

第二章 CSS基础

本章导读

网页是Web的基础。网页内容以HTML标记搭起框架，然后通过CSS修饰外观，达到美观。可以说，没有CSS，就没有异彩纷呈的网页。CSS可以修饰指定某个标记，也可以修饰相同的HTML标记，还可以修饰某个指定范围内的HTML标记。灵活地应用CSS范围规则，可以更好修饰网页。

目标：

- 1. 掌握CSS的含义；
- 2. 掌握CSS的定义位置；
- 3. 掌握常见的CSS项目；
- 4. 熟悉CSS的应用；

本章目录

- 第一节 理解CSS含义
- 第二节 CSS的位置
- 第三节 CSS作用范围
- 第四节 通配符的使用
- 第五节 A(链接)的CSS
- 第六节 网页标记定位
 - 1、position及left、right、top和bottom属性
 - 2、z-index属性
 - 3、display和visibility属性
 - 4、float和clear属性
- 第七节 CSS尺寸单位
- 第八节 CSS的复合属性和单一属性
- 第九节 更多CSS属性
 - 1、cursor属性
 - 2、clip
- 第十节 小结

在HTML中，style属性主要用于定义HTML标记的外在特征，style属性可设置的项目较多， 这些属性的集合称之为CSS(Cascading Style Sheet)。这些属性不仅用于设置HTML标记的外在特征， 更有父子标记之间的继承关系。在网页设计中，HTML标记用于搭建起网页框架， 然后通过style属性接受CSS数据，影响HTML标记外观。可以说没有CSS，就没有美观。

第一节 理解CSS含义

CSS是Cascading Style Sheet的简写，其含义是层叠式样式表。Cascading的英文含义是级联、串联、层叠， 其基本语义是下级继承上级定义。例程2-1的第6行定义了Div的font-size和color属性， 其管辖范围内的span继承了color属性值，但由于span重新定义了font-size， 所以其管辖的“CSS”三个英文字母都被设置为14px。图2-1显示了例程2-1的执行效果。Div中的其他文字沿用Div中的定义。

例程2-1

第1行	<HTML>
第2行	<HEAD>
第3行	<TITLE>理解Cascading Style Sheets的含义</TITLE>
第4行	</HEAD>
第5行	<BODY>
第6行	<Div style="font-size:24px;color:Red">
第7行	
第8行	CSS
第9行	

第10行	用于定义Style，通常用于描述HTML标记的外在特征。在网页设计中，
第11行	HTML标记用于搭建起网页框架，然后 用
第12行	
第13行	CSS
第14行	
第15行	定义HTML标记的外在特征。
第16行	没有
第17行	
第18行	CSS
第19行	
第20行	，就没有美观。
第21行	</Div>
第22行	</BODY>
第23行	</HTML>

在例程2-1的代码中，Div的Style定义了字体大小(font-size:24px)和文本颜色(color:Red)，Span被Div包含，或者说Div拥有下一级标记Span，其Style定义了字体大小(font-size:14px)。在图2-1中的显示效果中，Span继承了其父级的颜色特征，或者说Div影响了子级标记Span的Style特征。

在CSS中，父影响子，子影响孙；反之，子受到父，父受到爷的影响。这就是CSS的逐级继承特征。在CSS中，很多CSS属性都具有继承性，如color、font、text-align等，仅有少部分不具有继承性，如background，border等。

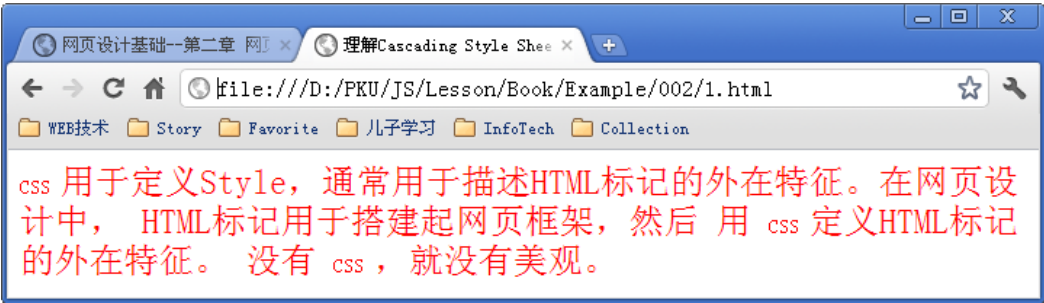


图2-1 CSS的继承特征

第二节 CSS的位置

CSS可以写在HTML标记的Style属性中，例程2-2的执行效果如图2-2所示。

例程2-2

第1行	<HTML>
第2行	<HEAD>
第3行	<TITLE>理解Cascading Style Sheets的含义</TITLE>
第4行	</HEAD>
第5行	<BODY>
第6行	《红楼梦》以女娲补天神话为引子，作者曹雪芹以金陵十二钗正册的
第7行	林黛玉、
第8行	薛宝钗、
第9行	贾元春、
第10行	贾探春、
第11行	史湘云、
第12行	妙玉、
第13行	贾迎春、
第14行	贾惜春、
第15行	王熙凤、

Head-Style的格式是<Style type="text/css">.....</Style>。在配对的Style中撰写其内容，每条定义必须用花括弧({})包围(在Inline-Style中，用双引号包围)，具体内容与Inline-Style完全相同。

在Inline-Style中，其定义的Style仅能作用于包含该定义的HTML标记，而在Head-Style中，可以一次定义，适用于多个标记，具体作用于那个标记，需要在Head-Style中指明。很明显，Inline-Style无需指明，就是作用于包含该Style的HTML标记。在Head-Style中，指明作用标记的方法基本有三种，这是其中一种，在例程2-3第5行，其含义是“本网页中所有span元素均采用该定义”。

将CSS写在Head中，还有利于实现内容实体与显示效果的分离。Body中主要包含网页框架及内容，Head-Style中主要定义网页框架及其显示方式。实现内容与效果分离且各自集中维护管理。

Style除了Inline-Style和Head-Style模式外，尚有File-Style模式，即将Style的内容保存在独立的文件中。一个网站通常由很多个网页组成，其中部分网页显示风格基本一致，此时可以将基本的显示风格定义在一个独立的CSS文件中，这就是File-Style。只要这些页面都引用相同的CSS文件，其风格就能保持一致，其格式如下。例程2-4和例程2-5两个独立网页都引用了myStyle.css文件，其风格一致(背景、字体、颜色等)。例程2-6是myStyle.css的内容。

```
<Link href="CSS文件的URL" rel="stylesheet" type="text/css">
```

例程2-4

第1行	<HTML>
第2行	<HEAD>
第3行	<TITLE>File-Style例程</TITLE>
第4行	<Link href="myStyle.css" rel="stylesheet" type="text/css">
第5行	</HEAD>
第6行	<BODY>
第7行	<Div> 如梦令</Div>
第8行	<P>林黛玉</P>
第9行	粉堕百花州,香残燕子楼。
第10行	一团团逐对成逑,飘泊亦如人命薄。
第11行	空缱绻,说风流。
第12行	草木也知愁,韶华竟白头!
第13行	叹今生谁舍谁收?
第14行	嫁与东风春不管,凭尔去,忍淹留。
第15行	</BODY>
第16行	</HTML>

例程2-4的执行效果如下。

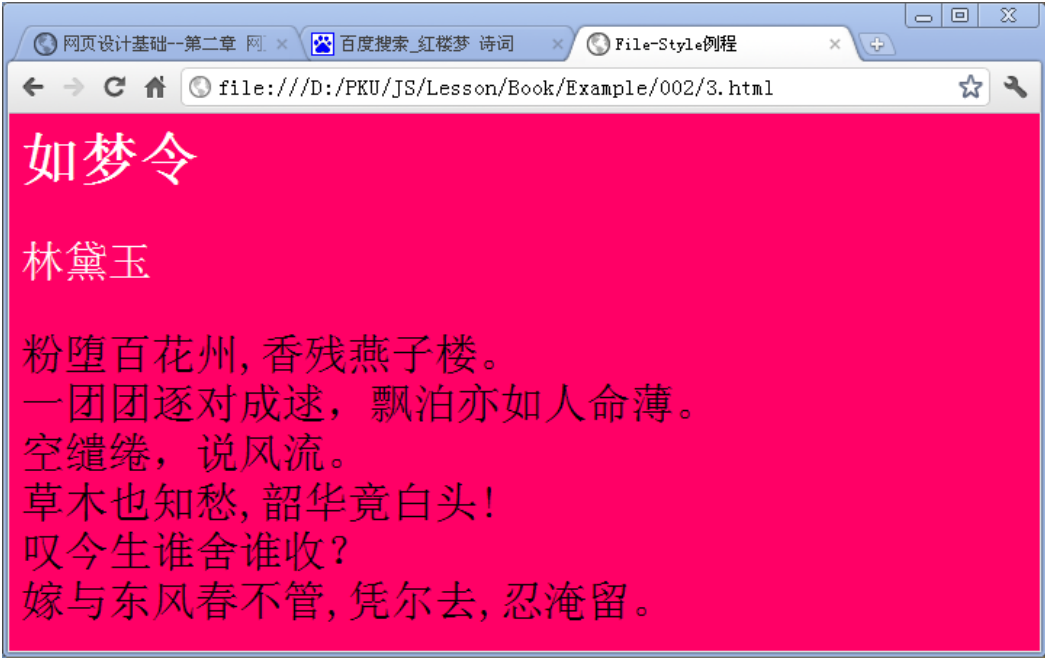


图2-3 例程2-4引用CSS文件后的执行效果

第1行	<HTML>
第2行	<HEAD>
第3行	<TITLE>File-Style例程</TITLE>
第4行	<Link href="myStyle.css" rel="stylesheet" type="text/css">
第5行	</HEAD>
第6行	<BODY>
第7行	<Div> 凝晖钟瑞匾额</Div>
第8行	<P>薛宝钗</P>
第9行	芳园筑向帝城西,华日祥云笼罩奇。
第10行	高柳喜迁莺出谷,修篁时待凤来仪。
第11行	文风已著宸游夕,孝化应隆归省时。
第12行	睿藻仙才盈彩笔,自惭何敢再为辞。
第13行	</BODY>
第14行	</HTML>

例程2-5的执行效果如下。

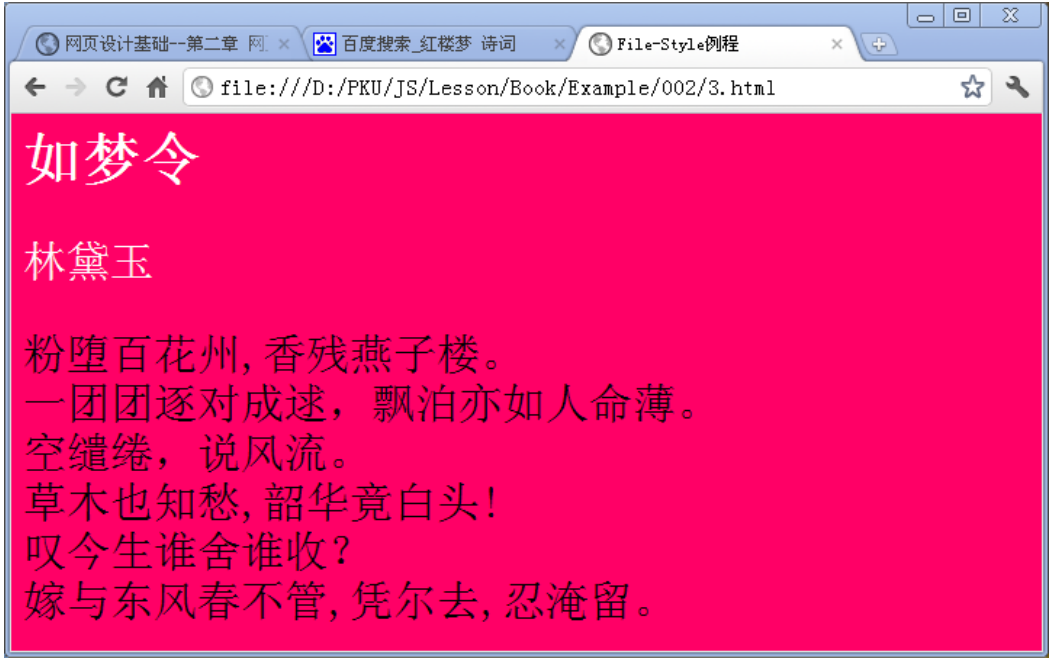


图2-4 例程2-5引用CSS文件后的执行效果

例程2-4和例程2-5所应用的CSS文件内容如下例程2-6所示。

第1行	body {background-color:#FF0066;font-size:30px;}
第2行	Div {font-size:40px;color:white;font-weight:bold}
第3行	P {color:white}

外部CSS文件通常放在Head区域，如例程2-4和例程2-5的第4行所示。采用File-Style模式的能使引用相同CSS文件的多个网页保持风格一致。当需要调整其显示效果时，仅仅修改CSS文件即可，无须每个网页逐一调整，减少工作量，降低维护难度。当采用File-Style方案时，如需局部调整，可采用Head-Style或Inline-Style模式重新某些HTML标记即可。如File-Style、Head-Style和Inline-Style都对同一个HTML标记做出定义，即发生冲突，此时，Inline-Style优先级最高，其次是Head-Style，最后是File-Style，即HTML标记的最终效果由优先级最高的定义确定。

当在File-Style中定义CSS时，头尾无需<Style type="text/css">……</Style>，如例程2-6所示。

第三节 CSS作用范围

在Head-Style和File-Style中定义CSS，除直接定义HTML标记外，还可以通过定义Class和ID的方式定义HTML标记的显示效果。如图2-

5所示，例程2-7是其代码。



图2-5 例程2-7执行效果

在例程2-7的第6行，定义了“.Odd”的CSS，在Body内，有多个P增加了Class=“Odd”。Class可以称之为分类，等号后引号内为类的名称。在Head-Style中，定义了Odd的CSS，则所有名为Odd的HTML标记都将呈现定义的效果，不仅仅是P标记。在例程2-7中，仅有P标记的Class属性被命名为Odd，因此也仅有这些标记显示出该效果。

例程2-7

```
第1行  <HTML>
第2行  <HEAD>
第3行  <TITLE>CSS作用范围</TITLE>
第4行  <Style type=text/css>
第5行  P{margin:10px 0px 0px 0px}
第6行  .Odd{font-size:20px;font-weight:bold;color:white;background-color:#336666}
第7行  </Style>
第8行  </HEAD>
第9行  <BODY>
第10行 <P class="Odd">王熙凤害死尤二姐——心狠手毒</P>
第11行 <P>林黛玉葬花——自叹命薄</P>
第12行 <P class="Odd">刘姥姥出大观园——满载而归</P>
第13行 <P>小葱拌豆腐——清的清白的白</P>
第14行 <P class="Odd">宋徽宗的鹰，赵子昂的马——都是好画儿</P>
第15行 <P>千里搭长棚——没有个不散的宴席</P>
第16行 <P class="Odd">狗咬吕洞宾——不识好歹</P>
第17行 <P>引风吹火——费力不多</P>
第18行 </BODY>
第19行 </HTML>
```

在例程2-7中，定义了P标记和Class=“Odd”的CSS，由于Odd都在P中，因此，Class为Odd的P标记受二者影响，是这二者综合总用的结果。当对相同的CSS属性进行定义，以Class为准，与定义的先后顺序无关。

在例程2-7中，奇数行和偶数行分别具有不同的效果，通过设定Class对P标记进行分类管理。如果觉得设定CSS不符合要求，在Head-Style中修改即可，所有具有相同Class的标记都将采用重新设定的结果，给维护带来了极大的方便。

还可以对例程2-7的代码进行修改，使维护奇数行和偶数行的效果更加方便，如例程2-8。

例程2-8

```
第1行  <HTML>
第2行  <HEAD>
```

第3行	<TITLE>CSS作用范围</TITLE>
第4行	<Style type=text/css>
第5行	P{Margin:10px 0px 0px 0px}
第6行	.Odd{font-size:20px;font-weight:bold;color:white;background-color:#336666;}
第7行	.Even{font-size:20px;color:black;}
第8行	</Style>
第9行	</HEAD>
第10行	<BODY>
第11行	<P class="Odd">王熙凤害死尤二姐——心狠手毒</P>
第12行	<P class="Even">林黛玉葬花——自叹命薄</P>
第13行	<P class="Odd">刘姥姥出大观园——满载而归</P>
第14行	<P class="Even">小葱拌豆腐——清的清白的白</P>
第15行	<P class="Odd">宋徽宗的鹰，赵子昂的马——都是好画儿</P>
第16行	<P class="Even">千里搭长棚——没有个不散的宴席</P>
第17行	<P class="Odd">狗咬吕洞宾——不识好歹</P>
第18行	<P class="Even">引风吹火——费力不多</P>
第19行	</BODY>
第20行	</HTML>

在例程2-8中，在Head-Style中定义了Odd和Even两类Class，分别对应奇数行和偶数行。如果要修改显示效果，将会更加方便，直接在Head-Style中修改即可。

一个HTML标记可以有多个Class(不一定是两个)，如例程2-9所示。

例程2-9

第1行	<HTML>
第2行	<HEAD>
第3行	<TITLE>CSS作用范围</TITLE>
第4行	<Style type=text/css>
第5行	.Common{Margin:10px 0px 0px 0px;font-size:20px}
第6行	.Odd{font-weight:bold;color:white;background-color:#336666;}
第7行	.Even{color:black;}
第8行	</Style>
第9行	</HEAD>
第10行	<BODY>
第11行	<P class="Common Odd">王熙凤害死尤二姐——心狠手毒</P>
第12行	<P class="Common Even">林黛玉葬花——自叹命薄</P>
第13行	<P class="Common Odd">刘姥姥出大观园——满载而归</P>
第14行	<P class="Common Even">小葱拌豆腐——清的清白的白</P>
第15行	<P class="Common Odd">宋徽宗的鹰，赵子昂的马——都是好画儿</P>
第16行	<P class="Common Even">千里搭长棚——没有个不散的宴席</P>
第17行	<P class="Common Odd">狗咬吕洞宾——不识好歹</P>
第18行	<P class="Common Even">引风吹火——费力不多</P>
第19行	</BODY>
第20行	</HTML>

在例程2-9中，每个P标记都有两个属性，分别是Common Odd和Common Even， 每一个P也是两个Class共同作用的结果。当定义两个Class定义冲突时，后面的Class覆盖前面的定义。这种方式，在设置“共同-个别”特征时，很是方便，通过第一个(或前面)Class定义通用CSS，后继的Class定义局部CSS特征。

有关Class的注意事项

1. 在Head-Style和File-Style中，定义Class的CSS时，其名称前必须有英文句点；
2. Class的命名最好不与HTML标记同名；
3. 当Class的定义CSS与HTML标记所定义向冲突时，Class的定义优先；
4. 一个HTML标记通常仅有一个Class属性，但每个Class属性可设置多个名称，每个名称之间用空格分开；
5. 当一个HTML标记有多个Class名称且定义CSS冲突时，后面的定义覆盖前面的定义；

在HTML和CSS中，还可以通过HTML标记的ID属性标记为一个HTML标记。在Head-Style和File-Style中，设置ID的CSS效果，将仅对该ID对应的标记有效果。例程2-10中使用HTML标记的ID属性。

例程2-10

第1行	<HTML>
第2行	<HEAD>
第3行	<TITLE>CSS作用范围</TITLE>
第4行	<Style type=text/css>
第5行	.Common{Margin:10px 0px 0px 0px;font-size:20px}
第6行	.Odd{font-weight:bold;color:white;background-color:#336666;}
第7行	.Even{color:black;}
第8行	#Caption{font-size:30px;font-weight:bold;margin:20 0px 10px 0px}
第9行	</Style>
第10行	</HEAD>
第11行	<BODY>
第12行	<P ID="Caption">红楼梦歇后语选编</P>
第13行	<P class="Common Odd">王熙凤害死尤二姐——心狠手毒</P>
第14行	<P class="Common Even">林黛玉葬花——自叹命薄</P>
第15行	<P class="Common Odd">刘姥姥出大观园——满载而归</P>
第16行	<P class="Common Even">小葱拌豆腐——清的清白的白</P>
第17行	<P class="Common Odd">宋徽宗的鹰，赵子昂的马——都是好画儿</P>
第18行	<P class="Common Even">千里搭长棚——没有个不散的宴席</P>
第19行	<P class="Common Odd">狗咬吕洞宾——不识好歹</P>
第20行	<P class="Common Even">引风吹火——费力不多</P>
第21行	</BODY>
第22行	</HTML>

在例程2-10的第8行和第12行，分别使用了ID，其中第12行是设置HTML标记的ID属性，绝大多数HTML标记都支持ID属性。第8行是对ID属性设置CSS。和定义Class属性的CSS不同，ID前面须增加英文#。图2-6是例程2-10的执行效果。

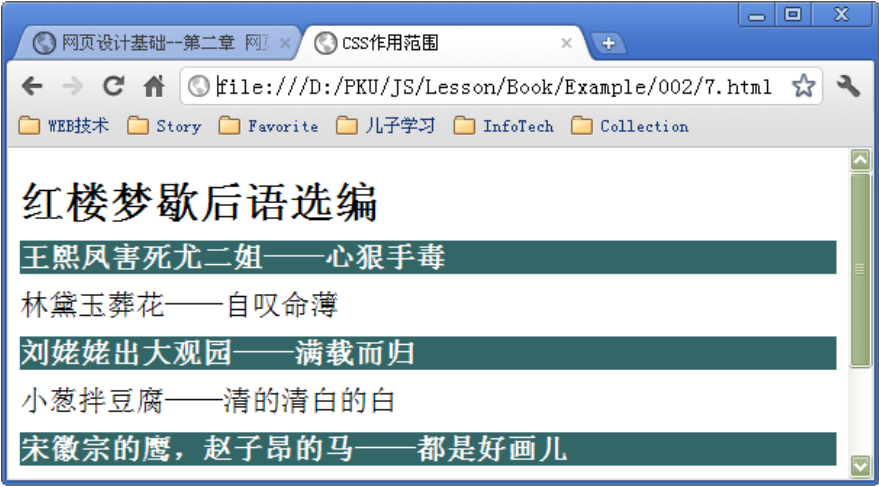


图2-6 HTML标记ID属性的使用

在Head-Style和File-Style中定义CSS时，还可以采用上下级(层叠)方式约束其范围，如例程2-11中第9行，p span的含义是包含在p

标记中的span标记（不一定仅是子级，孙级甚至跟深层下级均可）的CSS设置。在层叠式Style定义中，上级可以是ID,Class和HTML标记，下级通常是Class和HTML标记，很少ID，因为ID通常都是唯一，无需增加父级，约束其范围。

例程2-11

第1行	<HTML>
第2行	<HEAD>
第3行	<TITLE>CSS作用范围</TITLE>
第4行	<Style type=text/css>
第5行	.Common{Margin:10px 0px 0px 0px;font-size:20px}
第6行	.Odd{font-weight:bold;color:white;background-color:#336666;}
第7行	.Even{color:black;}
第8行	#Caption{font-size:30px;font-weight:bold;margin:20 0px 10px 0px}
第9行	p span{font-style:italic;font-weight:bold}
第10行	</Style>
第11行	</HEAD>
第12行	<BODY>
第13行	<P ID="Caption">红楼梦歇后语选编</P>
第14行	<P class="Common Odd">王熙凤害死尤二姐——心狠手毒</P>
第15行	<P class="Common Even">林黛玉葬花——自叹命薄</P>
第16行	<P class="Common Odd">刘姥姥出大观园——满载而归</P>
第17行	<P class="Common Even">小葱拌豆腐——清的清白的白</P>
第18行	<P class="Common Odd">宋徽宗的鹰，赵子昂的马——都是好画儿</P>
第19行	<P class="Common Even">千里搭长棚——没有个不散的宴席</P>
第20行	<P class="Common Odd">狗咬吕洞宾——不识好歹</P>
第21行	<P class="Common Even">引风吹火——费力不多</P>
第22行	</BODY>
第23行	</HTML>

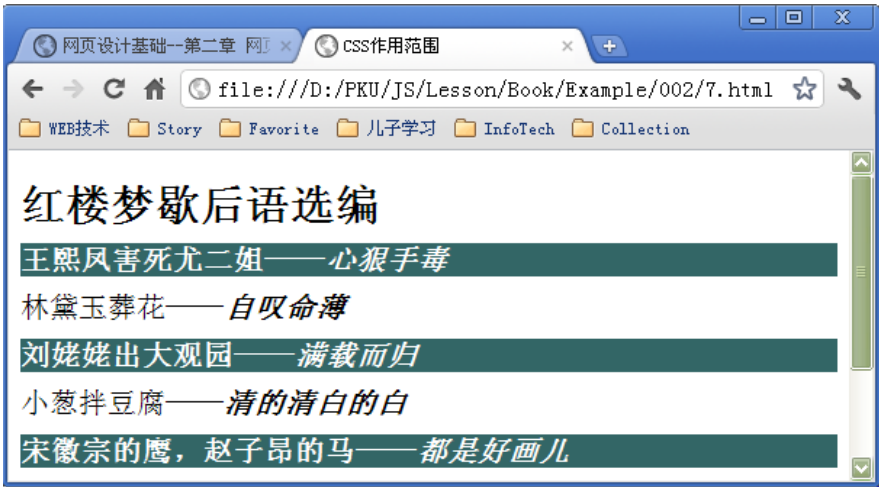


图2-7 例程2-11的执行效果

如果仅仅对p元素中的子级元素设置CSS，可以将“p span”调整为“p>span”即可， 仅对p标记的儿子辈span有效，其他级别的span将不在其约定范围内。

在定义Style过程中，可以通过HTML标记、Class和ID分别对指定标记定义Style。当在Head-Style和File-Style中定义Style内容时，HTML标记将对整个网页的相同HTML标记发挥作用，Class仅对拥有相同Class属性值的元素发挥作用，而ID通常是仅对一个元素发挥作用。

Style的定义还可以通过层级串联定义，也就是父子甚至多级定义以约束Style中CSS的作用范围。如“DIV SPAN”中DIV是父级，SPAN是子级。其含义是对Div中的SPAN进行Style定义，不在DIV中的SPAN则不受约束。父级可以是HTML标记，也可以使Class和ID，子级通常为HTML标记和 Class。通过层级串联定义，可以更加准确的限定Style的作用范围。

在定义Style时，还可能出现多个Class、ID以及HTML标记共有一个Style的情况，也就是分组，此时可以通过如下语法模式实现。

.XYZ, #ABC, DIV, A{color:Red;}

在上面的语法示例中，XYZ为一个类(Class)，ABC是一个ID，DIV和A是HTML标记，它们的颜色都设置Red(红色)。每一个都用逗号分开，每个独立部分与单独书写的语法完全一样，也支持层级串联等。

第四节 通配符的使用

在Head-Style和File-Style中，可以使用通配符*，表示所有HTML标记都以该设定为准，除非对某些HTML标记，或者Class(类)、或者ID再次定义。通配符通常用于设定全局默认值。如：设定所有文本的字体大小、颜色，或者margin、padding等。如

```
*{margin:0px;padding:0px;font-size:20px}
```

在上述代码中，表示所有元素margin被设为0px，padding被设为0px，字号为20px。此时，包括P标记在内，都没有默认的段前和段后空白。一般说来，很多浏览器在显示内容时，都会在上下左右流出一定的空白，通过上述设定，这些默认的空白也被消除。

第五节 A(链接)的CSS

A(链接)和其他HTML标记相比，其CSS有其特殊性。和A链接相比，其他HTML标记的CSS只能定义其静止状态的显示效果，而A还可以分别定义鼠标悬停和离开时的CSS，甚至是否访问(visited)或者活动(active)的状态的链接都可以定义不同的CSS。图2-8是A标记在鼠标悬停(hover)时的效果，例程2-12是其实现代码。



图2-8

从图2-8可以看出，“《红楼梦》诗词赏析”与其他各行效果不一样，其效果是通过例程2-12第6行实现。a:hover中的hover被称为“伪类(pseudo-class)”，与Class有些相似，但不是Class，其前面用“:”标记，而Class用“.”标记。与a标记相关的伪类有link、visited、hover、active，分别对应“未访问的链接”、“已访问的链接”、“鼠标指向的链接”、“选定的链接”，通常仅用link和hover。如果省略伪类，则对所有链接均有效。

例程2-12

第1行	<HTML>
第2行	<HEAD>
第3行	<TITLE>A的CSS</TITLE>
第4行	<Style type=text/css>
第5行	body{line-height:30px}
第6行	a:hover{font-size:20px;font-weight:bold;color:red}
第7行	</Style>
第8行	</HEAD>
第9行	<BODY>
第10行	<P>红楼梦</P>
第11行	小说《红楼梦》
第12行	《红楼梦》诗词赏析
第13行	《红楼梦》歇后语</BR>

第14行	<code>《红楼梦》人物探微</code>
第15行	<code></BODY></code>
第16行	<code></HTML></code>

在定义a的伪类时，定义顺序通常为Link、Visited、Hover、Active，简称LVHA。可以省略其中一个或多个，但如果调整顺序，可能在某些浏览器中效果不能达到设定目标。

鉴于A标记如此特征，有时还利用a标记做网页菜单，如图2-9所示，例程2-13是其实现代码。

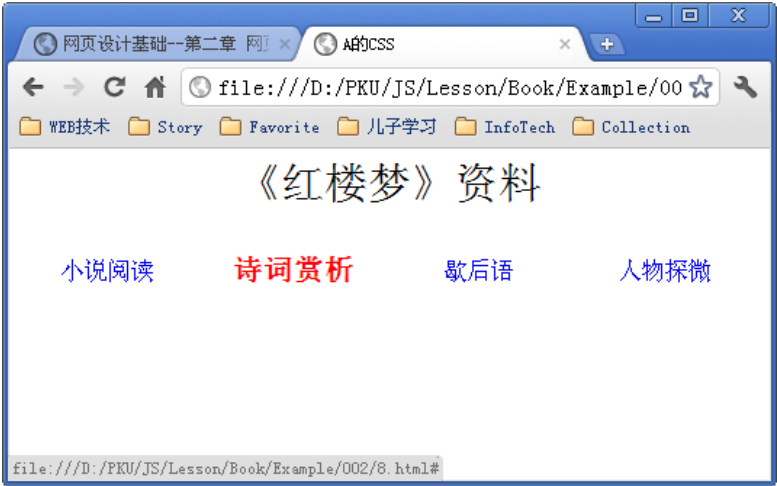


图2-9 用a标记做菜单

在图2-9中，“诗词赏析”明显不同于其他各个栏目。通过A(链接)实现了当鼠标移动到某个链接时，条目显示效果发生变化，当鼠标移出后，有恢复到原始状态，由此可以实现凸显某个菜单的目的。

例程2-13

第1行	<code><HTML></code>
第2行	<code><HEAD></code>
第3行	<code><TITLE>A的CSS</TITLE></code>
第4行	<code><Style type=text/css></code>
第5行	<code>body{line-height:30px}</code>
第6行	<code>P{font-size:30px;text-align:Center}</code>
第7行	<code>a{width:120px;text-align:Center;display:inline-block;text-decoration:none}</code>
第8行	<code>a:hover{font-size:20px;font-weight:bold;color:red;}</code>
第9行	<code></Style></code>
第10行	<code></HEAD></code>
第11行	<code><BODY></code>
第12行	<code><P>《红楼梦》资料</P></code>
第13行	<code>小说阅读</code>
第14行	<code>诗词赏析</code>
第15行	<code>歇后语</code>
第16行	<code>人物探微</code>
第17行	<code></BODY></code>
第18行	<code></HTML></code>

A在网页中应用非常普遍，除了用层级方式限定范围外，还可以通过Class方式同级限定，比如：A.XYZ 表示Class属性值为XYZ的A标记，通过这样的方式限定A的CSS作用范围。如例程2-14所示。

例程2-14

第1行	<code><HTML></code>
第2行	<code><HEAD></code>
第3行	<code><TITLE>A的CSS</TITLE></code>

第4行	<Style type=text/css>
第5行	body{line-height:30px}
第6行	P{font-size:30px;text-align:Center}
第7行	a.Menu{width:120px;text-align:Center;display:inline-block;text-decoration:none}
第8行	a.Menu:hover{font-size:20px;font-weight:bold;color:red;}
第9行	</Style>
第10行	</HEAD>
第11行	<BODY>
第12行	<P>《红楼梦》资料</P>
第13行	小说阅读
第14行	诗词赏析
第15行	歇后语
第16行	人物探微
第17行	</BODY>
第18行	</HTML>

在上例中，最后一个链接不同于其他链接，通过增加Class的方式约定其范围， 让A链接之间有所区分，在网页设计实践较为有用。如图2-10所示。

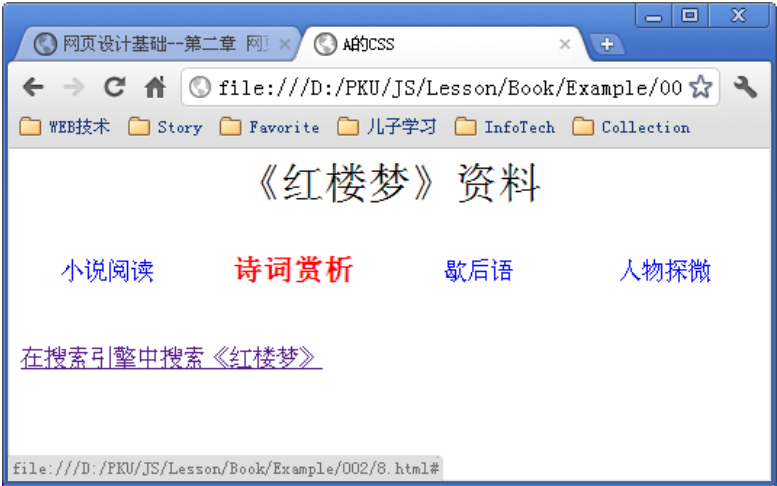


图2-10 用Class限定a标记范围

第六节 网页标记定位

一般说来，网页内容多按先后顺序依次呈现在浏览器中，但也能不时看见特殊定位的内容区块， 如图2-11所示，右下角的区块的底部还有其他内容，且移动滚动条、调整窗口大小，其位置都不变。 观察例程2-15，至少能发现“世事洞明皆学问”的内容实际上出现在“红楼梦”三字的后面， 但是其位置却在红楼梦之前。这些效果的实现，都依赖于CSS的定位属性。使用CSS定位属性， 能更好布局网页元素提供了方便。



图2-11 CSS定位示例

例程2-15

第1行	<HTML>
第2行	<HEAD>
第3行	<TITLE>CSS定位</TITLE>
第4行	</HEAD>
第5行	<BODY>
第6行	<Div style="position:absolute;left:80px;width:400px;
第7行	font-size:60px;text-align:center; background-color: #FF9966;
第8行	color:white;font-weight:bold">
第9行	红楼梦
第10行	</Div>
第11行	<Div style="position:absolute;left:20px;width:20px;
第12行	font-size:40px;font-weight:bold">
第13行	世事洞明皆学问
第14行	</Div>
第15行	<Div style="position:absolute;left:505px;width:60px;
第16行	font-size:40px;font-weight:bold">
第17行	人情练达即文章
第18行	</Div>
第19行	<Div style="position:fixed;bottom:10px;right:10px;border:5px ridge gray;
第20行	font-size:40px;background-color:#33FF99;opacity:0.5;color: red">
第21行	观看《红楼梦》电影
第22行	</Div>
第23行	</BODY>
第24行	</HTML>

定位需明确原点位置，或相对于父元素或其他元素，甚至浏览器窗口本身，原点位置的不同，也确定了元素出现位置的不同，使得元素能呈现在网页中任何位置。显然，这个功能非常强大，能出现很多让人意想不到的效果。

1、position及left、right、top和bottom属性

观察例程2-15的第6、11、15、19行，会发现都使用了position属性以及left、right、bottom等，其position取值分别有

absolute(决定)和fixed(固定)，除此以外，position还有其他取值，如表1所示。 上(top)、下(bottom)、左(left)、右(right)分别表示向下、向上、向右、向左的偏移量。

表1: position属性取值说明表

值	描述
absolute	绝对定位。相对于父元素进行定位。 元素的位置通过left、top、right 以及bottom中的一个或多个属性设定。
fixed	固定定位。相对于浏览器窗口进行定位。 元素的位置通过left、top、right 以及bottom中的一个或多个属性设定。
relative	相对定位。相对于本应出现的位置按left、top、top、bottom设定偏移。
static	静态定位，默认定位方式。 元素出现在常规位置，忽略 top, bottom, left, right以及z-index设定
inherit	继承定位。从父元素中继承position属性的值。

如图2-12所示，“红楼梦诗词选编”采用fixed(固定定位)，外部大的黑色框区采用absolute(绝对定位)， 而诗词主体所在框区采用relative(相对定位)，其代码如例程2-16所示。

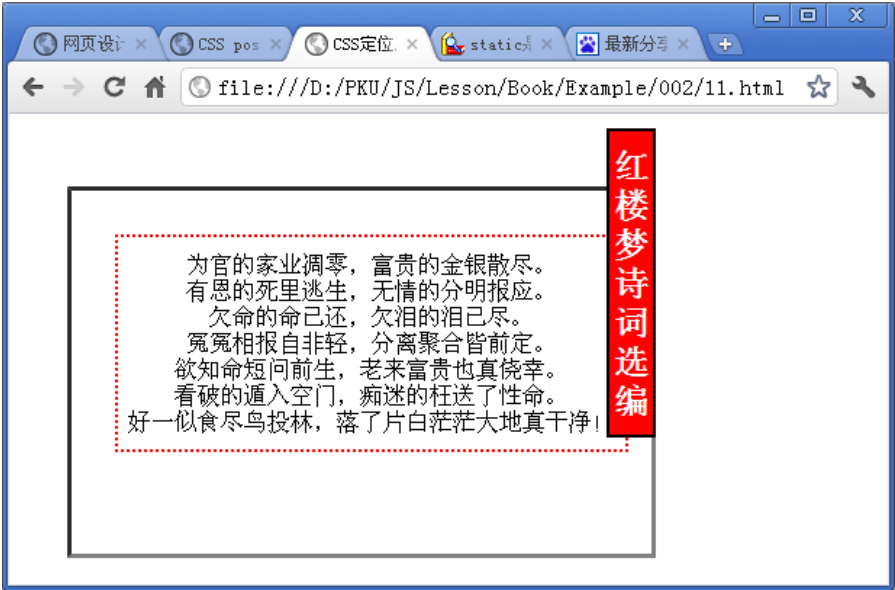


图2-12 CSS定位属性的使用

例程2-16

```
第1行  <HTML>
第2行  <HEAD>
第3行  <TITLE>CSS定位属性的使用</TITLE>
第4行  </HEAD>
第5行  <BODY>
第6行  <Div style="position:absolute;top:50px;left:40px;border:3px inset gray;
第7行  width:400px;height:250px">
第8行  <Div style="position:relative;left:30px;top:30px;width:350px;
第9行  border:2px dotted red;text-align:Center;padding:1
      0px 0px 10px 0px">
第10行  为官的家业凋零，富贵的金银散尽。<BR>
第11行  有恩的死里逃生，无情的分明报应。<BR>
第12行  欠命的命已还，欠泪的泪已尽。<BR>
第13行  冤冤相报自非轻，分离聚合皆前定。<BR>
第14行  欲知命短问前生，老来富贵也真侥幸。<BR>
第15行  看破的遁入空门，痴迷的枉送了性命。<BR>
```

```
第16行      好一似食尽鸟投林，落了片白茫茫大地真干净！
第17行      </Div>
第18行      </Div>
第19行      <Div style="position:fixed;top:10px;left:412px;border:2px solid black;width:30px;
第20行      text-align:center;background-color:red;font-size:
                24px;color:white;
第21行      font-weight:bold;padding:10px 0px 10px 0px">
第22行      红楼梦诗词选编
第23行      </Div>
第24行      </BODY>
第25行      </HTML>
```

CSS定位属性的使用，使网页布局能力大为增强。在网页设计中，经常采用双列或者三列式，图2-13是三列式，其实现代码如例程2-17所示。

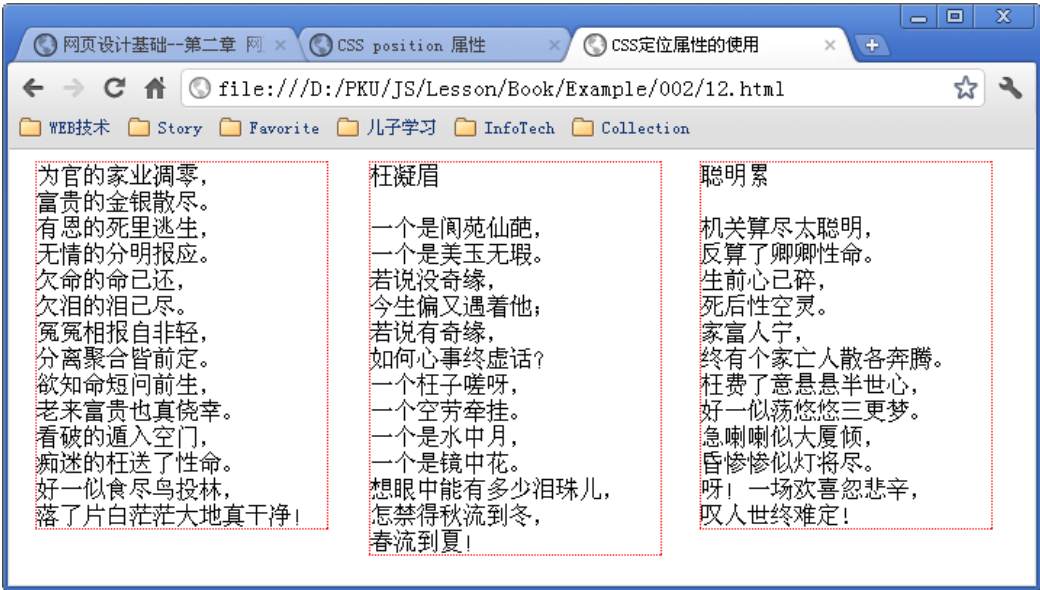


图2-13 CSS定位属性实现三列式网页布局

例程2-17的第7、16、26行分别用于定位三列，第一列采用left向右偏移，第二列left属性偏移量更大，因此更加向右偏移，第三列采用right方式，向左偏移。采用Div+CSS方式的网页布局，是较为流行的设计路线。

例程2-17

```
第1行      <HTML>
第2行      <HEAD>
第3行      <TITLE>CSS定位属性的使用</TITLE>
第4行      </HEAD>
第5行      <BODY>
第6行      <Div style="position:relative;width:680px">
第7行      <Div style="position:absolute;left:10px;border:1px dotted red;width:200px;">
第8行      为官的家业凋零，<BR>富贵的金银散尽。<BR>
第9行      有恩的死里逃生，<BR>无情的分明报应。<BR>
第10行     欠命的命已还，<BR>欠泪的泪已尽。<BR>
第11行     冤冤相报自非轻，<BR>分离聚合皆前定。<BR>
第12行     欲知命短问前生，<BR>老来富贵也真侥幸。<BR>
第13行     看破的遁入空门，<BR>痴迷的枉送了性命。<BR>
第14行     好一似食尽鸟投林，<BR>落了片白茫茫大地真干净！
第15行     </Div>
```


第16行	<Div style="position:absolute;left:240px;border:1px dotted red;width:200px;">
第17行	枉凝眉
第18行	一个是阆苑仙葩， 一个是美玉无瑕。
第19行	若说没奇缘， 今生偏又遇着他；
第20行	若说有奇缘， 如何心事终虚话？
第21行	一个枉子嗟呀， 一个空劳牵挂。
第22行	一个是水中月， 一个是镜中花。
第23行	想眼中能有多少泪珠儿，
第24行	怎禁得秋流到冬， 春流到夏！
第25行	</Div>
第26行	<Div style="position:absolute;right:10px;border:1px dotted red;width:200px;">
第27行	聪明累
第28行	机关算尽太聪明， 反算了卿卿性命。
第29行	生前心已碎， 死后性空灵。
第30行	家富人宁， 终有个家亡人散各奔腾。
第31行	枉费了意悬悬半世心，
第32行	好一似荡悠悠三更梦。
第33行	急喇喇似大厦倾， 昏惨惨似灯将尽。
第34行	呀！一场欢喜忽悲辛， 叹人世终难定！
第35行	</Div>
第36行	</Div>
第37行	</BODY>
第38行	</HTML>

2、z-index属性

z-index属性用于设置元素的叠放顺序，相当于设置z轴编号。z-index属性值高的元素位于低元素的顶部，其默认值为0。 z-index属性值为auto、number、inherit三种，其中auto为默认值，number为具体的数值，如1、5、8等。 inherit表示其属性值继承自父元素。当z-index为正值时，则离用户更近，为负数则表示离用户更远。 值得注意的是，z-index仅对position属性值为absolute、relative、fixed的元素有效。 图2-14是z-index的应用，例程2-18是其实现代码。



图2-14 z-index属性的使用

观察图2-14，“宝黛夜读西厢”位于图片之上，原因是在例程2-18第8行，将该Div的z-index属性设置1， 且调整其left和top属性值。如果将z-index设置-1，则这段文字将被置于图片之下，从而看不到文字。

例程2-18

第1行	<HTML>
-----	--------

第2行	<HEAD>
第3行	<TITLE>z-index属性的使用</TITLE>
第4行	</HEAD>
第5行	<BODY>
第6行	<Div style="width:200px">
第7行	
第8行	<Div style="position:absolute;z-index:1;left:8px;top:200px;background-color:#E FEFEF;
第9行	width:200px;text-align:Center;line-height:40px;fo nt-size:24px;">
第10行	宝黛夜读西厢
第11行	</Div>
第12行	</Div>
第13行	</BODY>
第14行	</HTML>

3、display和visibility属性

display和visibility都是显示与否的属性，当display属性值为none，或visibility属性值为hidden时， HTML标记将不显示，二者区别是display设为none时，元素不显示，其所占空间释放， 后继元素顺序占据其区域，而visibility设为hidden时，元素不被显示，但其所占区域不会被其他元素所占据。 图2-15是其演示效果。

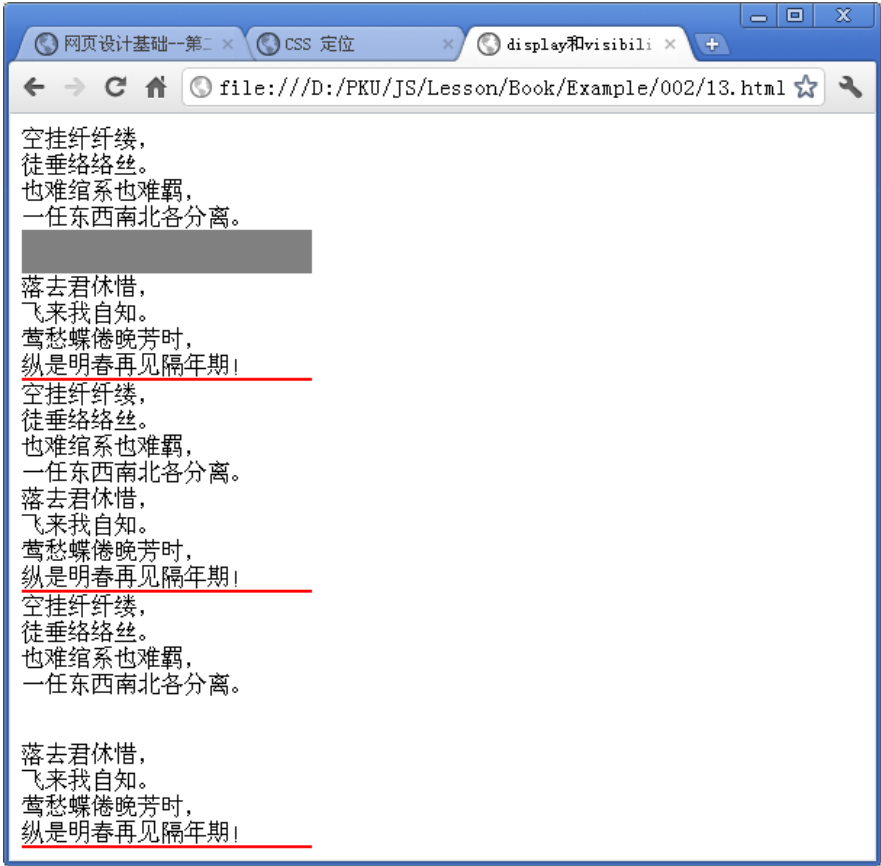


图2-15 display和visibility属性的使用

例程2-19是图2-15的实现代码。从代码中可以看出，第28、43行几乎完全相同， 不同的是第28行设置display属性为none，而第43行设置visibility属性为hidden。对应到图2-15， 就会发现采用display模式时，该区域完全隐藏，后面的文字自动上升占据空间； 而采用visibility则位置处为空白，内容已不再能看见。

例程2-19

第1行	<HTML>
第2行	<HEAD>

第3行	<TITLE>display和visibility属性的使用</TITLE>
第4行	</HEAD>
第5行	<BODY>
第6行	<Div style="width:200px;border-bottom:2px solid red">
第7行	<Div>
第8行	空挂纤纤缕，
第9行	徒垂络络丝。
第10行	也难绾系也难羁，
第11行	一任东西南北各分离。
第12行	</Div>
第13行	<Div style="height:30px;background-color:gray"></Div>
第14行	<Div>
第15行	落去君休惜，
第16行	飞来我自知。
第17行	莺愁蝶倦晚芳时，
第18行	纵是明春再见隔年期！
第19行	</Div>
第20行	</Div>
第21行	<Div style="width:200px;border-bottom:2px solid red">
第22行	<Div>
第23行	空挂纤纤缕，
第24行	徒垂络络丝。
第25行	也难绾系也难羁，
第26行	一任东西南北各分离。
第27行	</Div>
第28行	<Div style="display:none;height:30px;background-color:gray"></Div>
第29行	<Div>
第30行	落去君休惜，
第31行	飞来我自知。
第32行	莺愁蝶倦晚芳时，
第33行	纵是明春再见隔年期！
第34行	</Div>
第35行	</Div>
第36行	<Div style="width:200px;border-bottom:2px solid red">
第37行	<Div>
第38行	空挂纤纤缕，
第39行	徒垂络络丝。
第40行	也难绾系也难羁，
第41行	一任东西南北各分离。
第42行	</Div>
第43行	<Div style="visibility:hidden;height:30px;background-color:gray"></Div>
第44行	<Div>
第45行	落去君休惜，
第46行	飞来我自知。
第47行	莺愁蝶倦晚芳时，
第48行	纵是明春再见隔年期！

第49行	</Div>
第50行	</Div>
第51行	</BODY>
第52行	</HTML>

visibility属性值除了hidden外，还可以取visible(显示，默认)和inherit(继承父元素该属性值)。visible和hidden相反，hidden是隐藏内容，visible是显示内容。

display属性值除了none(隐藏)外，还可以不少属性值，常用的有inline(行显示)、block(块显示)、inline-block(行内块显示)以及inherit(继承父元素该属性值)等。

inline/block显示模式是HTML元素在浏览器中的两种显示方式，是HTML元素的两个重要特征。当元素采用inline模式时，该元素前后没有换行符，元素的内容与前后元素(如果存在)都保持在同一行上。当元素采用block模式时，将自动在该元素前后产生换行符，元素前后内容将分行显示。很多元素都有默认的inline或者block属性，如p、div默认是block元素，而span、img、a默认是inline元素。元素和inline/block特征，可以通过display属性调整。让inline元素有block特征，让block元素有inline特征。在图2-16中，“黑底白字”两个区域是用Div实现。Div是典型的Block元素，但由于将其设置为inline-block，所以两个元素并排排列，既有inline特征，也有block特征。而“灰底红字”两个区域，仍然是用Div实现，但将其设置为inline特征，可以发现“咏螃蟹”三字总是出现在行尾，这是同行显示特征导致，其余行分行是由于BR特征。最后的“黑底红字”连个区域是用Span实现，但将其display属性设置block，典型“行显示”元素具有了block特征。图2-16的代码如例程2-20所示。

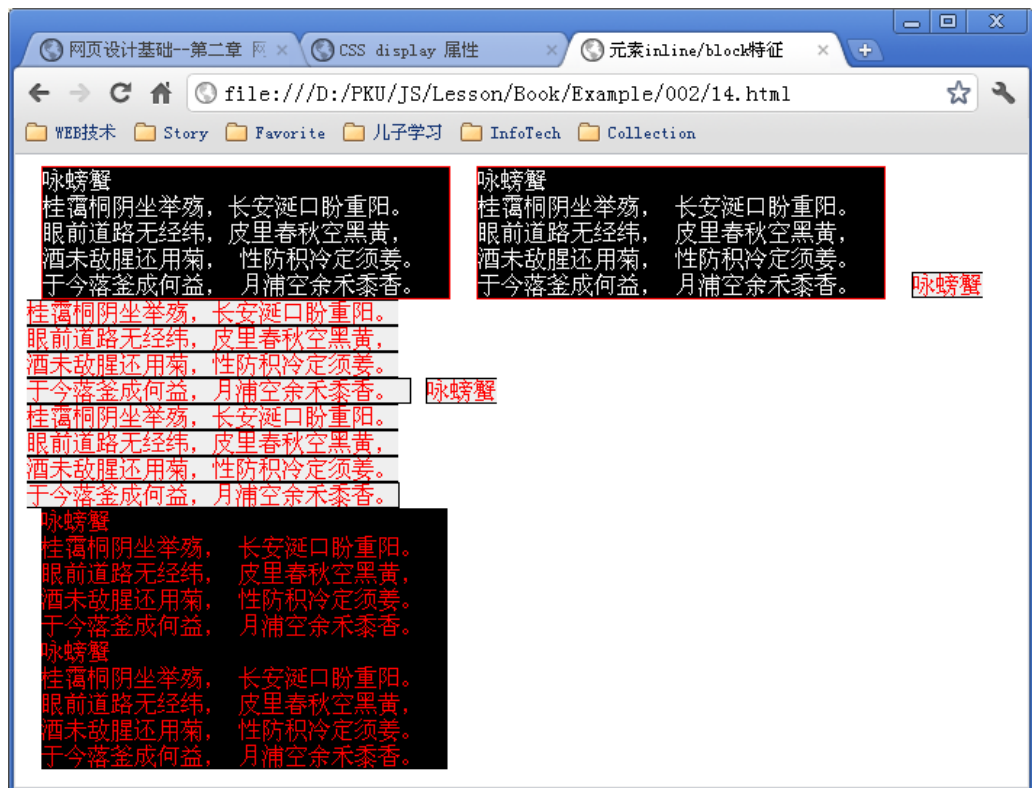


图2-16 元素的inline/block特征效果

例程2-20

第1行	<HTML>
第2行	<HEAD>
第3行	<TITLE>元素inline/block特征</TITLE>
第4行	<Style type=text/css>
第5行	. Inline_Block_Mode{display:inline-block;border:1px solid red;width:280px;
第6行	margin-left:10px;background-color:black;color:white}
第7行	
第8行	. Inline_Mode{display:inline;border:1px solid black;width:280px;
第9行	margin-left:10px;background-color:#EFEFEF;color:red}

第10行
第11行

第12行
第13行
第14行
第15行
第16行
第17行
第18行
第19行
第20行
第21行
第22行
第23行
第24行
第25行
第26行
第27行
第28行
第29行
第30行
第31行
第32行
第33行
第34行
第35行
第36行
第37行
第38行
第39行
第40行
第41行
第42行
第43行
第44行
第45行
第46行
第47行
第48行
第49行
第50行
第51行
第52行
第53行
第54行

```
span{display:block;width:280px;background-color:black;color:Red;margin-left:10p
x;}

</Style>

</HEAD>

<BODY>

<Div class="Inline_Block_Mode">

    咏螃蟹<BR>

    桂霨桐阴坐举殇，长安涎口盼重阳。<BR>

    眼前道路无经纬，皮里春秋空黑黄，<BR>

    酒未敌腥还用菊，性防积冷定须姜。<BR>

    于今落釜成何益，月浦空余禾黍香。

</Div>

<Div class="Inline_Block_Mode">

    咏螃蟹<BR>

    桂霨桐阴坐举殇，长安涎口盼重阳。<BR>

    眼前道路无经纬，皮里春秋空黑黄，<BR>

    酒未敌腥还用菊，性防积冷定须姜。<BR>

    于今落釜成何益，月浦空余禾黍香。

</Div>

<Div class="Inline_Mode">

    咏螃蟹<BR>

    桂霨桐阴坐举殇，长安涎口盼重阳。<BR>

    眼前道路无经纬，皮里春秋空黑黄，<BR>

    酒未敌腥还用菊，性防积冷定须姜。<BR>

    于今落釜成何益，月浦空余禾黍香。

</Div>

<Div class="Inline_Mode">

    咏螃蟹<BR>

    桂霨桐阴坐举殇，长安涎口盼重阳。<BR>

    眼前道路无经纬，皮里春秋空黑黄，<BR>

    酒未敌腥还用菊，性防积冷定须姜。<BR>

    于今落釜成何益，月浦空余禾黍香。

</Div>

<Span>

    咏螃蟹<BR>

    桂霨桐阴坐举殇，

    长安涎口盼重阳。

    眼前道路无经纬，

    皮里春秋空黑黄，

    酒未敌腥还用菊，

    性防积冷定须姜。

    于今落釜成何益，

    月浦空余禾黍香。

</span>

<Span>
```

第55行	咏螃蟹
第56行	桂霭桐阴坐举殇，
第57行	长安涎口盼重阳。
第58行	眼前道路无经纬，
第59行	皮里春秋空黑黄，
第60行	酒未敌腥还用菊，
第61行	性防积冷定须姜。
第62行	于今落釜成何益，
第63行	月浦空余禾黍香。
第64行	
第65行	</BODY>
第66行	</HTML>

观察例程2-20第8、9两行，可以发现虽然设置了margin-left和width属性，但是并没有起作用， 其原因是当元素为inline特征时，块属性对其不发挥作用。

4、float和clear属性

float属性定义元素的浮动方向，该属性的取值有left(左)、right(右)、 none(无浮动，默认)以及inherit(继承父元素该属性值)。

当HTML元素应用浮动属性时，元素会成为“块显示”， 无论该元素默认是“行显示”还是“块显示”。该属性常用于图像(其他元素当然也可)， 使文本围绕在图像周围。如图2-17， 文字两侧如有对联呈现，就是采用float属性实现， 例程2-21是其实现代码。



图2-17 float属性的使用

例程2-21

第1行	<HTML>
第2行	<HEAD>
第3行	<TITLE>float和clear属性</TITLE>
第4行	<Style type=text/css>
第5行	Div{margin:10px;border:5px inset red;font-size:24px;width:34px;text-align:Center}
第6行	P{font-size:16px;line-height:24px;text-indent:2em}
第7行	</Style>
第8行	</HEAD>
第9行	<BODY>

第10行

<Div style="float:right;">世事洞明皆学问</Div>

第11行

<Div style="float:left;">人情练达即文章</Div>

第12行

<P>红楼梦现存的版本有十几个，其中最早的是1754年（清乾隆十九年）的《脂砚斋重评石头记》。之后又有己卯本、庚辰本等古本。程甲本、程乙本以及人民文学出版社的版本前80回是曹雪芹著，后40回为高鹗续写。

<P>《红楼梦》是一部有高度思想性和高度艺术性的伟大作品，可以代表古典小说艺术的最高成就，也是中国古代文学的典范之作。它以荣国府的日常生活为中心，以宝玉、黛玉、宝钗的爱情婚姻悲剧及大观园中点滴琐事为主线，以金陵贵族名门贾、王、薛、史四大家族由鼎盛走向衰亡的历史为暗线。展现了穷途末路的封建社会终将走向灭亡的必然趋势。并以其曲折隐晦的表现手法，凄凉深切的情感格调，强烈高远的思想底蕴，在我国古代民俗，封建制度，社会图景，建筑金石等各领域皆有不可替代的研究价值，达到我国古典小说的高峰。关于红楼梦旨义思想的研究历来众说纷纭。自红楼梦成书之初脂砚斋即有评论：悼专制社会红颜之苦，故谓之悼红轩；索隐派的代表人物蔡元培《红楼梦索隐》：揭清之失，悼明之亡。考证派的胡适在《红楼梦考证》中说：红楼梦是曹雪芹隐去真实之后的自述。</P>

第13行

<P>红楼梦在流传初期是通过名为《石头记》的手抄本形式流传，自乾隆年间始，关于《红楼梦》的续作纷纷出笼。《红楼梦》被誉为“我国封建社会的百科全书”现在成为“红学”。</P>

第14行

</BODY>

第15行

</HTML>

clear属性是禁止float属性元素出现，其属性取值有none(不禁止浮动元素)、left(禁止左侧)、right(禁止右侧)、 both(左右都禁止)以及inherit(继承自父元素)。当禁止发生时，内容将出现在被禁止元素的下侧。 如图2-18所示。



图2-18 clear使用效果

图2-18中共有两个段落，都将clear属性设置为right，即清楚右侧的浮动元素。在第一段，有左侧浮动元素， 不在清楚范围。在第二段，由于设置了右侧清楚，右侧浮动无效，因此，其内容出现在右侧浮动元素之下。

例程2-22

第1行

<HTML>

第2行

<HEAD>

第3行

<TITLE>float和clear属性</TITLE>

第4行

<Style type=text/css>

第5行

Div{margin:20px;border:2px inset red;font-size:24px;width:200px;text-align:Center}

第6行

P{font-size:16px;line-height:24px;text-indent:2em}

| | |
|------|--|
| 第7行 | </Style> |
| 第8行 | </HEAD> |
| 第9行 | <BODY> |
| 第10行 | <Div style="float:left;">人情练达即文章</Div><P style="clear:right">红楼梦现存的版本有十几个，其中最早的是1754年（清乾隆十九年）的《脂砚斋重评石头记》。之后又有己卯本、庚辰本等古本。程甲本、程乙本以及人民文学出版社的版本前80回是曹雪芹著，后40回为高鹗续写。《红楼梦》是一部有高度思想性和高度艺术性的伟大作品，可以代表古典小说艺术的最高成就，也是中国古代文学的典范之作。它以荣国府的日常生活为中心，以宝玉、黛玉、宝钗的爱情婚姻悲剧及大观园中点滴琐事为主线，以金陵贵族名门贾、王、薛、史四大家族由鼎盛走向衰亡的历史为暗线。展现了穷途末路的封建社会终将走向灭亡的必然趋势。并以其曲折隐晦的表现手法，凄凉深切的情感格调，强烈高远的思想底蕴，在我国古代民俗，封建制度，社会图景，建筑金石等各领域皆有不可替代的研究价值，达到我国古典小说的高峰。关于红楼梦旨义思想的研究历来众说纷纭。自红楼梦成书之初脂砚斋即有评论：悼专制社会红颜之苦，故谓之悼红轩；索隐派的代表人物蔡元培《红楼梦索隐》：揭清之失，悼明之亡。考证派的胡适在《红楼梦考证》中说：红楼梦是曹雪芹隐去真实之后的自述。</P> |
| 第11行 | <Div style="float:right;">世事洞明皆学问</Div><P style="clear:right">红楼梦》在流传初期是通过名为《石头记》的手抄本形式流传，自乾隆年间始，关于《红楼梦》的续作纷纷出笼。《红楼梦》被誉为“我国封建社会的百科全书”现在成为“红学”。</P> |
| 第12行 | </BODY> |
| 第13行 | </HTML> |

第七节 CSS尺寸单位

尺寸单位是CSS属性常用的取值单位，如：font-size、height、width、margin、padding等。 在介绍上述属性时，常用px为单位，除了px外，还有其他单位可供选择。

在CSS中，尺寸单位可以分为绝对单位和相对单位。绝对单位如cm(厘米)、mm(毫米)、pt(点)等， 无论在什么样的输出环境(显示、打印等)都能保持一致。但是由于Web用户的分辨率不一致， 容易导致显示效果达不到预期，因此，绝对单位用的较少。相对单位包括px(像素)、 em(当前字体中字母M的宽度)、ex(当前字体中字母X的高度)，这些单位都是相对于屏幕分辨率， 或者相对于当前使用字号的某些字母的宽度或高度。下表是CSS中可用的相对和绝对尺寸单位。

表2：CSS尺寸单位表

单位	绝对/相对单位	描述
in	绝对单位	英寸，1in=2.53cm
cm	绝对单位	厘米
mm	绝对单位	毫米
pt	绝对单位	磅，1pt =0.0352806cm
pc	绝对单位	派卡，1pc=0.4233cm
em	相对单位	1em 等于当前的字体尺寸。英文中每个字的大小不一样，以M为标准。 2em 等于当前字体尺寸的两倍。 在CSS 中em是常用单位，可以自动适应用户所使用的字体。
ex	相对单位	ex相当于英文字母x的高度(x的高度通常是字体尺寸的一半)。
px	相对单位	像素(计算机屏幕上的一个点)。

除了绝对单位和相对单位外，还可以用%。%总是相对于其他值，其最终尺寸与参照对象有关。 如div的宽度默认整行宽度，如果设定某div的width:80%，则表示其宽度为整行宽度的80%， 而整行宽度的大小与包含该div的标记有关。例程2-15中的第6、7行都应用了%。在第6行中， div被包含在body中，其宽度为body的50%，body的宽度为图2-11中的深灰色区域， 白色区域为div的宽度。当调整浏览器的宽度时，body的宽度会被自动调整，div的宽度也会被自动调整。 在第7行，“好了歌”的大小为20px，而“解注”二字的大小为“好了歌”的60%。 当调整“好了歌”的字号大小时，“解注”二字的大小也随之调整。从这个意义上讲，%是相对尺寸。 例程2-15是多个相对单位、绝对单位以及%的使用，图2-11是其执行效果。

第1行	<HTML>
-----	--------

第2行	<HEAD>
第3行	<TITLE>CSS中%的使用</TITLE>
第4行	</HEAD>
第5行	<BODY style="background-color:#848484">
第6行	<Div style="width:50%;background-color:white" >
第7行	<P style="font-size:20px">好了歌解注</P>
第8行	陋室空堂，当年笏满床，衰草枯杨，曾为歌舞场。
第9行	蛛丝儿结满雕梁，绿纱今又糊在蓬窗上。
第10行	说什么脂正浓，粉正香</s
	pan>，
第11行	如何两鬓又成霜？
第12行	昨日黄土陇头送白骨，今宵红灯帐底卧鸳鸯。
第13行	金满箱，银满箱，展眼乞丐人皆谤。
第14行	正叹他人命不长，那知自己归来丧！
第15行	训有方，保不定日后作强梁。
第16行	择膏粱，谁承望流落在烟花巷！
第17行	因嫌纱帽小，致使锁枷杠，昨怜破袄寒，今嫌紫蟒长。
第18行	乱烘烘你方唱罢我登场，反认他乡是故乡。
第19行	甚荒唐，到头来都是为他人作嫁衣裳。
第20行	</Div>
第21行	</BODY>
第22行	</HTML>



图2-19 %的使用

第八节 CSS的复合属性和单一属性

在CSS中，属性可分为复合和单一两种，color是单一属性，但border属性实际上由border-color(边框颜色)、 border-style(边框样式)和border-width三个属性复合构成。在实际应用过程中，可以设置符合属性， 如border:1px solid red就是设置border-width为1px, border-style为solid, border-color为red， 也可以分别设置如: border-width:5px, 即仅将边框宽度设置为5px, border-color和border-style维持不变。 另外，border属性还可以单独对某一边进行设置， 如: border-bottom:1px solid red表示仅仅将底边设置宽1px、边框样式为solid，边框颜色为red。

border作为复合属性，可以单边设置，也可以分别对border-style、border-width和border-color进行设置(适用各边)。 border属性可以分解为border-top、border-right、border-bottom、border-left(适用单边)， 还可以继续分为border-style、 border-width、border-color(适用各边)、 也还可以分为：border-left-color、border-left-width、border-left-style(其他各边依此类推)。

border是典型的符合属性，出border外，符合属性还有：margin、padding、background、font等。 CSS中常见的符合属性如表3所示。

表3：CSS常用符合属性

Background(背景)属性	
background	在一个声明中设置所有的背景属性。
background-attachment	设置背景图像是否固定或者随着页面的其余部分滚动。
background-color	设置元素的背景颜色。
background-image	设置元素的背景图像。
background-position	设置背景图像的开始位置。
background-repeat	设置是否及如何重复背景图像。
Border(边框)属性	
border	在一个声明中设置所有的边框属性。
border-bottom	在一个声明中设置所有的下边框属性。
border-bottom-color	设置下边框的颜色。
border-bottom-style	设置下边框的样式。
border-bottom-width	设置下边框的宽度。
border-color	设置四条边框的颜色。
border-left	在一个声明中设置所有的左边框属性。
border-left-color	设置左边框的颜色。
border-left-style	设置左边框的样式。
border-left-width	设置左边框的宽度。
border-right	在一个声明中设置所有的右边框属性。
border-right-color	设置右边框的颜色。
border-right-style	设置右边框的样式。
border-right-width	设置右边框的宽度。
border-style	设置四条边框的样式。
border-top	在一个声明中设置所有的上边框属性。
border-top-color	设置上边框的颜色。
border-top-style	设置上边框的样式。
border-top-width	设置上边框的宽度。
border-width	设置四条边框的宽度。
Font(字体)属性	
font	在一个声明中设置所有字体属性。
font-family	规定文本的字体系列。
font-size	规定文本的字体尺寸。
font-size-adjust	为元素规定 aspect 值。
font-style	规定文本的字体样式。
font-variant	规定文本的字体样式。
font-weight	规定字体的粗细。

Magin(外边距)属性	
margin	在一个声明中设置所有外边距属性。
margin-bottom	设置元素的下外边距。
margin-left	设置元素的左外边距。
margin-right	设置元素的右外边距。
margin-top	设置元素的上外边距。
Padding(内边距)属性	
padding	在一个声明中设置所有内边距属性。
padding-bottom	设置元素的下内边距。
padding-left	设置元素的左内边距。
padding-right	设置元素的右内边距。
padding-top	设置元素的上内边距。

第九节 更多CSS属性

CSS是页面美观的基础。除了已讲述的CSS属性外，还有一些属性能提升网页美观。

1、cursor属性

cursor属性用于设定光标形状。浏览器根据元素性质会选择不同的默认光标样式，比如a(链接)通常是手形光标、文本框通常是I形光标等。通过CSS的cursor属性，可以设定光标样式。图2-19是各种鼠标样式及其cursor属性值，到鼠标移动到第3列时，能动态看到鼠标形状。例程2-23是图2-19是实现代码。

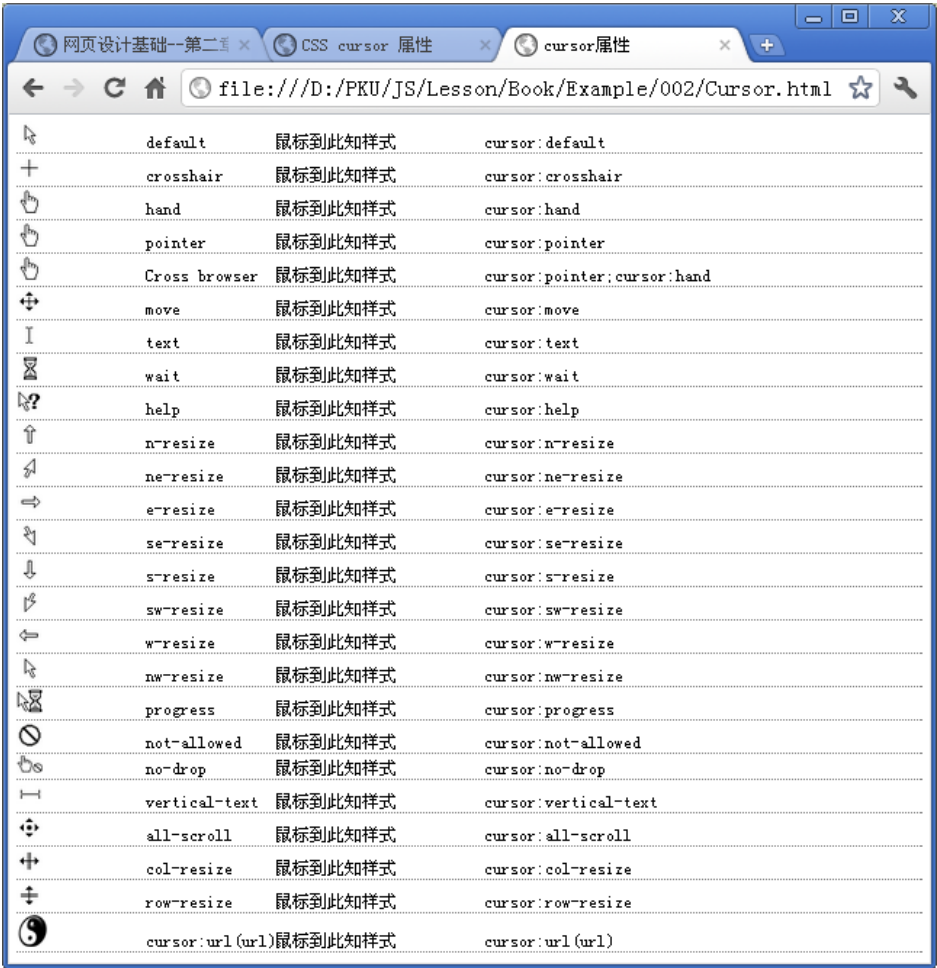


图2-20 cursor属性取值及其光标样式

第2行	<HEAD>
第3行	<TITLE>cursor属性</TITLE>
第4行	<Style type=text/css>
第5行	div{margin-bottom:2px;border-bottom:1px dotted gray}
第6行	span{display:inline-block;width:120px}
第7行	.LastSpan{width:200px}
第8行	</Style>
第9行	</HEAD>
第10行	
第11行	<BODY>
第12行	<div>
第13行	
第14行	default
第15行	鼠标到此知样式
第16行	cursor:default
第17行	</div>
第18行	<div>
第19行	
第20行	crosshair
第21行	鼠标到此知样式
第22行	cursor:crosshair
第23行	</div>
第24行	<div>
第25行	
第26行	hand
第27行	鼠标到此知样式
第28行	cursor:hand
第29行	</div>
第30行	<div>
第31行	
第32行	pointer
第33行	鼠标到此知样式
第34行	cursor:pointer
第35行	</div>
第36行	<div>
第37行	
第38行	Cross browser
第39行	鼠标到此知样式
第40行	cursor:pointer;cursor:hand
第41行	</div>
第42行	<div>
第43行	
第44行	move
第45行	鼠标到此知样式
第46行	cursor:move
第47行	</div>

第48行	<div>
第49行	
第50行	text
第51行	鼠标到此知样式
第52行	cursor:text
第53行	</div>
第54行	<div>
第55行	
第56行	wait
第57行	鼠标到此知样式
第58行	cursor:wait
第59行	</div>
第60行	<div>
第61行	
第62行	help
第63行	鼠标到此知样式
第64行	cursor:help
第65行	</div>
第66行	<div>
第67行	
第68行	n-resize
第69行	鼠标到此知样式
第70行	cursor:n-resize
第71行	</div>
第72行	<div>
第73行	
第74行	ne-resize
第75行	鼠标到此知样式
第76行	cursor:ne-resize
第77行	</div>
第78行	<div>
第79行	
第80行	e-resize
第81行	鼠标到此知样式
第82行	cursor:e-resize
第83行	</div>
第84行	<div>
第85行	
第86行	se-resize
第87行	鼠标到此知样式
第88行	cursor:se-resize
第89行	</div>
第90行	<div>
第91行	
第92行	s-resize
第93行	鼠标到此知样式

第94行	cursor:s-resize
第95行	</div>
第96行	<div>
第97行	
第98行	sw-resize
第99行	鼠标到此知样式
第100行	cursor:sw-resize
第101行	</div>
第102行	<div>
第103行	
第104行	w-resize
第105行	鼠标到此知样式
第106行	cursor:w-resize
第107行	</div>
第108行	<div>
第109行	
第110行	nw-resize
第111行	鼠标到此知样式
第112行	cursor:nw-resize
第113行	</div>
第114行	<div>
第115行	
第116行	progress
第117行	鼠标到此知样式
第118行	cursor:progress
第119行	</div>
第120行	<div>
第121行	
第122行	not-allowed
第123行	鼠标到此知样式
第124行	cursor:not-allowed
第125行	</div>
第126行	<div>
第127行	
第128行	no-drop
第129行	鼠标到此知样式
第130行	cursor:no-drop
第131行	</div>
第132行	<div>
第133行	
第134行	vertical-text
第135行	鼠标到此知样式
第136行	cursor:vertical-text
第137行	</div>
第138行	<div>


```
第139行      <span></span>
第140行      <span>all-scroll</span>
第141行      <span style="width:200px;cursor:all-scroll">鼠标到此知样式</span>
第142行      <span class="LastSpan">cursor:all-scroll</span>
第143行      </div>
第144行      <div>
第145行      <span></span>
第146行      <span>col-resize</span>
第147行      <span style="width:200px;cursor:col-resize">鼠标到此知样式</span>
第148行      <span class="LastSpan">cursor:col-resize</span>
第149行      </div>
第150行      <div>
第151行      <span></span>
第152行      <span>row-resize</span>
第153行      <span style="width:200px;cursor:row-resize">鼠标到此知样式</span>
第154行      <span class="LastSpan">cursor:row-resize</span>
第155行      </div>
第156行      <div>
第157行      <span></span>
第158行      <span>cursor:url(url)</span>
第159行      <span style="width:200px;cursor:url(wait.cur)">鼠标到此知样式</span>
第160行      <span class="LastSpan">cursor:url(url)</span>
第161行      </div>
第162行      </BODY>
第163行      </HTML>
```

各家浏览器都支持cursor属性，也支持绝大部分属性值，但cursor:hand部分浏览器不支持， 可以采用更加通用的pointer。

2、clip

当一幅图像的尺寸大于包含它的元素时，clip 属性(裁剪属性)允许规定元素的可见尺寸， 此时元素会被修剪并显示为这个形状。使用clip属性的元素要求绝对定位(position:absolute)。 此属性的取值有auto(默认)、rect(矩形)、inherit(继承自父元素)。显然，使用该元素时， rect是唯一合理的属性取值，其语法为clip:rect(上 右 下 左)，其中上右下左都是尺寸单位， 如 clip:rect(30px,180px,180px,30px)就是上右下左分别裁剪30px、180px、180px、30px。 图2-20是clip属性使用效果图，例程2-24是其实现代码。



图2-21

第1行	<HTML>
第2行	<HEAD>
第3行	<TITLE>clip属性的使用</TITLE>
第4行	<Style type=text/css>
第5行	Img{inline-block;}
第6行	</Style>
第7行	</HEAD>
第8行	<BODY>
第9行	
第10行	
第11行	</BODY>
第12行	</HTML>

text-indent(首行缩进)、letter-spacing(字母间距)、word-spacing(单词间距)

text-indent用于首行缩进，如:text-indent:20px表示首行文本缩进20px。letter-spacing用于调整字母间距，和word-spacing相比，letter-spacing调整的是一个单词内每个字母间的距离，而word-spacing则用于调整单词之间的距离。对于汉语来讲，letter-spacing用于调整汉字与汉字之间的距离，word-spacing则几乎没用。CSS判断word的标准是连续文字之间的空白，称之为word，而汉字通常是连续文本，很少产生文字间的空白，因此word-spacing通常是不适合汉语。

第十节 小结

CSS是网页美观的基础，通过CSS可以从不同角度美化网页。

1. CSS的书写位置。CSS可以写在元素标记的属性内称之为Inline-Style，也可以写在Head的Style标记内，称之为Head-Style，也可以写在一个专门的Style文件中，称之为File-Style；
2. 尽量不采用Inline-Style模式。Head-Style和File-Style更加易于维护管理。美容与样式的分离，是网页制作的良好习惯；
3. Inline-Style的作用范围就是其使用元素，通过HTML标记名称、Class、ID可以在Head-Style和File-Style修饰对应的HTML标记。在Head-Style和File-Style中，class名称前需要英文句点(.)，ID名称前须有英文#符号；
4. 在Head-Style和File-Style中，通常用通配符(*)设置基本属性，使网页内所有元素都相同的特征，如网页字体的默认大小、字体颜色等；
5. A(链接)元素还可以设置伪类CSS属性，其设置顺序简称LVHA，尤其Hover伪类，能设置鼠标指向链接时的显示特征，基于此，常用A元素做菜单效果；
6. HTML元素根据文本流中出现的先后顺序尤其默认位置，但可以通过position属性设置其定位属性。当将Position设置为relative和absolute属性时，可以通过left、right、top、bottom等属性确定其位置。另外，float和clear属性也与元素的定位密切相关。通过Div标记以及定位属性的配合，可以将网页分成多个独立的区域，完成网页布局；
7. 网页的尺寸可以分为绝对单位和相对单位，常用em和px，分别对应绝对尺寸和相对尺寸；
8. CSS属性可以分为符合属性和单一属性，复合属性是由多个单一属性所构成。复合属性可以一次设定多个属性，但也可以通过单一属性分别设置；
9. CSS属性较多，没有必要一次性全部牢记，但最好能逐一感知每个属性的效果。当需要某个效果，知道能通过CSS属性设置实现，并适当翻翻参考书了解其细节并逐步掌握。

学习建议：

1. 上机实践将大大促进学习效果，虽不必一次性记忆CSS全部属性，但通过上机实践感知每个属性的效果非常有必要；
2. 通过上机实践感知Head-Style与Inline-Style的异同，特别是当调整设计效果时，感知两种方式优劣性；
3. 通过多页面实践，感知File-Style在保持页面风格一致性方面的重要性；
4. 当前网页设计多Div+CSS实现多列式布局，宜通过上机实现其效果；
5. 当前已学习内容能实现绝大部分网站的静态效果，寻找知名网站并实现其首页效果，将提升学习兴趣。

更多了解

一、CSS版本