**Css-基本文本和字体样式**

用于样式文本的 CSS 属性通常可以分为两类：

* **字体样式**: 作用于字体的属性，会直接应用到文本中，比如使用哪种字体，字体的大小是怎样的，字体是粗体还是斜体，等等。
* **文本布局风格**: 作用于文本的间距以及其他布局功能的属性，比如，允许操纵行与字之间的空间，以及在内容框中，文本如何对齐。

1. 样式字体的属性
   1. **颜色**：[color](https://developer.mozilla.org/zh-CN/docs/Web/CSS/color) 属性设置选中元素的前景内容的颜色 (通常指文本，不过也包含一些其他东西，或者是使用 [text-decoration](https://developer.mozilla.org/zh-CN/docs/Web/CSS/text-decoration) 属性放置在文本下方或上方的线 (underline overline)。

color 也可以接受任何合法的 [CSS 颜色单位](https://developer.mozilla.org/zh-CN/Learn/CSS/Introduction_to_CSS/Values_and_units#Colors)。

\*text-decoration 这个 CSS 属性是用于设置文本排版（下划线、顶划线、删除线或者闪烁）。下划线和顶划线修饰于文本的下部，删除线位于文本之上。

文本修饰属性会延伸到后代元素。这意味着如果祖先元素指定了文本修饰属性，后代元素则不能将其删除。例如，在如下标记中<p>This text has <em>some emphasized words</em> in it.</p>，应用样式p { text-decoration: underline } 会对整个段落添加下划线，再添加样式 em { text-decoration: none } 则不会引起任何改变；整个段落仍然有下划线修饰。然而，新加样式 em { text-decoration: overline } 则会在<em>标记的文字上添加另一种新样式。

**语法**

/\*关键值\*/

text-decoration: none; /\*没有文本装饰\*/

text-decoration: underline red; /\*红色下划线\*/

text-decoration: underline wavy red; /\*红色波浪形下划线\*/

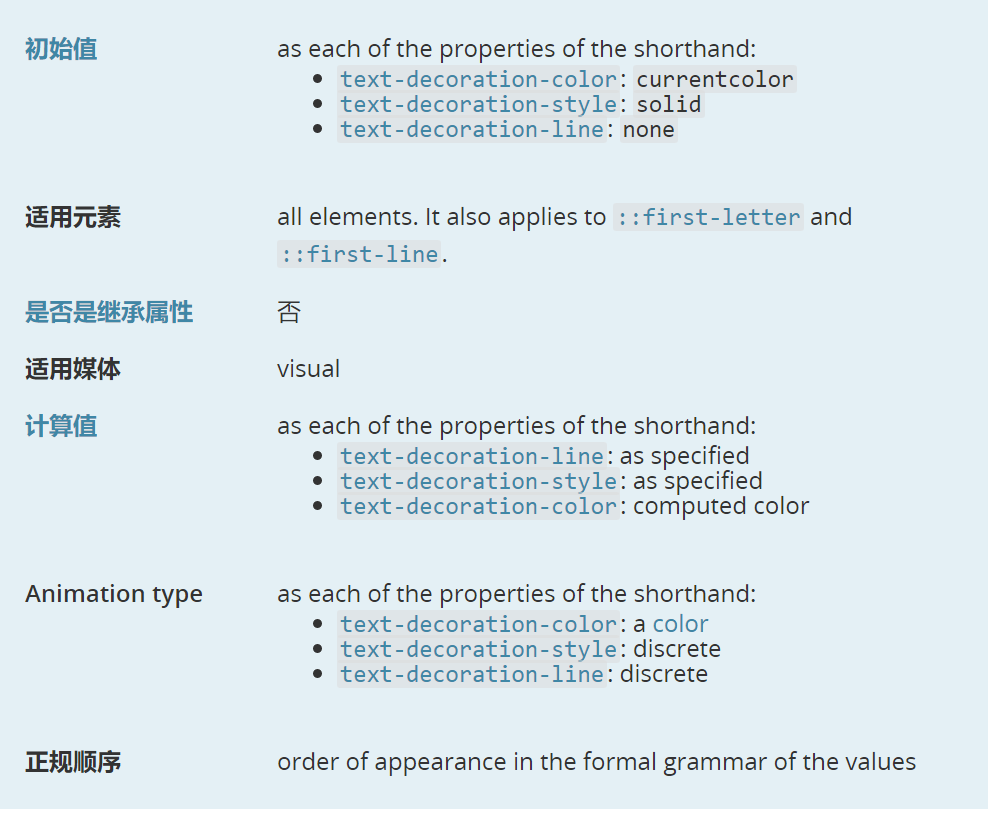
/\*全局值\*/

text-decoration: inherit;

text-decoration: initial;

text-decoration: unset;

text-decoration属性是一种简写属性，并且可以使用普通属性三个值中的任何一个。普通属性如下：[text-decoration-line](https://developer.mozilla.org/zh-CN/docs/Web/CSS/text-decoration-line)，[text-decoration-color](https://developer.mozilla.org/zh-CN/docs/Web/CSS/text-decoration-color" \o "CSS 属性 text-decoration-color 用于设置文本修饰线的颜色。文本修饰线是通过 text-decoration-line 属性指定的，修饰线包括下划线、上划线、删除线和波浪线，波浪线的典型用途是标明内容拼写错误（仅举例）。被指定的颜色会作用到该属性值有效范围内的所有修饰线上。)和[text-decoration-style](https://developer.mozilla.org/zh-CN/docs/Web/CSS/text-decoration-style" \o "CSS 属性 text-decoration-style 用于设置由 text-decoration-line 设定的线的样式。线的样式会应用到所有被 text-decoration-line 设定的线，不能为其中的每条线设置不同的样式。当要设置多个线修饰属性时，用 text-decoration 简写属性会比分别写多个属性更方便。)



* 1. **字体种类：**

[**font-family**](https://developer.mozilla.org/zh-CN/docs/Web/CSS/font-family)**属性：**规定元素的字体系列。

font-family 可以把多个字体名称作为一个“回退”系统来保存。如果浏览器不支持第一个字体，则会尝试下一个。也就是说，font-family 属性的值是用于某个元素的字体族名称或/及类族名称的一个优先表。浏览器会使用它可识别的第一个值。

有两种类型的字体系列名称：

指定的系列名称：具体字体的名称，比如："times"、"courier"、"arial"。

通常字体系列名称：比如："serif"、"sans-serif"、"cursive"、"fantasy"、"monospace"

提示：使用逗号分割每个值，并始终提供一个类族名称作为最后的选择。

注意：使用某种特定的字体系列（Geneva）完全取决于用户机器上该字体系列是否可用；这个属性没有指示任何字体下载。因此，强烈推荐使用一个通用字体系列名作为后路。

* 1. **字体大小：**

[**font-size**](https://developer.mozilla.org/zh-CN/docs/Web/CSS/font-size)**属性：**指定字体的大小。元素的 font-size 属性是从该元素的父元素继承的。

/\* <absolute-size>，绝对大小值 \*/

font-size: xx-small;

font-size: x-small;

font-size: small;

font-size: medium;

font-size: large;

font-size: x-large;

font-size: xx-large;

绝对值：

* 将文本设置为指定的大小
* 不允许用户在所有浏览器中改变文本大小（不利于可用性）
* 绝对大小在确定了输出的物理尺寸时很有用

/\* <relative-size>，相对大小值 \*/

font-size: larger;

font-size: smaller;

相对大小：

* 相对于周围的元素来设置大小
* 允许用户在浏览器改变文本大小

/\* <length>，长度值 \*/

font-size: 12px;

font-size: 0.8em;

/\* <percentage>，百分比值 \*/

font-size: 80%;

/\*继承\*/

font-size: inherit;

**注意：如果您没有规定字体大小，普通文本（比如段落）的默认大小是 16 像素 (16px=1em)。**

W3C 推荐使用 em 尺寸单位。

1em 等于当前的字体尺寸。如果一个元素的 font-size 为 16 像素，那么对于该元素，1em 就等于 16 像素。在设置字体大小时，em 的值会相对于父元素的字体大小改变。

浏览器中默认的文本大小是 16 像素。因此 1em 的默认尺寸是 16 像素。

可以使用下面这个公式将像素转换为 em：*pixels*/16=*em*

（注：16 等于父元素的默认字体大小，假设父元素的 font-size 为 20px，那么公式需改为：*pixels*/20=*em*）

以 em 为单位的文本大小与以像素计的文本是相同的。不过，如果使用 em 单位，则可以在所有浏览器中调整文本大小。

不幸的是，在 IE 中仍存在问题。在重设文本大小时，会比正常的尺寸更大或更小。

结合使用百分比和 EM

在所有浏览器中均有效的方案是为 body 元素（父元素）以百分比设置默认的 font-size 值：

**实例**

body {font-size:100%;}

h1 {font-size:3.75em;}

h2 {font-size:2.5em;}

p {font-size:0.875em;}

* 1. **字体样式，字体粗细，文本转换和文本装饰**

font-style: 用来打开和关闭文本 italic (斜体)。 可能的值如下 (你很少会用到这个属性，除非你因为一些理由想将斜体文字关闭斜体状态)：

属性值：

normal: 将文本设置为普通字体 (将存在的斜体关闭)

italic: 如果当前字体的斜体版本可用，那么文本设置为斜体版本；如果不可用，那么会利用 oblique 状态来模拟 italics。

oblique: 将文本设置为斜体字体的模拟版本，也就是将普通文本倾斜的样式应用到文本中。

font-weight: 设置文字的粗体大小。这里有很多值可选，防止你有好几个可用的字体。 (比如 -light, -normal, -bold, -extrabold, -black, 等等), 不过事实上你很少会用到 normal 和 bold以外的值：

属性值：

normal, bold: 普通或者加粗的字体粗细

lighter, bolder: 将当前元素的粗体设置为比其父元素粗体更细或更粗一步。100–900: 数值粗体值，如果需要，可提供比上述关键字更精细的粒度控制。

text-transform: 允许你设置要转换的字体。值包括：

none: 防止任何转型。

uppercase: 将所有文本转为大写。

lowercase: 将所有文本转为小写。

capitalize: 转换所有单词让其首字母大写。

full-width: 将所有字形转换成固定宽度的正方形，类似于等宽字体，允许对齐。拉丁字符以及亚洲语言字形（如中文，日文，韩文）

text-decoration: 设置/取消字体上的文本装饰 (你将主要使用此方法在设置链接时取消设置链接上的默认下划线。) 可用值为：

none: 取消已经存在的任何文本装饰。

underline: 文本下划线.

overline: 文本上划线

line-through: 穿过文本的线 strikethrough over the text.

你应该注意到 text-decoration 可以一次接受多个值，如果你想要同时添加多个装饰值， 比如 text-decoration: underline overline.。属性值不继承！！

**同时注意**[**text-decoration**](https://developer.mozilla.org/zh-CN/docs/Web/CSS/text-decoration)**是一个缩写形式，它由**[**text-decoration-line**](https://developer.mozilla.org/zh-CN/docs/Web/CSS/text-decoration-line)**,**[**text-decoration-style**](https://developer.mozilla.org/zh-CN/docs/Web/CSS/text-decoration-style)**和**[**text-decoration-color**](https://developer.mozilla.org/zh-CN/docs/Web/CSS/text-decoration-color)**构成。你可以使用这些属性值的组合来创建有趣的效果，比如 text-decoration: line-through red wavy.**

**注意：** 修饰的颜色由 "color" 属性设置。

text-decoration-color: #2AC845;

* 1. **文字阴影**

[**text-shadow**](https://developer.mozilla.org/zh-CN/docs/Web/CSS/text-shadow)**属性：**

**text-shadow: 4px 4px 5px red;**

**语法**

text-shadow: *h-shadow* *v-shadow* *blur*  *color*;

注释：text-shadow 属性向文本添加一个或多个阴影。该属性是逗号分隔的阴影列表，每个阴影有两个或三个长度值和一个可选的颜色值进行规定。省略的长度是 0。

|  |  |
| --- | --- |
| **值** | **描述** |
| ***h-shadow*** | **必需。水平阴影的位置。允许负值。** |
| ***v-shadow*** | **必需。垂直阴影的位置。允许负值。** |
| ***blur*** | **可选。模糊的距离。** |
| ***color*** | **可选。阴影的颜色。参阅**[**CSS 颜色值**](http://www.w3school.com.cn/cssref/css_colors_legal.asp)**。** |

**4 个属性如下:**

1. 阴影与原始文本的水平偏移，可以使用大多数的 CSS 单位 [length and size units](https://developer.mozilla.org/zh-CN/Learn/CSS/Introduction_to_CSS/Values_and_units#Length_and_size), 但是 px 是比较合适的。这个值必须指定。
2. 阴影与原始文本的垂直偏移;效果基本上就像水平偏移，除了它向上/向下移动阴影，而不是左/右。这个值必须指定。
3. 模糊半径 - 更高的值意味着阴影分散得更广泛。如果不包含此值，则默认为0，这意味着没有模糊。可以使用大多数的 CSS 单位 [length and size units](https://developer.mozilla.org/zh-CN/Learn/CSS/Introduction_to_CSS/Values_and_units#Length_and_size).
4. 阴影的基础颜色，可以使用大多数的 CSS 颜色单位 [CSS color unit](https://developer.mozilla.org/zh-CN/Learn/CSS/Introduction_to_CSS/Values_and_units#Colors). 如果没有指定，默认为 black.

注意: 正偏移值可以向右移动阴影，但也可以使用负偏移值来左右移动阴影，例如 -1px -1px.

**多种阴影**

您可以通过包含以逗号分隔的多个阴影值，将多个阴影应用于同一文本，例如：

text-shadow: -1px -1px 1px #aaa,

0px 4px 1px rgba(0,0,0,0.5),

4px 4px 5px rgba(0,0,0,0.7),

0px 0px 7px rgba(0,0,0,0.4);

* 1. **其他**

Font 样式:

font-variant: 在小型大写字母和普通文本选项之间切换。

font-kerning: 开启或关闭字体间距选项。

font-feature-settings: 开启或关闭不同的 OpenType 字体特性。

font-variant-alternates: 控制给定的自定义字体的替代字形的使用。

font-variant-caps: 控制大写字母替代字形的使用。

font-variant-east-asian: 控制东亚文字替代字形的使用, 像日语和汉语。

font-variant-ligatures: 控制文本中使用的连写和上下文形式。

font-variant-numeric: 控制数字，分式和序标的替代字形的使用。

font-variant-position: 控制位于上标或下标处，字号更小的替代字形的使用。

font-size-adjust: 独立于字体的实际大小尺寸，调整其可视大小尺寸。

font-stretch: 在给定字体的可选拉伸版本中切换。

text-underline-position: 指定下划线的排版位置，通过使用 text-decoration-line 属性的underline 值。

text-rendering: 尝试执行一些文本渲染优化。

* 1. **Font 简写**

许多字体的属性也可以通过 font 的简写方式来设置 . 这些是按照以下顺序来写的： font-style, font-variant, font-weight, font-stretch, font-size, line-height, and font-family.

如果你想要使用 font 的简写形式，在所有这些属性中，只有 font-size 和 font-family 是一定要指定的。

font-size 和 line-height 属性之间必须放一个正斜杠。

一个完整的例子如下所示：

font: italic normal bold normal 3em/1.5 Helvetica, Arial, sans-serif;

1. 文本布局
   1. **文本对齐**

[**text-align**](https://developer.mozilla.org/zh-CN/docs/Web/CSS/text-align)**属性：**定义行内内容（例如文字）如何相对它的块父元素对齐。text-align 并不控制块元素自己的对齐，只控制它的行内内容的对齐。

**值 描述**

left 把文本排列到左边。默认值：由浏览器决定。

right 把文本排列到右边。

center 把文本排列到中间。

justify 实现两端对齐文本效果。

inherit 规定应该从父元素继承 text-align 属性的值。

* 1. **行高**

[**line-height**](https://developer.mozilla.org/zh-CN/docs/Web/CSS/line-height)**属性：**line-height 属性设置行间的距离（行高）。注释：不允许使用负值。

**值 描述**

normal 默认。设置合理的行间距。

number 设置数字，此数字会与当前的字体尺寸相乘来设置行间距。

length 设置固定的行间距。

% 基于当前字体尺寸的百分比行间距。

inherit 规定应该从父元素继承 line-height 属性的值。

* 1. **字母和行间距**

[letter-spacing](https://developer.mozilla.org/zh-CN/docs/Web/CSS/letter-spacing) 和 [word-spacing](https://developer.mozilla.org/zh-CN/docs/Web/CSS/word-spacing) 属性允许你设置你的文本中的字母与字母之间的间距、或是字与字之间的间距。你不会经常使用它们，但是可能可以通过它们，来获得一个特定的外观，或者让较为密集的文字更加可读。它们可以接受大多数单位 [length and size units](https://developer.mozilla.org/zh-CN/Learn/CSS/Introduction_to_CSS/Values_and_units#Length_and_size).

* 1. **其他**

文本布局样式：

text-indent: 指定文本内容的第一行前面应该留出多少的水平空间，有继承性。

text-overflow: 定义如何向用户表示存在被隐藏的溢出内容。

white-space: 定义如何处理元素内部的空白和换行。

word-break: Specify whether to break lines within words.

direction: 定义文本的方向 (这取决于语言，并且通常最好让HTML来处理这部分，因为它是和文本内容相关联的。)

hyphens: 为支持的语言开启或关闭连字符。

line-break: Relax or strengthen line breaking for Asian languages.

text-align-last: 定义一个块或行的最后一行，恰好位于一个强制换行前时，如何对齐。

text-orientation: 定义行内文本的方向。

word-wrap: 指定浏览器是否可以在单词内换行以避免超出范围。

writing-mode: 定义文本行布局为水平还是垂直，以及后继文本流的方向。