HTML

# [HTML 简介](http://www.w3school.com.cn/html/html_intro.asp)

## 什么是 HTML？

HTML 是用来描述网页的一种语言。

HTML 指的是超文本标记语言 (Hyper Text Markup Language)

HTML 不是一种编程语言，而是一种标记语言 (markup language)

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 标签

HTML 标记标签通常被称为 HTML 标签 (HTML tag)。

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

<html>

<body>

<h1>My First Heading</h1>

<p>My first paragraph.</p>

</body>

</html>

例子解释

<html> 与 </html> 之间的文本描述网页

<body> 与 </body> 之间的文本是可见的页面内容

<h1> 与 </h1> 之间的文本被显示为标题

<p> 与 </p> 之间的文本被显示为段落

## 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

Eg：

<p>This is my first paragraph.</p>

这叫一个元素，<p>叫元素标签（<p>:开始标签,</p>:结束标签），“This is my first paragraph”叫元素内容。元素的内容可以是另一个元素，这叫嵌套。

### 空元素

没有内容的 HTML 元素被称为空元素。空元素是在开始标签中关闭的。

<br> 就是没有关闭标签的空元素（<br> 标签定义换行）。

在 XHTML、XML 以及未来版本的 HTML 中，所有元素都必须被关闭。

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

即使 <br> 在所有浏览器中都是有效的，但使用 <br /> 其实是更长远的保障。

### 使用小写标签

HTML 标签对大小写不敏感：<P> 等同于 <p>。许多网站都使用大写的 HTML 标签。

W3School 使用的是小写标签，因为万维网联盟（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 属性

HTML 标签可以拥有**属性**。属性提供了有关 HTML 元素的**更多的信息**。

属性总是以名称/值对的形式出现，比如：**name="value"**。

属性总是在 HTML 元素的**开始标签**中规定。

### 引号

始终为属性值加引号

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。

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='Bill "HelloWorld" Gates'

### 例子

* 链接： <a href="http://www.w3school.com.cn">This is a link</a>
* 居中对齐： <h1 align="center">
* 背景颜色： <body bgcolor="yellow">

# 具体属性的讲解

* 标题： <h1>This is a heading</h1>
* 水平分割线： <hr />
* HTML 注释： <!-- This is a comment -->
* Body

<body link="green" vlink="red"> 规定文档中未访问链接的默认颜色绿色 ，规定文档中已被访问链接的颜色红色。

<body text="green"> text 属性规定 HTML 文档的文本颜色。

<body bgcolor="#E6E6FA"> bgcolor 属性规定文档的背景颜色。

* <h1>

<h1 align="center">This is heading 1</h1>

|  |  |
| --- | --- |
| eft | 左对齐内容（默认值）。 |
| right | 右对齐内容。 |
| center | 居中对齐内容。 |
| justify | 对行进行伸展，这样每行都可以有相等的长度（就像在报纸和杂志中）。 |

* 段落

<p>This is<br />a para<br />graph with line breaks</p> 换行

段落中，将连续的空格（换行）视为一个空格。

* Style

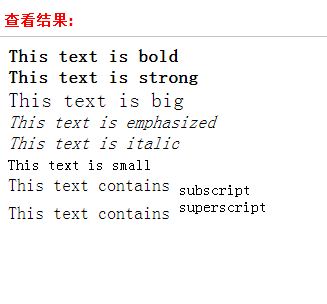
style 属性的作用：**提供了一种改变所有 HTML 元素的样式的通用方法。**

<p style="font-family:times; color:green; font-size:30px; background:yellow; text-align:center">This text is in Times and green</p>

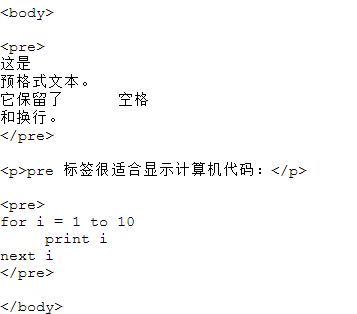
字体；颜色；大小, 背景颜色, 对齐方式。

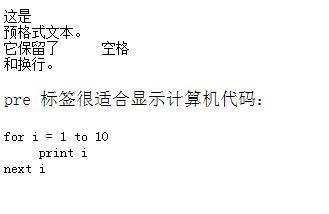
* 文本格式化

1. 文本格式化



1. 预格式文本





1. More

|  |  |
| --- | --- |
| 标签 | 描述 |
| [<b>](http://www.w3school.com.cn/tags/tag_font_style.asp) | 定义粗体文本。 |
| [<big>](http://www.w3school.com.cn/tags/tag_font_style.asp) | 定义大号字。 |
| [<em>](http://www.w3school.com.cn/tags/tag_phrase_elements.asp) | 定义着重文字。 |
| [<i>](http://www.w3school.com.cn/tags/tag_font_style.asp) | 定义斜体字。 |
| [<small>](http://www.w3school.com.cn/tags/tag_font_style.asp) | 定义小号字。 |
| [<strong>](http://www.w3school.com.cn/tags/tag_phrase_elements.asp) | 定义加重语气。 |
| [<sub>](http://www.w3school.com.cn/tags/tag_sup.asp) | 定义下标字。 |
| [<sup>](http://www.w3school.com.cn/tags/tag_sup.asp) | 定义上标字。 |
| [<ins>](http://www.w3school.com.cn/tags/tag_ins.asp) | 定义插入字。 |
| [<del>](http://www.w3school.com.cn/tags/tag_del.asp) | 定义删除字。 |

**“**计算机输出”标签

|  |  |
| --- | --- |
| **标签** | **描述** |
| [<code>](http://www.w3school.com.cn/tags/tag_phrase_elements.asp) | 定义计算机代码。 |
| [<kbd>](http://www.w3school.com.cn/tags/tag_phrase_elements.asp) | 定义键盘码。 |
| [<samp>](http://www.w3school.com.cn/tags/tag_phrase_elements.asp) | 定义计算机代码样本。 |
| [<tt>](http://www.w3school.com.cn/tags/tag_font_style.asp) | 定义打字机代码。 |
| [<var>](http://www.w3school.com.cn/tags/tag_phrase_elements.asp) | 定义变量。 |
| [<pre>](http://www.w3school.com.cn/tags/tag_pre.asp) | 定义预格式文本。 |

引用、引用和术语定义

|  |  |
| --- | --- |
| **标签** | **描述** |
| [<abbr>](http://www.w3school.com.cn/tags/tag_abbr.asp) | 定义缩写。 |
| [<acronym>](http://www.w3school.com.cn/tags/tag_acronym.asp) | 定义首字母缩写。 |
| [<address>](http://www.w3school.com.cn/tags/tag_address.asp) | 定义地址。 |
| [<bdo>](http://www.w3school.com.cn/tags/tag_bdo.asp) | 定义文字方向。 |
| [<blockquote>](http://www.w3school.com.cn/tags/tag_blockquote.asp) | 定义长的引用。 |
| [<q>](http://www.w3school.com.cn/tags/tag_q.asp) | 定义短的引用语。 |
| [<cite>](http://www.w3school.com.cn/tags/tag_phrase_elements.asp) | 定义引用、引证。 |
| [<dfn>](http://www.w3school.com.cn/tags/tag_phrase_elements.asp) | 定义一个定义项目。 |

* 引用

<http://www.w3school.com.cn/html/html_quotation_elements.asp>

* 链接

***Target属性：***

<a href="http://www.w3school.com.cn/" target="\_blank">Visit W3School!</a>

***Name属性：***

首先，我们在 HTML 文档中对锚进行命名（创建一个书签）：

<a name="tips">基本的注意事项 - 有用的提示</a>

然后，我们在同一个文档中创建指向该锚的链接：

<a href="#tips">有用的提示</a>

您也可以在其他页面中创建指向该锚的链接：

<a href="http://www.w3school.com.cn/html/html\_links.asp#tips">有用的提示</a>

在上面的代码中，我们将 # 符号和锚名称添加到 URL 的末端，就可以直接链接到 tips 这个命名锚了。

# Css学习笔记

## CSS层叠次序表：

**当同一个 HTML 元素被不止一个样式定义时，会使用哪个样式呢？**

一般而言，所有的样式会根据下面的规则层叠于一个新的虚拟样式表中，其中数字 4 拥有最高的优先权。

1. 浏览器缺省设置
2. 外部样式表
3. 内部样式表（位于 <head> 标签内部）
4. 内联样式（在 HTML 元素内部）

## CSS 语法CSS 基础语法

注意：

1. 如果值为若干单词，则要给值加引号：

p {font-family: "sans serif";}

1. 你应该在每行只描述一个属性，这样可以增强样式定义的可读性，就像这样：

p {

text-align: center;

color: black;

font-family: arial;

}

1. 是否包含空格不会影响 CSS 在浏览器的工作效果，同样，与 XHTML 不同，CSS 对大小写不敏感。不过存在一个例外：如果涉及到与 HTML 文档一起工作的话，class 和 id 名称对大小写是敏感的。

## 3. CSS 高级语法

继承：根据 CSS，子元素从父元素继承属性。但是它并不总是按此方式工作。

如果你不希望 "Verdana, sans-serif" 字体被所有的子元素继承，又该怎么做呢？比方说，你希望段落的字体是 Times。没问题。创建一个针对 p 的特殊规则，这样它就会摆脱父元素的规则：

Body {

font-family: Verdana, sans-serif;

}

td, ul, ol, ul, li, dl, dt, dd {

font-family: Verdana, sans-serif;

}

p {

font-family: Times, "Times New Roman", serif;

}

## 4. CSS 派生选择器

**通过依据元素在其位置的上下文关系来定义样式，你可以使标记更加简洁。**

Eg1：

li strong {

font-style: italic;

font-weight: normal;

}

请注意标记为 <strong> 的蓝色代码的上下文关系：

<p><strong>我是粗体字，不是斜体字，因为我不在列表当中，所以这个规则对我不起作用</strong></p>

<ol>

<li><strong>我是斜体字。这是因为 strong 元素位于 li 元素内。</strong></li>

<li>我是正常的字体。</li>

</ol>

Eg2：strong {

color: red;

}

h2 {

color: red;

}

h2 strong {

color: blue;

}

下面是它施加影响的 HTML：

<p>The strongly emphasized word in this paragraph is<strong>red</strong>. </p>

<h2>This subhead is also red.</h2>

<h2>The strongly emphasized word in this subhead is<strong>blue</strong>.</h2>

## 5. CSS id 选择器

**1. id 选择器可以为标有特定 id 的 HTML 元素指定特定的样式。**

下面的两个 id 选择器，第一个可以定义元素的颜色为红色，第二个定义元素的颜色为绿色：

#red {color:red;}

#green {color:green;}

下面的 HTML 代码中，id 属性为 red 的 p 元素显示为红色，而 id 属性为 green 的 p 元素显示为绿色。

<p id="red">这个段落是红色。</p>

<p id="green">这个段落是绿色。</p>

## ****6. css 类选择器****

1. 在 CSS 中，类选择器以一个点号显示：

.center {text-align: center}

在上面的例子中，所有拥有 center 类的 HTML 元素均为居中。在下面的 HTML 代码中，h1 和 p 元素都有 center 类。这意味着两者都将遵守 ".center" 选择器中的规则。

<h1 class="center">

This heading will be center-aligned

</h1>

<p class="center">

This paragraph will also be center-aligned.

</p>

**注意：**类名的第一个字符不能使用数字！它无法在 Mozilla 或 Firefox 中起作用。

2. 类选择器结合元素选择器：

p.important {color:red;}

选择器现在会匹配 class 属性包含 important 的所有 p 元素，但是其他任何类型的元素都不匹配，不论是否有此 class 属性。选择器 p.important 解释为：“其 class 属性值为 important 的所有段落”。 因为 h1 元素不是段落，这个规则的选择器与之不匹配，因此 h1 元素不会变成红色文本。

**3.多类选择器：**

**Eg： 我们假设 class 为 important 的所有元素都是粗体，而 class 为 warning 的所有元素为斜体，class 中同时包含 important 和 warning 的所有元素还有一个银色的背景 。就可以写作：**

<html>

<head>

<style type="text/css">

.important {font-weight:bold;}

.warning {font-style:italic;}

.important.warning {background:silver;}

</style>

</head>

<body>

<p class="important">This paragraph is very important.</p>

<p class="warning">This is a warning.</p>

<p class="important warning">This paragraph is a very important warning.</p>

</body>

</html>

**效果：**

**This paragraph is very important.**

*This is a warning.*

***This paragraph is a very important warning.***

4.id选择器与类选择器的区别：

***区别 1：只能在文档中使用一次 （同一个id只能被一种元素所有）***

与类不同，在一个 HTML 文档中，ID 选择器会使用一次，而且仅一次。

***区别 2：不能使用 ID 词列表***

不同于类选择器，ID 选择器不能结合使用，因为 ID 属性不允许有以空格分隔的词列表。

***区别 3：ID 能包含更多含义***

类似于类，可以独立于元素来选择 ID。有些情况下，您知道文档中会出现某个特定 ID 值，但是并不知道它会出现在哪个元素上，所以您想声明独立的 ID 选择器。例如，您可能知道在一个给定的文档中会有一个 ID 值为 mostImportant 的元素。您不知道这个最重要的东西是一个段落、一个短语、一个列表项还是一个小节标题。您只知道每个文档都会有这么一个最重要的内容，它可能在任何元素中，而且只能出现一个。在这种情况下，可以编写如下规则：

## 7. css 属性选择器

1. [title]{color:red;}

为带有 title 属性的所有元素设置样式

2. [title=W3School] {…}

为 title="W3School" 的所有元素设置样式

3. [title~=hello] { color:red; }

为包含指定值的 title 属性的所有元素设置样式。适用于由空格分隔的属性值

4. [lang|=en] { color:red; }

为带有包含指定值的 lang 属性的所有元素设置样式。适用于由连字符分隔的属性值

5. 注：属性选择器在为不带有 class 或 id 的表单设置样式时特别有用：

<html>

<head>

<style>

input[type="text"]

{

width:150px;

display:block;

margin-bottom:10px;

background-color:yellow;

font-family: Verdana, Arial;

}

input[type="button"]

{

width:120px;

margin-left:35px;

display:block;

font-family: Verdana, Arial;

}

</style>

</head>

<body>

<form name="input" action="" method="get">

<input type="text" name="Name" value="Bill" size="20">

<input type="text" name="Name" value="Gates" size="20">

<input type="button" value="Example Button">

</form>

</body>

</html>

注：



## 8.创建CSS

**1. 外部样式表**

当样式需要应用于很多页面时，外部样式表将是理想的选择。在使用外部样式表的情况下，你可以通过改变一个文件来改变整个站点的外观。每个页面使用 <link> 标签链接到样式表。<link> 标签在（文档的）头部：

<head>

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

</head>

浏览器会从文件 mystyle.css 中读到样式声明，并根据它来格式文档。

外部样式表可以在任何文本编辑器中进行编辑。文件不能包含任何的 html 标签。样式表应该以 .css 扩展名进行保存。下面是一个样式表文件的例子：

hr {color: sienna;}

p {margin-left: 20px;}

body {background-image: url("images/back40.gif");}

不要在属性值与单位之间留有空格。

假如你使用 “margin-left: 20 px” 而不是 “margin-left: 20px” ，它仅在 IE 6 中有效，但是在 Mozilla/Firefox 或 Netscape 中却无法正常工作。

**2. 内部样式表**

当单个文档需要特殊的样式时，就应该使用内部样式表。你可以使用 <style> 标签在文档头部定义内部样式表，就像这样:

<head>

<style type="text/css">

hr {color: sienna;}

p {margin-left: 20px;}

body {background-image: url("images/back40.gif");}

</style>

</head>

**3. 内联样式**

由于要将表现和内容混杂在一起，内联样式会损失掉样式表的许多优势。请慎用这种方法，例如当样式仅需要在一个元素上应用一次时。

要使用内联样式，你需要在相关的标签内使用样式（style）属性。Style 属性可以包含任何 CSS 属性。本例展示如何改变段落的颜色和左外边距：

<p style="color: sienna; margin-left: 20px">

This is a paragraph

</p>

注意：上者覆盖下者！



## 9. CSS 框模型概述 (**Box Model**)

## 10. CSS 定位 (Positioning)

## ****CSS 博客学习（加强理解）****

### http://images.cnitblog.com/blog/138012/201502/060828063908324.png1. 浏览器的工作过程

### 2. CSS的页面呈现

页面呈现可以分为两类——***文字，块***。

***文字***： 可以设置字体、字号、加粗、斜体、背景色等等；

***块***： 有盒子模型、浮动、定位、display、背景等；

### 3. 布局

经典例子：

1. 经典三列布局

2. Bootstrap栅格布局

3. 百度首页布局

4. 微博布局

5. 人人网布局

6. 瀑布流布局

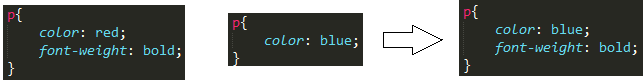
……

### 4. 层叠的概念 & 样式的来源



使用顺序应该是从上往下排列。（换句话说，层叠的顺序是上者覆盖下者）

例子：



特例（反例）：!important

如下图。虽然color:blue是后来者，但是它没有居上，因为前者color:red跟着!important。这就像得到一个尚方宝剑，有最高执行权，其他人统统让路，否则佛挡杀佛。



附录：来自于浏览器的默认样式：

1 html, address,blockquote,body, dd, div,dl, dt, fieldset, form,frame, frameset,h1, h2, h3, h4,h5, h6, noframes,ol, p, ul, center,dir, hr, menu, pre { display: block }/\*以上列表元素默认状态下一块状显示，未显示的将以内联元素显示，该列表针对HTML4版本，部分元素在XHTML1中将废弃\*/

2 li { display: list-item }/\*默认以列表显示\*/

3 head { display: none }/\*默认不显示\*/

4 table { display: table }/\*默认为表格显示\*/

5 tr { display: table-row }/\*默认为表格行显示\*/

6 thead { display: table-header-group }/\*默认为表格头部分组显示\*/

7 tbody { display: table-row-group }/\*默认为表格行分组显示\*/

8 tfoot { display: table-footer-group }/\*默认为表格底部分组显示\*/

9 col { display: table-column }/\*默认为表格列显示\*/

10 colgroup { display: table-column-group }/\*默认为表格列分组显示\*/

11 td, th { display: table-cell; }/\*默认为单元格显示\*/

12 caption { display: table-caption }/\*默认为表格标题显示\*/

13 th { font-weight: bolder; text-align: center }/\*默认为表格标题显示，呈现加粗居中状态\*/

14 caption { text-align: center }/\*默认为表格标题显示，呈现居中状态\*/

15 body { margin: 8px; line-height: 1.12 }

16 h1 { font-size: 2em; margin: .67em 0 }

17 h2 { font-size: 1.5em; margin: .75em 0 }

18 h3 { font-size: 1.17em; margin: .83em 0 }

19 h4, p, blockquote, ul, fieldset, form, ol, dl, dir, menu { margin: 1.12em 0 }

20 h5 { font-size: .83em; margin: 1.5em 0 }

21 h6 { font-size: .75em; margin: 1.67em 0 }

22 h1, h2, h3, h4, h5, h6, b,strong { font-weight: bolder }

23 blockquote { margin-left: 40px; margin-right: 40px }

24 i, cite, em,var, address { font-style: italic }

25 pre, tt, code, kbd, samp { font-family: monospace }

26 pre { white-space: pre }

27 button, textarea, input, object, select { display:inline-block; }

28 big { font-size: 1.17em }

29 small, sub, sup { font-size: .83em }

30 sub { vertical-align: sub }/\*定义sub元素默认为下标显示\*/

31 sup { vertical-align: super }/\*定义sub元素默认为上标显示\*/

32 table { border-spacing: 2px; }

33 thead, tbody, tfoot { vertical-align: middle }/\*定义表头、主体表、表脚元素默认为垂直对齐\*/

34 td, th { vertical-align: inherit }/\*定义单元格、列标题默认为垂直对齐默认为继承\*/

35 s, strike, del { text-decoration: line-through }/\*定义这些元素默认为删除线显示\*/

36 hr { border: 1px inset }/\*定义分割线默认为1px宽的3D凹边效果\*/

37 ol, ul, dir, menu, dd { margin-left: 40px }

38 ol { list-style-type: decimal }

39 ol ul, ul ol, ul ul, ol ol { margin-top: 0; margin-bottom: 0 }

40 u, ins { text-decoration: underline }

41 br:before { content: "A" }/\*定义换行元素的伪对象内容样式\*/

42 :before, :after { white-space: pre-line }/\*定义伪对象空格字符的默认样式\*/

43 center { text-align: center }

44 abbr, acronym { font-variant: small-caps; letter-spacing: 0.1em }

45 :link, :visited { text-decoration: underline }

46 :focus { outline: thin dotted invert }

47 /\* Begin bidirectionality settings (do not change) \*/

48 BDO[DIR="ltr"] { direction: ltr; unicode-bidi: bidi-override }/\*定义BDO元素当其属性为DIR="ltr"时的默认文本读写显示顺序\*/

49 BDO[DIR="rtl"] { direction: rtl; unicode-bidi: bidi-override }/\*定义BDO元素当其属性为DIR="rtl"时的默认文本读写显示顺序\*/

50 \*[DIR="ltr"] { direction: ltr; unicode-bidi: embed }/\*定义任何元素当其属性为DIR="ltr"时的默认文本读写显示顺序\*/

51 \*[DIR="rtl"] { direction: rtl; unicode-bidi: embed }/\*定义任何元素当其属性为DIR="rtl"时的默认文本读写显示顺序\*/

52 @media print { /\*定义标题和列表默认的打印样式\*/

53 h1 { page-break-before: always }

54 h1, h2, h3, h4, h5, h6 { page-break-after: avoid }

55 ul, ol, dl { page-break-before: avoid }

56 }