# CSS和文档

**元素**：是文档结构的根基。文档中的每个元素都对文档的表现起一定作用。可分为置换元素和非置换元素。

**元素的显示方式**：最常见的有块级和行内两种基本类型。

**把CSS应用到HTML上**：

①style元素：<style type=”text/css”>……</style>

②link标签：<link rel=”stylesheet” type=”text/css” href=”sheet1.css” media=”all”>（必须放在head元素中）

③@import指令：@import url(sheet2.css);（该指令放在style元素内部，且必须放在其他CSS规则前面）

④HTTP链接。

⑤行内样式。

**特性查询**：根据用户代理是否支持特定的CSS属性及其值来应用一段样式；渐进增强样式的完美方式；如果浏览器不支持特性查询则会跳过整段样式。

例：@supports(color: black) {

body {color: black;}

h1 {color: purple;}

}

# 选择符

### 2.5 根据文档结构选择

父子关系和祖先后代关系。

后代选择符：h1 em {color: gray;}（在这里即h1的后代元素em）。

选择子元素：h1 > strong {color: gray;}（直接子元素才会被渲染，在这里即h1的子元素strong）。

选择紧邻同胞元素：h1 + p {color: gray;}（同父且相邻的兄弟元素才会被渲染，在这里即h1和p是同一个父元素，且h1和p相邻）。

# 特指度和层叠

**特指度**：

选择符中的每个ID属性值加0,1,0,0；

选择符中的每个类属性值、属性选择或伪类加0,0,1,0；

选择符中的每个元素和伪元素加0,0,0,1；

行内声明的特指度为1,0,0,0；

连结符和通用选择符不增加特指度。

**层叠**：

①找到匹配特定元素的所有规则；

②按显式权重排序应用到特定元素上的所有声明。以!important标记的规则比没有这一标记的权重高；

③按来源排序应用到特定元素行的所有声明（创作人员、读者和用户代理）；

④按特指度排序应用到特定元素上的所有声明；

⑤按声明的前后位置应用到特定元素上的所有声明；

*在HTML中，向上传播有个例外：应用到body元素上的背景样式会传给html元素。*

*很多的属性也是不继承的，例如border属性、多数盒模型属性。其次，继承的值没有特指度，连零都没有。*

# 值和单位

**计算值**calc()：p {width: calc(90% - 2em);}（+、-运算符两侧必须有空白）

**自定义值**：

自定义标识符要以两个连字符开头（--），且是区分大小写的；调用方法是使用var()值类型；根据作用域动态替换CSS变量值。

例：html {--base-color: #639; --highlight-color: #AEA;}

h1 {color: var(--base-color);}

H2 {color: var(--highlight-color);}

# 字体

## 5.1 字体族

font-family

为了覆盖所有情况，CSS定义了五种通用字体族：衬线字体（Times）、无衬线字体（Helvetica）、等宽字体（Courier）、草书字体（Author）、奇幻字体（Western）。

衬线字体：字形宽度各异，有衬线，笔画的粗细会有所不同。

无衬线字体：字形宽度各异，无衬线，笔画粗细一致。

## 5.2 使用@font-face

必须的描述符：font-family和src。

其他字体描述符：

font-style：区分常规、斜体和倾斜字型。

font-weight：区分不同的字重（例如加粗）。

font-stretch：区分不同的字符宽度。

font-variant：区分众多字形变体（例如小号大写字母），和font-feature-settings很像。

font-feature-settings：直接访问OpenType的底层特性。

unicode-range：定义指定字体中可用的字符范围。

## 5.3 字重

font-weight。

## 5.4 字号

font-size的作用是为字体的em方框提供一个尺寸，所以显示出来的字符不可能完全是指定的大小。

**绝对大小**：有7个关键字：xx-small、x-small、small、medium、large、x-large、xx-large。这几个关键字没有固定大小，而是相对而言的。

**相对大小**：与字重的相对值不同，字号的相对值没有上下限。因此，字号可以小于xx-small或大于xx-large。

字号的继承问题：如果父元素设置为medium，有些浏览器通过继承向下传递的是关键词medium；而有些浏览器会覆盖medium默认的对应关系，根据父元素的字号计算实际值。因此文本在不同浏览器中显示的大小不一致。

解决方法：在font-family中写多一个‘serif’，这种方法适用于所有浏览器。

自动调整字号font-size-adjust。

## 5.5 字形

font-style：normal、italic、oblique。

## 5.6 字体拉伸

font-stretch。

## 5.7 字距调整

font-kerning。

## 5.8 字体变形

font-variant：normal、small-caps（CSS1、CSS2）。

## 5.9 字体特性

font-feature-settings。

## 5.10 字体合成

如果字体族中没有粗体或斜体，用户代理会自行计算或直接倾斜常规字型，从而生成粗体或斜体，不过这种实现的效果可能无法让人满意。

使用 font-synthesis: none 可以阻止用户代理在对应的元素上做字体合成。

## 5.11 font属性

通过群组选择符可以把各属性合并到一起，但是要注意的是，font属性必须的两个值是font-size和font-family，而且必须以这样的顺序编写，其他的值都是可选的。

也可以加入line-height行高属性，且font-size始终在line-height的前面，两值之间用斜线（/）分隔。

使用font属性时，省略的值都将重设为默认值。

# 文本属性

## 6.1 缩进和行内对齐

缩进文本：text-indent。

文本对齐：text-align。

## 6.2 块级对齐

行的高度：line-height。

行高的继承：根据父元素的字号计算。

纵向对齐文本：vertical-align。（只能用于行内元素和置换元素）

基线对齐：baseline，强制元素的基线与父元素的基线对齐。

上标和下标：sub把元素放在下标处，即元素的基线（对置换元素来说是底边线）低于父元素的基线；super则相反。

底端对齐：bottom（元素所在的行内框的底边与行框的底边对齐）；text-bottom（相对行中文本的底边对齐）。

顶端对齐：top；text-top。

中线对齐：middle（行内元素所在方框的中线与父元素基线向上偏移0.5ex处的线对齐，1ex等于父元素的font-size，不同浏览器向上偏移的量可能不同）。

## 6.3 单词间距和字符间距

单词间距：word-spacing。

字符间距：letter-spacing。

## 6.4 文本转换

text-transform。

uppercase：转大写字母。

lowercase：转小写字母。

capitalize：每个单词以大写字母开头。

none：默认。

## 6.5 文本装饰

text-decoration：none、underline、overline、line-through、blink。

如果后代元素没有自己的装饰，祖先元素上设置的装饰会“延伸”到后代元素中。不要求用户代理支持 blink。

## 6.6 文本渲染效果

text-rendering。

auto：当绘制文本时，浏览器会进行智能识别，何时应该从速度、清晰度和几何精度方 面进行优化。

optimizeSpeed：当绘制文本时，浏览器会着重渲染速度，而不是清晰度和几何精度。该属性值不 能用于字距调整和连字。

optimizeLegibility：当绘制文本时，浏览器会侧重文本的可读性（清晰度），而不是渲染速度和几何 精度。该属性值可以用于字距调整和连字。

geometricPrecision：当绘制文本时，浏览器会着重几何精度，而不是清晰度和渲染速度。字体的某些 方面，比如字间距并不是按照线性比例进行渲染的，因此该属性可以使得设置为 这些字体的文本看起来很整洁。

## 6.7 文本阴影

text-shadom。

每个阴影由一个可选的颜色和三个长度值定义，最后一个长度值是可选的。

第一个长度值设定横向偏移，第二个长度值设定纵向偏移，第三个长度值设定阴影的模糊半径。

*注意：大量阴影或模糊半径较大的文本阴影会损耗性能，尤其是在低功耗和CPU能力有限的情况下。*

## 6.8 处理空白

white-space：normal、nowrap、pre、pre-wrap、pre-line。

|  |  |  |  |
| --- | --- | --- | --- |
| **值** | **空白** | **换行符** | **自动换行** |
| **normal** | 折叠 | 保留 | 允许 |
| **nowrap** | 折叠 | 忽略 | 允许 |
| **pre** | 折叠 | 忽略 | 禁止 |
| **pre-wrap** | 保留 | 保留 | 禁止 |
| **pre-line** | 保留 | 保留 | 允许 |

设定制表符的宽度：tab-size（默认1个制表符为8个连续的空格）。

## 6.9 换行和断字

hyphens。

manual：默认值，只在文档中手动插入的连字符处断字，否则不断字。

none：即使有手动插入的连字符，也不断字。

auto：即使没有手动插入连字符，浏览器也会在单词中“合适的”位置断字。

word-break：用于控制不同语言处理文本转换行的方式，可以影响CJK的软换行方式。

normal：默认值，按正常方式换行，即在单词之间换行。

break-all：换行可能会出现在任何字符之间，即使是在一个词的内部；只用这个值是，连字符不显示，即使出现在断字的位置。

keep-all：禁止在字符之间软换行。

line-break：auto、loose、normal、strict。

文本换行overflow-wrap。

normal：默认值，按正常方式换行，即在单词之间换行，或者由所用的语言决定。

break-word：可以在单词内部换行。

## 6.10 书写模式

writing-mode。

horizontal-tb：行内方向为横向，块级方向为从上到下。

vertical-rl、vertical-lr。

改变文本方向text-orientation：mixed、upright、sideways。

# 视觉格式化基础

## 7.1 元素框基础

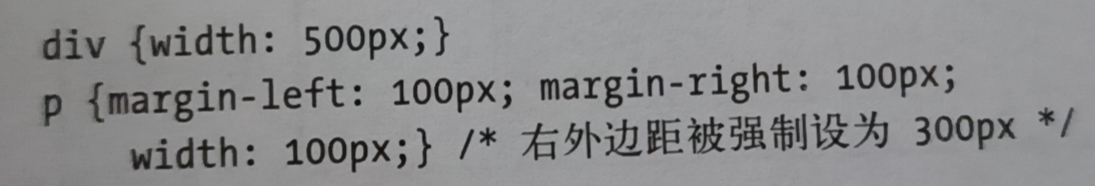
元素框：内边距、边框和外边距都是可选的，默认值是零。

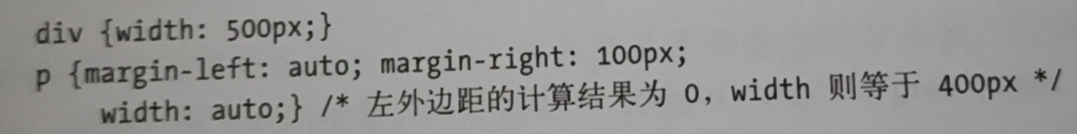
容纳块：父元素的块级框。

## 7.2 调整元素的显示方式

块级框box-sizing：content-box、padding-box、border-box（当设置为border-box时，width和height的值包括border-width和padding）。

使用auto：从某种意义上说，auto用于补全总和所缺的尺寸。在“过约束”情况下横向格式化属性会被强制设置为auto。但是有个例外：当置换元素的width为auto时，width等于内容自身的宽度。





七个横向属性之和始终等于父元素的width。

七个纵向属性中只有3个可以设为auto：元素的高度和上下外边距。上下内边距和边框必须设为具体的值，否则取默认值零（假设未声明边框式样）。

height属性要么设为auto（此时框体的高度恰好能撑起里面的内容），要么设为某种类型的非负值，决不能小于零。

相邻的纵向外边距会折叠。内边距和边框绝不与任何区域折叠。

list-style-position：设置在何处放置列表项标记。（inside、outside）

## 7.3 行内元素

改变断行行为box-decoration-break：slice、clone。

vertical-align属性的百分数值是相对元素的line-height计算的。如果元素的line-height没有值，就不能把纵向对齐的值（vertical-align）设为百分数了。

行内块级元素的底边默认与文本行的基线对齐，而且内部不会断行。

流动显示方式：

display属性的新旧值对照：

|  |  |
| --- | --- |
| 旧值 | 新值 |
| block | block flow |
| inline | inline flow |
| inline-block | inline flow-root |
| list-item | list-item block flow |
| inline-list-item | list-item inline flow |
| table | block table |
| inline-table | inline table |
| flex | block flex |
| inline-flex | inline flex |
| grid | block grid |
| inline-grid | inline grid |

contents显示方式：把display:contents应用到元素上之后，那个元素不再参与页面的格式化，相当于把它的子元素提升到当前的层级。

# 内边距、边框、轮廓和外边距

## 8.1 基本元素框

## 8.2 内边距

## 8.3 边框

## 8.4 轮廓

## 8.5 外边距

# 颜色、背景和渐变

## 9.1 颜色

前景色color。

某些Web浏览器通常根据操作系统的用户界面构建表单中的小组件，因此见到的复选框和勾号其实不是HTML文档中的内容，而是插入文档的用户界面小组件。而且表单输入框和图像一样，都是置换元素。理论上CSS无法装饰置换元素的内容。

## 9.2 背景

**背景色**background-color。该属性的值不继承，默认值为transparent。

**裁剪背景**background-clip：border-box、padding-box、content-box、text。

*背景相关的属性都不继承。*

**背景定位**：background-position。

**关键字**：top、bottom、left、right和center。

位置关键字的顺序随意，只要不超过两个；一个指定横向位置；一个指定纵向位置。如果只有一个关键字，另一个假定位center。

**百分数值**：与关键字十分接近，不过行为更加复杂。源图像的相对位置点与背景区域的相对位置点重合。如果只提供一个百分数值，那个值将作为横向偏移，而纵向偏移假定位50%。

**长度值**：是相对元素背景左上角的偏移。源图像上的偏移点是左上角。

**改变定位框**background-origin：border-box、padding-box、content-box。

**背景重复方式（或不重复）**：background-repeat。

|  |  |
| --- | --- |
| 单个关键字 | 等效的关键字 |
| Repeat-x | Repeat no-repeat |
| Repeat-y | no-repeat repeat |
| Repeat | Repeat repeat |
| No-repeat | no-repeat no-repeat |
| Space | Space Space |
| round | Round round |

space：确定沿某一轴能完全重复多少次，然后从背景区域的一边到对边均匀排列图像。得到的效果并不一定是正方形网格，即横向和纵向的间隔并不一定相同。均匀排列得到的是一种类似行和列的效果，横向和纵向的间隔可以不等。如果在某个方向放不下时，位置由background-position的值确定。如果在某个方向上可以重复多次时，那个方向的background-position值将被忽略。

round：如果从背景区域的一边到对边之间无法重复整数次，则将图像放大或缩小，使恰好重复整数次。图像沿各轴的缩放程度可以不同（使用该值时，如果不想让平铺的图像被裁剪，必须从四个角中的某一个开始平铺）。

**背景贴附**background-attachment：scroll、fixed、local。

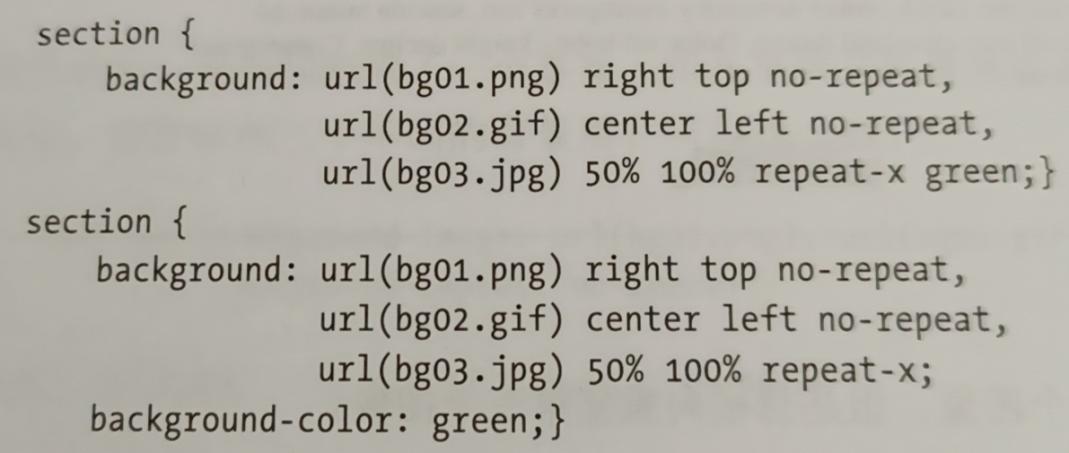
**控制背景图的尺寸**background-size。

**简写属性声明**background。

书写顺序没有严格要求，不过有三个限制规则：①background-size值必须紧随background-position值后面，而且二者之间要以一条斜线（/）隔开；②值的常用规则依然使用，即横向值在前，纵向值在后；③如果同时为background-origin和background-clip提供值，前一个分配给background-origin，后一个分配给background-clip。

简写时，省略的值自动使用对应属性的默认值。

**多个背景**：几乎所有背景属性都能接受以逗号分隔的多个值，如



这里唯一的限制是不能为background-color指定多个值，因为背景色只能有一个，因此背景色只能出现在最后一个背景层中。

## 9.3 渐变

线性渐变background-image:linear-gradient( [ [ <angle> | to <side-or-quadrant> ], ] ? [ <color-stop> [, <color-hint> ] ? ] #, <color-stop> )

径向渐变radial-gradient( [ <shape> | <size> ] [ at <position> ] ? , | at <position> , ] ? [ <color-stop> [, <color-hint> ] ? ] [, <color-stop> )+

## 9.4 盒子投影

box-shadow。

# 浮动及其形状

# 定位

# 弹性盒布局

# 栅格布局

# CSS中的表格布局

# 列表和生成的内容

# 变形

# 过渡

# 动画

# 滤镜、混合、裁剪和遮罩

# 针对特定媒体的样式