**CSS 规则**

CSS规则实际上就是一条完整的CSS 指令。规则声明了要修改的元素和要应用给该元素的样式。

有三种方法可以把CSS 添加到网页中，分别是写在元素标签里（也叫行内样式）写在<style>标签里（也叫嵌入样式）和写在单独的CSS 样式表中（也叫链接样式）。

**行内样式**

行内样式是写在特定HTML 标签的style 属性里的。行内样式的作用范围非常有限。行内样式只能影响它所在的标签，而且总会覆盖嵌入样式和链接样式。

**嵌入样式**

嵌入的CSS 样式是放在HTML 文档的head 元素中的。嵌入样式的应用范围仅限于当前页面。页面样式会覆盖外部样式表中的样式，但会被行内样式覆盖。

**链接样式**

在创建包含多个页面的网站时，需要把样式集中在一个单独的文件里，这个文件就叫样式表。样式表其实就是一个扩展名为.css 的文本文件。可以在任意多个HTML 页面中链接同一个样式表文件。链接样式的作用范围可以是整个网站。只要使用<link>标签把样式表链接到每个页面，相应的页面就可以使用其中的样式。

**@import 指令**（是一种at 规则）

@import 指令必须出现在样式表中其他样式之前，否则@import 引用的样式表不会被加载。

CSS 样式是通过<style>标签嵌入到页面里的。当浏览器遇到开标签<style>时，就会由解释HTML 代码切换为解释CSS 代码。等遇到闭标签</style>时，它会再切换回解释HTML 代码。

**CSS规则命名惯例**

CSS 规则由选择符和声明两部分组成，其中选择符用于指出规则所要选择的元素，声明则又由两部分组成：属性和值。属性指出要影响元素哪方面的样式，值就是属性的一个新状态。

基本语法结构：

第一种方法：多个声明包含在一条规则里。

p {color:red; font-size:12px; font-weight:bold;}

第二种方法：多个选择符组合在一起。如果想让<h1>、<h2>和<h3>的文本都变成蓝色，

粗体，可以这样分别写：

h1 {color:blue; font-weight:bold;}

h2 {color:blue; font-weight:bold;}

h3 {color:blue; font-weight:bold;}

把三个选择符组合在一起也可以，这样就能减少重复输入：

h1, h2, h3 {color:blue; font-weight:bold;}

第三种方法：多条规则应用给一个选择符。 假设，你在写完前面那条规则后，又想只把h3 变成斜体，那可以再为h3 写一条规则：

h1, h2, h3 {color:blue; font-weight:bold;}

h3 {font-style:italic;}

同一元素不同css属性样式取合集，同一元素同一属性，后续写的属性，或者是其他位置权重高的，会覆盖当前的css规则的。

**CSS选择符**

1. 上下文选择符。基于祖先或同胞元素选择一个元素。
2. ID 和类选择符。基于id 和class 属性的值（你自己设定）选择元素。
3. 属性选择符。基于属性的有无和特征选择元素。

**上下文选择符：**

CSS规范里称为**后代组合式选择符**（descendantcombinator selector）

* + 后代选取器(以空格分隔)，匹配所有指定元素的后代元素。div p｛｝，选取div下所有的p元素。
  + 子元素选择器(以大于号分隔），只能选择作为某元素子元素的元素。div>p{ }，选取div的子元素p，其他的子孙p不会被选择。
  + 相邻兄弟选择器（以加号分隔），选择紧接在另一元素后的元素，且二者有相同父元素。div+p{ }，选取了所有位于 div 元素后的第一个 p 元素。
  + 普通兄弟选择器（以破折号分隔），选取所有指定元素的相邻兄弟元素。div~p{ }，选取了所有 div 元素之后的所有相邻兄弟元素 。
  + 通用选择符（以星号表示），匹配任何元素。

**ID和类选择符：**

使用标签的id（唯一）和类名（可重复）来选取对应的元素。

**属性选择符：**

使用标签的属性（attribute）来选取对应的元素。有可分为用属性名和属性值两种方式，img[title](**带有title属性的image标签**)和img[title=“red”]（**带有title属性且属性值是red的image标签**）

**伪类和伪元素：**

伪类用于向某些选择器添加特殊的效果。

伪元素用于将特殊的效果添加到某些选择器中。

伪类有 :first-child,:link,visited :hover :active :foucs :lang

伪元素: ::first-line ::first-letter ::before ::after

|  |  |  |
| --- | --- | --- |
| **选择器** | **示例** | **示例说明** |
| [:link](https://www.w3cschool.cn/cssref/sel-link.html) | a:link | 选择所有未访问链接 |
| [:visited](https://www.w3cschool.cn/cssref/sel-visited.html) | a:visited | 选择所有访问过的链接 |
| [:active](https://www.w3cschool.cn/cssref/sel-active.html) | a:active | 选择正在活动链接 |
| [:hover](https://www.w3cschool.cn/cssref/sel-hover.html) | a:hover | 把鼠标放在链接上的状态 |
| [:focus](https://www.w3cschool.cn/cssref/sel-focus.html) | input:focus | 选择元素输入后具有焦点 |
| [:first-letter](https://www.w3cschool.cn/cssref/sel-firstletter.html) | p:first-letter | 选择每个<p> 元素的第一个字母 |
| [:first-line](https://www.w3cschool.cn/cssref/sel-firstline.html) | p:first-line | 选择每个<p> 元素的第一行 |
| [:first-child](https://www.w3cschool.cn/cssref/sel-firstchild.html) | p:first-child | 选择器匹配属于任意元素的第一个子元素的 <]p> 元素 |
| [:before](https://www.w3cschool.cn/cssref/sel-before.html) | p:before | 在每个<p>元素之前插入内容 |
| [:after](https://www.w3cschool.cn/cssref/sel-after.html) | p:after | 在每个<p>元素之后插入内容 |
| [:lang(*language*)](https://www.w3cschool.cn/cssref/sel-lang.html) | p:lang(it) | 为<p>元素的lang属性选择一个开始值 |

**CSS的工作原理—机制**

CSS 提供了三种机制：继承、层叠和特指。

**继承：**

CSS 中的祖先元素也会向后代传递一样东西：CSS 属性的值。CSS中有些属性能继承，有些属性不能继承。这些不能继承的属性主要涉及元素盒子的定位和显示方式，比如边框、外边距、内边距等。

**层叠：**

层叠，就是层叠样式表中的层叠，是一种样式在文档层次中逐层叠加的过程，目的是让浏览器面对某个标签特定属性值的多个来源，确定最终使用哪个值。层叠是CSS 的核心机制，最直观的作用是让文档外观在达到设计要求的同时，也给用户留下一些空间，让他们能根据需要更改文档的显示效果。

1、样式来源，

1.  浏览器默认样式表
2.  用户样式表
3.  作者链接样式表（按照它们链接到页面的先后顺序）
4.  作者嵌入样式
5.  作者行内样式

2、层叠规则，

层叠规则一：找到应用给每个元素和属性的所有声明

层叠规则二：按照顺序和权重排序。

层叠规则三：按特指度排序。

查理版简单层叠要点，这三条规则适合所有情况。

规则一：包含ID 的选择符胜过包含类的选择符，包含类的选择符胜过包含标签名的选择符。

规则二：如果几个不同来源都为同一个标签的同一个属性定义了样式，行内样式胜过嵌入样式，嵌入样式胜过链接样式。在链接的样式表中，具有相同特指度的样式，后声明的胜过先声明的。

规则一胜过规则二。换句话说，如果选择符更明确（特指度更高），无论它在哪里，都会胜出。

规则三：设定的样式胜过继承的样式，此时不用考虑特指度（即显式设定优先）

**清除浮动**

**清除浮动**

* 为父元素应用overflow:hidden
* 浮动父元素
* 在父元素内容的末尾添加非浮动元素，可以直接在标记中加，也可以通过给父元素添加clearfix 类来加（当然，样式表中得需要相应的clearfix 规则）

.clearfix:after {

content:".";

display:block;

height:0;

visibility:hidden;

clear:both;

}

**定位**

CSS 布局的核心是position 属性，对元素盒子应用这个属性，可以相对于它在常规文档流中的位置重新定位。position 属性有4 个值：static、relative、absolute、fixed，默认值为static。

**显示属性**

display，控制内容的显示以及显示方式。

* 块级元素，比如段落、标题、列表等，在浏览器中上下堆叠显示。
* 行内元素，比如a、span 和img，在浏览器中左右并排显示，只有前一行没有空间时才会显示到下一行

**背景 background**

CSS 规定以下与背景相关属性。

 background-color

 background-image

 background-repeat

 background-position

 background-size

 background-attachment

 background（简写属性）

 background-clip、background-origin、background-break（目前尚未得到广泛支持）

设定背景位置时可以使用三种值：关键字、百分比、绝对或相对单位的数值。可以使用两个值分别设定水平和垂直位置。有意思的是，还可以使用负值。这样就可以把图片的左上角定位到元素外部，从而在元素中只能看到部分图片。当然，给图片设定足够大的正值，也可以把图片的右下角推到元素外部，从而在元素中也只能看到部分图片。位于元素外部的那部分图片不会显示。

在CSS 中，我把每张图片的声明都单独放在了一行里，以逗号分隔，以便看清它们的位置、重复的设定值。为了防止图片加载失败时元素背景处于默认的透明状态，这里也在最后一条声明中加上了背景颜色（加粗的值）。要注意的是，代码中先列出的图片显示在上方，或者说，更接近前景。

多图片背景示例：

background:

url(images/turq\_spiral.png) 30px -10px no-repeat,

url(images/pink\_spiral.png) 145px 0px no-repeat,

url(images/gray\_spiral.png) 140px -30px no-repeat, #ffbd75;

**背景渐变：**

渐变是CSS 帮我们生成的背景图片。添加渐变可以使用background-image 属性，也可以使用简写background 属性。

**语法：**

**<linear-gradient>** = linear-gradient([ [ [<angle>](http://www.css88.com/book/css/values/angle/angle.htm) | to [<side-or-corner>](http://www.css88.com/book/css/values/image/linear-gradient%28%29.htm#side-or-corner) ] ,]? [<color-stop>](http://www.css88.com/book/css/values/image/linear-gradient%28%29.htm#color-stop)[, [<color-stop>](http://www.css88.com/book/css/values/image/linear-gradient%28%29.htm#color-stop)]+)

**<side-or-corner>** = [left | right] || [top | bottom]

**<color-stop>** = [<color>](http://www.css88.com/book/css/values/color/color.htm) [ [<length>](http://www.css88.com/book/css/values/length/length.htm) | [<percentage>](http://www.css88.com/book/css/values/numeric/percentage.htm) ]?

取值：

下述值用来表示渐变的方向，可以使用角度或者关键字来设置：

[<angle>](http://www.css88.com/book/css/values/angle/index.htm)：

用角度值指定渐变的方向（或角度）。

to left：

设置渐变为从右到左。相当于: 270deg

to right：

设置渐变从左到右。相当于: 90deg

to top：

设置渐变从下到上。相当于: 0deg

to bottom：

设置渐变从上到下。相当于: 180deg。这是默认值，等同于留空不写。

<color-stop> 用于指定渐变的起止颜色：

[<color>](http://www.css88.com/book/css/values/color/color.htm)：

指定颜色。

[<length>](http://www.css88.com/book/css/values/length/length.htm)：

用长度值指定起止色位置。不允许负值

[<percentage>](http://www.css88.com/book/css/values/numeric/percentage.htm)：

用百分比指定起止色位置。

background:linear-gradient(#e86a43, #fff); 默认从上到下

background:linear-gradient(left, #64d1dd, #fff); 从左到右

background:linear-gradient(-45deg, #e86a43, #fff); 左上到右下

**渐变点：**

渐变点就是渐变方向上的点，可以在这些点上设定颜色和不透明度。通过设定下一个渐变点的颜色值，就可以控制渐变的效果。可以添加任意多个渐变点。渐变点的位置一般使用整个渐变宽度的百分比来表示。图3-41 展示了使用渐变点后的四种渐变效果。

/\*例1：50%处有一个渐变点\*/

.gradient1 {

background:linear-gradient(#64d1dd, #fff 50%, #64d1dd);

}

/\*例2：20%和80%处有两个渐变点\*/

.gradient2 {

background:linear-gradient(#e86a43 20%, #fff 50%, #e86a43 80%);

}

/\*例3：25%、50%、75%处有三个渐变点\*/

.gradient3 {

background:linear-gradient(#64d1dd, #fff 25%, #64d1dd 50%, #fff 75%,#64d1dd);

}

/\*例4：为同一个渐变点设定两种颜色可以得到突变效果\*/

.gradient4 {

background:linear-gradient(#e86a43, #fff 25%, #64d1dd 25%,#64d1dd 75%,#fff 75%, #e86a43);

}

**放射性渐变**

放射性渐变比线性渐变复杂那么一点点，因为可用的控制点多一些。如果你写过程序，从属性值中的括号就可以看出，渐变属性其实是函数。什么是函数？函数可以接收参数，然后根据这些参数来生成渐变。在创建放射性渐变时，可以使用参数指定形状、位置、尺寸、颜色和不透明度。

下面的每一个例子都设定3 种颜色。

.gradient1 {

background: -webkit-radial-gradient(#fff, #64d1dd, #70aa25);

}

.gradient2 {

background: -webkit-radial-gradient(circle, #fff, #64d1dd, #e86a43);

}

.gradient3 {

background: -webkit-radial-gradient(50px 30px, circle, #fff, #64d1dd,

#4947ba);

}

这里虽然只声明了-webkit-前缀，但带有其他厂商前缀的属性也是必要的。

在图中，例1 展示了默认的渐变形状，即渐变效果会填充元素，这里的元素是

矩形。如果元素是正方形，那渐变就是圆形。

图3-42 三个三色放射性渐变。第一个是默认的填满图形渐变，第二个是圆形渐变，第三个是指定位置的圆形渐变

例2 设定了形状关键字circle，于是渐变的形状变得均匀，并在元素最近的边达到了终点，形成了圆形渐变。而长边剩下的区域则填充了终点的颜色。例3 中的位置参数50px 30px 把渐变的圆心放到了靠近左上角的位置。

## 字体和文本

网页中的字体有三个来源。

1、用户机器中安装的字体。（直到最近，这些字体还是能在网页中放心使用的唯一 一批字体。）

2、保存在第三方网站上的字体。最常见的是Typekit 和Google，可以使用link 标签把它们链接到你的页面上。

3、保存在你的Web 服务器上的字体。这些字体可以使用@font-face 规则随网页一起发送给浏览器。