# 第5天【CSS基础样式和CSS3选择器】

## 主要内容

1. 背景属性
2. 字体属性
3. 文本属性
4. 列表属性
5. 表格属性
6. 其他属性
7. 关系选择器
8. 伪类选择器
9. 伪对象选择器
10. 属性选择器

## 学习目标

|  |  |  |
| --- | --- | --- |
| 节数 | 知识点 | 要求 |
| 第一节 （背景属性） | 背景属性 | 掌握 |
| 第二节（字体和文本） | 字体属性 | 掌握 |
| 文本属性 | 掌握 |
| 第三节（列表和表格） | 列表属性 | 掌握 |
| 表格属性 | 熟悉 |
| 第四节（其他属性） | Line-height | 掌握 |
| Opacity | 熟悉 |
| Overflow | 掌握 |
| Letter-spacing | 熟悉 |
| Text-overflow | 掌握 |
| Vertical-align | 了解 |
| 第五节（选择器） | 关系选择器 | 掌握 |
| 伪类选择器 | 掌握 |
| 伪对象选择器 | 掌握 |
| 属性选择器 | 了解 |

1. **背景属性**

CSS背景属性主要有以下几个

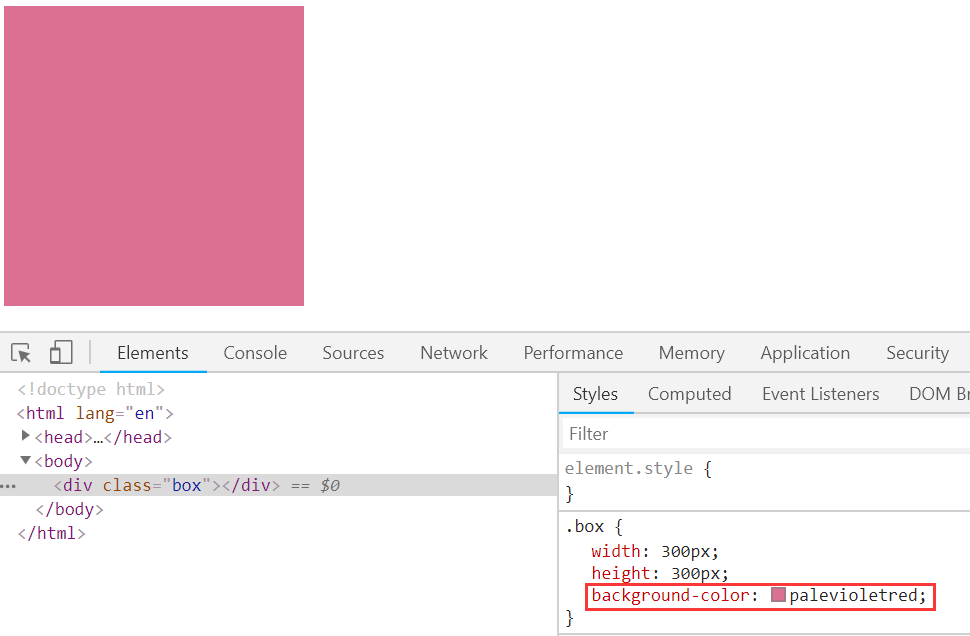


**1.1background-color属性**

该属性设置背景颜色



|  |
| --- |
| <div class="box"></div>  .box{  width: 300px;  height: 300px;  background-color: palevioletred;  } |



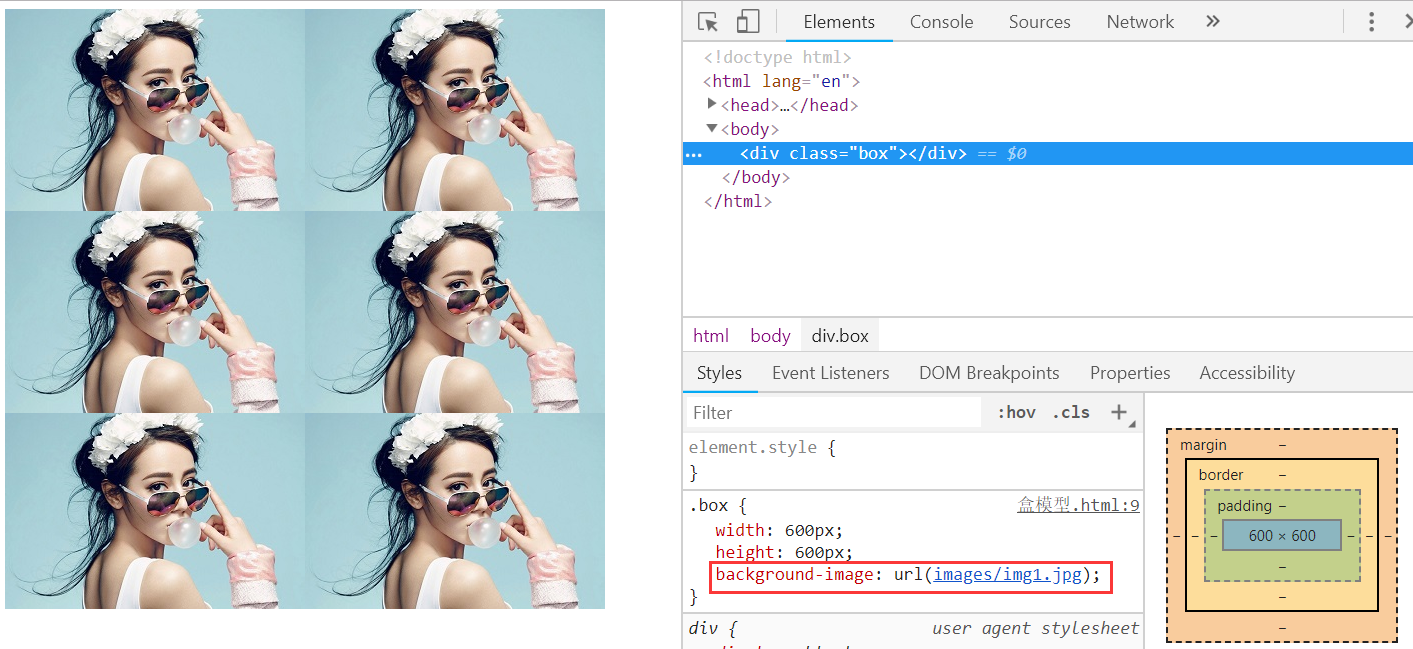
**1.2background-image属性**

设置元素的背景图像。

元素的背景是元素的总大小，包括填充和边界（不包括外边距）。默认情况下background-image属性放置在元素的左上角，如果图像不够大的话会在垂直和水平方向平铺图像，如果图像大小超过元素大小从图像的左上角显示元素大小的那部分。



|  |
| --- |
| <div class="box"></div>  .box{  width: 600px;  height: 600px;  background-image: url("images/img1.jpg");  } |

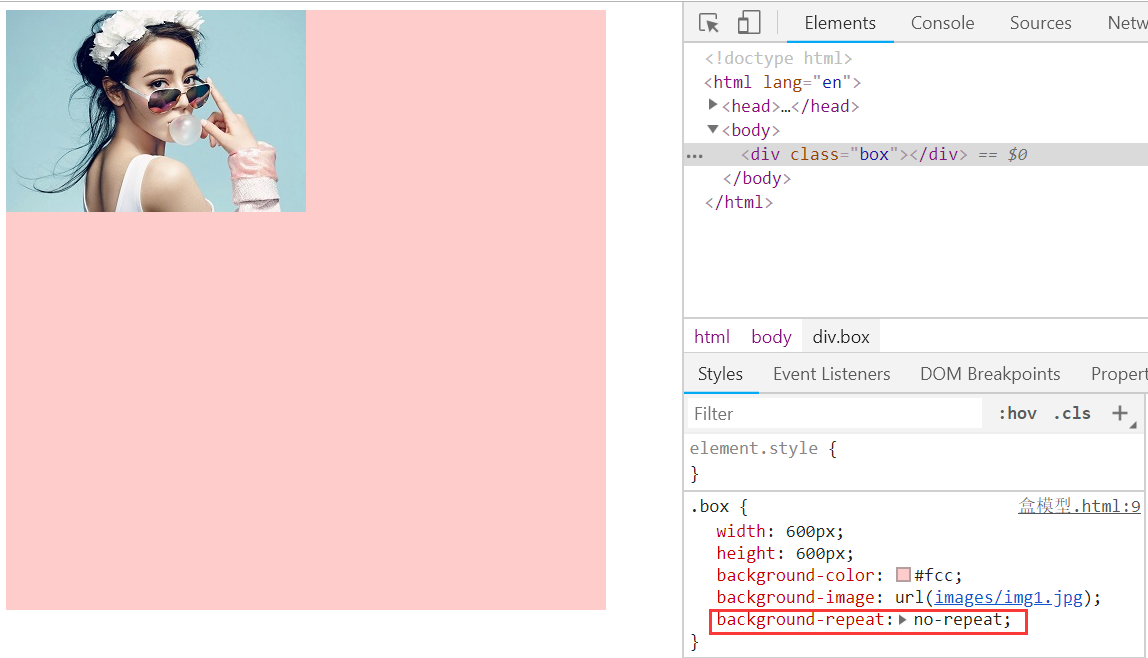


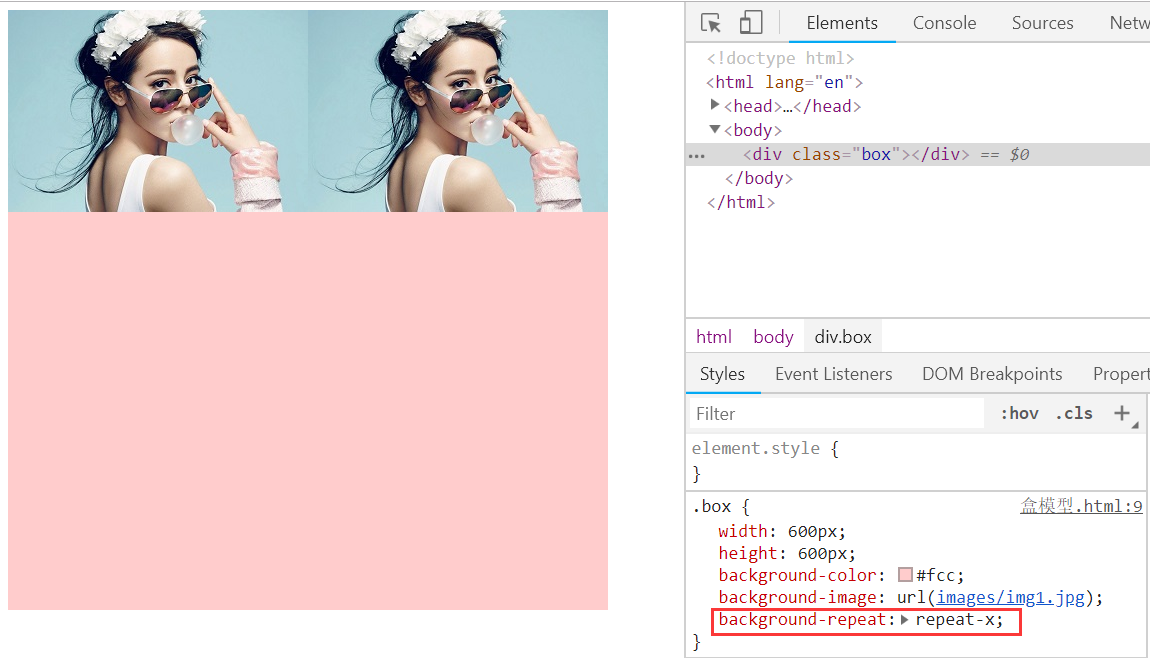
**1.3background-repeat属性**

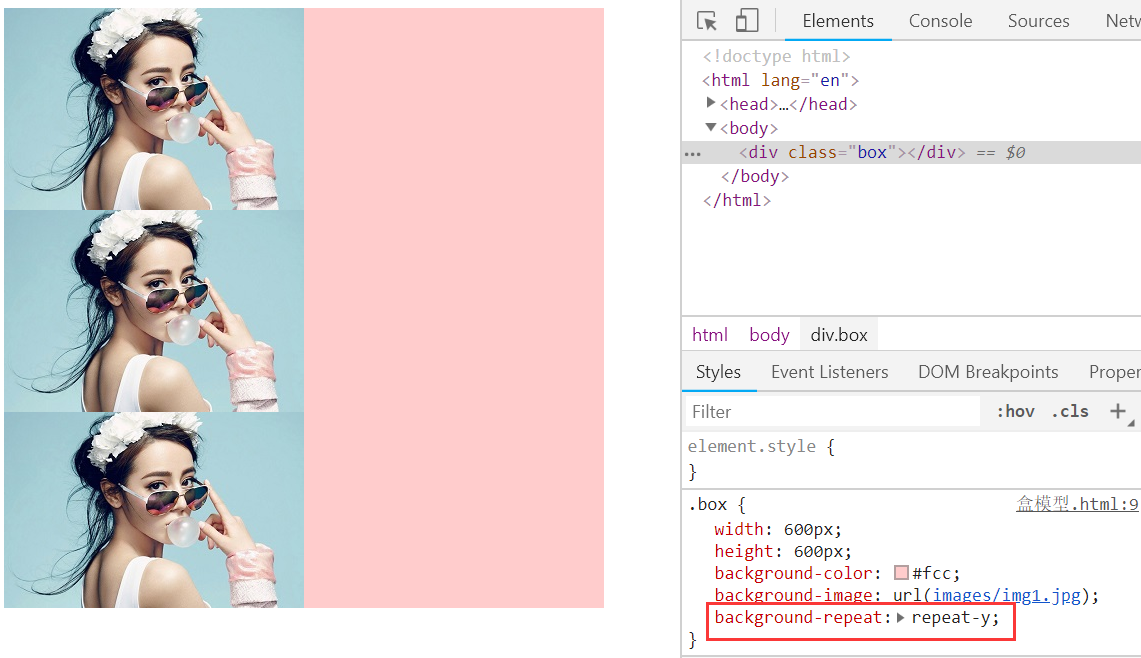
该属性设置如何平铺背景图像



|  |
| --- |
| .box{  width: 600px;  height: 600px;  background-color: #fcc;  background-image: url("images/img1.jpg");  background-repeat: no-repeat;  } |

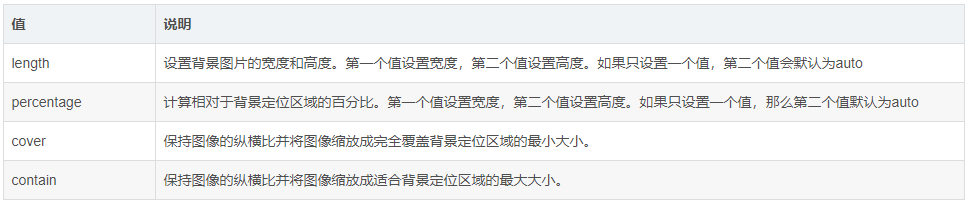




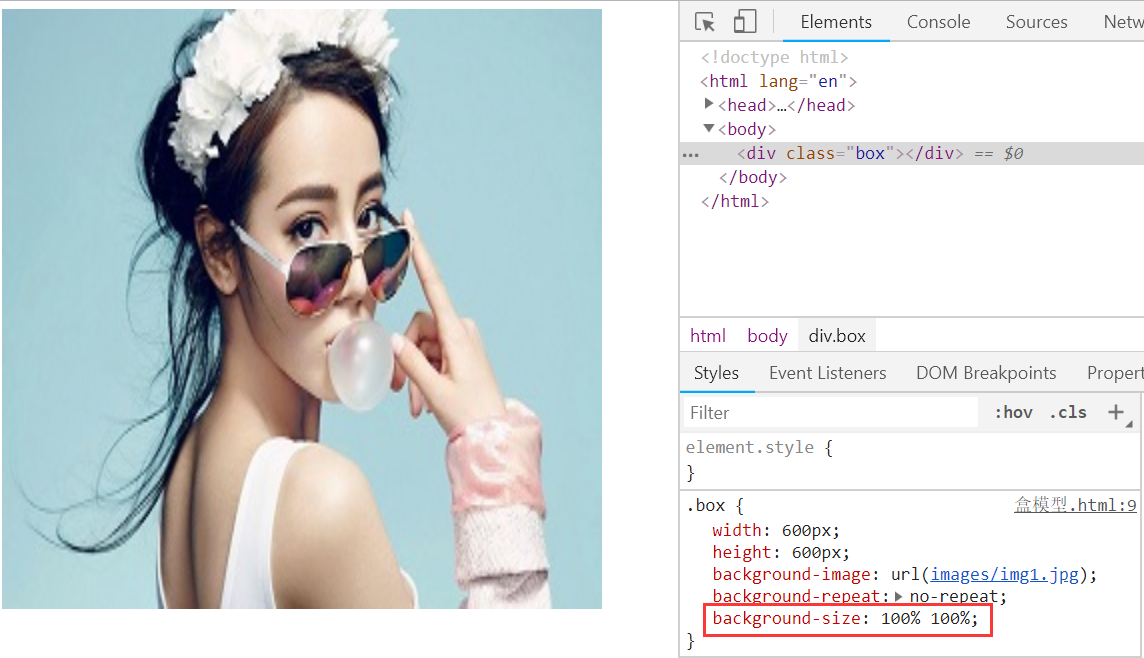


**1.4background-size属性**

该属性设置背景图像的大小



|  |
| --- |
| .box{  width: 600px;  height: 600px;  background-image: url("images/img1.jpg");  background-repeat: no-repeat;  background-size: 100% 100%;  } |

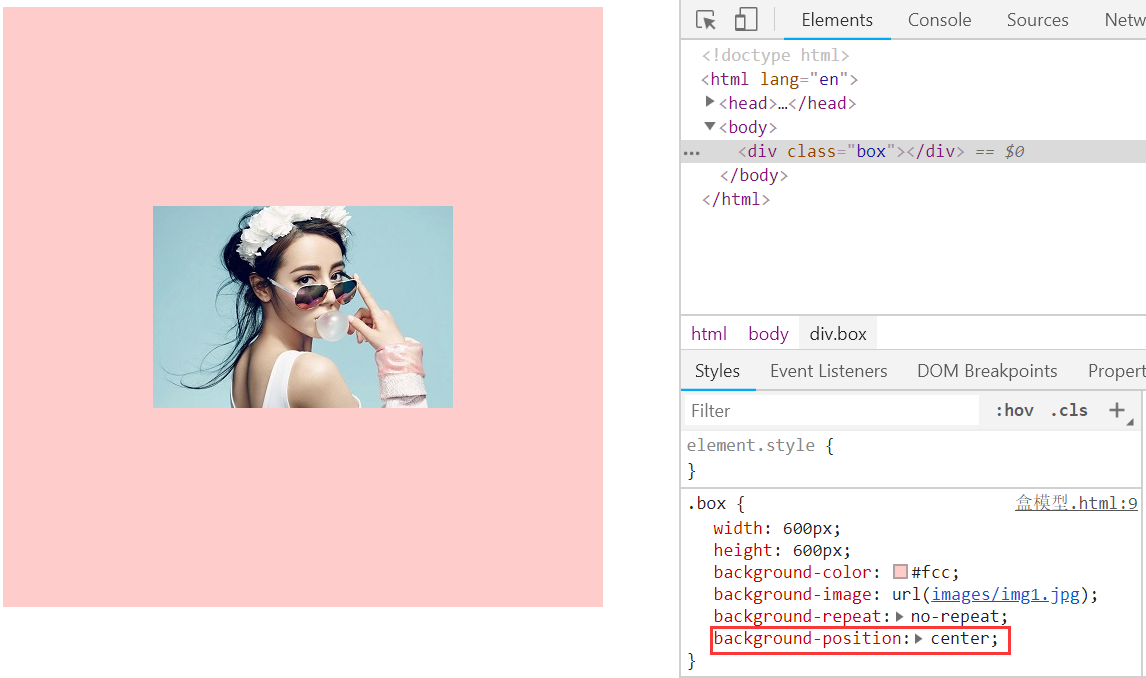


**1.5background-position属性**

该属性设置背景图像的起始位置，其默认值是：0% 0%。



|  |
| --- |
| .box{  width: 600px;  height: 600px;  background-color: #fcc;  background-image: url("images/img1.jpg");  background-repeat: no-repeat;  background-position: center;  } |



**1.6background-attachment属性**

该属性设置背景图像是否固定或者随页面滚动。简单来说就是一个页面有滚动条的话，滑动滚动条背景是固定的还是随页面滑动的。

属性值：



**1.7background属性**

background 简写属性在一个声明中设置所有的背景属性。

可以设置如下属性：

background-color

background-position

background-size

background-repeat

background-attachment

background-image

如果不设置其中的某个值，也不会出问题，取默认值，比如：

background:url('smiley.gif') no-repeat; 也是允许的

1. **字体属性**

CSS字体属性定义字体，加粗，大小，文字样式。



**2.1color**

规定文本的颜色

|  |
| --- |
| div{ color:red;}  div{ color:#ff0000;}  div{ color:rgb(255,0,0);}  div{ color:rgba(255,0,0,.5);} |

**2.2font-size**

设置文本的大小。

能否管理文字的大小，在网页设计中是非常重要的。但是，你不能通过调整字体大小使段落看上去像标题，或者使标题看上去像段落。

请务必使用正确的HTML标签，就<h1> - <h6>表示标题和<p>表示段落：

字体大小的值可以是绝对或相对的大小。

如果你不指定一个字体的大小，默认大小和普通文本段落一样，是16像素（16px=1em）。

|  |
| --- |
| h1 {font-size:40px;} h2 {font-size:30px;} p {font-size:14px;} |

**2.3font-weight**

设置文本的粗细



|  |
| --- |
| H1 {font-weight:normal;} div{font-weight:bold;} p{font-weight:900;} |

**2.4font-style**

指定文本的字体样式



**2.5font-family**

font - family属性指定一个元素的字体。

font-family 可以把多个字体名称作为一个"回退"系统来保存。如果浏览器不支持第一个字体，则会尝试下一个。

注意： 每个值用逗号分开。

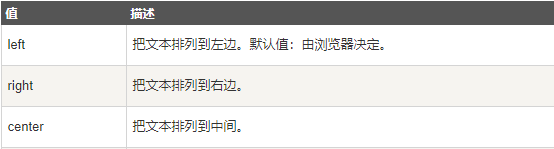
注意: 如果字体名称包含空格，它必须加上引号。

|  |
| --- |
| font-family:"Microsoft YaHei","Simsun","SimHei"; |

1. **文本属性**

**3.1text-align**

指定元素文本的水平对齐方式。



|  |
| --- |
| h1 {text-align:center} h2 {text-align:left} h3 {text-align:right} |

**3.2text-decoration**

text-decoration 属性规定添加到文本的修饰，下划线、上划线、删除线等。



|  |
| --- |
| h1 {text-decoration:overline}  h2 {text-decoration:line-through}  h3 {text-decoration:underline} |

**3.3text-transform**

text-transform 属性控制文本的大小写。



|  |
| --- |
| h1 {text-transform:uppercase;} h2 {text-transform:capitalize;} p {text-transform:lowercase;} |

**3.4text-indent**

text-indent 属性规定文本块中首行文本的缩进。

注意： 负值是允许的。如果值是负数，将第一行左缩进。

|  |
| --- |
| p{ text-indent:50px; } |

1. **列表属性**

在HTML中，有两种类型的列表：

无序列表 - 列表项标记用特殊图形（如小黑点、小方框等）

有序列表 - 列表项的标记有数字或字母

**4.1list-style-type**

list-style-type 属性设置列表项标记的类型。



|  |
| --- |
| ul.a {list-style-type: circle;}  ul.b {list-style-type: square;} |

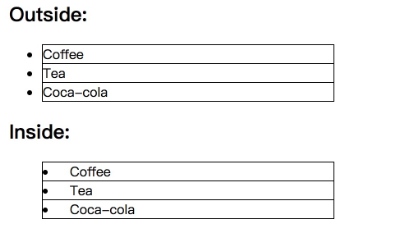
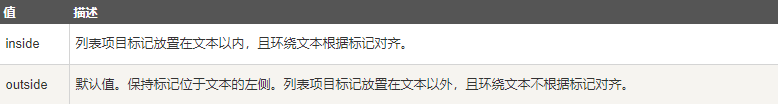
**4.2list-style-image**

list-style-image 属性使用图像来替换列表项的标记。

|  |
| --- |
| ul { list-style-image: url('sqpurple.gif'); } |

**4.3list-style-position**

list-style-position属性指示如何相对于对象的内容绘制列表项标记。



|  |
| --- |
| ul { list-style-position: inside; } |

**4.4 list-style**

list-style 简写属性在一个声明中设置所有的列表属性。

可以设置的属性（按顺序）： list-style-type, list-style-position, list-style-image.

可以不设置其中的某个值，比如 "list-style:circle inside;" 也是允许的。未设置的属性会使用其默认值。

|  |
| --- |
| ul { list-style: none;} |

1. **表格属性**

使用 CSS 可以使 HTML 表格更美观。

**5.1表格边框**

指定CSS表格边框，使用border属性。

下面的例子指定了一个表格的Th和TD元素的黑色边框：

|  |
| --- |
| table, th, td { border: 1px solid black; } |

**5.2折叠边框**

border-collapse 属性设置表格的边框是否被折叠成一个单一的边框或隔开：

|  |
| --- |
| table { border-collapse:collapse; }  table,th, td { border: 1px solid black; } |

**5.3表格宽度和高度**

Width和height属性定义表格的宽度和高度。

下面的例子是设置100％的宽度，50像素的th元素的高度的表格：

|  |
| --- |
| table { width:100%; }  th { height:50px; } |

**5.4表格文字对齐**

表格中的文本对齐和垂直对齐属性。

text-align属性设置水平对齐方式，向左，右，或中心：

|  |
| --- |
| td { text-align:right; } |

垂直对齐属性设置垂直对齐，比如顶部，底部或中间：

|  |
| --- |
| td { height:50px; vertical-align:bottom; } |

**5.5表格填充**

如果在表的内容中控制空格之间的边框，应使用td和th元素的填充属性：

|  |
| --- |
| td { padding:15px; } |

**5.6表格颜色**

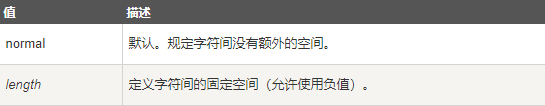
下面的例子指定边框的颜色，和th元素的文本和背景颜色：

|  |
| --- |
| table, td, th { border:1px solid green; } th { background-color:green; color:white; } |

1. **其他属性**

**6.1letter-spacing**

letter-spacing 属性增加或减少字符间的空白（字符间距）

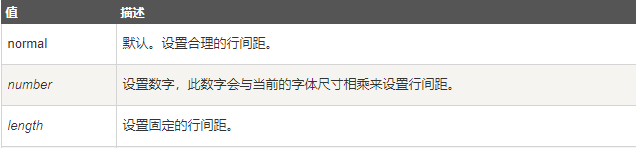


|  |
| --- |
| h1 {letter-spacing:2px} h2 {letter-spacing:-3px} |

**6.2line-height**

设置行高：.

注意： 负值是不允许的



|  |
| --- |
| p{  height: 30px;  line-height: 30px;  } |

**6.3white-space**

white-space属性指定元素内的空白怎样处理。



|  |
| --- |
| p {  white-space: nowrap;  overflow: hidden;  text-overflow: ellipsis;  } |

**6.4verticle-align**

vertical-align 属性设置一个元素的垂直对齐方式。

该属性定义行内元素的基线相对于该元素所在行的基线的垂直对齐。允许指定负长度值和百分比值。这会使元素降低而不是升高。在表单元格中，这个属性会设置单元格框中的单元格内容的对齐方式。

|  |
| --- |
| img{  vertical-align:middle;  } |

1. **选择器**

**7.1关系选择器**

**7.1.1后代选择器**

1、定义

选择所有被E元素包含的F元素，中间用空格隔开

2、语法

E F{}

3、例子

|  |
| --- |
| <ul>  <li>宝马</li>  <li>奔驰</li>  </ul>  <ol>  <li>奥迪</li>  </ol> |

|  |
| --- |
| ul li{color:green;} |



**7.1.2子代选择器**

1. 定义

选择所有作为E元素的直接子元素F，对更深一层的元素不起作用，用>表示

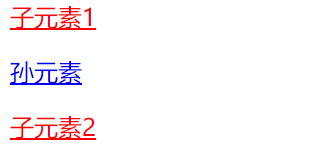
1. 语法

E>F{}

3、例子

|  |
| --- |
| <div>  <a href="#">子元素1</a>  <p> <a href="#">孙元素</a> </p>  <a href="#">子元素2</a>  </div> |

|  |
| --- |
| div>a{color:red} |



**7.1.3相邻兄弟选择器**

1、定义

选择紧跟E元素后的F元素，用加好表示，选择相邻的第一个兄弟元素。

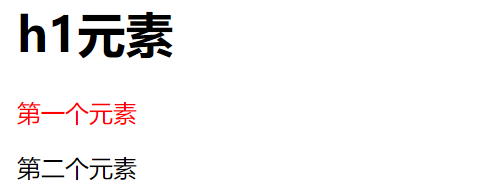
2、语法

E+F{}

3、例子

|  |
| --- |
| <h1>h1元素</h1>  <p>第一个元素</p>  <p>第二个元素</p> |

|  |
| --- |
| h1+p{color:red;} |



**7.1.4通用兄弟选择器**

1、定义

选择E元素之后的所有兄弟元素F，作用于多个元素，用~隔开

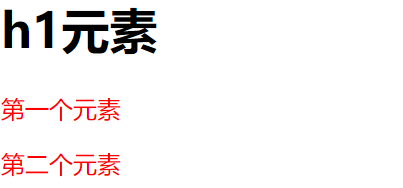
2、语法

E~F{}

3、例子

|  |
| --- |
| <h1>h1元素</h1>  <p>第一个元素</p>  <p>第二个元素</p> |

|  |
| --- |
| h1~p{color:red;} |



**7.2伪类选择器**

同一个标签，根据其不同的种状态，有不同的样式。这就叫做“伪类”。伪类用冒号来表示。

（1）:link “链接”：超链接点击之前（只适用于a）

（2）:visited “访问过的”：链接被访问过之后（只适用于a）

（3）:hover “悬停”：鼠标放到标签上的时候（适用于所有标签）

（4）:active “激活”： 鼠标点击标签，但是不松手时。（适用于所有标签）

|  |
| --- |
| /\*让超链接点击之前是红色\*/  a:link{  color:red;  }  /\*让超链接点击之后是橘色\*/  a:visited{  color:orange;  }  /\*鼠标悬停，放到标签上的时候是绿色\*/  a:hover{  color:green;  }  /\*鼠标点击链接，但是不松手的时候是黑色\*/  a:active{  color:black;  } |

记住，在css中，这四种状态必须按照固定的顺序写：

a:link 、a:visited 、a:hover 、a:active

如果不按照顺序，那么将失效。“爱恨准则”：love hate。必须先爱，后恨。

**7.2.1 :first-child 选择器**

1. 定义

:first-child 选择器匹配其父元素中的第一个子元素。

1. 实例

列表中的第一个 <li>元素选择的样式：

|  |
| --- |
| li:first-child{  background:yellow;  } |

**7.2.2 :last-child 选择器**

1、定义

:last-child选择器用来匹配父元素中最后一个子元素。

2、实例

指定最后一个p元素的背景色：

|  |
| --- |
| p:last-child{ background:#ff0000;  } |

**7.2.3 :nth-child ()选择器**

1、定义

:nth-child(n) 选择器匹配父元素中的第 n 个子元素，元素类型没有限制。

n 可以是一个数字，一个关键字，或者一个公式。

2、实例

奇数和偶数是可以作为关键字使用用于相匹配的子元素，其索引是奇数或偶数（该索引的第一个子节点是1）。 在这里，我们为奇数和偶数p元素指定两个不同的背景颜色：

|  |
| --- |
| p:nth-child(odd){  background:#ff0000;  }  p:nth-child(even){  background:#0000ff;  } |

**7.2.4 :only-child 选择器**

1、定义

:only-child 选择器匹配属于父元素中唯一子元素的元素。

2、实例

匹配属于父元素中唯一子元素的 p 元素:

|  |
| --- |
| p:only-child{  background:#ff0000;  } |

**7.2.5 :empty选择器**

1、定义

:empty选择器选择每个没有任何子级的元素（包括文本节点）。

2、实例

指定空的p元素的背景色：

|  |
| --- |
| p:empty{  background:#ff0000;  } |

**7.2.6 :not() 选择器**

1、定义

:not(selector) 选择器匹配每个元素是不是指定的元素/选择器。

2、实例

为每个并非<p>元素的元素设置背景颜色：

|  |
| --- |
| :not(p){  background:#ff0000;  } |

**7.2.7 :focus 选择器**

1、定义

:focus选择器用于选择具有焦点的元素。

2、实例

一个输入字段获得焦点时选择的样式：

|  |
| --- |
| input:focus{  background-color:yellow;  } |

**7.2.8 :checked 选择器**

1、定义

:checked 选择器匹配每个选中的输入元素（仅适用于单选按钮或复选框）。

2、实例

为所有选中的输入元素设置宽高：

|  |
| --- |
| input:checked {  height: 50px;  width: 50px;  } |

**7.3伪对象选择器**

伪对象也叫伪元素，在过去，伪类和伪元素都被书写成前面只加一个冒号，实际上应该是：  
：weilei ：：伪元素 而现在我们为了兼容旧的书写方式，用一个冒号引导伪类也是能被解析的。  
伪类一般反应无法在CSS中轻松或者可靠检测到的某个元素的状态或属性 ；  
伪元素表示DOM外部的某种文档结构 。

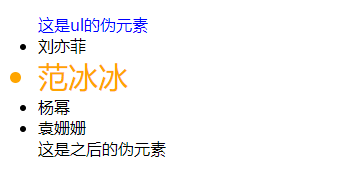
伪类更多的是定义元素的状态，而伪元素则是改变文档结构，在结构外另加一个没有实际存在的元素（伪元素）

常用伪元素：  
 1. E：before/E：：before  
 2. E：after/E：：after

1. E：before/E：：before  
    before 选择器在被选元素的“内容”前面插入内容，用来和content属性一起使用。  
   虽然E：before可转化为E::before，但是你写伪元素是要规范，用两个冒号。

|  |
| --- |
| <ul>  <li>刘亦菲</li>  <li>范冰冰</li>  <li>杨幂</li>  <li>袁姗姗</li>  </ul> |

|  |
| --- |
| li:nth-child(2){font-size: 30px; color: orange;}  ul::before{content:"这是ul的伪元素"; color: blue;}  ul::after{content:"这是之后的伪元素";} |



注意：是显示在被选中标签内容之前和之后。

2. E：after/E：：after

 after选择器在被选择元素的“内容”前面插入内容，用来和content属性一起使用

虽然E：before可转化为E：：before，但是你写伪元素时要规范，用两个冒号。

content属性与：：before及：：after伪元素配合使用，来插入生成内容。伪元素还可以添加图片。

|  |
| --- |
| li::after{  content: url(images/img1.jpg)  } |

**7.4属性选择器**

可以为拥有指定属性的 HTML 元素设置样式，而不仅限于 class 和 id 属性。

注释：只有在规定了 !DOCTYPE 时，IE7 和 IE8 才支持属性选择器。在 IE6 及更低的版本中，不支持属性选择。



**7.4.1 [attribute] 选择器**

1. 定义

[*attribute*] 选择器用于选取带有指定属性的元素。

2、实例

为带有 target 属性的 <a> 元素设置样式：

|  |
| --- |
| a[target]{  background-color:yellow;  } |

**7.4.2[attribute=value] 选择器**

1、定义

[attribute=value] 选择器用于选取带有指定属性和值的元素。

2、实例

为 target="\_blank" 的 <a> 元素设置样式：

|  |
| --- |
| a[target=\_blank]{  background-color:yellow;  } |

**7.4.3 [attribute~=value] 选择器**

1. 定义

[*attribute*~=*value*] 选择器用于选取属性值中包含指定词汇的元素。

1. 实例

选择 titile 属性包含单词 "flower" 的元素，并设置其样式：

|  |
| --- |
| [title~=flower]{  background-color:yellow;  } |

**7.4.4 [attribute|=value] 选择器**

1、定义

[attribute|=value] 选择器用于选取带有以指定值开头的属性值的元素。

注释：该值必须是整个单词，比如 lang="en"，或者后面跟着连字符，比如 lang="en-us"。

2、实例

选择 lang 属性值以 "en" 开头的元素，并设置其样式：

|  |
| --- |
| [lang|=en]{  background-color:yellow;  } |

**7.4.5[attribute^=value] 选择器**

1、定义

[attribute^=value] 选择器匹配属性值以指定值开头的每个元素。

2、实例

设置 class 属性值以 "test" 开头的所有 div 元素的背景色：

|  |
| --- |
| div[class^="test"]{  background:#ffff00;  } |

**7.4.6[attribute$=value] 选择器**

1、定义

[attribute$=value] 选择器匹配属性值以指定值结尾的每个元素。

2、实例

设置 class 属性值以 "test" 结尾的所有 div 元素的背景色：

|  |
| --- |
| div[class$="test"]{  background:#ffff00;  } |

**7.4.7[attribute\*=value] 选择器**

1、定义

[attribute\*=value] 选择器匹配属性值包含指定值的每个元素。

2、实例

设置 class 属性值包含 "test" 的所有 div 元素的背景色：

|  |
| --- |
| div[class\*="test"]{  background:#ffff00;  } |

1. **作业**