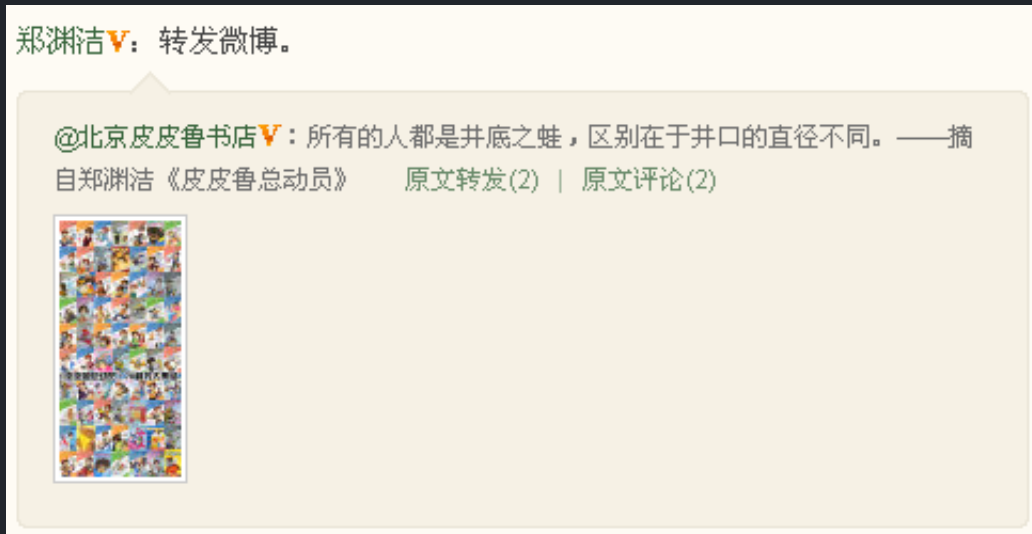


利用CSS3特性巧妙实现漂亮的DIV箭头和气泡

一、样例

DIV箭头用于表现DIV内容的指向，是使用非常普遍的一种表现形式，例如新浪微博的消息转发：



还有网站的导航条：



像上方这种箭头更需要多幅图片以表现箭头和hover的效果。

传统的实现方式都需要一副表示箭头的图片放在DIV上方来实现，例如新浪微博的相关CSS如下

```

.MIB_assign .MIB_asarrow_l {          index....7172238 (第 1 行)
    height: 11px;
    margin: 0 0 -2px 53px;
    overflow: hidden;
    position: relative;
    width: 19px;
}

.head_menu .menu_l, .head_menu .menu_r, skin.c...7172238 (第 39 行)
.head_menu .submit, .head_menu
img.icon_color, .head_menu li.cur
em.arrow, .userinfo .user_atten,
.MIB_asarrow_l, .MIB_asarrow_r,
.right_nav ul li, .right_nav
.empty_group .empty_btm, .smallNav
li.smallNavIcon, .smallNav
li.smallNavIcon2, .MIB_mblogbox .connBg
.postBtnBg a, .bottomLinks .bL_line,
.MIB_dot {
    background-image: url("images/fragmentbg.png");
}

.MIB_asarrow_l {          index....7172238 (第 1 行)
    background: url("../images/common/skin/fragmentbg.png")
    no-repeat scroll -163px 0 transparent;
}

```

二、div箭头

使用CSS3的特性，我们不需要图片就可以实现更加华丽的效果，这样做的好处还包括减少本地文件系统的读取、节省服务器带宽和连接数、避免文件下载失败带来的错误等等

实现的原理是

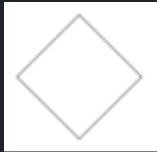
- 我们可以将箭头看作是一个矩形或者菱形的一个角，使用CSS3的属性transform来实现矩形的旋转
- 朝上的箭头需要将矩形旋转45度，我们使用transform: rotate(45deg)来实现，另外针对不同的浏览器还需要添加不同的hack，例如Opera的-o-transform、Firefox的-moz-transform

下面以Chrome浏览器为例实现一个样例。

1、定义一个arrow-shadow样式，内容如下：

```
.arrow-shadow {  
  -webkit-transform: rotate(45deg);  
  border: 1px solid #AAAAAA;  
  height: 40px;  
  position: absolute;  
  width: 40px;  
}
```

实现的效果：



2、为了更加饱满，我们加上CSS3的盒阴影

```
.arrow-shadow {  
  -webkit-transform: rotate(45deg);  
  -webkit-box-shadow: 0 0 10px 0 #aaa;  
  height: 40px;  
  position: absolute;  
  width: 40px;  
}
```

现在效果如下：



3、在外面加一层DIV控制它的高度和宽度

```
.arrow-outer {  
  height: 24px;  
  overflow: hidden;  
  position: absolute;  
  width: 60px;  
}
```

效果如下：



4、加上内部箭头和外部容器

现在我们已经有了一个漂亮的箭头，但是这还不够，为了表现得更加出色，我们再加一层内部的箭头，然后在外面套上容器

页面结构代码：

```
<div class="arrow-outer">
  <div class="arrow-shadow"></div>
</div>
<div class="arrow-inner"></div>
```

样式代码：

```
.arrow-outer {
  position: absolute;
  height: 24px;
  width: 60px;
  overflow: hidden;
}

.arrow-shadow {
  box-shadow: 0 0 10px 0 #aaaaaa;
  -webkit-transform: rotate(45deg);
  transform: rotate(45deg);
  background: none repeat scroll 0 0 #ffffff;
  height: 40px;
  left: 15px;
  position: absolute;
  top: 16px;
  width: 40px;
}

.arrow-inner {
  position: relative;
  left: 10px;
  top: 20px;
  height: 40px;
  width: 40px;
```

```
background: #fff;  
border: 5px solid #ededed;  
-webkit-transform: rotate(45deg);  
transform: rotate(45deg);  
}
```

最后完整的效果图：



三、div气泡

页面开发中常常需要一个气泡框，还需要添加圆角和指向性尖角，预览图如下



借助css的伪元素选择器可以实现上述效果

先写上盒子，上代码

HTML代码:

```
<div class="element"></div>
```

CSS代码:

```
.element {  
    width: 0;  
    height: 0;  
    border-top: 20px solid red;  
    border-right: 20px solid blue;  
    border-bottom: 20px solid yellow;  
    border-left: 20px solid green;  
}
```

实现效果如下:



具体的实现原理就是，一个盒子，作为一个块级元素，我们首先将它的宽和高都设置成0，再将边框的大小设置成20px，也就是形成了如上图这样的四等分三角形组成的一个正方形。

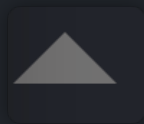
那我们要得到一个方向朝上的尖角，就必须让上，右，左的三角形“隐形”。

透明属性可以帮助我们做到这一点，透明的背景欺骗了我们的眼睛~让我们以为他们消失了，实际上它们只是看不见了而已。

将css改为

```
.element{  
    width: 0;  
    height: 0;  
    border-top: 20px solid transparent;  
    border-right: 20px solid transparent;  
    border-bottom: 20px solid #727171;  
    border-left: 20px solid transparent;  
}
```

效果



现在三角形已经出来了，如何“掏空”它变成透明的呢？这就是接下来我们要完成的工作了。

首先我们定义好外部的盒子wrapper

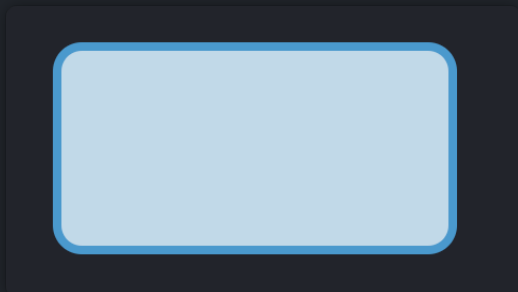
HTML代码：

```
<div class="wrapper"></div>
```

CSS代码：

```
.wrapper {  
  width: 200px;  
  height: 100px;  
  position: relative;  
  margin: 100px;  
  border: 5px solid #4999ce;  
  background-color: #c1d9e9;  
  border-radius: 15px;  
}
```

效果



然后我们需要为这个盒子添加上之前实现的三角形 bottom那个三角形与盒子颜色相同，其余的三角形选择“隐形”即可。这里可以使用伪元素**::before** 和 **::after**来实现这个效果。这两个元素被默认为行内元素，所以需要更改它们的属性为块级元素,绝对定位。再添加一个小点的白色

三角形遮盖上面的三角形，使其露出尖角边框。

CSS代码

```
.wrapper:before {  
  content: '';  
  width: 0;  
  height: 0;  
  border: 20px solid transparent;  
  border-bottom-color: #4999ce;  
  position: absolute;  
  top: 0;  
  right: 20%;  
  margin-top: -44px;  
}
```

效果



然后再次添加after选择器，用它来形成一个白色的三角形，并按照之前的写法把它定位到上边框上去。这里需要特别提醒一点，那就是这个白色的小三角形，要比之前的三角形更小一点，至于大小可以自己调整，出来的效果可以了就基本完成了。


```
.wrapper:after {  
  content: '';  
  width: 0;  
  height: 0;  
  border: 20px solid transparent;  
  border-bottom-color: #c1d9e9;  
  position: absolute;  
  top: 0;  
  right: 20%;  
  margin-top: -36px;  
}
```

最终效果



四、总结

CSS3、HTML5的发展和完善让我们编写代码更加简便和快捷，实现的效果也更加漂亮，例如背景的渐变不再需要背景图片的平铺就可以通过CSS代码直接实现，但是有时候需要经过一些转换，才能将这些新技术应用到我们常见的功能中去，这需要我们多加思考和保持思维的灵感。