

兄弟连：吴俏祥

邮箱：764217451@qq.com

css 基本语法及页面引用

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 }
```

css 页面引入方法：

1、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">
```

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>
```

3、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>
```

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';
- 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
- font-weight 设置文字是否加粗，如： font-weight:bold; 设置加粗 font-weight:normal 设置不加粗

- **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
- **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

css 颜色表示法

css 颜色值主要有三种表示方法：

- 1、颜色名表示，比如：**red** 红色，**gold** 金色
- 2、**rgb** 表示，比如：**rgb(255,0,0)**表示红色
- 3、16 进制数值表示，比如：**#ff0000** 表示红色，这种可以简写成 **#f00**

css 选择器

常用的选择器有如下几种：

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 -->
```

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、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>
```

4.选择器权重

Id 选择器权重 > 类选择器 > 标签选择器

5、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>
```

6、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>
```

7、伪类及伪元素选择器

常用的伪类选择器有 **hover**，表示鼠标悬浮在元素上时的状态，伪元素选择器有 **before** 和 **after**，它们可以通过样式在元素中插入内容。

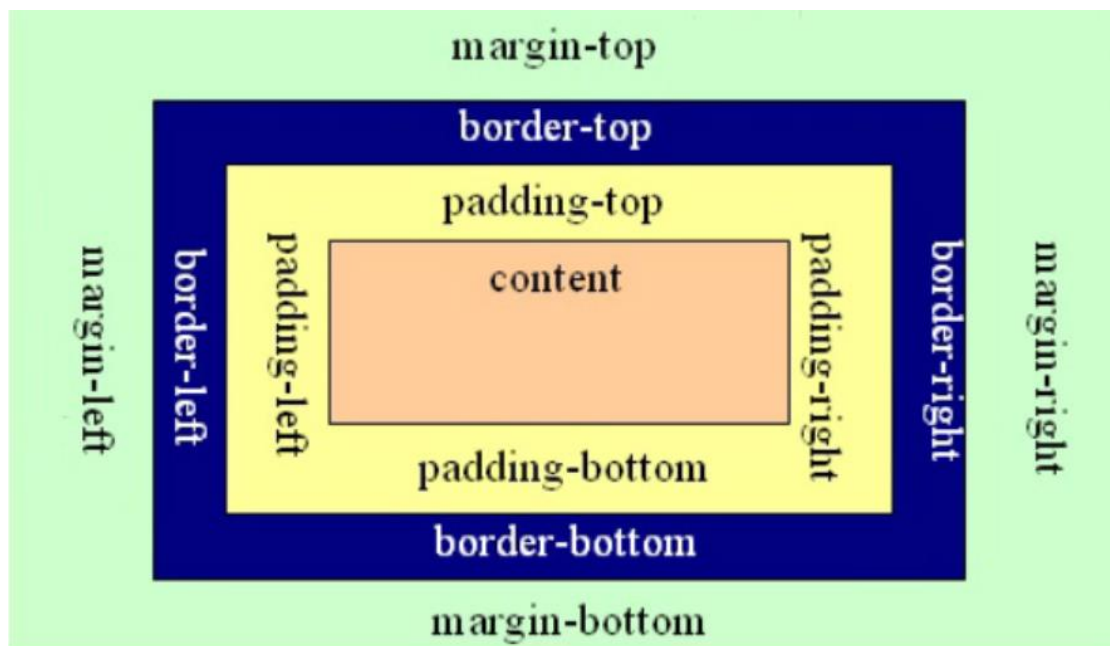
```
.box1:hover{color:red}
.box2:before{content:'行首文字';}
.box3:after{content:'行尾文字';}

<div class="box1">....</div>
<div class="box2">....</div>
<div class="box3">....</div>
```

CSS 盒子模型

1. 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)。

2. 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;
```

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;
```

3.设置内边距 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 */
```

4.设置外边距 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。

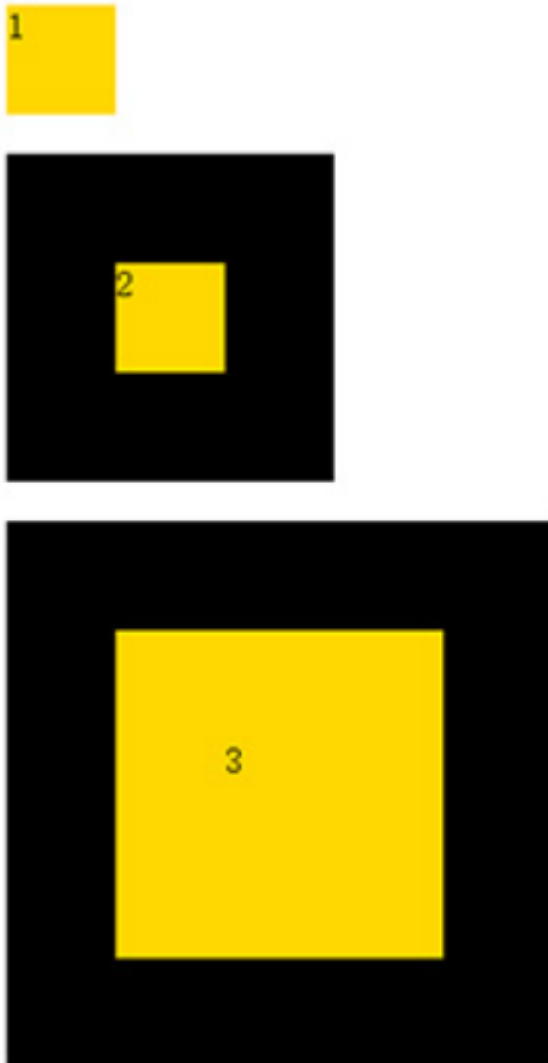
5.盒子模型的尺寸

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：

- 盒子宽度 = width + padding 左右 + border 左右
- 盒子高度 = height + padding 上下 + border 上下

6、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。

7.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。

8.盒子居中显示

9.行内标签垂直内外边距

CSS 元素溢出

当子元素的尺寸超过父元素的尺寸时，需要设置父元素显示溢出的子元素的方式，设置的方法是通过 `overflow` 属性来设置。

overflow 的设置项：

- 1、`visible` 默认值。内容不会被修剪，会呈现在元素框之外。
- 2、`hidden` 内容会被修剪，并且其余内容是不可见的，此属性还有清除浮动、清除 `margin-top` 塌陷的功能。
- 3、`scroll` 内容会被修剪，但是浏览器会显示滚动条以便查看其余的内容。
- 4、`auto` 如果内容被修剪，则浏览器会显示滚动条以便查看其余的内容。
- 5、`inherit` 规定应该从父元素继承 `overflow` 属性的值。

块元素、内联元素、内联块元素

元素就是标签，布局中常用的有三种标签，块元素、内联元素、内联块元素，了解这三种元素的特性，才能熟练的进行页面布局。

1.块元素

块元素，也可以称为行元素，布局中常用的标签如：**div**、**p**、**ul**、**li**、**h1~h6**、**dl**、**dt**、**dd** 等等都是块元素，它在布局中的行为：

- 支持全部的样式
- 如果没有设置宽度，默认的宽度为父级宽度 100%
- 盒子占据一行、即使设置了宽度

2.内联元素

内联元素，也可以称为行内元素，布局中常用的标签如：**a**、**span**、**em**、**b**、**strong**、**i** 等等都是内联元素，它们在布局中的行为：

- 支持部分样式（不支持宽、高、margin 上下、padding 上下）
- 宽高由内容决定
- 盒子并在一行
- 代码换行，盒子之间会产生间距
- 子元素是内联元素，父元素可以用 **text-align** 属性设置子元素水平对齐方式，用 **line-height** 属性值设置垂直对齐方式

解决内联元素间隙的方法

- 1、去掉内联元素之间的换行
- 2、将内联元素的父级设置 **font-size** 为 0，内联元素自身再设置 **font-size**

3.内联块元素

内联块元素，也叫行内块元素，是新增的元素类型，现有元素没有归于此类别的，**img** 和 **input** 元素的行为类似这种元素，但是也归类于内联元素，我们可以用 **display** 属性将块元素或者内联元素转化成这种元素。它们在布局中表现的行为：

- 支持全部样式
- 如果没有设置宽高，宽高由内容决定
- 盒子并在一行
- 代码换行，盒子会产生间距
- 子元素是内联块元素，父元素可以用 **text-align** 属性设置子元素水平对齐方式，用 **line-height** 属性值设置子元素垂直对齐方式

这三种元素，可以通过 **display** 属性来相互转化，不过实际开发中，块元素用得比较多，所以我们经常把内联元素转化为块元素，少量转化为内联块，而使用内联元素时，直接使用内联元素，而不用块元素转化了。

display 属性

display 属性是用来设置元素的类型及隐藏的，常用的属性有：

- 1、**none** 元素隐藏且不占位置
- 2、**block** 元素以块元素显示
- 3、**inline** 元素以内联元素显示
- 4、**inline-block** 元素以内联块元素显示

浮动

文档流

文档流，是指盒子按照 **html** 标签编写的顺序依次从上到下，从左到右排列，块元素占一行，行内元素在一行之内从左到右排列，先写的先排列，后写的排在后面，每个盒子都占据自己的位置。

浮动特性

- 1、浮动元素有左浮动(**float:left**)和右浮动(**float:right**)两种
- 2、浮动的元素会向左或向右浮动，碰到父元素边界、浮动元素、未浮动的元素才停下来
- 3、相邻浮动的块元素可以并在一行，超出父级宽度就换行
- 4、浮动让行内元素或块元素自动转化为行内块元素
- 5、浮动元素后面没有浮动的元素会占据浮动元素的位置，没有浮动的元素内的文字会避开浮动的元素，形成文字绕图的效果
- 6、父元素内整体浮动的元素无法撑开父元素，需要清除浮动
- 7、浮动元素之间没有垂直 **margin** 的合并

清除浮动

- 父级上增加属性 **overflow: hidden**
- 在最后一个子元素的后面加一个空的 **div**，给它样式属性 **clear:both**（不推荐）
- 使用成熟的清浮动样式类，**clearfix**
- `.clearfix:after,.clearfix:before{ content: "";display: table;}`
- `.clearfix:after{ clear:both;}`
- `.clearfix{zoom:1;}`

清除浮动的使用方法：

```
.con2{... overflow:hidden}  
或者  
<div class="con2 clearfix">
```

定位

属性：position

相对定位

绝对定位

固定定位

background 属性

属性解释

background 属性是 css 中应用比较多，且比较重要的一个属性，它是负责给盒子设置背景图片和背景颜色的，background 是一个复合属性，它可以分解成如下几个设置项：

- background-color 设置背景颜色
- background-image 设置背景图片地址
- background-repeat 设置背景图片如何重复平铺
- background-position 设置背景图片的位置
- background-attachment 设置背景图片是固定还是随着页面滚动条滚动

实际应用中，我们可以用 background 属性将上面所有的设置项放在一起，而且也建议这么做，这样做性能更高，而且兼容性更好，比如：“background: #00FF00 url(bgimage.gif) no-repeat left center fixed”，这里面的“#00ff00”是设置 background-color；“url(bgimage.gif)”是设置 background-image；“no-repeat”是设置 background-repeat；“left center”是设置 background-position；“fixed”是设置 background-attachment，各个设置项用空格隔开，有的设置项不写也是可以的，它会使用默认值。

举例：

下面这些例子使用下面这张图片做为背景图：



1、“background:url(bg.jpg)”，默认设置一个图片地址，图片会从盒子的左上角开始将盒子铺满。



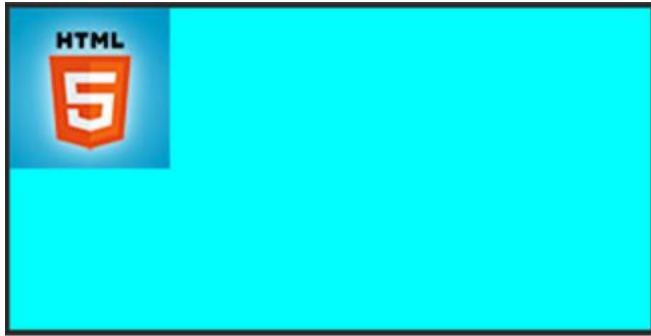
2、“background:cyan url(bg.jpg) repeat-x”，横向平铺盒子，盒子其他部分显示背景颜色“cyan”。



3、“background:cyan url(bg.jpg) repeat-y”，纵向平铺盒子，盒子其他部分显示背景颜色“cyan”。



4、“background:cyan url(bg.jpg) no-repeat”，背景不重复，背景和盒子左上角对齐，盒子其他部分显示背景颜色“cyan”。



5、“background:cyan url(bg.jpg) no-repeat left center”，背景不重复，背景和盒子左中对齐，盒子其他部分显示背景颜色“cyan”。



6、“background:cyan url(bg.jpg) no-repeat right center”，背景不重复，背景和盒子右中对齐，也就是背景图片的右边对齐盒子的右边，盒子其他部分显示背景颜色“cyan”。



相关代码：

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>test background</title>
  <style type="text/css">
    .backshow{
      width:320px;
```

```

        height:160px;
        border:3px solid #333;
        float:left;
        margin:10px;
    }
    .bg1{background:cyan url(bg.jpg);}
    .bg2{background:cyan url(bg.jpg) repeat-x;}
    .bg3{background:cyan url(bg.jpg) repeat-y;}
    .bg4{background:cyan url(bg.jpg) no-repeat;}
    .bg5{background:cyan url(bg.jpg) no-repeat left center;}
    .bg6{background:cyan url(bg.jpg) no-repeat right center;}
</style>
</head>
<body>
    <div class="backshow bg1"></div>
    <div class="backshow bg2"></div>
    <div class="backshow bg3"></div>
    <div class="backshow bg4"></div>
    <div class="backshow bg5"></div>
    <div class="backshow bg6"></div>
</body>
</html>

```

例子说明：

代码中使用到的背景图片，可以直接在页面上的背景图片上点右键另存下来，命名为：“bg.jpg”，并且和网页文件存在同一个目录下。

关于背景图片的 **background-position** 的设置，设置背景图片水平位置的有“left”、“center”、“right”，设置背景图片垂直位置的有“top”、“center”、“bottom”，水平和垂直的属性值两两组合，就可以把背景图设置到对齐盒子的四个角或者四个边的中心或者盒子的正中心位置。还可以设置具体的像素值来把背景图片精确地定位到盒子的某个位置，特别是对于背景图片尺寸大于盒子尺寸的这种情况，我们可以用数值定位，截取大图片的某一块做为盒子的背景。

比如说，我们想把下边的盒子用右边的图片作为背景，并且让背景显示图片中靠近底部的那朵花：

用上面中间那张图片作为左边那个比它尺寸小的盒子的背景，如果不设置 **background-position**，默认背景图的左上角会和盒子的左上角对齐，如果设置，可以用两个数值定位背景图，上面右边的实现效果设置为：“**background:url(location_bg.jpg) -110px -150px**”，第一个数值表示背景图相对于自己的左上角向左偏移 110px，负值向左，正值向右，第二个数值表示背景图相对于自己的左上角向上偏移 150px，负值向上，正值向下。

实现原理示意图：

对应代码：

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>test background</title>
  <style type="text/css">
    .backshow{
      width:320px;
      height:160px;
      border:3px solid #333;
      float:left;
      margin:10px;
    }
    .bg{width:94px;
      height:94px;
      border:3px solid #666;
      background:url(location_bg.jpg) -110px -150px;
    }
  </style>
</head>
<body>
  <div class="bg"></div>
</body>
</html>
```