



# CSS 基础教程

2008-5-7

# 一、CSS 概念

- >**CSS** stands for **Cascading Style Sheets**
- >Styles define **how to display** HTML elements
- >Styles are normally stored in **Style Sheets**
- >Styles were added to HTML 4.0 **to solve a problem**
- >**External Style Sheets** can save you a lot of work
- >External Style Sheets are stored in **CSS files**

- 以往，要定义字的颜色，必须在HTML文件中的字所在位置加上格式限定，如果网页中有十二行标题，都需要设定为红色，就只好每行标题处都作设定。

- 例如：

```
<font color= “red”>This is a title. </font>
```

# 样式表 Style Sheets

---

- 样式表Style Sheets，就是指对网页特定的内容一次过添加格式设置。
  - 用了样式表，就不必在每行需要格式设置的字的前后加上标签了。
  - 你可以设定好一个样式表，把它放在HTML文件的合适位置，这个样式表就会对整个文件要格式的相应部分起作用。
-

- 例如: `td{ font- color: # FF0000}`

- 该网页中所有带<td>标签的文字，即标题字，都会变成红色了，不必逐行格式设置。而当你想把红色改为黑色时，就直接把#FF0000改为# 000000，一次过就把所有的标题改了颜色。
- W3C从HTML3.2版本开始，样式表就开始推广使用了。

# 什么是CSS层叠样式表？

---

- CSS (Cascading Style Sheets)，就是“层叠样式表”。正如其名，就是将多种的格式设置象洋葱的皮一样，一层一层地加到网页上面去，每加一层，整个页面就象涂上了一层新的油漆，获得了新的格式设置。

# CSS的应用是网页编辑的一大革命。

---

- 通过设置CSS，我们可以随意地控制网页中字体的大小、颜色等，便于统一网站的**整体风格**；
  - 可以方便地为网页中的各个元素设置背景颜色和图片并进行精确的**定位控制**；
  - 可以为网页中的元素设置各种**滤镜**，从而产生诸如阴影、辉光、模糊等只有在图像处理软件中才能实现的效果；
  - 可以与脚本语言相结合，在网页中实现很多**动态滤镜效果**。
-

- **CSS**是网站设计必不可少的一个重要组成部分。



# CSS基本特性：分组、继承、层叠

- 分组：
- 通常我们需要同时改变几种样式的属性，**CSS**允许声明通过使用**分号**分隔其内容的方式来分组。
- 例如：**body,td,p {color:#000000;text-align:left;}**

# CSS基本特性：分组、继承、层叠

---

- 继承：
- 通过继承，**CSS**设置可以被应用到多个标识中。绝大部分的**CSS**声明可以通过封闭**CSS**选择器中的**HTML**标识来被继承。
- 例如通过一个**CSS**设置来改变整个页面的字体：  
**body { font-family: "Arial"; font-size: 14px; line-height: 18px; color: #000000 }**
- **<body>**标识---页面上所有元素的父标识。

# CSS基本特性：分组、继承、层叠

---

## ■ 层叠:

- **Cascading**描述了覆盖通用样式的局部样式的能力。在相同的方式中，一个普遍地应用到某一块文本上的**CSS**规则可以被其他应用到相同文本中某个更为特殊的部分的规则所覆盖。

附加CSS样式  
即链接一个外部CSS文件

新建CSS样式



删除CSS样式

编辑CSS样式

## 二、样式表语法

---

- HTML 语言由标志和属性构成, CSS 也是如此。
  - 样式表基本语法:
    - 选择器 {属性1: 值; 属性2: 值; .....}
    - **h3 {color: red; text-align: left; font-size: 8pt}**
-

# CSS如何起作用?

- 在 HTML 页面内直接引用样式表的方法。



```
<html>
  <head>
    <title>网页标题</title>
    <style>CSS样式表</style>
  </head>

  <body>
    网页的内容

    HTML标签或类名称与上面的CSS呼应
  </body>
</html>
```

# 三、样式表的放置

---

- 样式表放在不同的地方，产生作用的范围也不同。概括起来有四种方法。其中常用前三种。
  - 方法一：内部（或嵌入式）CSS 样式表
  - 方法二：外部导入CSS 样式表
  - 方法三：外部链接CSS 样式表
  - 方法四：行间写入式CSS 样式表.
-

## 方法一：内部（或嵌入式）CSS 样式表

---

- 直接把CSS代码粘贴到html中。
- 比如我们把下面一段代码粘贴入HTML代码的< **head** ></ **head** >之间就可以了。



```
<head>
<style type="text/css">
<!--
p { font-family: "宋体"; font-size: 12px; line-
    height: 10pt; color: #000000}
td { font-family: "宋体"; font-size: 12px; color:
    #000000}
a:link { color: #0000FF; text-decoration: none}
a:visited { color: #666666; text-decoration: none}
a:active {color:green;text-decoration: none;}
a:hover      {          color:          #ff0000;background-
    color:#FF9900;text-decoration: underline }
-->
</style>
</head>
```

## 方法二：外部导入CSS 样式表

---

- 就是将外部的\*.css文件导入到所需的网页面HTML文件上，放置于<head><style>...</style></head>之间。
  - 好处是你可以把所有的样式都放在一起，导入的样式表成为总样式表的一部分。
  - 但这种外部导入CSS 样式表也存在局限性，只有Internet Explorer浏览器才支持，在Netscape Navigator浏览器下不产生效果。
-

# 外部导入CSS 样式表HTML代码

---

```
<head>
<style type="text/css">
<!--
@import
    url(http://www.domain.com/mystyle.css);
h1 {color: blue}
-->
</style>
</head>
```

---

# 方法三：外部链接CSS样式表

---

- 最常用的方法
  - 就是先编写一个\*.css的文件(并非 HTML文件)，然后将这个css文件链接到所需的页面HTML文件上。
  - 可以把网站上所有页面都链接到同一个css文件，一旦这个css文件修改，那么所有的页面风格也随之改变。
-

# 外部链接CSS样式表HTML代码:

---

```
<head>
```

```
<link rel="stylesheet" type="text/css"  
      href="http://www.domain.com/mystyle.css"  
    ></head>
```

---

## mystyle.css的内容

---

- `body { line-height: 130pt}`  
`H1,H2,H3,H4,H5,H6 { color: red; text-decoration: underline; font-family: " 黑体 " }`  
`b { font-style: italic; color: #FF3333; text-decoration: underline }`

- 以上方法一、二、三都可在Dreamweaver MX 2004中找到相应设定好的应用方式。

- 方法一： 内部（或嵌入式）CSS 样式表
- 方法二： 外部导入CSS 样式表
- 方法三： 外部链接CSS 样式表

# 方法四：行间写入式CSS 样式表

---

- 就是把样式写入HTML代码需要格式化的所在行的位置。这种方法已很少使用，只是在特定需要定义的某行中才应用上。在Dreamweaver MX 2004中已没有与之相应设定好的应用方式，若要使用这方法，只有手动写代码了。
- 行间写入式CSS 样式表可放置于以下二地方：
- ·<span> . . . </span>标签之间



# 行间写入式CSS 样式表可放置于以下二地方：

- **<span> . . . </span>**标签之间

```
<span style="background: yellow">World  
Enterprises</span>
```

- 现存的**HTML**标签内部

```
< h1 style="color: blue; font-size: 30pt">Join the  
club! </h1>
```

```
<ul style="font-family: verdana">
```

```
<li>Events
```

```
<li>Meetings
```

```
</ul>
```

## 四、创建CLASS

- 以上所介绍的样式表规则，我们发现在一个HTML页中，每种HTML标志只能为其定义一种风格，然而我们有时候会碰到其他一些情况，比如：同一个HTML标志需要呈现不同的风格；有若干个不同的HTML标志采用相同的样式规则。

样式表提供了解决方法，创建类（CLASS）可以创建同一个HTML标志的多种风格。

# CLASS

---

- 其语法为：
- 标签.类名{属性：值；属性：值；.....属性：值}
- 引用方法是：
- < 标签 CLASS="类名">

- < html>
- < head>
- < title>This is a sample< /title>
- < style type="text/css"> < !-- p.first  
{ text-indent: 0.5in} p.second { text-indent: 1.0in} --> < /style>
- < /head>
- < body bgcolor="#FFFFFF"> .....
- < p class="first">这个段落将缩进0.5in< /p>  
< p class="second">这个段落将比上面缩进一倍距离< /p> .....
- < /body>
- < /html>

- 此外，可以直接定义CLASS，应用于HTML页面中的各个标记。其语法只是比上面的少了一个标志：
- .类名 {标签属性：值； 标签属性：值； .....标签属性：值}
- `< style type="text/css"> < !-- .main01 { font-size: 10pt; color: blue} --> < /style>`
- 该“.main01”类可用于任何**HTML**标签

- 该CLASS规定了字符的大小和颜色，当HTML文档中任何地方，无论是段落< p>、表格< table>，需要其字体大小为10pt颜色为红色时，就可以引用这个CLASS。引用的方法和上面一样：

## ■ < HTML标签 CLASS=“类名”>

- 例如要设置某单元格中的字符大小为main10所定义的风格，则可写为：

```
< td class="main01">
```

- 设置某一段落字符风格为main10，可写为：

```
< p class="main01">
```

我们可以在同一HTML文档中多次引用这个类，引用该类的地方都将呈现同一种风格。

# ID

- #id名 {标签属性: 值; 标签属性: 值; .....标签属性: 值}
- 
- ```
< style type="text/css">  
  <!--  
    #001 { font-size: 10pt; color: red}  
    #002 { font-size: 12pt; color: blue}  
  -->  
< /style>
```
- 引用的方法也相同: 

```
< 标签 ID="ID名">
```

# 五、边框属性

---

- 1、边距属性
- 2、填充属性
- 3、边界属性
- 4、尺寸属性
- 5、浮动和清除属性



## 五、边框属性--1、边距属性

---

- - 左边距 (margin-left)
  - 右边距 (margin-right)
  - 上边距 (margin-top)
  - 下边距 (margin-bottom)
  - 边距 (margin)
- 边距属性定义HTML页面中方框四边和其他元素之间的空白距离，每一个margin-side属性只影响到一条边，而margin则提供了一个同时设置四个边距的机会。

## 五、边框属性--2、填充属性

- ·左填充 (padding-left)
- 右填充 (padding-right)
- 上填充 (padding-top)
- 下填充 (padding-bottom)
- 填充 (padding)

■ 填充属性设置边框和内部元素的距离。它和边距属性十分相似，每个padding-side定义一个边框距离，padding则同时定义四个距离。

## 五、边框属性--3、边界属性

- ·上边界宽 (border-top-width)
- 右边界宽 (border-right-width)
- 下边界宽 (border-bottom-width)
- 左边界宽 (border-left-width)
- 边界宽 (border-width)

这几个属性定义边界宽度，用thin、medium和thick分别表示细、中等和粗，或者指定具体的数值来定义边界的宽度。

## 五、边框属性--3、边界属性

### ■ ·边界颜色 (border-color)

这个属性定义边界的颜色，可以用16种颜色名或RGB值来设置。如果指定一种颜色，则表示四个边框是一种颜色，指定两种颜色，则定义顺序为上下、左右；指定三种颜色，顺序为上、左右、下；指定四种颜色，顺序则为上、右、下、左。16种颜色名分别为：aqua, black, blue, fuchsia, gray, green, lime, maroon, navy, olive, purple, red, silver, teal, white, yellow。RGB值则表示为#RRGGBB或r%, g%, b%。



# 五、边框属性--3、边界属性

- ·边界样式 (border-style)

该属性用以定义边框的风格呈现式样。共有九种。

- none — 不显示边框，为缺省值

- dotted — 点线

- dashed — 虚线

- solid — 实线

- double — 双线

- groove — 凹线

- ridge — 凸线

- inset — 使整个方框凸起

- outset — 使整个方框凹陷

- 上边界 (border-top)

- 右边界 (border-right)

- 下边界 (border-bottom)

- 左边界 (border-left)

## 五、边框属性--3、边界属性

### ■ .边界 (border)

这几个是简写属性，它集合了上面几种属性的特点，可使用其给边框一次性定义宽度、式样和颜色。各属性值之间用空格隔开。

## 五、边框属性--4、尺寸属性

---

- 定义方框的宽度width和高度height。
- 例：`DIV.sample {width: 300px;  
height: 200px }`

## 五、边框属性--5、浮动和清除属性

- 浮动属性是指某一对象在母体元素内的浮动位置，清除属性和浮动属性一起使用，则表示同一母体内的其他对象可以流到浮动对象的旁边。

例：

- `.float1 { float: left; clear: right }`

假如这个样式定义在表格单元中的某个对象，则该对象将浮动在单元中的左边，而同一单元格中的其他对象在它右边的周围部分。



## 六、字体属性

- 字体属性共分五种：
- 字体家族（font-family）、
- 字体风格（font-style）、
- 字体变体（font-variant）、
- 字体黑度（font-weight）、
- 字体尺寸（font-size），
- 此外font属性是一个总体属性，可一一指定以上各种属性和属性值。

# 1、字体家族 (font-family)

- 通过font-family指定字体类型，其语法为：
- 标记 {font-family: 字体类型, 字体类型, 字体类型}
- 例: p {font-family: "Times New Roman", arial, serif }
- 假如字体类型之间有空格，就象Times New Roman，必须用引号将之括起，中文字体也须用括号括起，列出多种类型，以防浏览用户无前面的字体类型时，可以用后一种字体类型将之取代。

## 2、字体风格 (font-style)

---

- 属性值: normal (普通)、默认值
- italic (斜体)、
- oblique (倾斜)
- 例: h1 {font-style: italic}

### 3、字体变体 (font-variant)

---

- 属性值: normal (普通) 和 small-caps (小型大写字母), 缺省值为 normal。
- 字体变体属性是让字体以小型大定方式来显示。小型大写方式看起来是一般大写字母的75%—80%左右。
- 该变体在某些需要特殊表现的标题中比较有用。例:
- `H1 {font-variant: small-caps}`

## 4、字体加粗 (font-weight)

---

- 属性值: normal、bold、lighter、bolder、以及数值100—900。
  - 该属性用以指定字体粗细的显示，并不是所有字体都可以显示这些指定的加粗程度，因此有些情况下这些指定值会被替代，例如100到300被lighter替代，600到900则替换bolder，反之亦然。例：
    - `h1 < font-weight: lighter>`
    - `p < font-weight: 800>`
-

## 5、字体尺寸（font-size）

- 字体尺寸可设置绝对尺寸、相对尺寸、长度、百分比。
- - 绝对尺寸分为xx-small, x-small, small, medium, large, x-large, xx-large七种，它们以各种字体的medium尺寸成比例缩放，每一级1.5倍。
  - 相对尺寸有larger和smaller两种，根据字体原来的大小来决定缩放以后的大小。
  - 长度是以pt, points, cm, mm, inch等度量单位用具体数值来指定字体的尺寸大小。
  - 百分比是把字体设置成原来尺寸的百分比值，可以任意指定数值。
- 例如：

```
h1 {font-size: 150%}  
td {font-size: 10pt}  
p {font-size: smaller}
```

## 6、字体（font）

---

- 该属性使你可以把上述字体的各种属性合并到一行代码中指定，也即提供了一个实现字体属性的快捷方式，各种属性值之间用空格隔开。
- 例如：
- `td {font: italic bold 12pt "方正楷体简体", "宋体"}`

# 七、文本属性

---

## ■ 1、文本对齐text-align

- 属性值：left（左），
- right（右），
- center（居中），
- justify（两端对齐）。

- 例：h1 {text-align: center}
-



## ■ 2、文本缩进text-indent

---

- 该属性有效的控制了文本段落第一行的缩进，其值可以指定，是长度或段落宽度的百分比。

■ 例：

■ **p {text-indent: 1.0 in}**

### ■ 3、行高 line-height

---

- 该属性设置行与行之间的间距，其值可以为数值、长度或百分比，百分比以行高为基础。

■ 例：

■ **body {line-height: 120%}**

## ■ 4、字间距 letter-spacing

---

- 字间距设置字与字之间的距离，同样可以用数值、长度或百分比来指定，百分比以字符大小为基础。例：  
**body {letter-spacing: 0.5em}**
- 顺便提一下，文本属性中还有一个单词间距（**word-spacing**），设置每个单词之间的距离，对于中文页面来讲，可能很少用到。

## ■ 5、文本装饰 text-decoration

- 属性值：**underline**（下划线），
- **overline**（底线），
- **line-through**（线穿过），
- **blink**（闪烁）。

■ 例：

■ **h3 {text-decoration: underline}**

## ■ 6、垂直对齐 vertical-align

- 属性值：baseline（基准线），
- super（上标），
- sub（下标），
- top（顶部），
- text-top（文本顶部），
- middle（中），
- bottom（底部），
- text-bottom（文本底部）和百分比。

### ■ 例：

- `< p>平方值：3< font style="vertical-align:super">2< /font>< /p>`

## ■ 7、文本变换text-transform

---

- 属性值：capitalize（首字母大写），
- uppercase（大写），
- lowercase（小写）
- none（无）缺省值
- 例：
- `p {text-transform: capitalize}`

# 八、颜色与背景属性

- 1、颜色属性: color
- 属性值: 16种颜色名、
- 数值 (#RRGGBB或是r%, g%, b%) 。
- 在< font>、< p>、< body>、< table>及其单元元素、标题等对象都可以用到color属性。
- 例:
- ```
< style type="text/css">  
a { color: green}  
a:hover { color: red}  
< /style>
```

## ■ 2、背景属性

- background-color
- background-image
- background-repeat

属性值：no-repeat（无重复），缺省值  
repeat（重复），  
repeat-x（x方向重复），  
repeat-y（y方向重复），

指定背景图像的显示方式。该属性需与background-image和background-position组合使用。



## ■ background-attachment

- 属性值: scroll 随对象一起滚动, 缺省值
- fixed 固定

- 该属性指定对象的背景图像是否与对象一起滚动, 或是固定在页面上的某一个位置。这个属性与background-image组合使用。

## ■ background-position

- 属性值：
- 垂直位置vertical，指定top, center, bottom和具体数值、百分比；
- 水平位置horizontal，指定left, center, right和具体数值、百分比。定义背景图像的绝对或相对位置显示。

## ■ Background

- 这是一个简写属性，可以把上述所有背景属性归纳到一行代码中定义。 这里是一个较完整的例子：

```
body { background-image:
        url (abcsite/images/002.gif);
        background-repeat: no-repeat;
        background-position: 20px 50px;
        background-attachment: fixed}
```
- 用background属性简写为：

```
body { background: url (abcsite
        /images/002.gif) no-repeat 20px 50px fixed}
```

# 九、分类属性

---

- 分类属性控制了浏览器的显示方式，我们知道在HTML语言中，对于空格和换行的处理是缺陷所在，分类属性的到来使这些特性的控制成为可能。
  - 分类属性包括显示属性、空白属性和列表属性。
-

## ■ 1、显示属性display

- 通过显示属性控制HTML页面中被定义对象在浏览器中的显示方法。这个属性有四个值：

- - 块（block）——在该对象前后自动增加分行符

- 例：`text< img style="display: block" src="photo.jpg" >text`

- 图像前后的文字将与图像分行显示。



- ·内联（inline）——这个对象前后的元素与之在一行显示
- 这个属性值正好相反，当需要对象与前一元素强制性在同一行时使用该值。
- 
- ·列表项（list-item）——加一个列表项标记，其余同块
- 该属性值与< LI>、< DD>、< OL>、< UL>等列表标记一起使用。
- 
- ·无（none）——对象不显示
- 该属性值在某种情况下很有用，比如事先隐藏正确答案，直到经过某种操作以后才显示。

## ■ 2、空白属性 white space

- 制表符、空格和换行泛称空白，HTML页面显示时，通常根据浏览器窗口的大小自动折行，多余的空白符忽略成一个空格。< PRE>标记可以控制文本在浏览器中的显示，但该标记的限制性很大，只对一些非比例字体才有效，并且每次用到这个命令框起所有文本也相当麻烦，空白属性提供了这种方便。

## ■ 空白属性有三个值：



·普通（normal）——该属性值保持浏览器自动换行的特性。

■ ·保持原样（pre）——这个值与使用< PRE>标记一样，告诉浏览器按照文本输入原样显示，并不自动折行。



·不折行（nowrap）——这个值把文本中的回车、换行和多个空格都显示成一个空格，换行只由< BR>标记来控制。





- 
- 1) 列表一样式—类型 `list-style-type`
  - 该属性用以改变HTML中列表项的外形显示。
  - 它有下列值：`disc`（圆盘）、
  - `circle`（圆圈）、
  - `square`（正方）、
  - `decimal`（小数点）、
  - `lower-roman`（小写罗马数字）、
  - `upper-roman`（大写罗马数字）。

## ■ 2) 列表一样式—图像 list-style-image

- 这个属性用来指定图像作为列表标记，缺省值为无，当它被设置时，list-style-type属性不显示，可使用绝对或相对位置指定图像。



■ 例：

```
< ul style=list-style-  
image:url(http://www.163.com/abc.gif)">
```

### ■ 3) 列表一样式—位置 list-style-position

- 该属性决定了列表项标记放在列表项的什么位置，  
值：**inside**（内部）列表下一行不缩进显示  
**outside**（外部）缺省值，列表下一行缩进显示

显示效果如下：

- 内部  
内部显示
- 外部  
外部显示

## ■ 4) 列表一样式 list-style

- 该属性提供了上述列表属性的快捷方法，其书写格式为：
- 元素 {list-style:[type]  
[position] [url] }
- 括号表示任选一种。

*THE END*  
*THANK YOU!*

