

本章前几节重点讲解了 CSS3 中的高级应用，包括过渡、变形及动画等。为了使读者更好地理解这些应用，并能够熟练运用相关属性实现元素的过渡、平移、缩放、倾斜、旋转及动画等特效，本节将通过案例的形式分步骤地制作工作日天气预报的主题页面，其默认效果如图 9-24 所示。

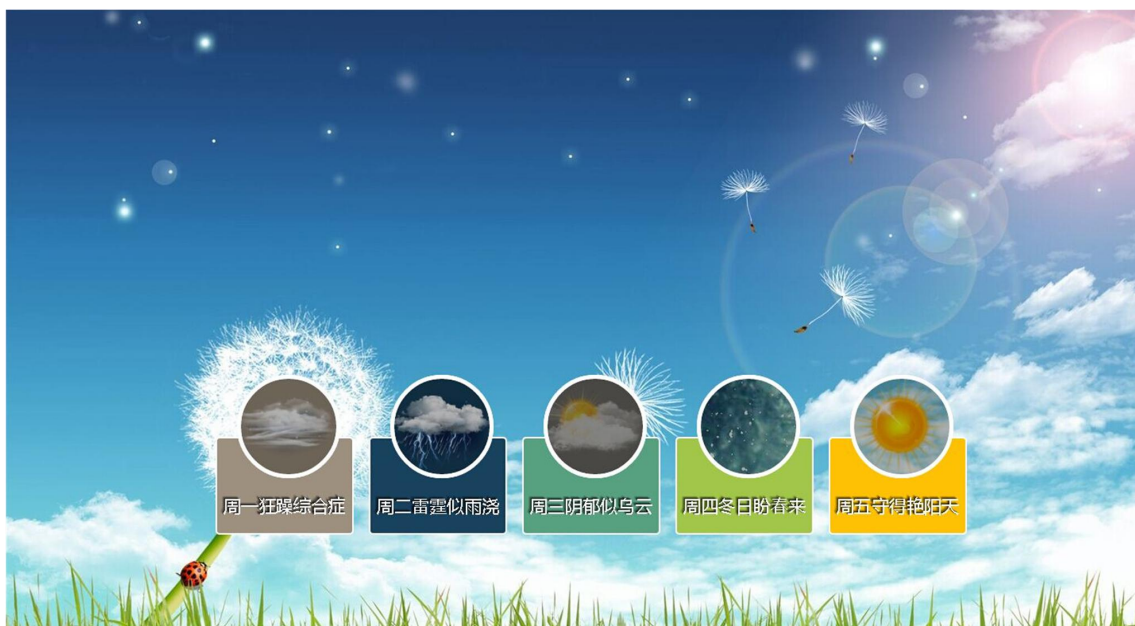


图 9-24 工作日天气预报效果图

当鼠标移上网页中的圆形天气图标时，图标中的图片将会变亮，效果如图 9-25 所示。



图 9-25 鼠标移上时图标变亮

当鼠标点击网页中的天气图标时，网页中的背景图片将发生改变，且切换背景图片时会产生不同的动画效果。如图 9-26 所示即为点击“周四冬日盼春来”图标时的网页效果。



图 9-26 点击图标时背景图片发生改变

9.1.1 分析效果图

1、结构分析

观察效果图 9-24 不难看出，整个页面可以分为背景图片和天气图标两部分，这两部分内容均嵌套在 `<section>` 标记内部，其中背景图片模块由 `` 标记定义。天气图标模块整体上由无序列表 `` 布局，并由 `` 标记嵌套 `<a>` 标记构成，每个 `<a>` 标记代表天气图标中的圆角矩形模块。效果图 9-24 对应的结构如图 9-27 所示。

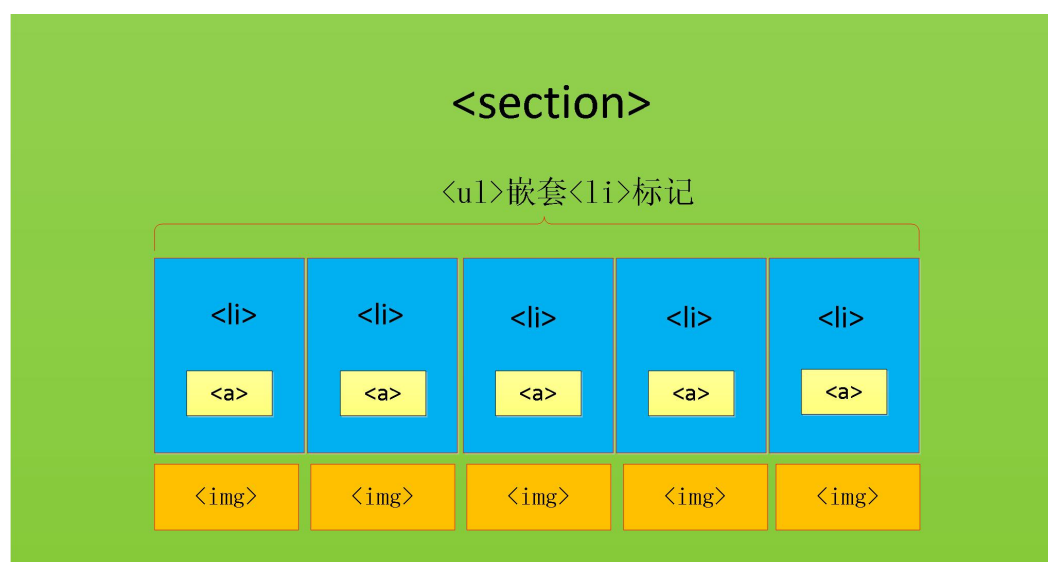


图 9-27 工作日天气预报页面结构图

2、样式分析

控制效果图 9-24 的样式比较复杂，主要分为 6 个部分，具体如下：

- (1) 整体控制背景图片的样式，需要对其设置宽高 100%，固定定位、层叠性最低。

- (2) 整体控制元素，需要设置宽度 100%，绝对定位方式、文字居中及层叠性最高。
- (3) 控制每个标记的样式，需要转化为行内块元素，并设置宽高、外边距样式。
- (4) 控制每个<a>标记的样式，需要设置文本及边框样式，并设置为相对定位。另外，需要单独控制每个<a>元素的背景色。
- (5) 通过:after 伪元素选择器在<a>标记之后插入五张不同的天气图片，设置为圆形图标。同时，使用绝对定位方式控制其位置、层叠性。
- (6) 通过:before 伪元素选择器为圆形图标添加不透明度并设置鼠标移上时的不透明度为 0。

3、动画分析

通过案例演示可以看出，第一张背景图片的切换效果为从左向右移动；第二张背景图片的切换效果为从下向上移动；第三张背景图片的切换效果为由小变大展开；第四张背景图片的切换效果为由大变小缩放；第五张背景图片的切换效果为由小变大旋转。具体实现步骤如下：

- (1) 通过@keyframes 属性分别设置每一个背景图切换时的动画效果。并分别设置元素在 0%和 100%处的动画状态。
- (2) 通过使用:target 选择器控制 animation 属性来定义背景图切换动画播放的时间和次数。

9.1.2 制作页面结构

根据上面的分析，使用相应的 HTML 标记搭建网页结构，如例 9-14 所示。

例 9-14 example14.html

```

1  <!doctype html>
2  <html>
3  <head>
4  <meta charset="utf-8">
5  <title>工作日天气预报</title>
6  </head>
7  <body>
8  <section>
9      <ul class="slider">
10         <li><a href="#bg1">周一狂躁综合症</a></li>
11         <li><a href="#bg2">周二雷霆似雨浇</a></li>
12         <li><a href="#bg3">周三阴郁似乌云</a></li>
13         <li><a href="#bg4">周四冬日盼春来</a></li>
14         <li><a href="#bg5">周五守得艳阳天</a></li>
15     </ul>
16     
17     
18     
19     
20     
21 </section>
22 </body>
23 </html>

```

在例 9-14 中，最外层使用<section>标记对页面进行整体控制。另外，分别定义 class 为 slider 的标记，来搭建天气图标模块的结构。同时，通过标记控制每一个具体的天气图标，并嵌套<a>标记来制作天气图标中的圆角矩形模块。此外，分别添加 5 个标记来搭建背景图片的结构。

运行例 9-14，效果如图 9-28 所示。

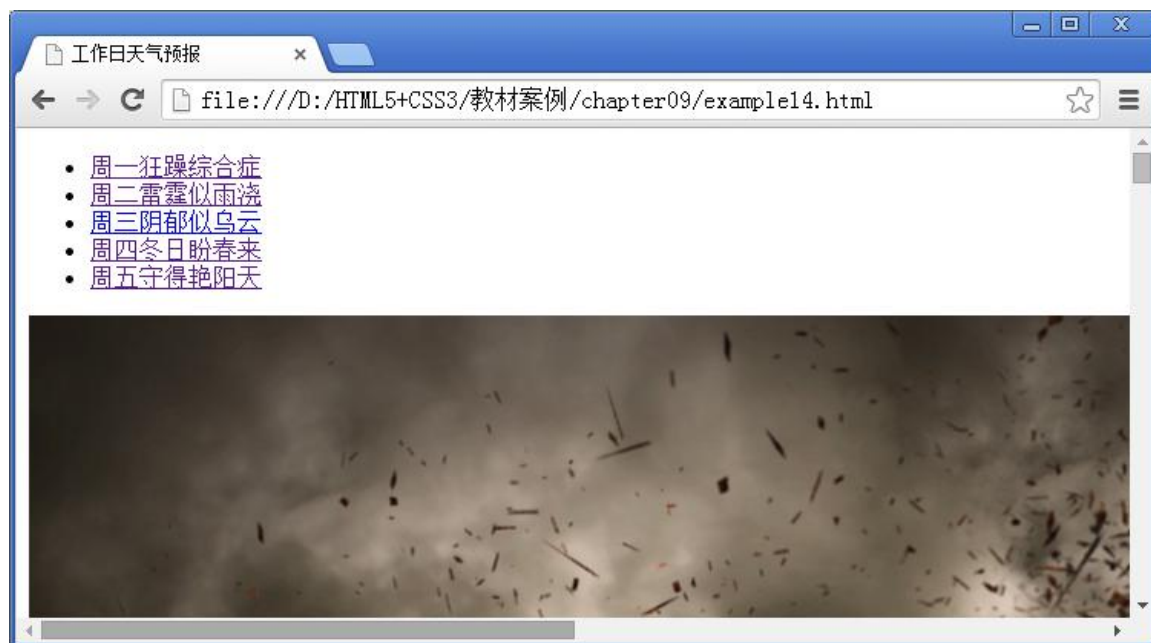


图 9-28 HTML 页面结构效果

9.1.3 定义 CSS 样式

搭建完页面的结构，接下来为页面添加 CSS 样式。本节采用从整体到局部的方式实现图 9-24 所示的效果，具体如下。

1、定义基础样式

首先定义页面的全局样式，具体 CSS 代码如下：

```
/*重置浏览器的默认样式*/
body, ul, li, p, h1, h2, h3, img {margin:0; padding:0; border:0; list-style:none;}
/*全局控制*/
body{font-family:'微软雅黑';}
a:link,a:visited{text-decoration:none;}
```

2、控制背景图片的样式

制作页面结构时，我们将五个定义为同一个类名 bg，来实现对网页背景图片的统一控制。通过 CSS 样式设置其宽度和高度 100%显示，并设置“min-width”为 1024 像素。另外，设置背景图片依据浏览器窗口来定义自己的显示位置，同时定义层叠性为 1。具体 CSS 代码如下：

```
img.bg {
    width: 100%;
    height: auto!important;
    min-width: 1024px;
    position: fixed; /*固定定位*/
```



```
z-index:1;          /*设置 z-index 层叠等级为 1;*/
}
```

3、整体控制每个天气图标的样式

观察效果图 9-24 可以看出，页面上包含五个样式相同的天气图标，分别由五个标记搭建结构。由于五个天气图标在一行内并列显示，需要将标记转为行内块元素并设置宽高属性。另外，为了使各个天气图标间拉开一定的距离，需要设置合适的外边距。具体 CSS 代码如下：

```
.slider li {
    display: inline-block;  /*将块元素转为行内块元素*/
    width: 170px;
    height: 130px;
    margin-right: 15px;
}
```

4、绘制天气图标的圆角矩形

每个天气图标由一个圆形图标和一个圆角矩形组成。对于圆角矩形模块，可以将<a>元素转为行内块元素来设置宽度和不同的背景色，并且通过边框属性设置圆角效果。另外，由于每个圆角矩形模块中都包含说明性的文字，需要设置文本样式，并通过 text-shadow 属性设置文字阴影效果。此外，圆形图标需要依据圆角矩形进行定位，所以将圆角矩形设置为相对定位。具体 CSS 代码如下：

```
.slider a {
    width: 170px;
    font-size:22px;
    color:#fff;
    display:inline-block;  /*将行内元素转为行内块元素*/
    padding-top:70px;
    padding-bottom:20px;
    border:2px solid #fff;
    border-radius:5px;      /*设置圆角边框*/
    position:relative;     /*相对定位*/
    cursor:pointer;        /* 光标呈现为指示链接的手型指针*/
    text-shadow:-1px -1px 1px rgba(0, 0, 0, 0.8),-2px -2px 1px rgba(0, 0, 0, 0.3),-3px
-3px 1px rgba(0, 0, 0, 0.3);
}

/*分别控制每个天气图标圆角矩形的背景色*/
.slider li:nth-of-type(1) a {background-color:#9d907f;}
.slider li:nth-of-type(2) a {background-color:#19425e;}
.slider li:nth-of-type(3) a {background-color:#58a180;}
.slider li:nth-of-type(4) a {background-color:#a1c64a;}
.slider li:nth-of-type(5) a {background-color: #ffc103;}
```

5、设置天气图标的圆形图标

对于天气图标的圆形图标，是将天气图片设置为圆角效果形成的，所以需要在结构中插入天气图片。首先，使用 after 伪元素可以在<a>标记之后插入天气图片。然后，通过 CSS3 中的边框属性设置天气图片显示为圆形。最后，设置圆形天气图标相对于圆角矩形模块绝对定位。具体 CSS 代码如下：

```
/* 设置 after 伪元素选择器的样式*/
```

```
.slider a::after {
    content: "";
    display: block;
    height: 120px;
    width: 120px;
    border: 5px solid #fff;
    border-radius: 50%;
    position: absolute; /*相对与<a>元素绝对定位*/
    left: 50%;
    top: -80px;
    z-index: 9999; /*设置 z-index 层叠等级为 9999;*/
    margin-left: -60px;
}
/* 使用 after 伪元素在<a>标记之后插入内容*/
.slider li:nth-of-type(1) a::after {
    background: url(images/sbg1.jpg) no-repeat center;
}
.slider li:nth-of-type(2) a::after {
    background: url(images/sbg2.jpg) no-repeat center;
}
.slider li:nth-of-type(3) a::after {
    background: url(images/sbg3.jpg) no-repeat center;
}
.slider li:nth-of-type(4) a::after {
    background: url(images/bg4.jpg) no-repeat center;
}
.slider li:nth-of-type(5) a::after {
    background: url(images/sbg5.jpg) no-repeat center;
}
```

6、设置圆形天气图标鼠标移上状态

当鼠标移上网页中的天气图标时，天气图标中的图片将会变亮，需要使用 `before` 伪元素在 `<a>` 标记之前插入一个和圆形天气图标大小、位置相同的盒子，并且设置其背景的不透明度为 0.3。当鼠标移上时，将其不透明度设置为 0，以实现图片变亮的效果。具体 CSS 代码如下：

```
/*设置 before 伪元素选择器的样式*/
.slider a::before {
    content: "";
    display: block;
    height: 120px;
    width: 120px;
    border: 5px solid #fff;
    border-radius: 50%;
    position: absolute; /*相对与<a>元素绝对定位*/
    left: 50%;
```

```

top: -80px;
margin-left: -60px;
z-index: 99999; /*设置 z-index 层叠等级为 9999;*/
background: rgba(0,0,0,0.3);
}
/*设置鼠标移上时 before 伪元素的样式*/
.slider a:hover::before {opacity:0;}

```

至此，我们完成了效果图 9-24 所示的工作日天气预报页面的 CSS 样式部分，将该样式应用于网页后，效果如图 9-29 所示。当鼠标移上网页中的圆形天气图标时，天气图标中的图片将会变亮。

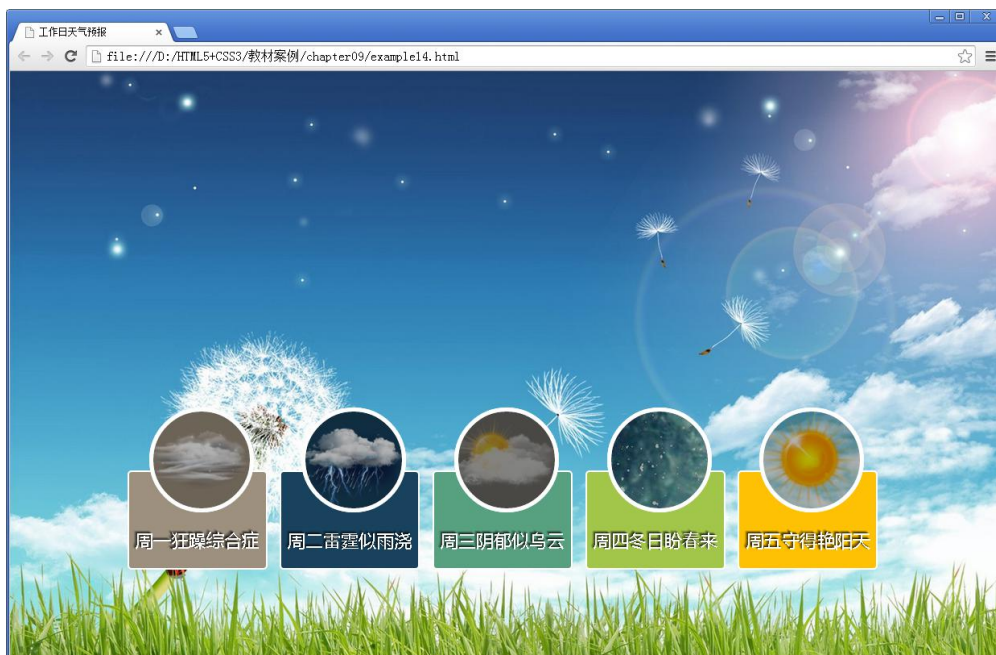


图 9-29 工作日天气预报效果

9.1.4 制作 CSS3 动画

本节分步骤来实现效果图 9-24 中所示的各个背景图切换的动画效果，继续在 CSS 样式中添加代码，具体如下。

1、设置第一个背景图切换的动画效果

通过案例演示可以看出，第一个背景图片切换效果为从左向右移动，可以通过@keyframes 属性设置元素在 0%和 100%处的 left 属性值，指定当前关键帧在应用动画过程中的位置。另外，使用:target 选择器控制 animation 属性定义单击链接时执行 1 秒钟播放完成 1 次切换动画。同时，设置其 z-index 层叠性为 100，具体代码如下：

```

@keyframes 'slideLeft' {
    0% { left: -500px; }
    100% { left: 0; }
}
@-webkit-keyframes 'slideLeft' {
    0% { left: -500px; }

```

```
    100% { left: 0; }
}
@-moz-keyframes 'slideLeft' {
    0% { left: -500px; }
    100% { left: 0; }
}
@-o-keyframes 'slideLeft' {
    0% { left: -500px; }
    100% { left: 0; }
}
@-ms-keyframes 'slideLeft' {
    0% { left: -500px; }
    100% { left: 0; }
}
/*当单击链接时，为所链接到的内容指定样式*/
.slideLeft:target {
    z-index: 100;
    animation: slideLeft 1s 1;
    -webkit-animation: slideLeft 1s 1;
    -moz-animation: slideLeft 1s 1;
    -ms-animation: slideLeft 1s 1;
    -o-animation: slideLeft 1s 1;
}
```

2、设置第二个背景图切换的动画效果

第二个背景图片切换效果为从下向上移动，可以通过@keyframes 属性设置元素在 0%和 100%处的 top 属性值，指定当前关键帧在应用动画过程中的位置。另外，使用:target 选择器控制 animation 属性来定义单击链接时切换动画播放的时间和次数。具体代码如下：

```
@keyframes 'slideBottom' {
    0% { top: 350px; }
    100% { top: 0; }
}
@-webkit-keyframes 'slideBottom' {
    0% { top: 350px; }
    100% { top: 0; }
}
@-moz-keyframes 'slideBottom' {
    0% { top: 350px; }
    100% { top: 0; }
}
@-ms-keyframes 'slideBottom' {
    0% { top: 350px; }
    100% { top: 0; }
}
```



```
@-o-keyframes 'slideBottom' {
  0% { top: 350px; }
  100% { top: 0; }
}
/*当单击链接时，为所链接到的内容指定样式*/
.slideBottom:target {
  z-index: 100; /*设置 z-index 层叠等级 100;*/
  animation: slideBottom 1s 1; /*定义动画播放时间和次数*/
  -webkit-animation: slideBottom 1s 1;
  -moz-animation: slideBottom 1s 1;
  -ms-animation: slideBottom 1s 1;
  -o-animation: slideBottom 1s 1;
}
```

3、设置第三个背景图切换的动画效果

第三个背景图片切换效果为由小变大展开，需要通过@keyframes 属性设置元素在 0%处的动画状态为元素缩小为 10%；100%处的动画状态为元素正常显示。并且，使用 animation 属性来定义单击链接时切换动画播放的时间和次数。具体代码如下：

```
@keyframes 'zoomIn' {
  0% { -webkit-transform: scale(0.1); }
  100% { -webkit-transform: none; }
}
@-webkit-keyframes 'zoomIn' {
  0% { -webkit-transform: scale(0.1); }
  100% { -webkit-transform: none; }
}
@-moz-keyframes 'zoomIn' {
  0% { -moz-transform: scale(0.1); }
  100% { -moz-transform: none; }
}
@-ms-keyframes 'zoomIn' {
  0% { -ms-transform: scale(0.1); }
  100% { -ms-transform: none; }
}
@-o-keyframes 'zoomIn' {
  0% { -o-transform: scale(0.1); }
  100% { -o-transform: none; }
}
.zoomIn:target { /*当单击链接时，为所链接到的内容指定样式*/
  z-index: 100; /*设置 z-index 层叠等级为 100;*/
  animation: zoomIn 1s 1;
  -webkit-animation: zoomIn 1s 1;
  -moz-animation: zoomIn 1s 1;
  -ms-animation: zoomIn 1s 1;
}
```

```
-o-animation: zoomIn 1s 1;
}
```

4、设置第四个背景图切换的动画效果

第四个背景图片切换效果为由大变小缩放，需要通过@keyframes 属性设置元素在 0%处的动画状态为元素放大两倍，100%处的动画状态为元素正常显示。具体代码如下：

```
@keyframes 'zoomOut' {
    0% { -webkit-transform: scale(2); }
    100% { -webkit-transform: none; }
}
@-webkit-keyframes 'zoomOut' {
    0% { -webkit-transform: scale(2); }
    100% { -webkit-transform: none; }
}
@-moz-keyframes 'zoomOut' {
    0% { -moz-transform: scale(2); }
    100% { -moz-transform: none; }
}
@-ms-keyframes 'zoomOut' {
    0% { -ms-transform: scale(2); }
    100% { -ms-transform: none; }
}
@-o-keyframes 'zoomOut' {
    0% { -o-transform: scale(2); }
    100% { -o-transform: none; }
}
.zoomOut:target { /*当单击链接时，为所链接到的内容指定样式*/
    z-index: 100; /*设置 z-index 层叠等级 100;*/
    animation: zoomOut 1s 1;
    -webkit-animation: zoomOut 1s 1;
    -moz-animation: zoomOut 1s 1;
    -ms-animation: zoomOut 1s 1;
    -o-animation: zoomOut 1s 1;
}
```

5、设置第五个背景图切换的动画效果

第五个背景图片切换效果为由小变大旋转。通过@keyframes 属性设置元素在 0%处的动画状态为逆时针旋转 360 度，元素缩小为 10%；100%处的动画状态为元素正常显示。并且，使用 animation 属性定义单击链接时 1 秒钟播放完成 1 次切换动画。具体代码如下：

```
@keyframes 'rotate' {
    0% { -webkit-transform: rotate(-360deg) scale(0.1); }
    100% { -webkit-transform: none; }
}
@-webkit-keyframes 'rotate' {
    0% { -webkit-transform: rotate(-360deg) scale(0.1); }
```

```
    100% { -webkit-transform: none; }
}
@-moz-keyframes 'rotate' {
    0% { -moz-transform: rotate(-360deg) scale(0.1); }
    100% { -moz-transform: none; }
}
@-ms-keyframes 'rotate' {
    0% { -ms-transform: rotate(-360deg) scale(0.1); }
    100% { -ms-transform: none; }
}
@-o-keyframes 'rotate' {
    0% { -o-transform: rotate(-360deg) scale(0.1); }
    100% { -o-transform: none; }
}
.rotate:target { /*当单击链接时，为所链接到的内容指定样式*/
    z-index: 100; /*设置 z-index 层叠等级为 100;*/
    animation: rotate 1s 1;
    -webkit-animation: rotate 1s 1;
    -moz-animation: rotate 1s 1;
    -ms-animation: rotate 1s 1;
    -o-animation: rotate 1s 1;
}
```

6、实现背景图交互性切换效果

为了使背景图可以有序地进行切换，需要排除当前单击链接时的元素，并为其他元素执行 1 秒钟播放完成 1 次的背景切换动画。另外，通过@keyframes 属性定义元素在 0%和 100%处的层叠性，设置单击链接后的背景图处于当前背景图片的下一层，实现背景图交互性切换效果。具体代码如下：

```
@keyframes 'notTarget' {
    0% { z-index: 75; } /*动画开始时的状态*/
    100% { z-index: 75; } /*动画结束时的状态*/
}
@-webkit-keyframes 'notTarget' {
    0% { z-index: 75; }
    100% { z-index: 75; }
}
@-moz-keyframes 'notTarget' {
    0% { z-index: 75; }
    100% { z-index: 75; }
}
@-ms-keyframes 'notTarget' {
    0% { z-index: 75; }
    100% { z-index: 75; }
}
@-o-keyframes 'notTarget' {
```

```
0% { z-index: 75; }
100% { z-index: 75; }
}
.bg:not(:target) { /*排除当前单击链接时的 target 元素，为其他 target 元素元素指定动画样式*/
  animation: notTarget 1s 1;
  -webkit-animation: notTarget 1s 1;
  -moz-animation: notTarget 1s 1;
  -ms-animation: notTarget 1s 1;
  -o-animation: notTarget 1s 1;
}
```

保存 CSS 样式文件，刷新页面，点击天气图标时，背景图片发生改变，效果如图 9-30 所示。



图 9-30 CSS3 动画效果

至此，工作日天气预报主题页面的 HTML 结构、CSS 样式以及动画特效已经全部制作完成。通过本案例的制作，相信读者已经对 CSS3 中的过渡和动画有了更深入的认识，并能够在实际项目开发中熟练运用。