

CSS层叠样式表

ZhangXu

2019 年 1 月 23 日

第一部分 CSS基础教程

1 CSS概述

- CSS 指层叠样式表 (Cascading Style Sheets)
- 样式定义如何显示 HTML 元素
- 样式通常存储在样式表中
- 把样式添加到 HTML 4.0 中，是为了解决内容与表现分离的问题
- 外部样式表可以极大提高工作效率
- 外部样式表通常存储在 CSS 文件中
- 多个样式定义可层叠为一

1.1 多重样式层叠为一个

样式表允许以多种方式规定样式信息。样式可以规定在单个的 HTML 元素中，在 HTML 页的头元素中，或在一个外部的 CSS 文件中。甚至可以在同一个 HTML 文档内部引用多个外部样式表。

1.2 层叠次序

一般而言，所有的样式会根据下面的规则层叠于一个新的虚拟样式表中，其中数字 4 拥有最高的优先权

- 1 浏览器缺省设置

2 外部样式表

3 内部样式表（位于 `<head>` 标签内部）

4 内联样式（在 HTML 元素内部）

2 基础语法

2.1 语法

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

selectordeclaration1; declaration2; ...declarationN

- 选择器通常是您需要改变样式的 HTML 元素
- 每条声明由一个属性和一个值组成。属性（property）是您希望设置的样式属性（style attribute）。每个属性有一个值。属性和值被冒号分开。
- *selectorproperty : value* (如果值为若干单词，则要给值加引号)

3 CSS高级语法

3.1 选择器分组

你可以对选择器进行分组，这样，被分组的选择器就可以分享相同的声明。用逗号将需要分组的选择器分开

3.2 继承

根据 CSS，子元素从父元素继承属性。通过 CSS 继承，子元素将继承最高级元素（在本例中是 `body`）所拥有的属性（这些子元素诸如 `p`, `td`, `ul`, `ol`, `li`, `dl`, `dt`, 和 `dd`）。

如果你不希望 “`Verdana, sans-serif`” 字体被所有的子元素继承，又该怎么做呢？比方说，你希望段落的字体是 Times。没问题。创建一个针对 `p` 的特殊规则，这样它就会摆脱父元素的规则：*`pfont-family : Times, "TimesNewRoman", serif;`*

4 CSS派生选择器

通过依据元素在其位置的上下文关系来定义样式，你可以使标记更加简洁。

5 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。id 选择器以“#”来定义。（注：id 属性只能在每个 HTML 文档中出现一次。）

5.1 id选择器和派生选择器

在现代布局中，id 选择器常常用于建立派生选择器。

一个选择器，多种用法 即使被标注为 sidebar 的元素只能在文档中出现一次，这个 id 选择器作为派生选择器也可以被使用很多次。

单独的选择器 id 选择器即使不被用来创建派生选择器，它也可以独立发挥作用

6 类选择器

在 CSS 中，类选择器以一个点号显示（类名的第一个字符不能使用数字！它无法在 Mozilla 或 Firefox 中起作用）

和id一样，class也可用作派生选择器 类名为某某的的元素内部

元素也可以基于他们的类而被选择 如`element.class`====< `tdclass = "fancy"` >

7 属性选择器

对带有指定属性的 HTML 元素设置样式
可以为拥有指定属性的 HTML 元素设置样式，而不仅限于 class 和 id 属性

7.1 属性选择器

`[title = title]...`

7.2 属性和值选择器-多个值

适用于由空格分隔的属性值 `[title = hello]color : red;`

适用于由连字符分隔的属性值 `[lang| = en]color : red;`

设置表单的样式 属性选择器在为不带有 class 或 id 的表单设置样式时特别有用 `input[type = "text"]`

8 如何创建CSS

如何插入样式表?当读到一个样式表时,浏览器会根据它来格式化 HTML 文档。插入样式表的方法有三种

8.1 外部样式表

当样式需要应用于很多页面时,外部样式表将是理想的选择。在使用外部样式表的情况下,你可以通过改变一个文件来改变整个站点的外观。每个页面使用 `< link >` 标签链接到样式表。`< link >` 标签在(文档的)头部:`< head >< linkrel = "stylesheet" type = "text/css" href = "mystyle.css" / >< /head >` 浏览器会从文件 `mystyle.css` 中读到样式声明,并根据它来格式文档(不要在属性值与单位之间留有空格)

8.2 内部样式表

当单个文档需要特殊的样式时,就应该使用内部样式表。你可以使用 `<style>` 标签在文档头部定义内部样式表,就像这样

```
< head >< styletype = "text/css" > ... < /style >< /head >
```

8.3 内联样式

由于要将表现和内容混杂在一起,内联样式会损失掉样式表的许多优势。请慎用这种方法,例如当样式仅需要在一个元素上应用一次时。

要使用内联样式,你需要在相关的标签内使用样式(style)属性。Style 属性可以包含任何 CSS 属性。

```
< pstyle = "..." > thisisparagraph < /p >
```

8.4 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。

第二部分 CSS样式

9 背景

CSS 允许应用纯色作为背景，也允许使用背景图像创建相当复杂的效果。CSS 在这方面的能力远远在 HTML 之上。

9.1 背景色

可以使用 `background-color` 属性为元素设置背景色。这个属性接受任何合法的颜色值

9.2 背景图像

要把图像放入背景，需要使用 `background-image` 属性。`background-image` 属性的默认值是 `none`，表示背景上没有放置任何图像。

10 CSS文本

CSS 文本属性可定义文本的外观。通过文本属性，您可以改变文本的颜色、字符间距，对齐文本，装饰文本，对文本进行缩进，等等。

11 CSS字体

CSS 字体属性定义文本的字体系列、大小、加粗、风格（如斜体）和变形（如小型大写字母）。

11.1 CSS字体系列

在 CSS 中，有两种不同类型的字体系列名称：

- 通用字体系列 - 拥有相似外观的字体系统组合（比如 "Serif" 或 "Monospace"）
- 特定字体系列 - 具体的字体系列(比如 "Times" 或 "Courier")

除了各种特定的字体系列外，CSS 定义了 5 种通用字体系列：

- Serif 字体
- Sans-serif 字体
- Monospace 字体
- Cursive 字体
- Fantasy 字体

11.2 指定字体系列

使用 `font-family` 属性定义文本的字体系列

使用通用字体系列 `body font-family: sans-serif;`

指定字体系列 除了使用通用的字体系列，您还可以通过 `font-family` 属性设置更具体的字体。

这样的规则同时会产生另外一个问题，如果用户代理上没有安装 Georgia 字体，就只能使用用户代理的默认字体来显示 `h1` 元素。

我们可以通过结合特定字体名和通用字体系列来解决这个问题：`h1 font-family: Georgia, serif;`

11.3 字体风格

`font-style` 属性最常用于规定斜体文本

- normal - 文本正常显示
- italic - 文本斜体显示
- oblique - 文本倾斜显示

11.4 字体变形

`font-variant` 属性可以设定小型大写字母。小型大写字母不是一般的大写字母，也不是小写字母，这种字母采用不同大小的大写字母。

11.5 字体加粗

`font-weight` 属性设置文本的粗细；使用 `bold` 关键字可以将文本设置为粗体；关键字 `100 900` 为字体指定了 9 级加粗度；

11.6 字体大小

`font-size` 属性设置文本的大小。请始终使用正确的 HTML 标题，比如使用 `<h1>` – `<h6>` 来标记标题，使用 `<p>` 来标记段落。

`font-size` 值可以是绝对或相对值。

绝对值：

- 将文本设置为指定大小
- 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

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
- 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`）

11.7 使用像素设置字体大小

通过像素设置文本大小，可以对文本大小进行完全控制

11.8 使用em来设置字体大小

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。

12 CSS链接

链接的四种状态:

- a:link - 普通的、未被访问的链接
- a:visited - 用户已访问的链接
- a:hover - 鼠标指针位于链接的上方
- a:active - 链接被点击的时刻

文本修饰 `a : linktext - decoration : none;`

背景色 `a : linkbackground - color : #B2FF99;`

13 CSS列表

CSS 列表属性允许你放置、改变列表项标志, 或者将图像作为列表项标志

13.1 列表

从某种意义上讲, 不是描述性的文本的任何内容都可以认为是列表。人口普查、太阳系、家谱、参观菜单, 甚至你的所有朋友都可以表示为一个列表或者是列表的列表。

列表类型 要影响列表的样式, 最简单(同时支持最充分)的办法就是改变其标志类型。

例如, 在一个无序列表中, 列表项的标志 (marker) 是出现在各列表项旁边的圆点。在有序列表中, 标志可能是字母、数字或另外某种计数体系中的一个符号。

要修改用于列表项的标志类型, 可以使用属性 `list-style-type: ullist - style - type : square`

列表项图像 常规的标志是不够的。你可能想对各标志使用一个图像, 这可以利用 `list-style-image` 属性做到`ul``list - style - image : url(www.gif)`

列表标志位置 标志出现在列表项内容之外还是内容内部。这是利用 `list-style-position` 完成的

简写列表样式 为简单起见，可以将以上 3 个列表样式属性合并为一个方便的属性：`list-style`，就像这样：`list-style: url(example.gif)squareinside`

14 CSS表格

14.1 表格边框

如需在 CSS 中设置表格边框，请使用 `border` 属性。下面的例子为 `table`、`th` 以及 `td` 设置了蓝色边框：`table,th,tdborder:1pxsolidblue;`

14.2 折叠边框

`border-collapse` 属性设置是否将表格边框折叠为单一边框

14.3 表格宽度和高度

通过 `width` 和 `height` 属性定义表格的宽度和高度

14.4 表格文本对齐

`text-align` 和 `vertical-align` 属性设置表格中文本的对齐方式；
`text-align` 属性设置水平对齐方式，比如左对齐、右对齐或者居中；
`vertical-align` 属性设置垂直对齐方式，比如顶部对齐、底部对齐或居中对齐

14.5 表格内边距

如需控制表格中内容与边框的距离，请为 `td` 和 `th` 元素设置 `padding` 属性

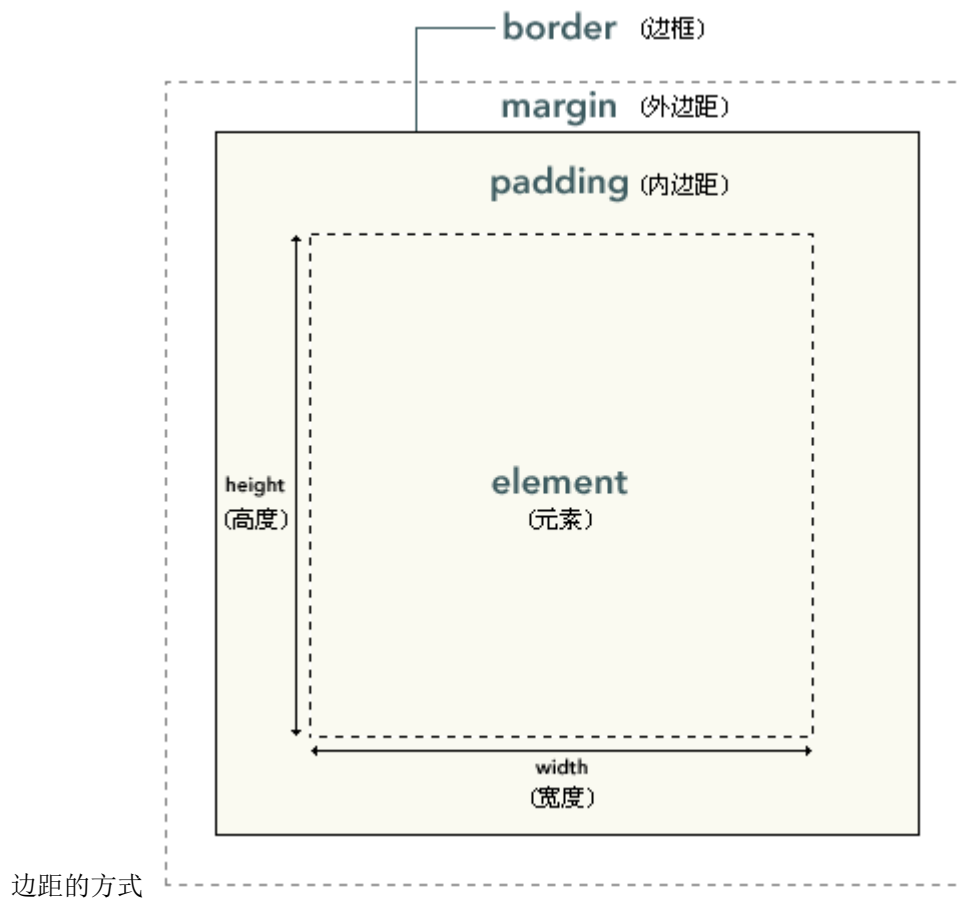
14.6 表格颜色

15 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。CSS `outline` 属性规定元素轮廓的样式、颜色和宽度。

第三部分 CSS框模型

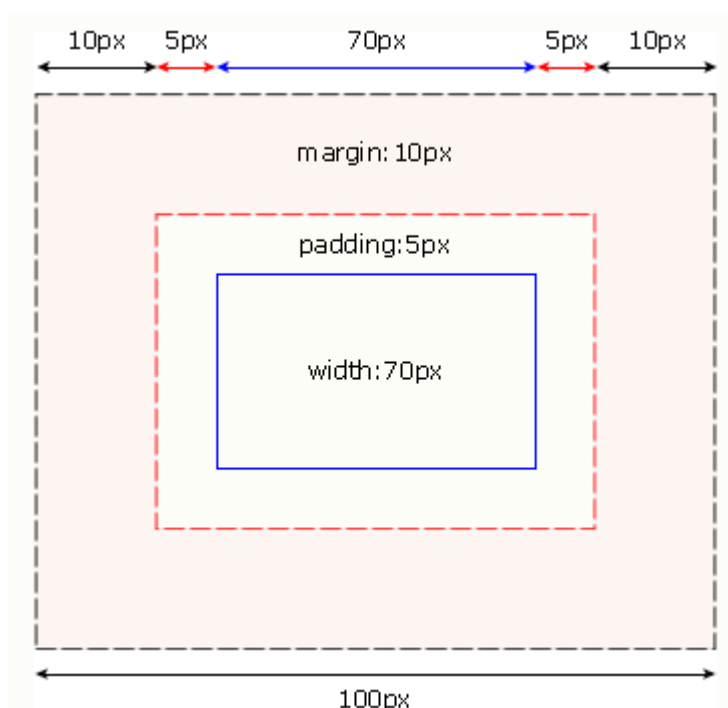
CSS 框模型 (Box Model) 规定了元素框处理元素内容、内边距、边框和外



- 元素框的最内部分是实际的内容，直接包围内容的是内边距
- 内边距呈现了元素的背景
- 内边距的边缘是边框
- 边框以外是外边距，外边距默认是透明的，因此不会遮挡其后的任何元素
- 提示：背景应用于由内容和内边距、边框组成的区域。
- 内边距、边框和外边距都是可选的，默认值是零。但是，许多元素将由用户代理样式表设置外边距和内边距。可以通过将元素的 `margin` 和 `padding`

设置为零来覆盖这些浏览器样式。这可以分别进行，也可以使用通用选择器对所有元素进行设置：

- CSS 中，width 和 height 指的是内容区域的宽度和高度。增加内边距、边框和外边距不会影响内容区域的尺寸，但是会增加元素框的总尺寸
- 假设框的每个边上有 10 个像素的外边距和 5 个像素的内边距。如果希望这个元素框达到 100 个像素，就需要将内容的宽度设置为 70 像素，请看下图：



提示：内边距、

边框和外边距可以应用于一个元素的所有边，也可以应用于单独的边。

提示：外边距可以是负值，而且在很多情况下都要使用负值的外边距。

16 内边距

元素的内边距在边框和内容区之间。控制该区域最简单的属性是 padding 属性。CSS padding 属性定义元素边框与元素内容之间的空白区域。

16.1 padding属性

padding 属性定义元素的内边距。padding 属性接受长度值或百分比值，但不允许使用负值。

所有 h1 元素的各边都有 10 像素的内边距：`h1padding: 10px;`

按照上、右、下、左的顺序分别设置各边的内边距，各边均可以使用不同的单位或百分比值：`h1padding: 10px 0.25em 2ex 20%;`

16.2 单边内边距属性

也通过使用下面四个单独的属性，分别设置上、右、下、左内边距：

- padding-top
- padding-right
- padding-bottom
- padding-left

`h1padding-top: 10px; padding-right: 0.25em; padding-bottom: 2ex; padding-left: 20%;`

16.3 内边距的百分比数值

可以为元素的内边距设置百分数值。百分数值是相对于其父元素的 width 计算的，这一点与外边距一样。（例如：如果一个段落的父元素是 div 元素，那么它的内边距要根据 div 的 width 计算）

17 CSS边框

元素的边框 (border) 是围绕元素内容和内边距的一条或多条线。CSS border 属性允许你规定元素边框的样式、宽度和颜色。

17.1 边框

每个边框有 3 个方面：宽度、样式，以及颜色

边框与背景 元素的背景是内容、内边距和边框区的背景

边框样式 CSS 的 border-style 属性定义了 10 个不同的非 inherit 样式，包括 none

定义多种样式 您可以为一个边框定义多个样式。我们又看到了这里的值采用了 top-right-bottom-left 的顺序，讨论用多个值设置不同内边距时也见过这个顺序

17.2 定义单边样式

- border-top-style
- border-right-style
- border-bottom-style
- border-left-style

17.3 边框宽度

您可以通过 border-width 属性为边框指定宽度

注释：CSS 没有定义 3 个关键字的具体宽度，所以一个用户代理可能把 thin 、medium 和 thick 分别设置为等于 5px、3px 和 2px，而另一个用户代理则分别设置为 3px、2px 和 1px。

17.4 定义单边宽度

您可以按照 top-right-bottom-left 的顺序设置元素的各边边框：您也可以通过下列属性分别设置边框各边的宽度：

17.5 没有边框

这是因为如果边框样式为 none，即边框根本不存在，那么边框就不可能有宽度，因此边框宽度自动设置为 0，而不论您原先定义的是什么。因此，如果您希望边框出现，就必须声明一个边框样式。

17.6 边框颜色

CSS 使用一个简单的 border-color 属性，它一次可以接受最多 4 个颜色值。还有一些单边边框颜色属性。它们的原理与单边样式和宽度属性相同

17.7 透明边框

引入了边框颜色值 transparent。这个值用于创建有宽度的不可见边框

18 外边距

围绕在元素边框的空白区域是外边距。设置外边距会在元素外创建额外的“空白”。设置外边距的最简单的方法就是使用 `margin` 属性，这个属性接受任何长度单位、百分数值甚至负值。

18.1 `margin`属性

`margin` 属性接受任何长度单位，可以是像素、英寸、毫米或 `em`。`margin` 可以设置为 `auto`。更常见的做法是为外边距设置长度值。下面的声明在 `h1` 元素的各个边上设置了 1/4 英寸宽的空白

18.2 值复制

CSS 定义了一些规则，允许为外边距指定少于 4 个值。

- 如果缺少左外边距的值，则使用右外边距的值。
- 如果缺少下外边距的值，则使用上外边距的值。
- 如果缺少右外边距的值，则使用上外边距的值。

18.3 单边外边距属性

您可以使用单边外边距属性为元素单边上的外边距设置值。

19 外边距合并

外边距合并指的是，当两个垂直外边距相遇时，它们将形成一个外边距。合并后的外边距的高度等于两个发生合并的外边距的高度中的较大者。

20 CSS定位 (Positioning)

20.1 一切皆为框

`div`、`h1` 或 `p` 元素常常被称为块级元素。这意味着这些元素显示为一块内容，即“块框”。而 `span` 和 `strong` 等元素称为“行内元素”，这是因为它们的内容显示在行中，即“行内框”。

您可以使用 `display` 属性改变生成的框的类型。这意味着，通过将 `display` 属

性设置为 `block`，可以让行内元素（比如 `jai` 元素）表现得像块级元素一样。还可以通过把 `display` 设置为 `none`，让生成的元素根本没有框。这样的话，该框及其所有内容就不再显示，不占用文档中的空间。

20.2 CSS定位机制

CSS 有三种基本的定位机制：**普通流、浮动和绝对定位**