

CSS基础

CSS Fundamental

DAY01

主讲: Mr.King

Q Q: 382771946

博客: <http://www.phpfamily.org>

微博: 新浪微博@Mr_King_Hs

- Cascading Style Sheets -- 层叠样式表



CSS优势

- 实现页面结构与表现样式的分离
- 便于文档的维护与更新

structure



presentation



CSS发展历史

- CSS1 -- 作为W3C推荐，发布于1996年12月
- CSS2 -- 作为W3C推荐，发布于1999年1月
- CSS3 -- 作为W3C草案，发布于2001年5月

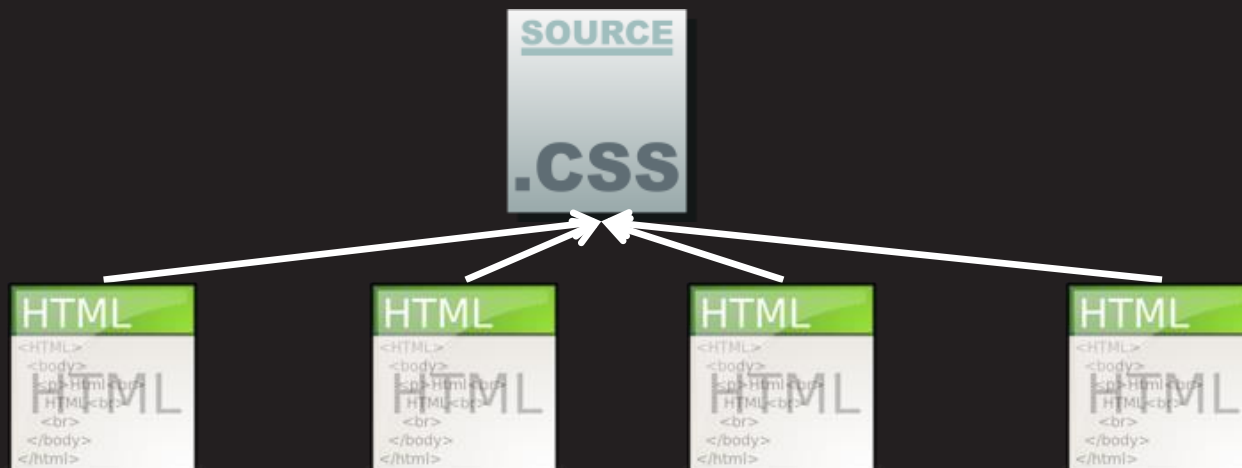


CSS使用方法

- 链接到外部CSS文档

```
<link rel="stylesheet" type="text/css" href="style/style.css" media="all"/>
```

| 媒体类型 | 描述 |
|--------|------|
| all | 所有媒体 |
| print | 打印机 |
| screen | 屏幕 |



CSS书写方法

- 书写于标记内部

```
<body>  
<div style="width:100px;height:100px;background:#e2e2e2">CSS技术</div>  
</body>
```



CSS使用方法

- 书写于文档头部

```
<style type= "text/css">
#context{
    width : 150px;
    height : 100px;
    border : 2px solid #900;
}
</style>
</head>
<body>
<div id="context">CSS技术核心</div>
</body>
</html>
```



CSS使用方法

• 链接到外部CSS文档

1.Style标记：

`<style type="text/css"></style>` 标签用于为HTML文档定义样式信息，可以通过style标记规定浏览器中如何呈现HTML文档。

type属性是必须的，定义style元素的内容，style元素位于head部分中。

2.Link标记：

`<link />` 标记定义文档与外部资源的关系，最常见的用途是链接样式表。

link标记是空元素，它仅包含属性。它只能出现在head部分，不过它可出现任何次数。

`<link />` 元素可选属性：

href: 定义链接文档的位置

charset: 定义链接文档的编码方式

media: 定义被链接文档将显示在什么设备上

rel: 定义当前文档与被链接文档之间的关系

`<link href="CSS文档URL" rel="stylesheet" media="all|screen|tv|print"/>`

rel="stylesheet" 文档的外部样式表

media="all|screen|tv|print", all代表适用于所有设备，screen计算机屏幕，tv电视机，print打印机

CSS语法结构

- selector { property : value ; property : value ; }



选择器



属性



值

Declaration

声明

```
#container {
  width : 100px;
  height : 40px;
  border : 1px solid #f00;
}
```



通配选择器(Universal selectors)

- 通配选择器 -- *
- 匹配所有元素

以下规则将应用文档树的所有元素

```
* {
  font : 12px / 1.5em 微软雅黑 ;
  color : #333;
}
```



标记选择器(Type selectors)

- 标记选择器 -- tag
- 匹配指定标记

```
<style type="text/css">
h1 { font : 12px/1.8em 微软雅黑;} /*仅匹配H1标记*/
</style>
</head>
<body>
<h1>Web前端技术</h1>
<h2>XHTML技术</h2>
<h2>CSS技术核心</h2>
</body>
</html>
```



类选择器(Class selectors)

- 类选择器 -- .
- 匹配指定class属性的标记
- 多个属性值之间以空格分隔

```
<style type="text/css">
.pastoral { color : blue; } /*匹配class="pastoral"的标记*/
h1.pastoral { color : green } /*仅匹配H1标记且存在class="pastoral"的标记*/
.title { font-size : 18px; } /*匹配class="title"的标记*/
</style>
</head>
<body>
<h1 class="pastoral title">Web前端技术</h1>
<h2 class="pastoral">XHTML技术</h2>
<h2 class="pastoral">CSS技术</h2>
<h2 class="pastoral">JavaScript技术</h2>
```



ID选择器(ID selectors)

- ID选择器 -- #名称
- 匹配文档唯一标记

```
<style type="text/css">
#pastoral { color : blue; } /*匹配id="pastoral"的标记*/
</style>
</head>
<body>
<h1 id="pastoral">Web前端技术</h1>
<h2>XHTML技术</h2>
<h2>CSS技术</h2>
<h2>JavaScript技术</h2>
```



ID选择器(ID selectors)

- 注意：
- 1》ID选择器在文档中使用一次
- 2》不能使用ID词列表，也就是说ID选择器不能结合使用，因为ID属性不允许有以空格分隔的词列表
- 3》ID能包含更多含义
- 标记中的ID可以应用于以下三种情况：
- CSS样式
- Javascript-》 `Document.getElementById("#ID")`
- JQuery-> `$("#ID")`



后代选择器(Descendant selectors)

- 后代选择器 -- selector selector ...
- 选择器之间至少为父子关系

```
<style type="text/css">
strong em { color : blue; } /*匹配strong的后代为em的标记*/
</style>
</head>
<body>
<p><strong>华盛顿</strong></p>
<p><em>林肯</em></p>
<p><strong><em>罗斯福</em></strong></p>
<p><em><strong>杜鲁门</strong></em></p>
<p><strong><span><em>肯尼迪</em></span></strong></p>
```



子代选择器(Child selectors)

- 子代选择器 -- selector > selector
- 选择器之间只能为父子关系
- 提示：IE6不支持

```
<style type="text/css">
strong > em { color : blue; } /*匹配strong的子代为em的标记*/
</style>
</head>
<body>
<p><strong>华盛顿</strong></p>
<p><em>林肯</em></p>
<p><strong><em>罗斯福</em></strong></p>
<p><em><strong>杜鲁门</strong></em></p>
<p><strong><span><em>肯尼迪</em></span></strong></p>
```



选择器群组(Groups of selectors)

- 选择器群组 – selector , selector,...
- 具有相同属性选择器的简写方式

```
<style type="text/css">
h1, p { color:green; } /*匹配h1,p标记*/
h1 { font-size : 18px;}
</style>
</head>
<body>
<h1>美国历任总统</h1>
<p>华盛顿</p>
<p>林肯</p>
<p>罗斯福</p>
<p>杜鲁门</p>
```



相邻兄弟选择器

- 可选择紧接在另一元素后的元素，且二者具有相同父元素。
- 选择器 + 选择器，
- 注意：
- 用一个结合符只能选择两个相邻兄弟中的第二个子元素。



伪类

- CSS伪类用于向某些选择器添加特殊效果。
- 语法：
- `selector:pseudo-class{property:value}`
- `: active`：向被激活的元素添加样式
- `: hover`：当鼠标悬浮在元素上方时，向元素添加样式
- `: link`：向未被访问的链接添加样式
- `: visited`：向已被访问的链接添加样式
- `: focus`：向拥有键盘输入焦点的元素添加样式(IE6浏览器不支持此样式)
- `: first-child`：向元素的第一个子元素添加样式



链接属性

| 属性 | 描述 |
|------------------|-------------|
| a:link | 普通的、未被访问的链接 |
| a:visited | 用户已访问的链接 |
| a:hover | 鼠标指针位于链接的上方 |
| a:active | 链接被点击的时刻 |

注意：伪类选择器

LVHA伪类,样式按**LVHA优先级顺序**从右至左覆盖，不同的顺序会产生不同的效果。

a:link - 默认链接样式

a:visited - 已访问链接样式

a:hover - 鼠标悬停样式

a:active - 鼠标点击样式



伪元素

- 伪元素用于向某些选择器设置特殊效果.
- 语法：selector:pseudo-element{property:value;}
- 1>:before:伪元素可以在元素的内容前面插入新内容。
- 2>:after:伪元素可以在元素的内容之后插入新内容
- 形式为
- p:before{content:"内容"}
- p:after{content:url(图片或者视频URL);}



CSS优先级问题

- CSS优先级问题：
- 外部样式External style sheet < 内部样式Internal style sheet < 内联样式inline style
- 注意：如果外部样式表放在内部样式表后面，外部样式表将覆盖内部样式表
- 选择器的优先权：
- 内联样式[1000]：权值1000
- id选择器[100]：权值100
- 类选择器[10]：权值10
- 元素选择器[1]：权值1
- 1.选择器都有一个权值，权值大的先执行
- 2.当权值相等时，
- 后出现的样式表设置要优于先出现的样式表设置
- 3.网页设置的CSS高于浏览器的所有样式
- 4.继承的CSS样式不如后来指定的CSS样式
- 5.在同一组属性设置
- 中有!important规则的优先级最大。

```
<style type="text/css">
#test p{ /*100 +1 =101*/
    color:red;
}
#test .test1 em{ /*100+10+1*/
    color:green;
}
#test p span em{ /*100+ 1+ 1+ 1=103*/
    color:yellow;
}|
</style>
</head>
<body>

<div id="test">
    <p class="test1">aaaaa
        <span><em>bbbbbb</em></span>
    </p>
    <p>cccccc</p>
</div>
```



单位(Units)

- 1》长度(Length)
- px:相对长度单位。像素Pixel
- 像素是相对于显示器屏幕分辨率而言的。例如Windows的用户所使用的分辨率一般是96像素/英寸。而MAC的用户所使用的分辨率一般是72像素/英寸
- em:相对长度单位。相对于当前对象内文本的字体尺寸。
- 如当前行内文本的字体尺寸未被人设置，则相对于浏览器默认字体尺寸
-



单位(Units)

- 2》颜色(Color)
- rgb(R,G,B):
- 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
- 并非所有浏览器都支持百分数的值
- #RRGGBB
- RR:红色值。十六进制正整数
- GG : 绿色值。十六进制正整数
- BB : 蓝色值。十六进制正整数
- 参数取值范围为00-FF
- 参数必须是两位数。对于只有一位的，应在前面补零。如果每个参数各自在两位上的数字相同，那么本单位也可缩写为#RGB的方式。
- Color Name :
- 颜色单位。颜色名称。不同的浏览器会有不同的预定义颜色名称。



Div和span

- div用于标记块级元素(block),span用于标记内联元素(inline)
- 他们是没有意义的元素，除了对其应用样式。
- 块级元素和内联元素区别在于块级对象默认宽度是100%，继承父元素，如果没有采用float样式，相邻的两个对象就会分排在不同的两行上。
- 内联对象的宽度取决于内部元素的宽度与padding样式值之和，不可直接指定其宽度与高度(强行转换后可以定义display:block或者float : left/right)，相邻的两个内联对象会排在同一行上。
- 块级元素：
body,p,div,h1,h2,h3,h4,h5,h6,ul,ol,li,dl,dt,dd,form,table,tr,td,th



Box模型



Box模型

Box模型(Box Model)

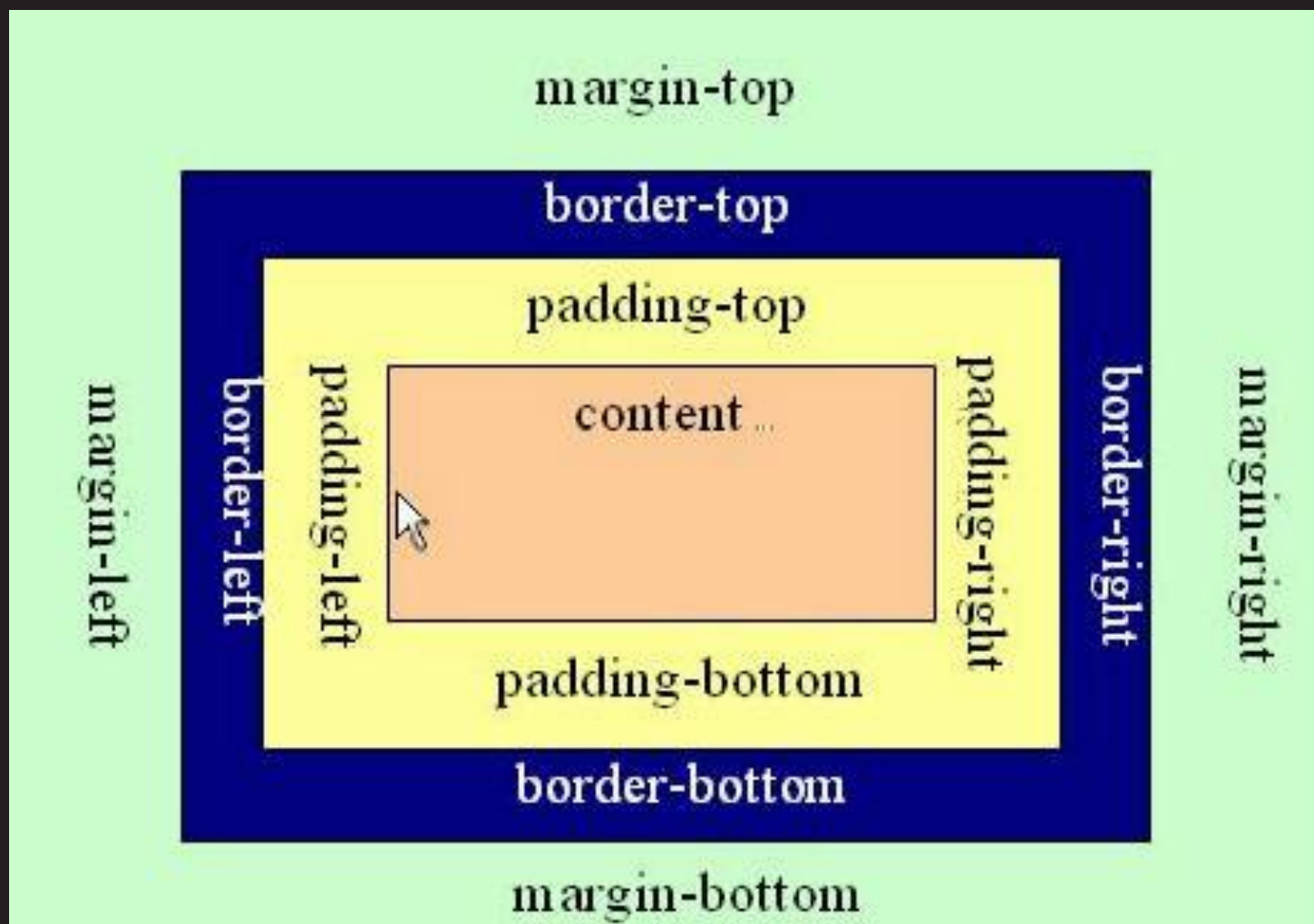


素材网 www.nipic.com qq: 621118202

www.danetech.com



Box模型(Box Model)



布局属性



width

height

margin

padding

clear

float

布局属性

| 属性 | 描述 |
|----------------|----------------------|
| Width | 宽度 |
| Height | 高度 |
| Padding | 简写属性，内边距，即内容与边框之间的距离 |
| Padding-top | 上内边距 |
| Padding-right | 右内边距 |
| Padding-bottom | 下内边距 |
| Padding-left | 左内边距 |
| Margin | 简写属性，边框与边框之间的距离 |
| Margin-top | 上外边距 |
| Margin-right | 右外边距 |
| Margin-bottom | 下外边距 |
| Margin-left | 左外边距 |

外边距合并指的是当两个垂直外边距相遇时，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外边距的高度中的较大者。



width属性

- 设置对象宽度

```
<style type="text/css">
#container { width : 200px ; border: 1px solid #f00;}
</style>
</head>
<body>
<div id="container">CSS技术</div>
```

说明:块级对象的默认宽度为其父对象的100%。



height属性

- 设置对象高度

```
<style type="text/css">
#container { width : 200px ; height : 150px; border: 1px solid #f00;}
</style>
</head>
<body>
<div id="container">CSS技术</div>
```



margin属性

- 设置外边距

```
<style type="text/css">
#container { width : 200px ; height : 150px; margin:0 auto; border: 1px solid #f00;}
</style>
</head>
<body>
<div id="container">CSS技术</div>
```

```
margin : value;
margin : value value ;
margin : value value value ;
margin : value value value value ;
```



padding属性

- 设置内边距

```
<style type="text/css">
#container { width : 200px ; height : 150px; padding:0 10px; border: 1px solid #f00;}
</style>
</head>
<body>
<div id="container">CSS技术</div>
```

```
padding : value;
padding : value value ;
padding : value value value ;
padding : value value value value ;
```



float属性

- 设置浮动
- float : left | right | none;

```
<style type="text/css">
#navi { border : 1px solid #f00 ; }
#navi li { float :left ;}
</style>
</head>
<body>
<ul id="navi">
  <li>全部商品分类</li>
  <li>首页</li>
  <li>服装城</li>
</ul>
```

说明：如果子对象存在浮动属性，那么其父对象的高度将清除为零。



clear属性

- 清除浮动
- `clear : left | right | both ;`
- 解决如果子对象存在浮动属性，那么父对象的高度将清除为零。
- 清除浮动包括清除子元素的浮动和清除上级元素的浮动，其中清除上级元素的浮动，只需设置clear为both就可以了，而清除子元素的浮动则可以用空标签法、overflow方法。因清除上级元素的浮动比较简单，而空标签法清除子元素浮动会增加额外标签，可以通过以下方式清楚浮动。



清除浮动方式

- 1》为父对象赋予明确的高度值。
- 2》定义名称为.clear的CSS类
- .clear:after{
 - height:0px;
 - display:block;
 - visibility:hidden;
 - overflow:hidden;
 - content:" " ;
 - clear:both;
- }
- .clear{
 - Clear:both;
 - Zoom:1;/*IE专有属性*/
- }



清除浮动方式

- 3》.clear:after{
- Content:" " ;
- Display:block;
- Overflow:hidden;
- Clear:both;
- }
- .clear{
- Clear:both;
- Zoom:1;
- }



清除浮动方式

- 4》对其父类应用样式
- #父类名称{
 - Overflow:auto;
 - Zoom:1;
- }
- 或者：
- #父类名称{
 - Overflow:hidden;
 - Zoom:1;
- }



边框属性

| 属性 | 描述 |
|---------------------|---|
| Border | 简写属性 |
| Border-style | 用于设置元素所有边框的样式，none无边框；hidden隐藏；dotted定义点状边框；dashed定义虚线；solid定义实现；double定义双线；groove定义3D凹槽边框；ridge定义3D垄状边框；inset定义3Dinset边框；outset定义3Doutset边框； |
| Border-width | 简写属性，用于设置所有边框的宽度 |
| Border-color | 简写属性，用于设置所有边框的颜色 |
| Border-bottom | 简写属性，把下边框的所有属性设置到一个声明中 |
| Border-bottom-color | 下边框的颜色 |
| Border-bottom-style | 下边框的样式 |
| Border-bottom-width | 下边框的宽度 |
| Border-left | 简写属性，左边框属性 |
| Border-left-color | 左边框颜色 |
| Border-left-style | 左边框样式 |
| Border-left-width | 设置左边框的宽度 |
| Border-right | 简写属性，设置右边框属性 |
| Border-right-color | 设置右边框的颜色 |
| Border-right-style | 右边框样式 |
| Border-right-width | 右边框的宽度 |
| Border-top-color | 简写属性 |
| Border-top-style | 设置上边框样式 |
| Border-top-width | 设置上边框宽度 |



轮廓属性

| 属性 | 描述 |
|---------------|--|
| Outline | 简写，在一个设置中设置所有的轮廓属性 Outline:<轮廓颜色> <轮廓样式> <轮廓宽度> |
| Outline-color | 轮廓颜色 |
| Outline-style | 轮廓样式。none默认，无轮廓；dotted定义点状的轮廓；dashed定义虚线轮廓；solid定义实现轮廓；double定义双线轮廓；groove定义3D凹槽轮廓；ridge定义3D突槽轮廓；inset定义3D凹边轮廓；outset定义3D凸边轮廓； |
| Outline-width | 轮廓的宽度 |



背景属性

| 属性 | 描述 |
|-----------------------|--|
| Background | 简写属性 |
| Background-color | 背景颜色 |
| Background-image | 背景图像；background:<背景颜色> <背景图片> <背景重复> <背景依附方式> 背景位置 |
| Background-repeat | 设置背景图像重复。repeat、repeat-x、repeat-y、no-repeat |
| Background-position | 背景图像位置，横向left、center、right；纵向top、center、bottom |
| Background-attachment | 设置背景图像的依附方式,scroll滚动、fixed固定 |



定位属性

| 属性 | 描述 |
|--------------------------------|---|
| Position | <p>把元素设置到一个静态的、相对的、绝对的、或固定的位置中。</p> <p>Absolute：生成绝对定位的元素，元素的位置通过left、top、right、bottom进行规定，已经脱离文档流，也就是不占用页面的位置。默认情况下absolute定位是相对于窗口的左上角。如果父对象存在relative定位，那么子对象的absolute是相对于父对象而言。</p> <p>Fixed：生成绝对定位元素，相对于浏览器窗口进行定位。元素的位置通过left、top、right、及bottom进行规定</p> <p>Relative：生成相对定位的元素，相对于其正常位置进行定位</p> <p>Static：默认值，没有定位，元素出现在正常的流中</p> |
| Top | 规定元素的顶部边缘 |
| Right | 规定元素的右侧边缘 |
| Bottom | 规定元素的底部边缘 |
| Left | 规定元素的左侧边缘 |
| Overflow、overflow-x、overflow-y | <p>设置当元素的内容溢出时的处理</p> <p>Visible：默认值，内容不会被修剪，会呈现在元素边框之外</p> <p>Hidden：内容会被修剪，其余内容是不可见的</p> <p>Scroll：内容会被修剪，但是浏览器会显示滚动条来显示其余内容</p> <p>Auto：如果内容被修剪，则浏览器会显示滚动条来显示其余内容</p> |



定位属性

| | |
|------------|--|
| z-index | <p>设置元素的堆叠顺序。</p> <p>Auto: 默认。堆叠顺序与父元素相等</p> <p>Number: 设置元素的堆叠顺序</p> |
| Display | <p>block:块对象的默认值。将对象强制作为块对象呈递，为对象之后添加新行</p> <p>none:隐藏对象。与visibility属性的hidden值不同，其不为被隐藏的对象保留其物理空间</p> <p>line: 内联对象的默认值。将对象强制作为内联对象呈递，从对象中删除行</p> |
| visibility | <p>visible: 对象可视</p> <p>hidden: 对象隐藏</p> <p>设置或检索是否显示对象，与display属性不同，此属性为隐藏的对象保留其占据的物理空间</p> |

透明颜色：

Opacity : value[0~1];
Filter:alpha(opacity=value[0~100]),IE专用



字体属性

| 属性 | 描述 |
|--------------|---|
| Font | 简写。把所有针对字体的属性设置在一个声明中 font[<字体风格> <字体变形> <字体粗细>]?<字体大小>[/<行高>]?<字体族科> |
| Font-family | 设置字体系列，如times、serif |
| Font-size | 字体大小 |
| Font-style | 设置字体风格.normal标准，italic斜体,oblique倾斜 |
| Font-variant | 以小型大写字体或者正常字体显示文本。normal默认值，显示标准字体;small-caps浏览器会显示小型大写字母的字体； |
| Font-weight | 设置字体的粗细。normal默认值；bold定义粗体；bolder更粗的字体；lighter更细的字体；100~900定义值，400=normal，700=bold； |



文本属性

| 属性 | 描述 |
|-----------------|---|
| Color | 设置文本颜色 |
| Direction | 设置文本方向,ltr默认从左到右,rtl右到左,inherit继承父元素 |
| Text-indent | 文本缩进,可以为正值,也可以为负值 |
| Text-align | 对齐方式, left center right justify |
| Word-spacing | 改变单词之间的标准间隔, 数值, 正负均可 |
| Letter-spacing | 字母间隔, 数值, 正负均可 |
| Text-transform | 设置文本大小写。 none uppercase lowercase capitalize |
| Text-decoration | 文本装饰.none underline overline line-through blink,可以组合使用既有underline又有overline |
| Line-height | 设置行高, 数值, 只能为正值。默认值normal |



表格属性

| 属性 | 描述 |
|------------------------|--|
| Border-collapse | 设置是否把表格边框合并为单一的边框。 separate默认值，边框会分开；collapse如果可能边框会合并为一个单一的边框； |
| Border-spacing | 设置分隔单元格边框的距离，数值 |
| Caption-side | 设置表格标题的位置，top，bottom |
| Empty-cells | 设置是否显示表格中的空单元格，hide隐藏； show显示 |
| Table-layout | 设置显示单元、行和列的算法，automatic默认， 列宽度由单元格内容设定；fixed列宽由表格宽度和列宽设定； |



列表属性

| 属性 | 描述 |
|---------------------|---|
| List-style | 简写属性；list-style:<列表图像> <列表形式> <列表位置> |
| List-style-type | 列表类型,none无标记；disc默认，实心圆；circle空心圆；square实心方块；decimal数字；decimal-leading-zero0开头的标记；lower-roman小写罗马数字；upper-roman大写罗马数字；lower-alpha：小写英文字母；upper-alpha大写英文字母；lower-greek小写希腊字母；lower-latin小写拉丁字母；upper-latin大写拉丁字母等 |
| List-style-image | 列表标志，用图像代表列表项的标记 |
| List-style-position | 设置在何处设置列表项的标记。inside列表项目标记放置在文本以内，且环绕文本根据标记对齐；outside默认值，保持标记位于文本的左侧。 |



CSS Hack

- 1. 简介
 - CSS hack由于不同的浏览器，比如Internet Explorer 6, Internet Explorer 7, Mozilla Firefox等，对CSS的解析认识不一样，因此会导致生成的页面效果不一样，得不到我们所需要的页面效果。这个时候我们就需要针对不同的浏览器去写不同的CSS，让它能够同时兼容不同的浏览器，能在不同的浏览器中也能得到我们想要的页面效果。
- 2. CSS Hack表现形式
 - 1》CSS类内部Hack
 - 2》选择器Hack
 - 3》HTML头部引用if IE...Hack



CSS Hack

• 3 . Css Hack应用

- 1》IE6能识别_和*
- 2》IE7能识别*
- 3》firefox两个都不能识别
- 4》IE6不能识别!important;
- 5》IE7能识别!important;
- 6》条件选择加载指定IE下的CSS
- <!--[if IE]>引入IE的CSS<![endif]-->
- <!--[if IE 7]>引入IE7的CSS<![endif]-->
- <!--[if gt IE 7]>引入IE7以上的CSS<![endif]-->
- <!--[if lt IE 7]>引入IE7以下的CSS<![endif]-->
- <!--[if gte IE 7]>引入IE7及IE7以上的CSS<![endif]--

>



```
1.区别IE和非IE浏览器
#tip {
background:blue; /*非IE 背景蓝色*/
background:red \9; /*IE6、IE7、IE8背景红色*/
}
```

2.区别IE6,IE7,IE8,FF

【区别符号】：「\9」、「*」、「_」
【示例】：

```
#tip {
background:blue; /*Firefox 背景变蓝色*/
background:red \9; /*IE8 背景变红色*/
*background:black; /*IE7 背景变黑色*/
_background:orange; /*IE6 背景变橘色*/
}
```

【说明】：因为IE系列浏览器可读「\9」，而IE6和IE7可读「*」（米字号），另外IE6可辨识「_」（底线），因此可以依照顺序写下来，就会让浏览器正确的读取到自己看得懂得CSS语法，所以就可以有效区分IE各版本和非IE浏览器(像是Firefox、Opera、Google Chrome、Safari等)。

3.区别IE6、IE7、Firefox (方法 1)

【区别符号】：「*」、「_」

【示例】：

```
#tip {
background:blue; /*Firefox背景变蓝色*/
*background:black; /*IE7 背景变黑色*/
_background:orange; /*IE6 背景变橘色*/
}
```

【说明】：IE7和IE6可读「*」（米字号），IE6又可以读「_」（底线），但是IE7却无法读取「_」，至于Firefox(非IE浏览器)则完全无法辨识「*」和「_」，因此就可以透过这样的差异性来区分IE6、IE7、Firefox

4.区别IE6、IE7、Firefox (方法 2)

【区别符号】：「*」、「!important」

【示例】：

```
#tip {
background:blue; /*Firefox 背景变蓝色*/
*background:green !important; /*IE7 背景变绿色*/
*background:orange; /*IE6 背景变橘色*/
}
```

【说明】：IE7可以辨识「*」和「!important」，但是IE6只可以辨识「*」，却无法辨识「!important」，至于Firefox可以读取「!important」但不能辨识「*」因此可以透过这样的差异来有效区隔IE6、IE7、Firefox。

5.区别IE7、Firefox

【区别符号】：「*」、「!important」

【示例】：

```
#tip {
background:blue; /*Firefox 背景变蓝色*/
*background:green !important; /*IE7 背景变绿色*/
}
```

【说明】：因为Firefox可以辨识「!important」但却无法辨识「*」，而IE7则可以同时看懂「*」、「!important」，因此可以两个辨识符号来区隔IE7和Firefox。

6.区别IE6、IE7 (方法 1)

【区别符号】：「*」、「_」

【示例】：

```
#tip {
*background:black; /*IE7 背景变黑色*/
_background:orange; /*IE6 背景变橘色*/
}
```

【说明】：IE7和IE6都可以辨识「*」（米字号），但IE6可以辨识「_」（底线），IE7却无法辨识，透过IE7无法读取「_」的特性就能轻松区隔IE6和IE7之间的差异。

7.区别IE6、IE7 (方法 2)

【区别符号】：「!important」

【示例】：

```
#tip {
background:black !important; /*IE7 背景变黑色*/
background:orange; /*IE6 背景变橘色*/
}
```

【说明】：因为IE7可读取「!important;」但IE6却不行，而CSS的读取步骤是从上到下，因此IE6读取时因无法辨识「!important」而直接跳到下一行读取CSS，所以背景色会呈现橘色。

8.区别IE6、Firefox

【区别符号】：「_」

【示例】：

```
#tip {
background:black; /*Firefox 背景变黑色*/
_background:orange; /*IE6 背景变橘色*/
}
```

【说明】：因为IE6可以辨识「_」（底线），但是Firefox却不行，因此可以透过这样的差异来区隔Firefox和IE6，有效达成CSS hack。

