# 第三章 css基础

本节所讲内容：

3.1 css概述

3.2 css基本语法及页面引用

3.3 css文本设置

3.4 css颜色表示法

3.5 css选择器

3.6 CSS盒子模型

## 3.1 css概述

为了让网页元素的样式更加丰富，也为了让网页的内容和样式能拆分开，CSS由此思想而诞生，CSS是 Cascading Style Sheets 的首字母缩写，意思是层叠样式表。

Cascading [kæˈskeɪdɪŋ]：层叠

Sheet:[ʃiːt]:床单



css 优点：

1.表现和内容相分离将设计部分剥离出来放在一个独立样式文件中，HTML文件中只存放文本信息，这样的页面对搜索引擎更加友好。

2.提高页面浏览速度对于同一个页面的视觉效果，采用DIV+CSS重构的页面容量要比TABLE编码的页面文件容量小的多，前者一般只有后者的1/2大小，浏览器就不用去编译大量冗长的标签。

3.易于维护和改版，只要简单的修改几个CSS文件就可以重新设计整个网站的页面。

## 3.2 css基本语法及页面引用

### 3.2.1 css基本语法

css的定义方法是：

选择器 { 属性:值; 属性:值; 属性:值;}

选择器是将样式和页面元素关联起来的名称，属性是希望设置的样式属性每个属性有一个或多个值。代码示例：

div{ width:100px; height:100px; color:red }

### 3.2.2 css页面引入方法：

1、内联式：通过标签的style属性，在标签上直接写样式。

<div style="width:100px; height:100px; color:red ">......</div>

特点：仅作用于本标签

2、嵌入式：通过style标签，在网页上创建嵌入的样式表。

<style type="text/css">

div{ width:100px; height:100px; color:red }

......

</style>

特点：作用于当前整个页面

3、外联式：通过link标签，链接到外部样式表到页面中。

<link rel="stylesheet" type="text/css" href="css/main.css">

特点：作用于整个网站，工作常用这种方式。

注意:他们的优先级：内联式 > 嵌入式 > 外联式

当样式冲突时，就是采用就近原则，是指css属性离被修饰的内容最近的为主。若没有样式冲突则采用叠加效果。

样式冲突: 就近原则 优先级：内联式 > 嵌入式 > 外联式

没有样式冲突采用叠加效果。

## 3.3 css文本设置

常用的应用文本的css样式：

color 设置文字的颜色，如： color:red;

font-size 设置文字的大小，如：font-size:12px;

font-family 设置文字的字体，如：font-family:'微软雅黑'; windows下存放字体的路径为：C:\Windows\Fonts

font-style 设置字体是否倾斜，如：font-style:'normal'; 设置不倾斜，font-style:'italic';设置文字倾斜 'normal' [ˈnɔːml]:正常。

font-weight 设置文字是否加粗，如：font-weight:bold; 设置加粗（相当于数字值700） font-weight:normal 设置不加粗（相当于数字值400）。

font 同时设置文字的几个属性，写的顺序有兼容问题，建议按照如下顺序写： font：是否加粗 字号/行高 字体；如： font:normal 12px/36px '微软雅黑';

line-height 设置文字的行高，如：line-height:24px;

text-decoration 设置文字的下划线，如：text-decoration:none; 将文字下划线去掉，underline下画线，line-through贯穿线

text-indent 设置文字首行缩进，如：text-indent:24px; 设置文字首行缩进24px

text-align 设置文字水平对齐方式，如text-align:center 设置文字水平居中，一般有：left center right

text-shadow: 文本的文字是否有阴影及模糊效果 text-shadow: h-shadow v-shadow blur color;

h-shadow：必需。水平阴影的位置。允许负值。

v-shadow：必需。垂直阴影的位置。允许负值。

blur：可选。模糊的距离。

Color：可选。阴影的颜色

<!DOCTYPE html>  
<html>  
<head>  
 <meta charset="utf-8">  
 <title>文本属性</title>  
 <link rel="stylesheet" type="text/css" href="./style.css">  
</head>  
<body>  
<p>  
 For老师讲解文本属性-段落标签  
</p>  
<a href="">下划线的去除</a>  
<ul>  
 <li>文字的模糊</li>  
</ul>  
</body>  
</html>

## 创建一个style.css样式

p{  
 color:red;  
 text-indent:30px;  
 text-align:center;  
 text-decoration:line-through;  
 */\*每个单词的间距\*/* word-spacing:30px;  
 */\*每个字母的间距\*/* letter-spacing:2px;  
 */\*direction:ltr;\*/* width:200px;  
 height:30px;  
 background:green;  
 */\*处理空白不换行\*/* white-space:nowrap;  
 */\*多余的文字以...结束\*/* text-overflow:ellipsis;  
 */\*当前的元素溢出元素框的时候\*/* overflow:hidden;  
}  
a{  
 text-decoration:none;  
}  
ul li{  
 text-shadow:0px 0px 3px #ff0000;  
 font-size:35px;  
 line-height:20px;  
}

运行结果如下：



## 3.4 css颜色表示法

3.4.1.HSL颜色

通过对色调(H)、饱和度(S)、亮度(L)三个颜色通道的变化以及它们相互之间的叠加来得到各式各样的颜色。

H:Hue(色调)。0(或360)表示红色，120表示绿色，240表示蓝色，也可取其他数值来指定颜色。取值为：0 - 360。

S:Saturation(饱和度)。取值为：0.0% - 100.0%。

L:Lightness(亮度)。取值为：0.0% - 100.0%。

例： color: hsl(240,100%,50%);

3.4.2.HSLA颜色

色调(H)、饱和度(S)、亮度(L)、透明度(A)。

H:Hue(色调)，0(或360)表示红色，120表示绿色，240表示蓝色，也可取其他数值来指定颜色。取值为0 - 360 。

S:Saturation(饱和度)。取值为：0.0% - 100.0%。

L:Lightness(亮度)。取值为：0.0% - 100.0%。

A:Alpha透明度。取值0~1之间。

此色彩模式与HSL相同，只是在HSL模式上新增了Alpha透明度。

例：color:hsla(0,100%,50%,0.2);

3.4.3.RGB颜色

红(R)、绿(G)、蓝(B)三个颜色通道的变化

R:红色值。正整数 | 百分数

G:绿色值。正整数 | 百分数

B:蓝色值。正整数 | 百分数

以上三个参数，正整数值得取值范围为：0-255，百分比数值的取值范围为：0.0%-100.0%，超出范围的数值将被截至其最接近的取值极限。

如：rgb(300,0,0)会被解析rgb(255,0,0)

正整数值255对应百分比数值100%，如：rgb(255,255,255) = rgb(100%,100%,100%) = #FFFFFF = #FFF，RGB色彩是通过对红(R)、绿(G)、蓝(B)三个颜色通道的变化和它们相互之间的叠加来得到各式各样的颜色的。

例：color: rgba(200,100,0);

3.4.4 RGBA颜色

:红(R)、绿(G)、蓝(B)、透明度(A)

R:红色值。正整数 | 百分数

G:绿色值。正整数 | 百分数

B:蓝色值。正整数 | 百分数

A:Alpha透明度。取值0~1之间

此色彩模式与[RGB](http://www.css88.com/book/css/values/color/rgb.htm)相同，只是在[RGB](http://www.css88.com/book/css/values/color/rgb.htm)模式上新增了Alpha透明度。

例： color: rgba(0,0,0,0.5)

图片透明度的设置

3.4.5.十六进制色彩的表示方式

颜色构成：红绿蓝

取值范围：0-9 a-f

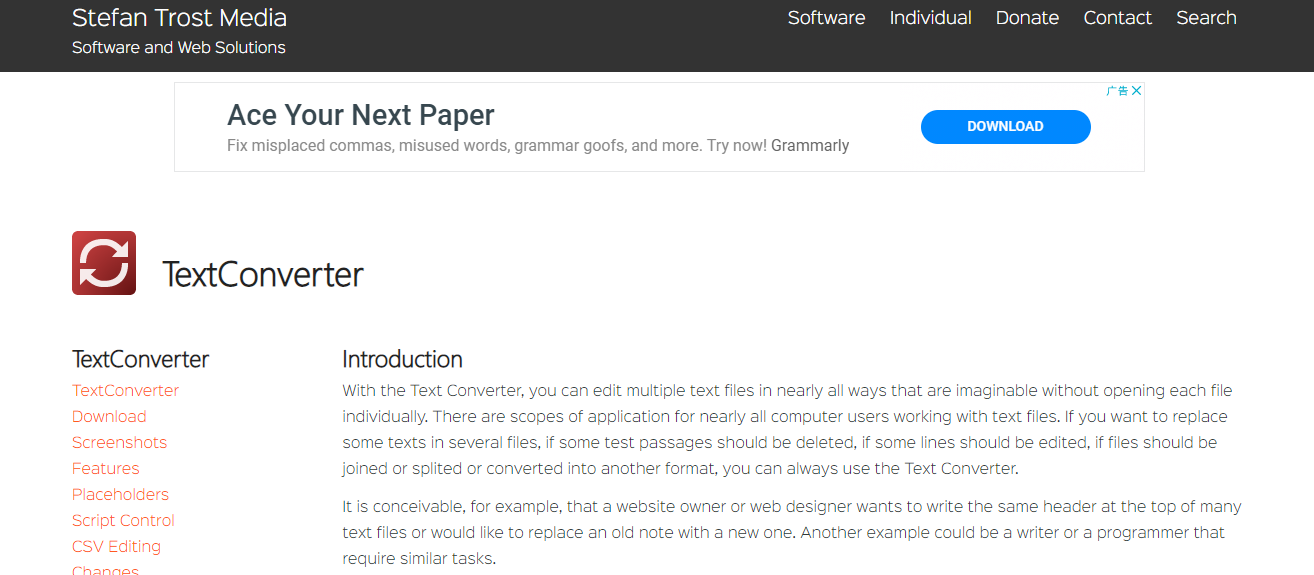
3.4.6.英文单词

颜色构成：red blue green

<!DOCTYPE html>  
<html>  
<head>  
 <meta charset="utf-8">  
 <title>color属性</title>  
 <style type="text/css">  
 p{  
 */\*  
 color:red;  
 green blue yellow pink  
 color:hsl(120,50%,50%);  
 color:hsla(120,50%,50%,0.2);  
 color:rgb(0,255,0);  
 color:rgba(0,255,0,0.1);  
 color:#000000;  
 \*/* color:#c81623;  
 }  
 </style>  
</head>  
<body>  
<p>  
 段落文字颜色  
</p>  
</body>  
</html>

配置颜色软件下载网站：<https://www.sttmedia.com/download=PipetteWin>

线上颜色对比：<https://www.baidu.com/search/hi/hi_css.html>



## 3.5 css选择器

### 3.5.1 标签选择器

标签选择器，此种选择器影响范围大，建议尽量应用在层级选择器中。  
举例：

\*{margin:0;padding:0}

div{color:red}

<div>....</div> <!-- 对应以上两条样式 -->

<div class="box">....</div> <!-- 对应以上两条样式 -->

### 3.5.2 id选择器

通过id名来选择元素，元素的id名称不能重复，所以一个样式设置项只能对应于页面上一个元素，不能复用，id名一般给程序使用，所以不推荐使用id作为选择器。  
举例：

#box{color:red}

<div id="box">....</div> <!-- 对应以上一条样式，其它元素不允许应用此样式 -->

### 3.5.3 类选择器

通过类名来选择元素，一个类可应用于多个元素，一个元素上也可以使用多个类，应用灵活，可复用，是css中应用最多的一种选择器。  
举例：

.red{color:red}

.big{font-size:20px}

.mt10{margin-top:10px}

<div class="red">....</div>

<h1 class="red big mt10">....</h1>

<p class="red mt10">....</p>

### 3.5.4 层级选择器

主要应用在选择父元素下的子元素，或者子元素下面的子元素，可与标签元素结合使用，减少命名，同时也可以通过层级，防止命名冲突。  
举例：

.box span{color:red}

.box .red{color:pink}

.red{color:red}

<div class="box">

<span>....</span>

<a href="#" class="red">....</a>

</div>

<h3 class="red">....</h3>

### 3.5.5 组选择器

多个选择器，如果有同样的样式设置，可以使用组选择器。  
举例：

.box1,.box2,.box3{width:100px;height:100px}

.box1{background:red}

.box2{background:pink}

.box2{background:gold}

<div class="box1">....</div>

<div class="box2">....</div>

<div class="box3">....</div>

### 3.5.6 伪类及伪元素选择器

常用的伪类选择器有hover，active等表示鼠标悬浮在元素上时的状态，伪元素选择器很多，它们可以通过样式在元素中实现不同的效果。

a:link {color: #FF0000} /\* 未访问的链接 \*/

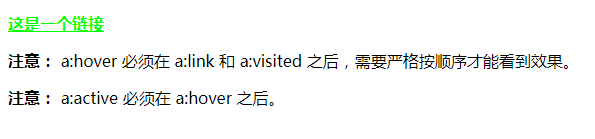
a:visited {color:#00FF00;} /\* 已访问的链接 \*/

a:hover {color: #FF00FF} /\* 鼠标移动到链接上 \*/

a:active {color: #0000FF} /\* 选定的链接 \*/

<!DOCTYPE html>  
<html>  
<head>  
 <meta charset="utf-8">  
 <title>FOR的伪类</title>  
 <style>  
 a:link {color:#000000;} */\* 未访问链接\*/* a:visited {color:#00FF00;} */\* 已访问链接 \*/* a:hover {color:#FF00FF;} */\* 鼠标移动到链接上 \*/* a:active {color:#0000FF;} */\* 鼠标点击时 \*/* </style>  
</head>  
<body>  
<p><b><a href="1.html" target="\_blank">这是一个链接</a></b></p>  
<p><b>注意：</b> a:hover 必须在 a:link 和 a:visited 之后，需要严格按顺序才能看到效果。</p>  
<p><b>注意：</b> a:active 必须在 a:hover 之后。</p>  
</body>  
</html>

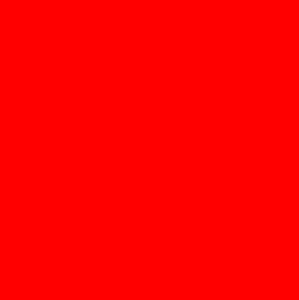
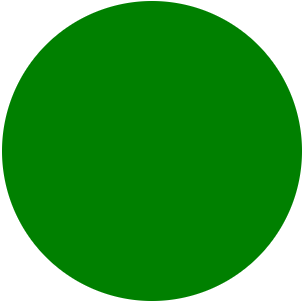
运行结果如下：



<!DOCTYPE html>  
<html>  
<head>  
 <meta charset="utf-8">  
 <title>文本属性</title>  
 <link rel="stylesheet" href="style.css">  
</head>  
<body>  
<div class="box1"></div>  
<div class="box2">....</div>  
<div class="box3">....</div>  
</body>  
</html>

Style.css

.box1{  
 width: 300px;  
 height: 300px;  
 margin: 0 auto;  
 background: red;  
 margin-top: 50px;  
 transition: all 0.6s;  
 */\*设置鼠标小手\*/* cursor: pointer;  
}  
.box1:hover{  
 width: 300px;  
 height: 300px;  
 margin: 0 auto;  
 background: green;  
 margin-top: 50px;  
 */\*添加圆角边框\*/* border-radius: 50%;  
 */\*变化时间\*/  
 /\*transition: all 0.6s;\*/*}  
.box2:before{content:'行首文字';}  
.box3:after{content:'行尾文字';}

注意:自己定义了元素属性，就听自己的，没定义就去继承父集元素的属性

注意:相同的选择器相同的属性 后面的代码会覆盖前面的代码

注意:css样式优先级问题

CSS三大特性— 继承、优先级和层叠。

优先级

行内样式>ID选择器 > 类选择器 > 标签 > 通配符 > 继承 > 浏览器默认属性

继承：即子类元素继承父类的样式

自己定义了元素属性，就听自己的，没定义就去继承父集元素的属性

层叠

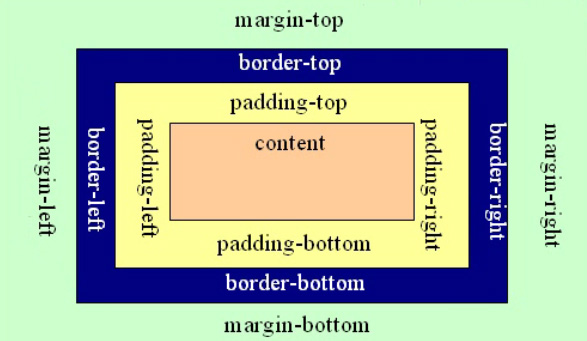
相同的选择器相同的属性 后面的代码会覆盖前面的代码

注意：

优先级相同时，则采用就近原则，选择最后出现的样式; 继承得来的属性，其优先级最低;

## 3.6 CSS盒子模型

****盒子模型解释****  
元素在页面中显示成一个方块，类似一个盒子，CSS盒子模型就是使用现实中盒子来做比喻，帮助我们设置元素对应的样式。盒子模型示意图如下：



把元素叫做盒子，设置对应的样式分别为：盒子的边框(border)、盒子内的内容和边框之间的间距(padding)、盒子与盒子之间的间距(margin)。

****设置边框****  
设置一边的边框，比如顶部边框，可以按如下设置：

border-top-color:red; /\* 设置顶部边框颜色为红色 \*/

border-top-width:10px; /\* 设置顶部边框粗细为10px \*/

border-top-style:solid; /\* 设置顶部边框的线性为实线，常用的有：solid(实线) dashed(虚线) dotted(点线); \*/

上面三句可以简写成一句：

border-top:10px solid red;

solid [ˈsɒlɪd]：实体的

dashed [dæʃt]：急奔

dotted [ˈdɑːtɪd]：有斑点的

设置其它三个边的方法和上面一样，把上面的'top'换成'left'就是设置左边，换成'right'就是设置右边，换成'bottom'就是设置底边。

四个边如果设置一样，可以将四个边的设置合并成一句：

border:10px solid red;

****设置内间距padding****

设置盒子四边的内间距，可设置如下：

padding-top：20px; /\* 设置顶部内间距20px \*/

padding-left:30px; /\* 设置左边内间距30px \*/

padding-right:40px; /\* 设置右边内间距40px \*/

padding-bottom:50px; /\* 设置底部内间距50px \*/

上面的设置可以简写如下：

padding：20px 40px 50px 30px; /\* 四个值按照顺时针方向，分别设置的是 上 右 下 左

四个方向的内边距值。 \*/

padding后面还可以跟3个值，2个值和1个值，它们分别设置的项目如下：

padding：20px 40px 50px; /\* 设置顶部内边距为20px，左右内边距为40px，底部内边距为50px \*/

padding：20px 40px; /\* 设置上下内边距为20px，左右内边距为40px\*/

padding：20px; /\* 设置四边内边距为20px \*/

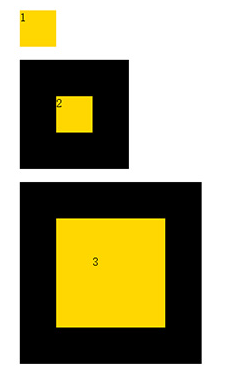
****设置外间距margin****

外边距的设置方法和padding的设置方法相同，将上面设置项中的'padding'换成'margin'就是外边距设置方法。

****盒子模型的尺寸****

按照下面代码制作页面：

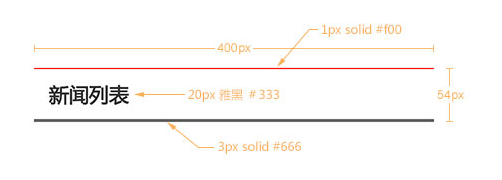
<!DOCTYPE html>  
<html lang="en">  
<head>  
 <meta charset="UTF-8">  
 <title>盒子的真实尺寸</title>  
 <style type="text/css">  
 .box01{width:50px;height:50px;background-color:gold;}  
 .box02{width:50px;height:50px;background-color:gold;border:50px  
 solid #000}  
 .box03{width:50px;height:50px;background-color:gold;border:50px  
 solid #000;padding: 50px;}  
 </style>  
</head>  
<body>  
<div class="box01">1</div>  
<br />  
<div class="box02">2</div>  
<br />  
<div class="box03">3</div>  
</body>  
</html>

页面显示效果如下：  


通过上面的页面得出结论：盒子的width和height设置的是盒子内容的宽和高，不是盒子本身的宽和高，盒子的真实尺寸计算公式如下：

1. 盒子宽度 = width + padding左右 + border左右
2. 盒子高度 = height + padding上下 + border上下

****思考题：****  
1.在布局中，如果我想增大内容和边框的距离，又不想改变盒子显示的尺寸，应该怎么做？

****课堂练习：****  
请制作图中所示的标题：  


<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Document</title>

<style>

div {

border-top: 1px solid #f00;

border-bottom: 3px solid #666;

width: 370px;

font:bold 20px/20px 雅黑;

padding: 15px;

margin: auto;

overflow: hidden;

}

</style>

</head>

<body>

<div>新闻列表</div>

</body>

</html>

****margin相关技巧：****  
1、设置元素水平居中： margin:xpx auto;  
2、margin负值让元素位移及边框合并

****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。解决方法如下：

1. 使用这种特性
2. 设置一边的外边距，一般设置margin-top
3. 将元素浮动或者定位

****margin-top 塌陷****

在两个盒子嵌套时候，内部的盒子设置的margin-top会加到外边的盒子上，导致内部的盒子margin-top设置失败，解决方法如下：

1. 外部盒子设置一个边框
2. 外部盒子设置 overflow:hidden
3. 使用伪元素类：

.clearfix:before{

content: '';

display:table;

}

总结：

3.1 css概述

3.2 css基本语法及页面引用

3.3 css文本设置

3.4 css颜色表示法

3.5 css选择器

3.6 css盒子模型