



## 第 5 章

# CSS 文本常用属性

### 本章学习目标

- 掌握 CSS 常用的文本样式和排版属性。
- 理解 CSS 伪类的概念及作用。
- 掌握利用 CSS 设置超链接的 4 种状态的方法。
- 了解其他伪类的用法。

在掌握了 CSS 的语法规则和应用方法之后,想要书写结构层和样式层分离的页面,就要掌握 CSS 的属性,本章将讲解 CSS 文本的常用属性。

## 5.1 文本属性



### 5.1.1 文本颜色

文本颜色(color)被用来设置文字的颜色,属性的取值方法如下。

- (1) 十六进制值,如 #FF0000。
- (2) RGB 值,如 RGB(255,0,0)。
- (3) 颜色的名称,如 red。

如例 5.1 所示,分别使用以上 3 种方法为文本颜色进行赋值,结果页面中的所有一级标题为红色,六级标题为绿色,段落文本为蓝色。

#### 例 5.1

---

```
1 <!DOCTYPE html >
2 <html lang = "en">
3 <head >
4     <meta charset = "UTF-8">
5     <title>文本颜色</title>
6     <style type = "text/css">
7         h1{
8             color: # ff0000;
9         }
10        h6{
11            color:rgb(0,255,0);
```

---

```

12     }
13     p{
14         color:blue;
15     }
16 </style>
17 </head>
18 < body>
19     < h1 >一级标题(红色)</h1 >
20     < h6 >六级标题(绿色)</h6 >
21     < p>段落文本(蓝色)</p>
22 </body>
23 </html >

```

### 5.1.2 文本字体

font-family 用来设置文本字体。font-family 可以用逗号隔开多个字体名称。假如客户端不认识第 1 种字体,就自动切换到第 2 种字体,第 2 种字体不认识就切换到第 3 种,以此类推。如果所有列出的字体都不认识,就使用默认字体显示。

```
body{font-family:"Helvetica Neue", Helvetica, Arial, "Microsoft Yahei", "Hiragino Sans GB", "Heiti SC", "WenQuanYi Micro Hei", sans-serif}
```

如上例所示,定义了多个字体,一般将中文字体写在英文字体后面。因为英文字体只能渲染英文、数字和一些特殊符号,当遇到中文文本时,会向后寻找中文字体。

Web 页面中,常用的中文字体有宋体、微软雅黑、华文细黑(Mac OS 中);常用的英文字体有 Tahoma、Helvetica、Arial、sans-serif。其中, Tahoma 和 Arial 分别用于早期 Windows 和现在 Windows 系统; Helvetica 用于 Mac OS; sans-serif 用于 Linux 系统。

字体文件存储于计算机磁盘中的 C:/window/fonts 位置。

### 5.1.3 文本字号

文本字号(font-size)属性可能的取值形式如表 5.1 所示。

表 5.1 font-size 属性可能的取值形式

值	描 述
length(常用)	把 font-size 设置成一个固定的值。如 14px、16px
%	把 font-size 设置为基于父元素的一个百分比值
xx-small	把字体尺寸设置为不同的尺寸,从 xx-small 到 xx-large。 默认值: medium
x-small	
small	
medium	
large	
x-large	
xx-large	

续表

值	描 述
smaller	把 font-size 设置为比父元素更小的尺寸
larger	把 font-size 设置为比父元素更大的尺寸
inherit	规定从父元素继承字体大小

如例 5.2 所示,将段落文本字号设置为 12px。

### 例 5.2

```

1  <!DOCTYPE html >
2  <html lang = "en">
3  <head >
4      <meta charset = "UTF-8">
5      <title>文本字号</title>
6      <style type = "text/css">
7          p{
8              font-size:12px;
9          }
10     </style>
11 </head >
12 <body >
13     <p>段落一</p>
14     <p>段落二</p>
15     <p>段落三</p>
16 </body >
17 </html >

```



## 5.1.4 字体风格

字体风格(font-style)属性用于将字体设置为斜体、倾斜体或正常字体。该属性可能的取值如表 5.2 所示。

表 5.2 font-style 属性可能的取值

值	描 述
normal	默认值。浏览器显示标准字体样式
italic	浏览器显示斜体字体样式
oblique	浏览器显示倾斜字体样式
inherit	从父元素继承字体样式

如例 5.3 所示,分别为段落文本定义了普通样式、斜体样式和倾斜体样式,效果如图 5.1 所示。

## 例 5.3

```

1  <!DOCTYPE html >
2  <html lang = "en">
3  <head >
4      <meta charset = "UTF-8">
5      <title>文本风格</title>
6      <style type = "text/css">
7          p.normal {font-style:normal}
8          p.italic {font-style:italic}
9          p.oblique {font-style:oblique}
10     </style>
11 </head>
12 <body >
13     <p class = "normal">段落一</p>
14     <p class = "italic">段落二</p>
15     <p class = "oblique">段落三</p>
16 </body >
17 </html >

```



图 5.1 font-style 属性正常样式、斜体样式和倾斜体样式效果

### 5.1.5 小型大写字母

设置小型大写字母(font-variant)的字体显示文本时,所有的小写字母均会被转换为大写字母。但是,小型的大写文本与其余文本相比,字体尺寸更小。这个属性可能的取值如表 5.3 所示。

表 5.3 font-variant 属性可能的取值

值	描 述
normal	默认值。显示标准字体
small-caps	显示小型大写字母的字体
inherit	从父元素继承属性值



如例 5.4 所示,分别为两个段落设置了不同的字母显示形式,效果如图 5.2 所示。

### 例 5.4

```
1 <!DOCTYPE html >
2 <html lang = "en">
3 <head >
4     <meta charset = "UTF-8">
5     <title>小型大写字母</title>
6     <style type = "text/css">
7         p.noraml{
8             font-variant:normal;
9         }
10        p.small{
11            font-variant:small-caps;
12        }
13    </style>
14 </head >
15 <body >
16     <p class = "normal"> This is a paragraph.</p>
17     <p class = "small"> This is a paragraph.</p>
18 </body >
19 </html >
```

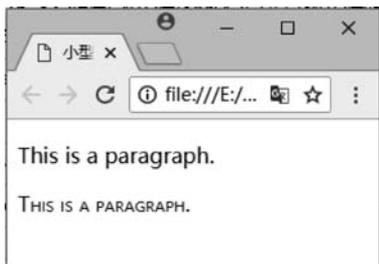


图 5.2 小型大写字母和正常英文字体对比



## 5.1.6 文本粗细

文本粗细(font-weight)用来设置文本显示的粗细,可能的取值如表 5.4 所示。

表 5.4 font-weight 属性可能的取值

值	描 述
normal	默认值。显示标准字符样式
bold	粗体字符
bolder	更粗体字符
lighter	更细体字符

续表

值	描 述
100	由细到粗的字符。其中,400 等同于 normal,700 等同于 bold
200	
300	
400	
500	
600	
700	
800	
900	
inherit	

如例 5.5 所示,分别为段落文本定义了正常粗细字体、粗体和细体,显示效果如图 5.3 所示。

### 例 5.5

```

1  <!DOCTYPE html >
2  <html lang = "en">
3  <head >
4      <meta charset = "UTF-8">
5      <title>文本粗细</title >
6      <style type = "text/css">
7          p.normal{
8              font-weight:normal;
9          }
10         p.cu{
11             font-weight: bold;
12         }
13         p.xi{
14             font-weight: lighter;
15         }
16     </style>
17 </head >
18 <body >
19     <p class = "normal">段落文本,正常粗细</p >
20     <p class = "cu">段落文本,粗体</p >
21     <p class = "xi">段落文本,细体</p >
22 </body >
23 </html >

```



图 5.3 不同文本粗细效果对比



## 5.1.7 行高

用于设置行间的距离,即行高(line-height),不允许为负值。行高属性可能的取值如表 5.5 所示。

表 5.5 line-height 属性可能的取值

值	描 述
normal	默认值。设置合理的行间距
number	数字,用此数字与当前的字体尺寸相乘来设置行间距
length	设置固定行距
%	基于当前字体尺寸的百分比行距
inherit	从父元素继承值

如例 5.6 所示,分别使用固定像素值、百分比和数值的形式设置了行距,效果如图 5.4 所示。

### 例 5.6

```

1  <!DOCTYPE html >
2  <html lang = "en">
3  <head >
4      <meta charset = "UTF-8">
5      <title>行距</title>
6      <style type = "text/css">
7          p.big{line-height: 35px;}
8          p.small{line-height: 15px;}
9          p.persents{line-height: 200 %;}
10         p.number{line-height: 2;}
11     </style >
12 </head >
13 <body >
14     <p>这是默认行高<br>多数浏览器的默认行高为 20px <br>认行高段落 <br>默认行高段落 <br>
15     </p>
16     <p class = "big">这是段落 35 像素行高<br>这个段落文本行高较大 <br>这个段落文本行高较大 <br>
17     </p>
18     <p class = "small">这是段落 15 像素行高<br>这个段落文本行高较小 <br>这个段落文本行高较小 <br>
19     </p>
20     <p class = "persents">这是段落 200 % 行高<br>这个段落文本行高较大 <br>这个段落文本行高较大 <br>
21     </p>
22     <p class = "number">这是段落 2 倍字号行高<br>这个段落文本行高较大 <br>这个段落文本行高较大 <br>
23     </p>
24 </body >
25 </html >

```



图 5.4 line-height 设置不同值的效果

注意：line-height 与 font-size 之间的差值(即行间距)分为两半,分别加到一个文本行内容的顶部和底部,包含这些内容的最小框就是行框。

### 5.1.8 font 综合设置属性

font 综合设置属性在一个声明中设置所有字体的属性,其语法格式如下:

```
font: font-style font-variant font-weight font-size/line-height font-family
```

在 CSS 中,属性值可以拆分成多个属性表示的称为复合属性,如下所示。

---

```
font: 13px/22px Arial, Helvetica, sans-serif;
```

---

上述示例就是使用 font 复合属性同时设置了字号、行高和字体。字号和行高属性值之间有个斜杠,等同于下面的代码。

---

```
font-size:13px;
line-height:22px;
font-family:Arial, Helvetica, sans-serif;
```

---

在使用 font 综合设置属性时,不需要设置的属性可以省略(取默认值)。但是,必须保留 font-size 和 font-family 属性,否则 font 属性将不起作用。





## 5.1.9 文本对齐

文本对齐(text-align)属性设置元素中文本的水平对齐方式,该可能的取值如表 5.6 所示。

表 5.6 text-align 属性可能的取值

值	描 述	值	描 述
left	左对齐,默认值	center	居中
right	右对齐	justify	两端对齐

如例 5.7 所示,设置段落文本的 3 种对齐方式,效果如图 5.5 所示。

### 例 5.7

```

1  <!DOCTYPE html >
2  <html lang = "en">
3  <head>
4      <meta charset = "UTF-8">
5      <title>水平对齐方式</title>
6      <style type = "text/css">
7          p.right{text-align: right;}
8          p.center{text-align: center;}
9          p.left{text-align: left;}
10     </style>
11 </head>
12 <body>
13     <p>默认对齐方式文本</p>
14     <p class = "left">设置了左对齐文本</p>
15     <p class = "right">设置了右对齐文本</p>
16     <p class = "center">设置了居中对齐文本</p>
17 </body>
18 </html >

```

在例 5.7 中,第 13 行代码未设置 text-align 属性与第 14 行代码设置的 text-align 值为 left 对齐方式效果相同,可见默认的水平对齐方式为左对齐。



图 5.5 text-align 水平对齐方式不同值的效果

### 5.1.10 文本装饰

文本装饰(text-decoration)属性规定添加到文本的修饰,如上画线、下画线和中画线等,可能的取值如表 5.7 所示。



表 5.7 text-decoration 属性可能的取值

值	描 述	值	描 述
none	标准文本,默认值	overline	带上画线
underline	带下画线	line-through	带中画线

在例 5.8 中,分别为 3 个三级标题定义了文本装饰为带下画线、带上画线、带中画线,效果如图 5.6 所示。

#### 例 5.8

```

1  <!DOCTYPE html >
2  <html lang = "en">
3  <head >
4      <meta charset = "UTF-8">
5      <title>文本装饰</title>
6      <style type = "text/css">
7          h3.unline{text-decoration: underline;}
8          h3.upline{text-decoration: overline;}
9          h3.cenline{text-decoration: line-through;}
10     </style>
11 </head >
12 <body >
13     <h3>标准文本</h3 >
14     <h3 class = "unline">带下画线文本</h3 >
15     <h3 class = "upline">带上画线文本</h3 >
16     <h3 class = "cenline">带中画线文本</h3 >
17 </body >
18 </html >

```



图 5.6 text-decoration 不同属性值效果



### 5.1.11 文本缩进

文本缩进(text-indent)属性定义文本块中首行文本的缩进。该属性值允许为负值。如果为负值,则首行会被缩进到左边。该属性可能的取值如表 5.8 所示。

表 5.8 text-indent 属性可能的取值

值	描 述
length	固定的缩进。默认值为 0
%	基于父元素百分比缩进

在中文段落排版中,一般设置文字首行缩进两个中文字符。如例 5.9 所示,定义了段落标记缩进值为 2em。由于采用了 em 为单位,左缩进 2 倍字号大小,达到缩进两个字符的效果,如图 5.7 所示。注意:在段落的第 1 行会缩进,而换行之后的行不会缩进。

#### 例 5.9

```

1  <!DOCTYPE html >
2  <html lang = "en">
3  <head >
4      <meta charset = "UTF-8">
5      <title>段落缩进</title>
6      <style type = "text/css">
7          p{text-indent: 2em;}
8      </style>
9  </head >
10 <body >
11  <p>这是第一个段落<br>第一个段落换行之后的文本<br>第一个段落再换行之后的文本</p>
12  <p>这是第二个段落<br>第二个段落换行之后的文本<br>第二个段落再换行之后的文本</p>
13 </body >
14 </html >

```



图 5.7 text-indent 设置首行缩进两个字符效果

### 5.1.12 字间隔



字间隔(word-spacing)属性定义元素中字之间插入多少空白符。“字”指的是由空白符包围的一个字符串,取值可以为正值或负值,0 等同于取值为 normal。

如例 5.10 所示,分别为两个段落定义了宽松的字间隔和紧凑的字间隔。从图 5.8 可以看出,当 word-spacing 取值为正时,字间隔较为宽松;当 word-spacing 取值为负时,字间隔紧凑甚至缩在一起。

例 5.10

```

1  <!DOCTYPE html >
2  <html lang = "en">
3  <head >
4      <meta charset = "UTF-8">
5      <title>字间距</title>
6      <style type = "text/css">
7          p.big{word-spacing: 40px;}
8          p.small{word-spacing: -1em;}
9      </style >
10 </head >
11 <body >
12     <p class = "big"> This is some text.</p >
13     <p class = "small"> This is some text.</p >
14 </body >
15 </html >

```



图 5.8 word-spacing 设置正值、负值效果

### 5.1.13 字母间隔

字母间隔(letter-spacing)属性修改的是字母之间的间隔,取值可为正值、负值或 normal(0)。取值可使字母之间的间隔增加或减少指定的值。

如例 5.11 所示,定义了两个四级标题的字母间隔,取正值时字母间隔稀疏,取负值时紧凑在一起甚至文本显示不全,效果如图 5.9 所示。

## 例 5.11

```

1  <!DOCTYPE html >
2  <html lang = "en">
3  <head >
4      <meta charset = "UTF-8">
5      <title>字母间隔</title>
6      <style type = "text/css">
7          h4.big{letter-spacing: 16px;}
8          h4.small{letter-spacing: -1em;}
9      </style >
10 </head >
11 <body >
12     <h4 class = "big">This is the header 4.</h4 >
13     <h4 class = "small">This is the header 4.</h4 >
14 </body >
15 </html >

```



图 5.9 letter-spacing 设置正值和负值时的效果



## 5.1.14 处理空白符

处理空白符(white-space)属性设置布局过程中如何处理元素内的空白。在编辑代码时,会手动输入空格、换行等;在浏览器渲染页面时,默认情况会合并手动输入的空格,对于按 Enter 键输入的换行则忽略,但是可以识别<br>标记进行换行,且内容会根据容器大小在边界换行。white-space 属性可能的取值及处理这些情况的方法如表 5.9 所示。

表 5.9 white-space 属性可能的取值及处理这些情况的方法

值	源码空格	源码换行	 换行	容器边界换行
normal	合并	忽略	换行	换行
pre	保留	换行	换行	不换行
nowrap	合并	忽略	换行	不换行
pre-wrap	保留	换行	换行	换行
pre-line	合并	换行	换行	换行

white-space 属性会导致文本排版形式的变化,经常与 overflow 属性一起使用。

在例 5.12 中,从第 7 行代码开始,定义<p>标记宽度为 100px,overflow:hidden 表示将超出定义大小的内容隐藏;第 11 行代码中 text-overflow:ellipsis 表示溢出的文本将被截断并添加省略号,效果如图 5.10 所示。

## 例 5.12

```
1 <!DOCTYPE html >
2 <html lang = "en">
3 <head >
4     <meta charset = "UTF-8">
5     <title>文本溢出显示省略号</title>
6     <style >
7         p{
8             width:100px;
9             overflow: hidden;
10            white-space:nowrap;
11            text-overflow: ellipsis;
12        }
13    </style >
14 </head >
15 <body >
16     <p>前端开发订阅号专注前端开发和技术分享,如果描述超过 100 像素,则会隐藏,添加省略号</p >
17 </body >
18 </html >
```



图 5.10 溢出文本显示省略号效果

## 5.2 应用案例

制作如图 5.11 所示的页面效果,其代码如例 5.13 所示。



图 5.11 文字排版效果

其中,发布日期之前的“0.775”为红色,之后的发布信息为比标题和正文浅些的灰色;在段落正文“空白折叠现象……”“文档流指的是元素……”中行距为 1.5em,还有个别词组需要加粗显示。

## 例 5.13

---

```
1  <!DOCTYPE html >
2  <html lang = "en">
3  <head >
4      <meta charset = "UTF-8">
5      <title>CSS 标准文档流-EASY BOOK </title>
6      <style type = "text/css">
7          h1{
8              color: # 404040;
9              font-size:30px;
10             font-weight:700;
11         }
12         h1,p{
13             color: # 404040;
14         }
15         p img.author{
16             width:44px;
17             height:44px;
18         }
19         p.info{
20             color: # 969696;
21             font-size: 12px;
22             line-height: 0.1em;
23         }
24         .red{
25             color:red;
26         }
27         .tit1{
28             line-height: 1.5em;
29         }
30         .content{
31             line-height: 1.5em;
32         }
33         .bold{
34             font-weight: bold;
35         }
36     </style>
37 </head >
38 <body >
39     <h1 >CSS 标准文档流</h1 >
40     <p >
41         <img src = "images/iconh.jpg" alt = "" align = "left" hspace = "8px" class = "author">
42         天天向上
```

---



## 例 5.14

```
1 <!DOCTYPE html >
2 <html >
3 <head >
4     <meta charset = "UTF-8">
5     <title>超链接的 4 种状态</title>
6     <style type = "text/css">
7         a:link{color: # f00;} /* 红色 */
8         a:visited{color: # 0f0;} /* 绿色 */
9         a:hover{color: # f0f;} /* 洋红 */
10        a:active{color: # 00f;} /* 蓝色 */
11    </style >
12 </head >
13 <body >
14     <a href = "http://www.baidu.com">百度</a >
15 </body >
16 </html >
```

注意：a:hover 的定义必须位于 a:link 和 a:visited 之后才能生效；a:active 必须位于 a:hover 之后才能生效。

伪类可以与 CSS 的选择器配合使用，如例 5.15 所示。

## 例 5.15

```
1 <!DOCTYPE html >
2 <html >
3 <head >
4     <meta charset = "UTF-8">
5     <title>伪类与其他选择器配合</title>
6     <style type = "text/css">
7         a.noline:link{
8             text-decoration: none;
9         }
10    </style >
11 </head >
12 <body >
13     <a href = "http://www.baidu.com">普通超链接</a >
14     <br/>
15     <a class = "noline" href = "http://www.sina.com.cn">使用 noline 类样式的超链接</a >
16 </body >
17 </html >
```



图 5.12 伪类与类选择器配合使用的效果

只有使用了 noline 类样式的 <a> 标记才没有下划线，效果如图 5.12 所示。

如果想制作如图 5.13 所示仿百度搜索结果页面。大标题的超链接访问过与未访问过的显示颜色不同，且这两种样式中的关键字都显示红色；站点名称的文本样式也不同。具体代码如例 5.16 所示。



图 5.13 仿百度搜索结果页面

注意：搜索关键字标红显示，与所在标记文本颜色不同，将这类文本嵌套在`<em>`等这样的标记中；再应用定义好的类样式，如例 5.16 中第 7~10 行代码和第 34、35 行代码所示。制作类似效果的时候也可以用`<span>`和`<strong>`等类似的行内标记（详见本书 6.1.2 节）。

## 例 5.16

---

```

1  <!DOCTYPE html >
2  <html lang = "en">
3  <head >
4      <meta charset = "UTF-8">
5      <title>超链接标记 CSS </title>
6      <style type = "text/css">
7          a em, p em{
8              color:red;
9              font-style: normal;
10         }
11         p. des{
12             color: # 333;
13             font:12px/1.5em "Microsoft Yahei";
14             //微软雅黑字体
15         }
16         .date{
17             color: # 666;
18         }
19         .icon{
20             width:14px;
21         }
22         .website{
23             color:green;
24             font-size: 12px;

```

---

```
25     }
26     a.m:visited,a.m:link{
27         font-size:12px;
28         color:#666;
29     }
30 </style>
31 </head>
32 <body>
33     <p>
34         <a href="#">关于<em>css 超链接</em>样式的小<em>案例</em>weishaohang 的
           专栏-CSDN 博客
35     </a>
36 </p>
37 <p class="des">
38 <!--span 标记本身不会产生任何样式上的变化 -->
39     <span class="date">2013 年 3 月 28 日 -</span> 通过一个简单的对文本设置<em>
           css</em> a 样式情况了解学习 CSS a 锚文本样式。1、案例<em>css</em>...一、
           HTMLHTML: 超文本标记语言."超文本": 页面中包含图片、链接等...
40 </p>
41 <p>
42     
43     <a class="website" href="#">CSDN 技术社区</a>
44      -
45     <a href="#" class="m">百度快照</a>
46 </p>
47 <p> &nbsp;</p>
48 <p>
49     <a href="#123">HTML<em>超链接</em>a 标记<em>CSS</em>样式写法示例 -
           <em>CSS</em>教程_Web 开发者 - 专业的开发...
50 </a>
51 </p>
52 <p class="des">
53 <!--span 标记本身不会产生任何样式上的变化 -->
54     <span class="date">2017 年 7 月 23 日 -</span> 下面我们介绍对于<em>css</em>
           初学者会有一些帮助了,主要告诉你 HTML<em>超链接</em>标记<em>CSS</em>样
           式写法,这里简单举了一个实例。定义链接样式的四个伪类 1 :link 2 :visite...
55 </p>
56 <p>
57     <a class="website" href="#">www.webkfz.com/WEBkF/C...</a>
58      -
59     <a href="#" class="m">百度快照</a>
60 </p>
61 </body>
62 </html>
```



### 5.3.2 其他伪类

在 CSS2 之前的版本中,定义的伪类如表 5.10 所示。

表 5.10 CSS2 之前版本定义的伪类

属 性	描 述	CSS 版本
:link	向未被访问的超链接添加样式	1
:visited	向被访问过的超链接添加样式	1
:hover	当鼠标指针悬停在元素上方时,向元素添加样式	1
:active	向被激活的元素添加样式	1
:focus	向拥有键盘输入焦点的元素添加样式	2
:first-child	向元素的第一个子元素添加样式	2
:lang	向带有指定 lang 属性的元素添加样式	2

如例 5.17 所示,演示了 :first-child 使用效果。p>em:first-child 表示选择<p>标记的第 1 个<em>元素。

例 5.17

```

1  <!DOCTYPE html >
2  <html >
3  <head >
4      <meta charset = "UTF-8">
5      <title> firstchild 用法</title >
6      <style type = "text/css">
7          p>em:first-child{
8              font-weight: bold;
9          }
10     </style >
11 </head >
12 <body >
13     <p><em>:hover"悬停"</em>: 鼠标<em>放到标记上</em>的时候。</p>
14     <p><em>:active"激活"</em>: 鼠标<em>单击标记,但是不松开</em>鼠标。</p>
15 </body >
16 </html >

```

在例 5.17 中,第 8 行代码 font-weight 是字体粗细属性,bold 表示粗体。由于选择器只选中了<p>标记中的第 1 个<em>标记,只有第 1 对<em>标记内文本加粗显示,效果如图 5.14 所示。



图 5.14 使用 em:first-child 选择第 1 个<em>子元素