C4</u> |  |
| <u>MistyRose</u>      | <u>#FFE4E1</u> |  |
| <u>BlanchedAlmond</u> | <u>#FFEBCD</u> |  |
| <u>PapayaWhip</u>     | <u>#FFEFD5</u> |  |
| <u>LavenderBlush</u>  | <u>#FFF0F5</u> |  |
| <u>SeaShell</u>       | <u>#FFF5EE</u> |  |
| <u>Cornsilk</u>       | <u>#FFF8DC</u> |  |
| <u>LemonChiffon</u>   | <u>#FFFACD</u> |  |
| <u>FloralWhite</u>    | <u>#FFFAF0</u> |  |
| <u>Snow</u>           | <u>#FFFAFA</u> |  |
| <u>Yellow</u>         | <u>#FFFF00</u> |  |
| <u>LightYellow</u>    | <u>#FFFFE0</u> |  |
| <u>Ivory</u>          | <u>#FFFFF0</u> |  |
| <u>White</u>          | <u>#FFFFFF</u> |  |

## HTML 脚本

---

JavaScript 是可插入 HTML 页面的编程代码。

JavaScript 使 HTML 页面具有更强的动态和交互性。

JavaScript 插入 HTML 页面后，可由所有的现代浏览器执行。

---

### 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。

JavaScript 最常用于图片操作、表单验证以及内容动态更新。

下面的脚本会向浏览器输出 "Hello World!"：

#### 实例

```
<script>
document.write("Hello World!")
</script>
```

**注释：**如果使用 "src" 属性，则 <script> 元素必须是空的。



**Tip:** 学习更多关于 Javascript 教程，请查看 [JavaScript 教程](#)!

---

### 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>Sorry, your browser does not support JavaScript!</noscript>
```



**Tip:** <noscript> 标签支持全局属性, 查看完整属性表 [HTML 全局属性](#)!

---

## JavaScript 体验(来自本站 JavaScript 教程)

JavaScript 实例代码:

JavaScript 可以直接在 HTML 输出:

```
document.write("<p>This is a paragraph</p>");
```

JavaScript 事件响应:

```
<button type="button" onclick = "myFunction()">Click Me!</button>
```

JavaScript 处理 HTML 样式:

```
document.getElementById("demo").style.color = "#ff0000";
```

---

## 如何应付老式的浏览器

如果浏览器压根没法识别 `<script>` 标签，那么 `<script>` 标签所包含的内容将以文本方式显示在页面上。为了避免这种情况发生，你应该将脚本隐藏在注释标签当中。那些老的浏览器（无法识别 `<script>` 标签的浏览器）将忽略这些注释，所以不会将标签的内容显示到页面上。而那些新的浏览器将读懂这些脚本并执行它们，即使代码被嵌套在注释标签内。

### 实例

JavaScript :

```
<script type="text/javascript">
<!--
document.write("Hello World!")
//-->
</script>
```

## HTML 脚本标签

| 标签                                   | 描述               |
|--------------------------------------|------------------|
| <u><code>&lt;script&gt;</code></u>   | 定义了客户端脚本         |
| <u><code>&lt;noscript&gt;</code></u> | 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。

一些在键盘上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。您不能使用包含这些字符的文本。

在 HTML 中不能使用小于号 (<) 和大于号 (>)，这是因为浏览器会误认为它们是标签。


如果希望正确地显示预留字符，我们必须在 HTML 源代码中使用字符实体 (character entities)。字符实体类似这样：

```
&entity_name;
```

或

```
&#entity_number;
```

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

 **提示：** 使用实体名而不是数字的好处是，名称易于记忆。不过坏处是，浏览器也许并不支持所有实体名称（对实体数字的支持却很好）。

## 不间断空格 (Non-breaking Space)

HTML 中的常用字符实体是不间断空格( )。

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)"。

一些变音符号, 如 尖音符 ( ) 和 抑音符 ( )。

变音符号可以出现字母的上面和下面，或者字母里面，或者两个字母间。

变音符号可以与字母、数字字符的组合来使用。

以下是一些实例:

| 音标符 | 字符 | Construct | 输出结果 |
|-----|----|-----------|------|
|     | a  | à         | à    |
|     | a  | á         | á    |
|     | a  | a&#770;   | â    |
|     | a  | a&#771;   | ã    |



|  |   |         |   |
|--|---|---------|---|
|  | o | O&#768; | Ö |
|  | o | O&#769; | Ó |
|  | o | O&#770; | Ô |
|  | o | O&#771; | Õ |

## 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; (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;  |

查看本站完整的 HTML 实体：请点击 [HTML 实体参考手册](#)。

## HTML 统一资源定位器 (Uniform Resource Locators)

---

URL 是一个网页地址。

URL 可以由字母组成，如 "lenovo.com.cn"，或互联网协议（IP）地址：120.79.88.157。大多数人进入网站使用网站域名来访问，因为名字比数字更容易记住。

---

### URL - 统一资源定位器

Web 浏览器通过 URL 从 Web 服务器请求页面。

当您点击 HTML 页面中的某个链接时，对应的 <a> 标签指向万维网上的一个地址。

一个统一资源定位器 (URL) 用于定位万维网上的文档。

一个网页地址实例：[//www.lenovo.com.cn/html/html-tutorial.html](http://www.lenovo.com.cn/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 lenovo.com.cn

: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

---

## 常见的 URL Schemes

以下是一些 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   |           | 您计算机上的文件。              |

---

## URL 字符编码

URL 只能使用 [ASCII 字符集](#)。

URL 编码会将字符转换为可通过因特网传输的格式。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。

URL 不能包含空格。URL 编码通常使用 + 来替换空格。

---

## 在线实例

### 实例

如果您点击“提交”按钮，浏览器会在发送输入之前对其进行 URL 编码。服务器上的页面会显示出接收到的输入。

```
<form name="input" target="_blank" action="form_action.php" method="get">
  <input type="text" value="lenovo.com.cn" name="text" size="30">
  <input type="submit" value="提交">
</form>
```

点击 "尝试一下" 按钮查看在线实例

 **提示:** [JavaScript](#)、[PHP](#)、[ASP](#) 都提供了对字符串进行 URL 编码的函数。

## URL 编码实例

| 字符 | URL 编码 |
|----|--------|
| €  | %80    |
| £  | %A3    |
| ©  | %A9    |
| ®  | %AE    |
| À  | %C0    |
| Á  | %C1    |
| Â  | %C2    |
| Ã  | %C3    |
| Ä  | %C4    |
| Å  | %C5    |

如需完整的 URL 编码参考, 请访问我们的 [URL 编码参考手册](#)。

## HTML 速查列表

HTML 速查列表. 你可以打印它, 以备日常使用。

## 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 (Basic Tags)

```
<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 (Formatting)

```
<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 (Links)

普通的链接: `<a href="链接地址">链接文本</a>`

图像链接: `<a href="http://www.example.com/" rel="external nofollow" target="_blank" ></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 (Images)

``

## 样式/区块 (Styles/Sections)

`<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 (Tables)

```
<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)

```
<iframe src="demo_iframe.htm"></iframe>
```

## 表单 (Forms)

```
<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 (Entities)

< 等同于 <

> 等同于 >

@ 等同于 @