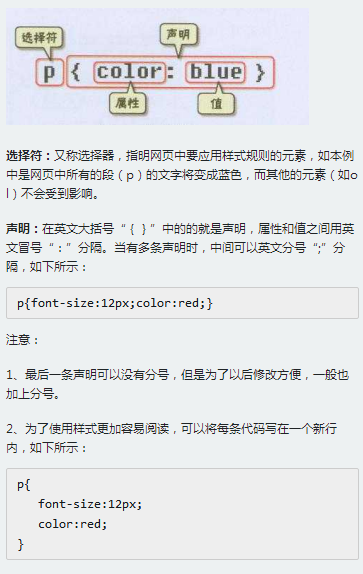
CSS代码语法



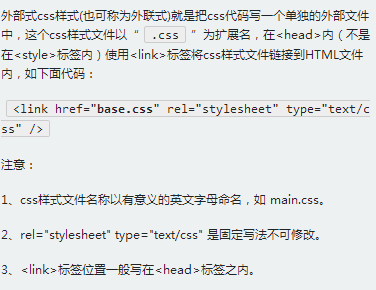
内联式css样式，直接写在现有的HTML标签中



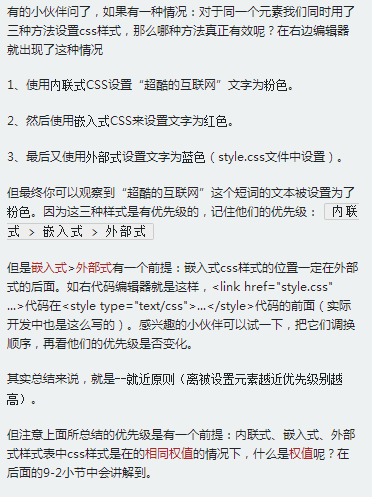
嵌入式css样式，写在当前的文件中



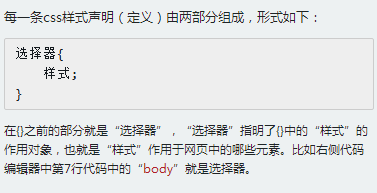
外部式css样式，写在单独的一个文件中



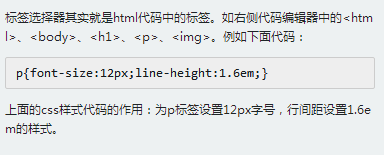
三种方法的优先级



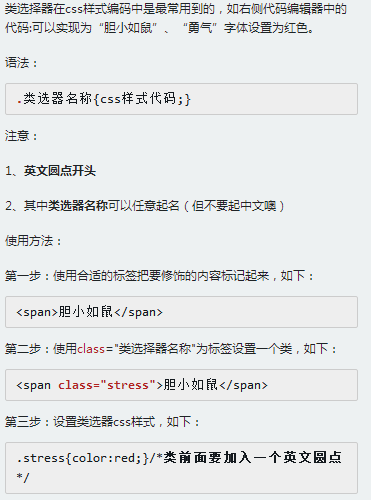
什么是选择器？



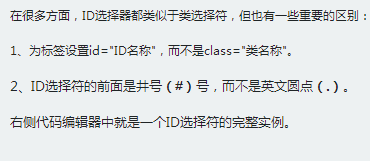
标签选择器



类选择器



ID选择器



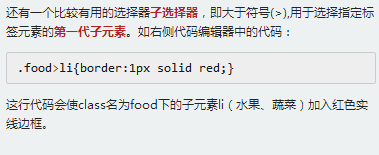
类和ID选择器的区别

相同点：可以应用于任何元素

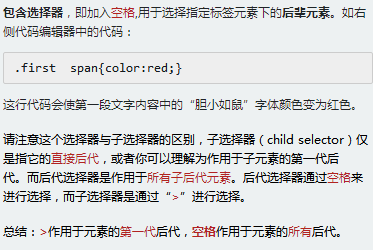
不同点：

1. ID选择器只能在文档中使用一次。与类选择器不同，在一个HTML文档中，ID选择器只能使用一次，而且仅一次。而类选择器可以使用多次。
2. 可以使用类选择器词列表方法为一个元素同时设置多个样式。我们可以为一个元素同时设多个样式，但只可以用类选择器的方法实现，ID选择器是不可以的（不能使用 ID 词列表）。

子选择器



包含(后代)选择器

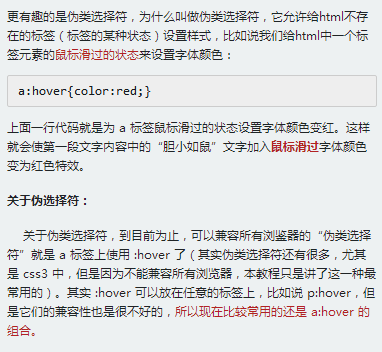


通用选择器

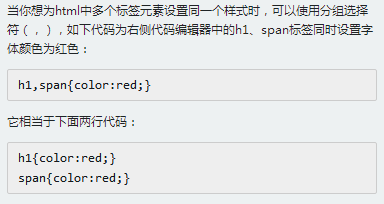
通用选择器是功能最强大的选择器，它使用一个（\*）号指定，它的作用是匹配html中所有标签元素，如下使用下面代码使用html中任意标签元素字体颜色全部设置为红色：

\* {color:red;}

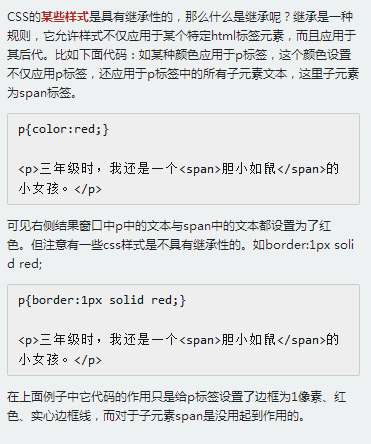
伪类选择符



分组选择符



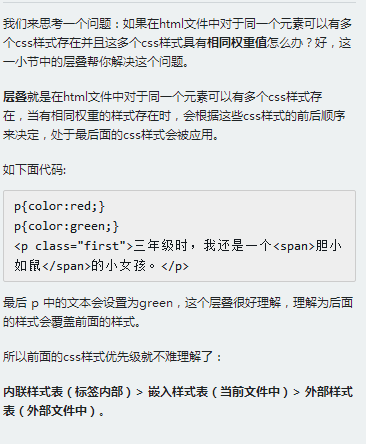
继承



特殊性



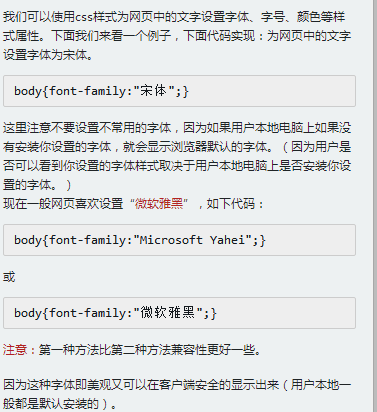
层叠



重要性



文字排版--字体



文字排版--字号、颜色

可以使用下面代码设置网页中文字的字号为12像素，并把字体颜色设置为#666(灰色)：

body{font-size:12px;color:#666}

文字排版--粗体

我们还可以使用css样式来改变文字的样式：粗体、斜体、下划线、删除线，可以使用下面代码实现设置文字以粗体样式显示出来。

p span{font-weight:bold;}

在这里大家可以看到，如果想为文字设置粗体是有单独的css样式来实现的，再不用为了实现粗体样式而使用h1-h6或strong标签了。

文字排版--斜体

以下代码可以实现文字以斜体样式在浏览器中显示：

p a{font-style:italic;}

<p>三年级时，我还是一个<a>胆小如鼠</a>的小女孩。</p>

文字排版--下划线

有些情况下想为文字设置为下划线样式，这样可以在视觉上强调文字，可以使用下面代码来实现：

p a{text-decoration:underline;}

<p>三年级时，我还是一个<a>胆小如鼠</a>的小女孩。</p>

文字排版--删除线

.oldPrice{text-decoration:line-through;}

段落排版--缩进

中文文字中的段前习惯空两个文字的空白，这个特殊的样式可以用下面代码来实现：

p{text-indent:2em;}

<p>1922年的春天，一个想要成名名叫尼克卡拉威（托比?马奎尔Tobey Maguire 饰）的作家，离开了美国中西部，来到了纽约。那是一个道德感渐失，爵士乐流行，走私为王，股票飞涨的时代。为了追寻他的美国梦，他搬入纽约附近一海湾居住。</p>

注意：2em的意思就是文字的2倍大小。

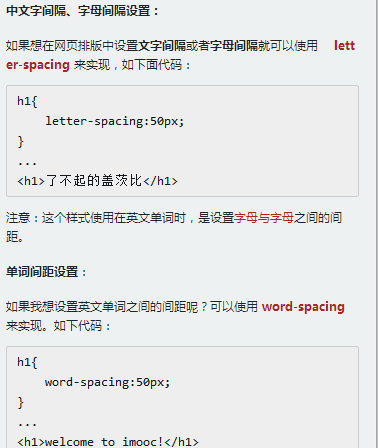
段落排版--行间距（行高）

这一小节我们来学习一下另一个在段落排版中起重要作用的行间距（行高）属性（line-height），如下代码实现设置段落行间距为1.5倍。

p{line-height:1.5em;}

<p>菲茨杰拉德，二十世纪美国文学巨擘之一，兼具作家和编剧双重身份。他以诗人的敏感和戏剧家的想象为"爵士乐时代"吟唱华丽挽歌，其诗人和梦想家的气质亦为那个奢靡年代的不二注解。</p>

段落排版--中文字间距、字母间距



段落排版--对齐



元素分类

在讲解CSS布局之前，我们需要提前知道一些知识，在CSS中，html中的标签元素大体被分为三种不同的类型：块状元素、内联元素(又叫行内元素)和内联块状元素。

常用的块状元素有：

<div>、<p>、<h1>...<h6>、<ol>、<ul>、<dl>、<table>、<address>、<blockquote> 、<form>

常用的内联元素有：

<a>、<span>、<br>、<i>、<em>、<strong>、<label>、<q>、<var>、<cite>、<code>

常用的内联块状元素有：

<img>、<input>

元素分类--块级元素

什么是块级元素？在html中<div>、 <p>、<h1>、<form>、<ul> 和 <li>就是块级元素。设置display:block就是将元素显示为块级元素。如下代码就是将内联元素a转换为块状元素，从而使a元素具有块状元素特点。

a{display:block;}

块级元素特点：

1、每个块级元素都从新的一行开始，并且其后的元素也另起一行。（真霸道，一个块级元素独占一行）

2、元素的高度、宽度、行高以及顶和底边距都可设置。

3、元素宽度在不设置的情况下，是它本身父容器的100%（和父元素的宽度一致），除非设定一个宽度。

元素分类--内联元素

在html中，<span>、<a>、<label>、 <strong> 和<em>就是典型的内联元素（行内元素）（inline）元素。当然块状元素也可以通过代码display:inline将元素设置为内联元素。如下代码就是将块状元素div转换为内联元素，从而使 div 元素具有内联元素特点。

内联元素特点：

1、和其他元素都在一行上；

2、元素的高度、宽度及顶部和底部边距不可设置；

3、元素的宽度就是它包含的文字或图片的宽度，不可改变。

元素分类--内联块状元素

内联块状元素（inline-block）就是同时具备内联元素、块状元素的特点，代码display:inline-block就是将元素设置为内联块状元素。(css2.1新增)，<img>、<input>标签就是这种内联块状标签。

inline-block 元素特点：

1、和其他元素都在一行上；

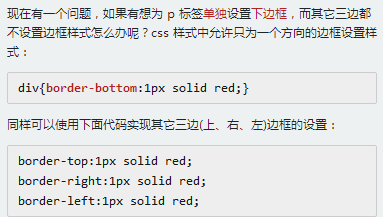
2、元素的高度、宽度、行高以及顶和底边距都可设置。

盒模型--边框（一）

盒子模型的边框就是围绕着内容及补白的线，这条线你可以设置它的粗细、样式和颜色(边框三个属性)。



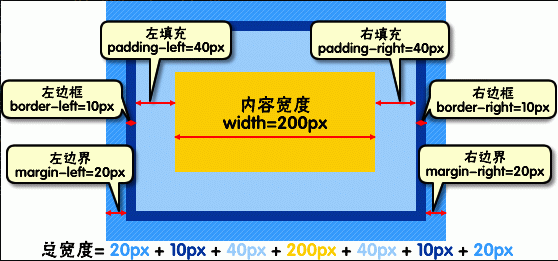
盒模型--边框（二）

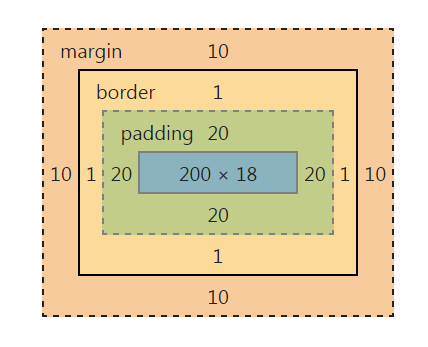
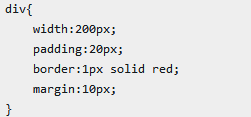


盒模型--宽度和高度

盒模型宽度和高度和我们平常所说的物体的宽度和高度理解是不一样的，css内定义的宽（width）和高（height），指的是填充以里的内容范围。

因此一个元素实际宽度（盒子的宽度）=左边界+左边框+左填充+内容宽度+右填充+右边框+右边界。





盒模型--填充

元素内容与边框之间是可以设置距离的，称之为“填充”。填充也可分为上、右、下、左(顺时针)。如下代码：

div{padding:20px 10px 15px 30px;}

顺序一定不要搞混。可以分开写上面代码：

div{

padding-top:20px;

padding-right:10px;

padding-bottom:15px;

padding-left:30px;

}

如果上、右、下、左的填充都为10px;可以这么写

div{padding:10px;}

如果上下填充一样为10px，左右一样为20px，可以这么写：

div{padding:10px 20px;}

盒模型--边界



css布局模型

清楚了CSS 盒模型的基本概念、 盒模型类型， 我们就可以深入探讨网页布局的基本模型了。布局模型与盒模型一样都是 CSS 最基本、 最核心的概念。 但布局模型是建立在盒模型基础之上，又不同于我们常说的 CSS 布局样式或 CSS 布局模板。如果说布局模型是本，那么 CSS 布局模板就是末了，是外在的表现形式。

CSS包含3种基本的布局模型，用英文概括为：Flow、Layer 和 Float。

在网页中，元素有三种布局模型：

1、流动模型（Flow）

2、浮动模型 (Float)

3、层模型（Layer）

流动模型（一）

先来说一说流动模型，流动（Flow）是默认的网页布局模式。也就是说网页在默认状态下的 HTML 网页元素都是根据流动模型来分布网页内容的。

流动布局模型具有2个比较典型的特征：

1.块状元素都会在所处的包含元素内自上而下按顺序垂直延伸分布，因为在默认状态下，块状元素的宽度都为100%。实际上，块状元素都会以行的形式占据位置。如右侧代码编辑器中三个块状元素标签(div，h1，p)宽度显示为100%。

2.在流动模型下，**内联元素**都会在所处的包含元素内从左到右水平分布显示。（内联元素可不像块状元素这么霸道独占一行）

右侧代码编辑器中内联元素标签a、span、em、strong都是内联元素。

浮动模型

块状元素这么霸道都是独占一行，如果现在我们想让两个块状元素并排显示，怎么办呢？不要着急，设置元素浮动就可以实现这一愿望。

任何元素在默认情况下是不能浮动的，但可以用 CSS 定义为浮动，如 div、p、table、img 等元素都可以被定义为浮动。如下代码可以实现两个 div 元素一行显示。

div{

width:200px;

height:200px;

border:2px red solid;

float:left;

}

<div id="div1"></div>

<div id="div2"></div>

什么是层模型？

什么是层布局模型？层布局模型就像是图像软件PhotoShop中非常流行的图层编辑功能一样，每个图层能够精确定位操作，但在网页设计领域，由于网页大小的活动性，层布局没能受到热捧。但是在网页上局部使用层布局还是有其方便之处的。下面我们来学习一下html中的层布局。

如何让html元素在网页中精确定位，就像图像软件PhotoShop中的图层一样可以对每个图层能够精确定位操作。CSS定义了一组定位（positioning）属性来支持层布局模型。

层模型有三种形式：

1、绝对定位(position: absolute)

2、相对定位(position: relative)

3、固定定位(position: fixed)

层模型--绝对定位

如果想为元素设置层模型中的绝对定位，需要设置position:absolute(表示绝对定位)，这条语句的作用将元素从文档流中拖出来，然后使用left、right、top、bottom属性相对于其最接近的一个具有定位属性的父包含块进行绝对定位。如果不存在这样的包含块，则相对于body元素，即相对于浏览器窗口。

如下面代码可以实现div元素相对于浏览器窗口向右移动100px，向下移动50px。

div{

width:200px;

height:200px;

border:2px red solid;

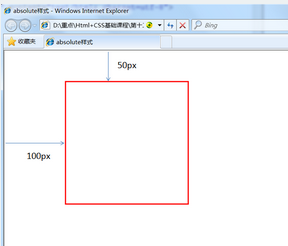
position:absolute;

left:100px;

top:50px;

}

<div id="div1"></div>



层模型--相对定位

如果想为元素设置层模型中的相对定位，需要设置position:relative（表示相对定位），它通过left、right、top、bottom属性确定元素在正常文档流中的偏移位置。相对定位完成的过程是首先按static(float)方式生成一个元素(并且元素像层一样浮动了起来)，然后相对于以前的位置移动，移动的方向和幅度由left、right、top、bottom属性确定，偏移前的位置保留不动。

如下代码实现相对于以前位置向下移动50px，向右移动100px;

层模型--固定定位

fixed：表示固定定位，与absolute定位类型类似，但它的相对移动的坐标是视图（屏幕内的网页窗口）本身。由于视图本身是固定的，它不会随浏览器窗口的滚动条滚动而变化，除非你在屏幕中移动浏览器窗口的屏幕位置，或改变浏览器窗口的显示大小，因此固定定位的元素会始终位于浏览器窗口内视图的某个位置，不会受文档流动影响，这与background-attachment:fixed;属性功能相同。以下代码可以实现相对于浏览器视图向右移动100px，向下移动50px。并且拖动滚动条时位置固定不变。

Relative与Absolute组合使用

小伙伴们学习了12-6小节的绝对定位的方法：使用position:absolute可以实现被设置元素相对于浏览器（body）设置定位以后，大家有没有想过可不可以相对于其它元素进行定位呢？答案是肯定的，当然可以。使用position:relative来帮忙，但是必须遵守下面规范：

1、参照定位的元素必须是相对定位元素的前辈元素：

<div id="box1"><!--参照定位的元素-->

<div id="box2">相对参照元素进行定位</div><!--相对定位元素-->

</div>

从上面代码可以看出box1是box2的父元素（父元素当然也是前辈元素了）。

2、参照定位的元素必须加入position:relative;

#box1{

width:200px;

height:200px;

position:relative;

}

3、定位元素加入position:absolute，便可以使用top、bottom、left、right来进行偏移定位了。

#box2{

position:absolute;

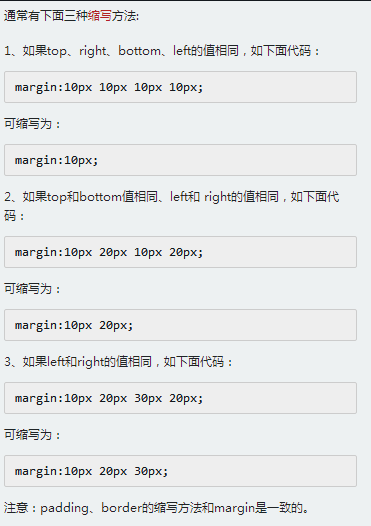
top:20px;

left:30px;

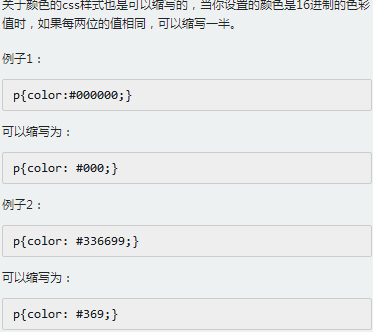
}

这样box2就可以相对于父元素box1定位了（这里注意参照物就可以不是浏览器了，而可以自由设置了）。

盒模型代码简写



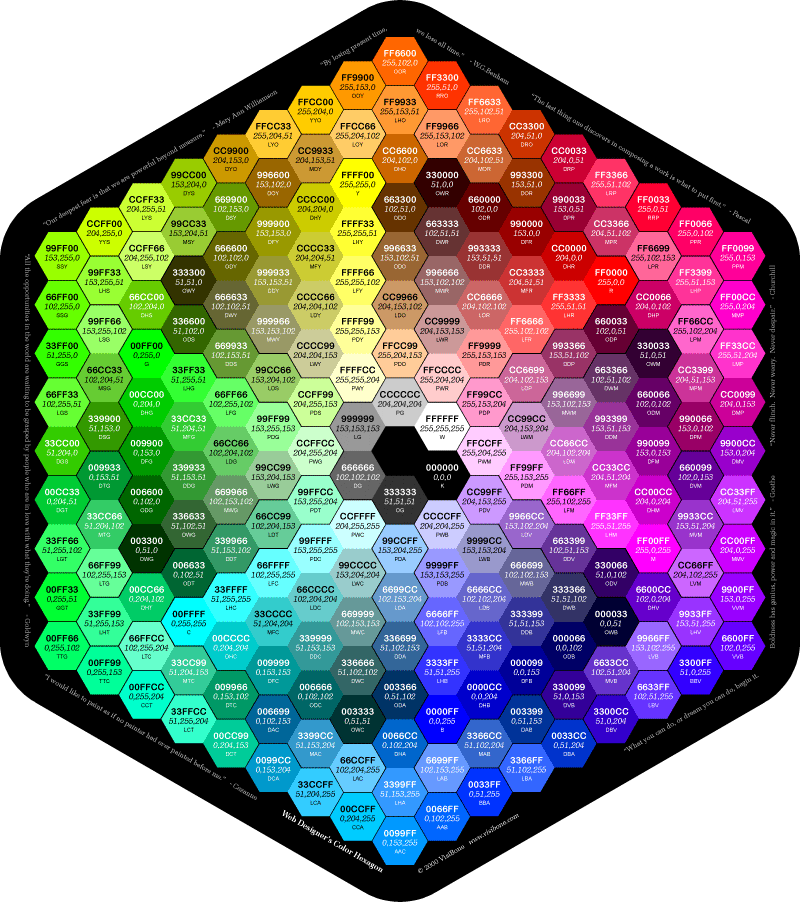
颜色值缩写



字体缩写



颜色值



长度值

长度单位总结一下，目前比较常用到px（像素）、em、% 百分比，要注意其实这三种单位都是相对单位。

1、像素

像素为什么是相对单位呢？因为像素指的是显示器上的小点（CSS规范中假设“90像素=1英寸”）。实际情况是浏览器会使用显示器的实际像素值有关，在目前大多数的设计者都倾向于使用像素（px）作为单位。

2、em

就是本元素给定字体的 font-size 值，如果元素的 font-size 为 14px ，那么 1em = 14px；如果 font-size 为 18px，那么 1em = 18px。如下代码：

p{font-size:12px;text-indent:2em;}

上面代码就是可以实现段落首行缩进 24px（也就是两个字体大小的距离）。

下面注意一个特殊情况：

但当给 font-size 设置单位为 em 时，此时计算的标准以 p 的父元素的 font-size 为基础。如下代码：

html:

<p>以这个<span>例子</span>为例。</p>

css:

p{font-size:14px}

span{font-size:0.8em;}

结果 span 中的字体“例子”字体大小就为 11.2px（14 \* 0.8 = 11.2px）。

3、百分比

p{font-size:12px;line-height:130%}

设置行高（行间距）为字体的130%（12 \* 1.3 = 15.6px）。