

CSS学习笔记

肖桐 PB18000037

本文摘抄自[菜鸟教程](#)

CSS 即层叠样式表(Cascading Style Sheets)。样式表定义如何显示 HTML 元素，就像 HTML 中的字体标签和颜色属性所起的作用那样。

一、CSS 语法

1. CSS 规则

CSS 规则由两个主要的部分构成：选择器，以及一条或多条声明。如：

```
1 | h1 {color:blue; font-size:12px;}
```

选择器通常是需要改变样式的 HTML 元素。每条声明由一个属性和一个值组成(类似于 HTML 中标签的属性)，属性和值被冒号分开(HTML 中用等号分开)。

[CSS选择器参考手册](#) [CSS属性参考](#) [CSS单位](#)

CSS 声明总是以分号(;)结束，声明总是以大括号({ })括起来。

2. CSS 注释

CSS 注释以 /* 开始，以 */ 结束(类似 C)。

二、CSS 选择器

[CSS选择器参考手册](#)

1. id 选择器

id 选择器可以为标有特定 id 的 HTML 元素指定特定的样式。

HTML 元素以 id 属性来设置 id 选择器，CSS 中 id 选择器以 # 来定义。如以下的样式规则应用于元素属性 id="para1"：

```
1 | #para1 {text-align:center; color:red;}
```

2. class 选择器

class 选择器用于描述一组元素的样式，class 选择器有别于 id 选择器，class 可以在多个元素中使用。

class 选择器在 HTML 中以 class 属性表示，在 CSS 中，类选择器以一个点 . 号显示：

在以下的例子中，所有拥有 center 类的 HTML 元素均为居中。

```
1 | .center {text-align:center;}
```

也可以指定**特定的 HTML 元素使用 class**。

在以下实例中, 所有的 p 元素使用 `class="center"` 让该元素的文本居中:

```
1 p.center {text-align:center;}
```

三、CSS 创建

当读到一个样式表时, 浏览器会根据它来格式化 HTML 文档。

1. 如何插入样式表

(1). 外部样式表

通过使用 `<link>` 标签链接到(导入)样式表。 `<link>` 标签应该在头部使用:

```
1 <head>
2   <link rel="stylesheet" type="text/css" href="mystyle.css">  <!-- rel属性和
   type属性有什么用? -->
3 </head>
```

(2). 内部样式表

当文档需要特殊的样式时, 可以使用内部样式表。使用 `<style>` 标签在文档头部定义内部样式表, 如:

```
1 <head>
2   <style>
3     hr {color:sienna;}
4     p {margin-left:20px;}
5     body {background-image:url("images/back40.gif")}
6   </style>
7 </head>
```

(3). 内联样式

当样式仅需要在一个元素上应用一次时可用内联样式。要使用内联样式, 可在相关的标签内使用 `style` 属性, `style` 属性可以包含任何 CSS 属性, 如:

```
1 <p style="color:sienna; margin-left=20px">这是一个段落</p>
```

(4). 多重样式

如果某些属性在不同的样式表中被同样的选择器定义, 那么属性值将从**更具体的样式表**中被继承过来。

例如, **外部样式表**拥有针对 h3 选择器的三个属性:

```
1 h3
2 {
3   color:red;
4   text-align:left;
5   font-size:8pt;
6 }
```

而内部样式表拥有针对 h3 选择器的两个属性：

```
1 h3
2 {
3     text-align:right;
4     font-size:20pt;
5 }
```

假如拥有内部样式表的这个页面同时与外部样式表链接，那么 h3 得到的样式是：

```
1 h3
2 {
3     color:red;
4     text-align:right;
5     font-size:20pt;
6 }
```

即颜色属性将被继承于外部样式表，而文字排列（text-alignment）和字体尺寸（font-size）会被内部样式表中的规则取代。（因为内部样式表更“具体”，优先级更高。）

一般情况下，优先级如下：

（内联样式）Inline style > （内部样式）Internal style sheet > （外部样式）External style sheet > 浏览器默认样式

四、CSS 背景

CSS 背景属性用于定义 HTML 元素的背景。

[CSS背景](#)

1. 背景颜色

background-color 属性定义了元素的背景颜色。

页面的背景颜色使用在 body 的选择器中：

```
1 body {background-color:#b0c4de}
```

CSS 中，颜色值可以以以下方式定义：

- 十六进制 - 如："#ff0000"
- RGB - 如："rgb(255, 0, 0)"
- 颜色名称 - 如：red

2. 背景图像

background-image 属性描述了元素的背景图像。

默认情况下，背景图像进行平铺重复显示，以覆盖整个元素实体。如：

```
1 body {background-image:url('paper.gif');}
```

(1). 水平或垂直平铺

默认情况下 `background-image` 属性会在页面的水平或者垂直方向平铺。

设置图像只在水平方向平铺：

```
1 body
2 {
3     background-image:url("...");
4     background-repeat:repeat-x;
5     /* 在垂直方向平铺则是 repeat-y, 不平铺则是 no-repeat */
6 }
```

(2). 设置位置

以上实例中，背景图像与文本显示在同一个位置，为了让页面排版更加合理，不影响文本的阅读，我们可以改变图像的位置。

可以利用 `background-position` 属性改变图像在背景中的位置：

```
1 body
2 {
3     background-image:url("...");
4     background-repeat:no-repeat;
5     background-position:right top;
6 }
```

(3). 简写属性

我们也可以将以上的属性合并并在同一个属性中，背景颜色的简写属性为 `background`。如：

```
1 body {background: #ffffff url("...") no-repeat right top;} /* 分别为颜色、图
   像、平铺、位置 */
```

简写时的属性值顺序为：

- `background-color`
- `background-image`
- `background-repeat`
- `background-attachment`
- `background-position`

五、CSS 文本格式

[CSS 文本格式](#)

1. 文本修饰

`text-decoration` 属性用来设置或删除文本的装饰。

从设计的角度看 `text-decoration` 属性主要是用来删除链接的下划线：

```
1 a {text-decoration:none;}
```

也可以用来装饰文字：

```
1 h1 {text-decoration:overline;}
2 h2 {text-decoration:line-through;}
3 h3 {text-decoration:underline;}
```

2. 文本转换

文本转换属性是用来指定在一个文本中的大写和小写字母。

可用于所有字句变成大写或小写字母，或每个单词的首字母大写。如：

```
1 p.uppercase {text-transform:uppercase;} /* 通过class属性使用 如<p
   class="uppercase">...</p> */
2 p.lowercase {text-transform:lowercase;}
3 p.capitalize {text-transform:capitalize;} /* 有什么用? */
```

3. 文本缩进

文本缩进属性是用来指定文本的第一行的缩进：

```
1 p {text-indent:50px;}
```

4. 指定文本之间的空间

(1). 字符之间的空间

使用 `letter-spacing` 属性来指定字符之间的空间：

```
1 h1 {letter-spacing:2px;}
2 h2 {letter-spacing:-3px;}
```

汉字之间的空间也是使用 `letter-space` 属性进行调整的。

(2). 行与行之间的空间

`line-height` 属性用于决定行与行之间的空间，如：

```
1 p.small {line-height:70%;}
2 p.big {line-height:200%;}
```

(3). 单词之间的空间

`word-spacing` 属性用于修改单词之间的空间，如：

```
1 p {word-spacing:30px;}
```

5. 文本阴影

`text-shadow` 用于设置文本阴影，如：

```
1 h1 {text-shadow:2px 2px 0px #FF0000;} /* 实际上就是将改变颜色之后的字体进行偏移，
    第一个长度表示左右偏移，第二个长度表示上下偏移，第三个单位表示阴影的模糊程度 */
```

六、CSS 字体

[CSS 字体](#)

1. CSS 字体系列

`font-family` 属性用于设置文本的字体系列。如：

```
1 p {font-family:"Times New Roman", "Times", "serif";}
```

七、CSS 链接

[CSS 链接](#)

四个链接状态：

- `a:link` - 正常，未访问过的链接
- `a:visited` - 用户已访问过的链接
- `a:hover` - 当用户鼠标放在链接上时
- `a:active` - 链接被点击的那一刻

可设置这四种状态的链接颜色：

```
1 a:link {color:#000000;}
2 a:visited {color:#00FF00;}
3 a:hover {color:#FF00FF;}
4 a:active {color:#0000FF;}
```

当设置为若干链路状态的样式，也有一些顺序规则：

- `a:hover` 必须跟在 `a:link` 和 `a:visited` 之后
- `a:active` 必须跟在 `a:hover` 后面

也可以删除或修改链接的下划线，通过 `text-decoration` 属性实现。如：

```
1 a:link {text-decoration:none;}
2 a:visited {text-decoration:none;}
3 a:hover {text-decoration:underline;}
4 a:active {text-decoration:underline;}
```

还可以通过属性 `background-color` 改变链接背景颜色，如：

```
1 a:link {background-color:#B2FF99;}
2 a:visited {background-color:#FFFF85;}
3 a:hover {background-color:#FF704D;}
4 a:active {background-color:#FF704D;}
```

八、CSS 列表

1. 不同的列表项标记

`list-style-type` 属性指定列表项标记的类型是：

```
1 ul.a {list-style-type:circle;}
2 ul.b {list-style-type:square;}
3
4 ol.c {list-style-type:upper-roman;}
5 ol.d {list-style-type:lower-alpha;}
```

2. 作为列表项标记的图像

可以使用属性 `list-style-image` 属性将列表项标记改为图像：

```
1 ul {list-style-image:url("sqpurple.gif");}
```

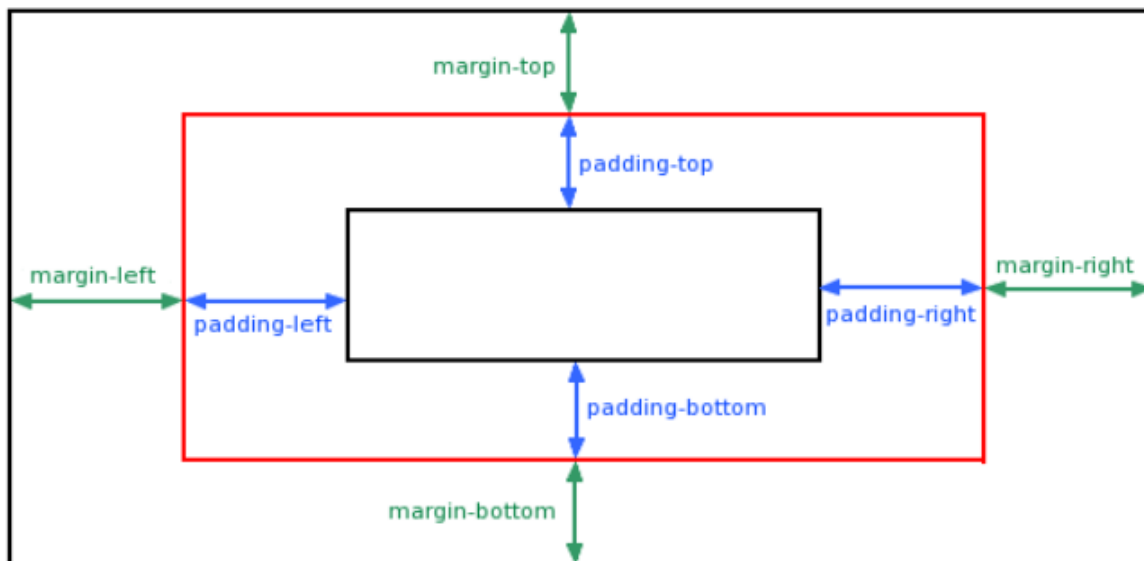
九、CSS表格

十、CSS盒子模型

[CSS盒子模型](#)

CSS 盒模型本质上是一个盒子，封装周围的 HTML 元素，它包括：

- `Margin` 外边距 - 清除边框外的区域，外边距是透明的。
- `Border` 边框 - 围绕在内边距和内容外的边框。
- `Padding` 填充(内边距) - 清除内容周围的区域，内边距是透明的。
- `Content` 实际内容 - 盒子的内容，显示文本和图像。



元素的宽度和高度

通过对 `div` 标签进行配置，可以达到指定元素宽度和高度的目的，如：

```

1  div
2  {
3      width: 220px;    /* 内容(content)的宽度 */
4      padding: 10px;  /* 内容距离两边界的宽度，内边距 */
5      border: 5px solid gray; /* 边距宽度 */
6      margin: 0px;    /* */
7  }
8  /* 上边距和下边距? */

```

总元素宽度的计算方法：总元素宽度 = $width + 2 * padding + 2 * border + 2 * margin$

1. CSS border 边框

[CSS边框](#)

(1). CSS 边框样式

css 边框属性允许指定一个元素边框的样式和颜色。通过 `border-style` 属性来指定。有：

```

1  border-style:none; /* 无边框 */
2  border-style:dotted; /* 定义一个点线边框 */
3  border-style:dashed; /* 定义一个虚线边框 */
4  border-style:solid; /* 定义实线边框 */
5  border-style:double; /* 定义两个边框，两个边框的宽度与 border-width 的值相同 */
6  border-style:groove; /* 定义3D沟槽边框，可指定边框颜色 */
7  border-style:ridge; /* 定义3D脊边框，可指定边框颜色 */
8  border-style:inset; /* 定义一个3D的嵌入边框，可指定边框颜色 */
9  border-style:outset; /* 定义一个3D突出边框，可指定颜色 */

```

边框颜色可以通过 `border-color` 属性决定。

(2). 边框宽度

可以通过 `border-width` 属性来为边框指定宽度。如：

```

1  p.one
2  {
3      border-style:solid;
4      border-width:5px;
5  }
6  p.two
7  {
8      border-style:solid;
9      border-width:medium; /* 3个关键字: thick, medium, thin */
10 }

```

(3). 单独设置各边

可以指定不同的侧边不同的边框，如：


```
1 p
2 {
3     border-top-style:dotted;
4     border-right-style:solid;
5     border-bottom-style:dotted;
6     border-left-style:solid;
7 }
```

可见，可通过附加 `top`、`right`、`bottom`、`left` 来改变特定边的属性。`border-width`、`border-color` 也相同。

2. CSS outline 轮廓

[CSS 轮廓](#)

轮廓 (`outline`) 是绘制于元素周围的一条线，位于边框边缘的外围，可起到突出元素的作用。通过 `outline-color`、`outline-style`、`outline-width` 属性来决定轮廓样式。如：

```
1 p
2 {
3     outline-color:red;
4     outline-style:none; /* 样式和CSS边框相同 */
5     outline-width:2px;
6 }
```

3. CSS margin 外边距

[CSS margin](#)

`margin` 清除周围的（外边框）元素区域。`margin` 没有背景颜色，是完全透明的。

`margin` 可以单独改变元素的上，下，左，右边距，也可以一次改变所有的属性。

Q：有什么用？怎么用？

A：

(1). 单边外边距属性

类似于 `CSS border`，可以使用 `top`、`bottom`、`left`、`right` 指定不同侧面不同的边距。如：

```
1 p.margin
2 {
3     margin-top:100px;
4     margin-bottom:100px;
5     margin-left:50px;
6     margin-right:50px;
7 }
```

4. CSS padding 填充

[CSS 填充](#)

当元素的 `padding`（填充）内边距被清除时，所释放的区域将会受到元素背景颜色的填充。

单独使用 `padding` 属性可以改变上下左右的填充。

(1). 单边内边距属性

如同 CSS border 边框一样，使用 top、bottom、left、right 可以改变四个方向的填充大小。如：

```
1 p.padding
2 {
3     padding-top:25px;
4     padding-bottom:25px;
5     padding-left:50px;
6     padding-right:50px;
7 }
```

十一、CSS 分组和嵌套选择器

[CSS 分组和嵌套选择器](#)

1. 分组选择器

在 CSS 中有很多具有相同样式的元素，为避免以下冗长的代码：

```
1 h1 {color:green;}
2 h2 {color:green;}
3 p {color:green;}
```

可以使用以下写法：

```
1 h1,h2,p {color:green;}
```

2. 嵌套选择器

它可能适用于选择器内部的选择器的样式。

- p {} 为所有 p 元素指定一个样式
- .marked {} 为所有 class="marked" 的元素指定一个样式
- .marked p {} 为所有 class="marked" 元素内的 p 元素指定一个样式
- p.marked {} 为所有 class="marked" 的 p 元素指定一个样式

如：

```
1 <head>
2 <style>
3 p
4 {
5     color:blue;
6     text-align:center;
7 }
8 .marked
9 {
10     background-color:red;
11 }
12 .marked p
13 {
14     color:white;
15 }
```

```

16  p.marked{
17      text-decoration:underline;
18  }
19  </style>
20  </head>
21  <body>
22  <p>这个段落是蓝色文本，居中对齐。</p>
23  <div class="marked">
24  <p>这个段落不是蓝色文本。</p>
25  </div>
26  <p>所有 class="marked"元素内的 p 元素指定一个样式，但有不同的文本颜色。</p>
27
28  <p class="marked">带下划线的 p 段落。</p>
29  </body>

```

十二、CSS 尺寸

十三、CSS 显示(Display)和可见性(Visibility)

[CSS显示和可见性](#)

1. 隐藏元素

`display` 属性设置一个元素应如何显示，`visibility` 属性指定一个元素应可见还是隐藏。

`visibility:hidden` 可以隐藏某个元素，但隐藏的元素仍需占用与未隐藏之前一样的空间。也就是说，该元素虽然被隐藏了，但仍然会影响布局。如：

```
1  h1.hidden {visibility:hidden;}
```

`display:none` 可以隐藏某个元素，且隐藏的元素不会占用任何空间。也就是说，该元素不但被隐藏了，而且该元素原本占用的空间也会从页面布局中消失。如：

```
1  h1.hidden {display:none;}
```

2. CSS Display - 块和行内元素

块元素是一个元素，占用了全部宽度，在前后都是换行符。

块元素拥有如下特点：

1. 每个块级元素都是独自占一行
2. 元素的高度、宽度、行高和边距都是可以设置的
3. 元素的宽度如果不设置的话，默认为父元素的宽度

块元素的例子：

- `<h1>`
- `<p>`
- `<div>`

行内元素只需要必要的宽度，不强制换行。

行内元素的特点：

1. 每一个行内元素可以和别的行内元素共享一行，相邻的行内元素会排列在同一行里，直到一行排不下了，才会换行。
2. 行内元素设置 `width`，`height` 无效，宽度随元素的内容而变化。
3. 行内水平方向的 `padding-left` 和 `padding-right` 都会产生边距效果，但是竖直方向上的 `padding-top` 和 `padding-bottom` 都不会产生边距效果。

行内元素的例子：

- ``
- `<a>`

3. 改变元素显示

可以更改行内元素和块元素，反之亦然，可以使页面看起来是以一种特定的方式组合，并仍然遵循web标准。

下面的示例把列表项显示为行内元素：

```
1 | li {display:inline;}
```

下面的示例把span元素作为块元素：

```
1 | span {display:block;}
```