

模块三

图文类网页

问题提出：与文字不同，图像是一种视觉语言形式，既能传达特定的信息，又能丰富页面组成、美化网页的整体风格，还能使页面充满一定的感情色彩。为了使页面内容更直观、更丰富、更漂亮，具有较高的浏览与交互体验，在网页中如何设计图像、如何进行图文混合设计、如何利用 CSS3 的新特性设计动态效果呢？本模块将基于“盐城文化风情”网站首页与图片展示页面 Picture.html 讲解其核心知识点与设计网页的具体过程和方法。

核心概念：HTML5 图像标签，CSS 背景，CSS 圆角边框，CSS 过渡，CSS 阴影，CSS 渐变，CSS 变形，CSS 动画，复合选择器。

HTML5 图像标签： 标签与 <figure> 标签，可以使页面更加生动、多彩。

CSS 背景：background，允许应用纯色作为背景，也允许使用背景图像甚至多个背景图像创建复杂的效果。

CSS 圆角边框：border-radius，实现多样圆角边框效果。

CSS3 过渡：transition，可平滑地改变一个元素的 CSS 值，使元素从一个样式逐渐过渡到另一个样式。

CSS 阴影：box-shadow，可实现给区块添加一个或多个阴影。

CSS 渐变：linear-gradient 或 radial-gradient，在两个或多个指定的颜色之间实现平稳过渡。

CSS 变形：transform，对元素应用 2D 或 3D 转换，允许对元素进行旋转、缩放、移动或倾斜。

CSS 动画：包含一组定义的动画关键帧和描述该动画的 CSS 声明。可以逐渐地从一个值变化到另一个值，比如尺寸大小、数量、百分比和颜色。

复合选择器：是基本选择器通过不同的连接方式构成的，通过复合选择器可以精确匹配页面元素。

学习目标：

- 掌握两种图像标签的使用方法。
- 熟悉 CSS 背景的设置方法。
- 领会 CSS 圆角边框的设置方法。
- 掌握 CSS 过渡的使用方法。
- 掌握 CSS 复合选择器的使用规则。
- 掌握 CSS 阴影的设置方法。
- 领会渐变的语法规则及使用方法。
- 掌握动画的语法规则及使用方法。

- 领会变形的语法规则及使用方法。
- 能综合应用 CSS 圆角边框、阴影、渐变、过渡、动画、变形、背景、复合选择器设计多样化的图像效果。
- 搜集、整理、分析、归纳家乡文化(如盐城)特色,进而了解家乡文化,热爱家乡文化,培养家国情怀、建设家乡的责任感和使命感;同时通过对表现家乡文化图片的搜集、选取及优化处理,提升审美能力和艺术修养。

项目 3.1 “盐城文化风情”网站首页设计

3.1.1 项目描述

“盐城文化风情”网站主要介绍盐城的风土人情、文化特色,主要包括红色文化、绿色文化和白色文化,以及地方美食、特色经典等。其首页(index.html)效果如图 3-1 所示。



图 3-1 “盐城文化风情”网站首页效果

3.1.2 项目分析

1. 页面整体结构

“盐城文化风情”网站首页页面结构如图 3-2 所示。

2. 具体实现细节

页面分为四个部分:< header>、< footer>和两个< section>,具体细节如下。

(1) < header>中包含一个< nav>导航条。

(2) 第一个< section>标签包含四个< div>,每个< div>有一幅背景图像,基于 transition 过渡以及 JavaScript 实现四幅图像依次淡入淡出,形成简单的轮播图效果。

(3) 第二个< section>标签包含三个< article>标签,每个< article>标签包含一个 h3 标题和一个列表,列表外边框为圆角边框效果。

(4) < footer >标签中包含一个< p >标签。

另外,为了便于方便快捷地定位某个元素,引入了复合选择器。

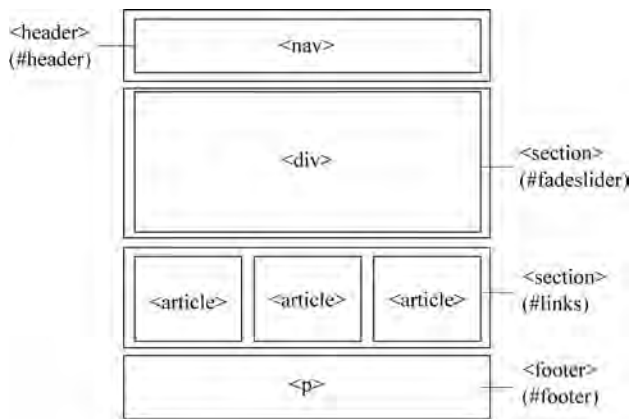


图 3-2 “盐城文化风情”网站首页页面结构

3.1.3 项目知识点分解

通过项目分析可知,此网页包含 HTML5 图像标签、背景图像、圆角边框、复合选择器。本项目涉及的知识点如图 3-3 所示。

3.1.4 知识点解析

1. HTML5 图像标签

在网页设计中,若只有文字排版的页面会显得过于单调和拥挤,为了提升浏览者阅读的积极性,可以在页面中插入图像。下面介绍两种 HTML5 中常用的图像标签。

1) < img >标签

< img >标签用于定义网页中的图像,语法格式如下。

```
< img src = "图片路径" alt = "图片无法显示时显示的文字">
```

其中,src 属性和 alt 属性是< img >标签的必需属性。src 属性用来指定图像源,即图像的 URL,该路径可以是相对路径,也可以是绝对路径,建议使用相对路径;alt 属性可在 Web 浏览器不支持图像显示或用户关闭图像下载功能时,告诉浏览者该处是一幅什么样的图像,推荐在文档的每幅图像中都使用这个属性。

2) < figure >标签和< figcaption >标签

< figure >标签是 HTML5 中的新标签。< figure > 标签规定独立的流内容(图像、图表、照片、代码等)。figure 元素的内容应该与主内容相关,但如果被删除,则不对文档流产生影响。所有主流浏览器都支持 < figure > 标签。

< figcaption >标签是嵌套在< figure >标签中使用的,用于指定流内容的标注。

例如,流内容是一幅安丰古镇图片,就可以用< figcaption >标签来标注“东台安丰镇风光”,具体使用方法如下。

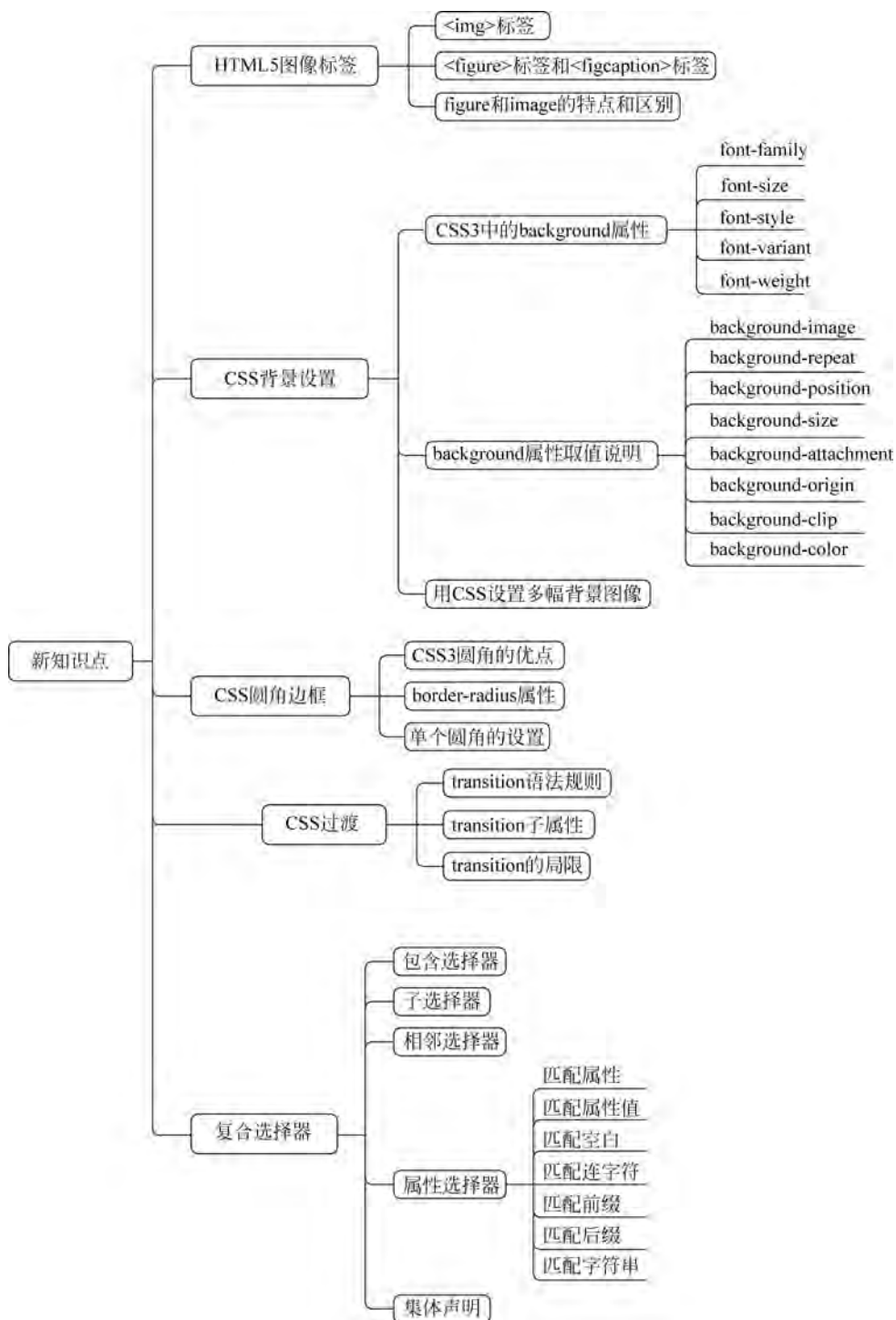


图 3-3 项目 3-1 知识点分解

demo3-1/图像标签.html:

```
<!doctype html>
<html>
<head>
<meta charset = "utf - 8">
<title>HTML 图像标签</title>
</head>

<body>
<figure>
  <figcaption>东台市安丰镇</figcaption>
  <p>安丰这块古老的地方,有着悠久的历史,最早的文字记载起于距今 1280 年的唐开元元年(713 年)。据考,安丰初名东淘,地濒沧海,常遭海涛侵袭,致使地方不宁,民不聊生。至北宋仁宗天圣五年(1027 年),西溪盐仓监范仲淹率民夫,修海堤,以挡海潮,方改名安丰。“安”寓安居乐业之意,“丰”寄丰衣足食之愿。为实现安居乐业,丰衣足食之愿望,安丰人民搏击海涛,烧凶煎盐,推动了历史前进。</p>
  <img src = "anfeng.jpg" alt = "抱歉,您的图片不能显示!" />
</figure>
</body>
</html>
```

用浏览器打开此实例,页面效果如图 3-4 所示。



图 3-4 demo3-1 页面效果

3) figure 和 image 的特点和区别

特点:

(1) figure 元素是一种元素的组合,可带有标题。<figure>标签用来表示网页上一块

独立的内容,将其从网页上移除后不会对网页上的其他内容产生影响。figure 所表示的内容可以是图片、统计图或代码示例。figure 拥有一个子标签——<figcaption>标签。

(2) img 元素向网页中嵌入一幅图像。这个标签并不会在网页中插入图像,而是从网页上链接图像。标签创建的是被引用图像的占位空间。

区别: figure 用于对元素进行组合,里面可以添加很多其他标签,包括标签,多用于图片与图片描述组合。而只是一个图片元素,可以嵌套在<figure>中使用。

课堂小实践

参考图像标签的例子,创建一个介绍家乡的页面,既包含<figure>标签,也包含<image>标签,注意布局与美化。

2. CSS 背景设置

CSS 背景是指 CSS 对对象设置背景属性,即通过 CSS 设置背景的各种样式。

1) CSS3 中的 background 属性

```
background: background-image || background-position/background-size || background-repeat ||  
background-attachment || background-clip || background-origin || background-color
```

也可以分解写成:

```
background-image: url();  
background-position: <length> || <per>  
background-size: <length> || <per>  
background-repeat: repeat || repeat-x || repeat-y || no-repeat;  
background-attachment: scroll || fixed;  
background-clip: padding-box || border-box || content-box;  
background-origin: padding-box || border-box || content-box;  
background-color: color 值 || RGBA 值;
```

如果使用连写方式时,background-size 需跟在 background-position 的后面,并用“/”隔开,即“background-position/background-size”。

建议将 CSS3 中的属性拆分出来单独书写,因为它们在不同浏览器下需要加上自己的前缀,如下。

```
background: background-color || background-image || background-repeat || background-  
attachment || background-position;  
background-size: <length> || <per>  
background-clip: padding-box || border-box || content-box;  
background-origin: padding-box || border-box || content-box;
```

建议在运用后面三个属性时,将各自私有的前缀加上。为什么要加前缀呢?

浏览器核心的部分称为“浏览器内核”,不同的浏览器内核对网页编写语法的解释也有不同,因此同一网页在不同内核的浏览器里的渲染(显示)效果也可能不同。为了让 CSS3 样式兼容,需要将某些样式加上浏览器前缀,可根据不同的浏览器内核添加不同的前缀,例如 Mozilla 内核(Firefox 浏览器)需要加上“-moz”,而 Webkit(Safari、Chrome 浏览器)内核需要加上“-webkit”,IE 浏览器需要加上“-ms”。



2) background 属性取值说明

(1) background-image。background-image 属性用来设置元素的背景图片,可以使用相对地址或绝对地址索引背景图片;元素的背景占据了元素的全部尺寸,包括内边距和边框,但不包括外边距。默认时,背景图像位于元素的左上角,并在水平方向和垂直方向上重复。

(2) background-repeat。background-repeat 属性用来设置元素的背景图片的平铺方式,默认情况下,背景图像在水平方向和垂直方向上重复,其五个属性值及其描述如表 3-1 所示。

表 3-1 background-repeat 属性值及其描述

属 性 值	描 述
repeat	默认。背景图像将在垂直方向和水平方向重复
repeat-x	背景图像将在水平方向重复
repeat-y	背景图像将在垂直方向重复
no-repeat	背景图像将仅显示一次
inherit	规定应该从父元素继承 background-repeat 属性的设置

(3) background-position。background-position 属性用来设置元素的背景图片的定位起点,其默认值为 left top; 具体属性值如表 3-2 所示。

表 3-2 background-position 属性值及其描述

属 性 值	描 述
top left top center top right center left center center center right bottom left bottom center bottom right	如果仅规定了一个关键词,那么第二个值将是 center。 默认值: 0 0
x% y%	第一个值是水平位置,第二个值是垂直位置。 左上角是 0 0。右下角是 100% 100%。 如果仅规定了一个值,另一个值将是 50%
xpos ypos	第一个值是水平位置,第二个值是垂直位置。 左上角是 0 0。单位是像素(0px 0px)或任何其他 CSS 单位。 如果仅规定了一个值,另一个值将是 50%。 可以混合使用%和 position 值



(4) background-size。background-size 属性用来设置元素的背景图片的尺寸大小,其默认值为 auto。CSS3 中可以指定背景图片,可以重新在不同的环境中指定背景图片的大小。可以指定像素或百分比大小,指定的大小是相对于父元素的宽度和高度的百分比大小。

语法规则如下：

background-size: length|percentage|cover|contain;

background-size 具体属性值及其描述如表 3-3 所示。

表 3-3 background-size 具体属性值及其描述

属 性 值	描 述
length	设置背景图片的高度和宽度。第一个值设置宽度,第二个值设置高度。如果只给出一个值,第二个值设置为 auto(自动)
percentage	将计算相对于背景定位区域的百分比。第一个值设置宽度,第二个值设置高度。如果只给出一个值,第二个值设置为 auto(自动)
contain	按比例调整背景图片,使得其图片宽高比自适应整个元素的背景区域的宽高比,所以若指定的图片尺寸过大,而背景区域的整体宽高不能恰好包含背景图片,那么其背景某些区域可能会有空白
cover	按比例调整背景图片,这个属性值与 contain 正好相反,背景图片会按照比例自适应铺满整个背景区域。假如背景区域不足以包含背景图片,那么那些部分就无法显示在背景区域中

下面通过一个案例验证 background-size 取不同属性值对应的效果。

demo3-2/background-size. html:

```
<!doctype html >
<html >
<head >
<meta charset = "utf - 8">
<title > background - size </title >
<style >
.length {
    background: url(bgsizel.jpg) no-repeat;
    background-size: 300px 300px;
}
.per {
    background: url(bgsizel.jpg) no-repeat;
    background-size: 50 % 60 % ;
}
.con {
    background: url(bgsizel.jpg) no-repeat;
    background-size: contain;
    width: 400px;
    height: 200px;
    border: 1px solid rgba(236,10,14,1.00);
}
.cov {
    background: url(bgsizel.jpg) no-repeat;
    background-size: cover;
    width: 400px;
    height: 200px;
    border: 1px solid rgba(236,10,14,1.00);
}
</style >
```



```
</head>
<body>
<div class = "length"> background - size 属性值为 length 时<br>
```

background-size 用来设置元素的背景图片的尺寸大小,其默认值为 auto。CSS3 以前,背景图像大小由图像的实际大小决定。CSS3 中可以指定背景图片,可以重新在不同的环境中指定背景图片的大小。可以指定像素或百分比大小,指定的大小是相对于父元素的宽度和高度的百分比的大小。

```
</div>
<br>
<br>
<div class = "per"> background - size 属性值为 percentage 时<br>
    background - size 此值用来设置元素的背景图片的尺寸大小,其默认值为 auto。...
</div>
<br>
<br>
<div class = "con"> background - size 属性值为 contain 时<br>
    <br>
    <br>
</div>
<br>
<div class = "cov"> background - size 属性值为 cover 时<br>
</div>
</body>
</html>
```

(5) background-attachment。background-attachment 属性用来设置元素的背景图片是否为固定显示,其默认值为 scroll。background-attachment 具体属性值及其描述如表 3-4 所示。

表 3-4 background-attachment 具体属性值及其描述

属 性 值	描 述
scroll	默认值。背景图像会随着页面其余部分的滚动而移动
fixed	背景图像静止。即当页面的其余部分滚动时,背景图像不会移动
inherit	规定应该从父元素继承 background-attachment 属性的设置

(6) background-origin。background-origin 属性用来控制元素的背景图像 position 的默认起始点,其默认值为 padding-box。background-origin 具体属性值及其描述如表 3-5 所示。

表 3-5 background-origin 具体属性值及其描述

属 性 值	描 述
border-box	背景图像相对于边框盒来定位
padding-box	背景图像相对于内边距框来定位
content-box	背景图像相对于内容框来定位

下面通过一个案例验证 background-origin 取不同属性值对应的效果。
demo3-3/background-origin. html:

```
<!doctype html>
```

```

<html>
<head>
<meta charset = "utf - 8">
<title> background - origin</title>
<style>
# example1 {
    border: 10px dotted black;
    padding: 35px;
    width: 400px;
    height: 400px;
    background: url(huahai. jpg) no - repeat yellow;
    - moz - background - origin: border - box;
    - webkit - background - origin: border - box;
    background - origin: border - box;
}
# example2 {
    border: 10px dotted black;
    padding: 35px;
    width: 400px;
    height: 400px;
    background: url(huahai. jpg) no - repeat yellow;
    - moz - background - origin: padding - box;
    - webkit - background - origin: padding - box;
    background - origin: padding - box;
}
# example3 {
    border: 10px dotted black;
    padding: 35px;
    width: 400px;
    height: 400px;
    background: url(huahai. jpg) no - repeat yellow;
    - moz - background - origin: content - box;
    - webkit - background - origin: content - box;
    background - origin: content - box;
}
</style>
</head>
<body>
<p> background - origin: border - box, 背景图像相对于边框盒来定位。 </p>
<div id = "example1">
    <h2>大丰荷兰花海</h2>
    <p>盐城荷兰花海度假区位于盐城市区东南斗龙港生态组团,大丰城区北的新丰镇,紧邻徐大高速大丰北出口,距大丰城区 5 分钟车程,距国家一类口岸大丰港 15 分钟车程,距盐城南洋国际机场 25 分钟车程。系盐城市统筹城乡发展的试点,江苏省四星级乡村旅游点,江苏省智慧旅游和电子商务示范基地。荷兰花海深度挖掘“民国村镇规划第一村”的历史底蕴,以 1919 年荷兰水利专家特莱克规划新丰农田水利为渊源,秉持“立足盐城、承接上海、辐射长三角”的功能定位,围绕“观光旅游、婚纱摄影、健康养年”三大产业布局,全力打造中国连片种植郁金香面积最大、种类最多的“中国郁金香第一花海”。</p>
</div>
<p> background - origin: padding - box, 背景图像相对于内边距框来定位。 </p>
<div id = "example2">

```

```
<h2>大丰荷兰花海</h2>
<p>盐城荷兰花海度假区位于盐城市区东南斗龙港生态组团……</p>
</div>
<p>background-origin: content-box, 背景图像相对于内边距框来定位。</p>
<div id="example3">
  <h2>大丰荷兰花海</h2>
  <p>盐城荷兰花海度假区位于盐城市区东南斗龙港生态组团……</p>
</div>
</body>
</html>
```

(7) background-clip。background-clip 属性用来控制元素的背景图片显示区域,其默认值为 border-box; background-clip 具体属性值及其描述如表 3-6 所示。

表 3-6 background-clip 具体属性值及其描述

属 性 值	描 述
border-box	背景被裁剪到边框盒
padding-box	背景被裁剪到内边距框
content-box	背景被裁剪到内容框

下面通过一个案例验证 background-clip 取不同属性值对应的效果。
demo3-4/background-clip. html:

```
<!doctype html >
<html >
<head>
<meta charset="utf-8">
<title>background-clip</title>
<style>
#example1 {
  border: 10px dotted black;
  padding: 35px;
  width: 600px;
  height: 402px;
  background: url(ddh.jpg) no-repeat yellow;
  -moz-background-clip: border-box;
  -webkit-background-clip: border-box;
  background-clip: border-box;
}
#example2 {
  border: 10px dotted black;
  padding: 35px;
  width: 600px;
  height: 402px;
  background: url(ddh.jpg) no-repeat yellow;
  -moz-background-clip: padding-box;
  -webkit-background-clip: padding-box;
  background-clip: padding-box;
}
#example3 {
```

```
border: 10px dotted black;
padding: 35px;
width: 600px;
height: 402px;
background: url(ddh.jpg) no-repeat yellow;
-moz-background-clip: content-box;
-webkit-background-clip: content-box;
background-clip: content-box;
}
</style>
</head>
<body>
<p> background-clip 属性值为 border-box, 背景被裁剪到边框盒。 </p>
<div id="example1">
  <h2>射阳丹顶鹤保护区</h2>
  <p>射阳丹顶鹤保护区全称江苏盐城国家级珍禽自然保护区, 又称“联合国教科文组织盐城生物圈保护区”。由江苏省人民政府于 1983 年批准建立, 1992 年经国务院批准晋升为国家级自然保护区, 同年 11 月被联合国教科文组织世界人与生物圈协调理事会批准为生物圈保护区, 成为中国第九个“世界生物圈保护区网络成员”, 1999 年被纳入“东亚——澳大利亚迁徙涉禽保护网络”。</p>
</div>
<p> background-clip 属性值为 padding-box, 背景被裁剪到内边距框。 </p>
<div id="example2">
  <h2>射阳丹顶鹤保护区</h2>
  <p>射阳丹顶鹤保护区全称江苏盐城国家级珍禽自然保护区……</p>
</div>
<p> background-clip 属性值为 content-box, 背景被裁剪到内容框。 </p>
<div id="example3">
  <h2>射阳丹顶鹤保护区</h2>
  <p>射阳丹顶鹤保护区全称江苏盐城国家级珍禽自然保护区……</p>
</div>
</body>
</html>
```

background-origin 与 background-clip 虽然两者看上去实现的效果差不多,但是它们的原理是不同的。background-origin 定义的是背景位置(background-position)的起始点;而 background-clip 是对背景(图片和背景色)的切割。

(8) background-color。background-color 属性用来设置元素的背景色。background-color 具体属性值及其描述如表 3-7 所示。

表 3-7 background-color 具体属性值及其描述

属 性 值	描 述
color_name	规定颜色值为颜色名称的背景颜色(比如 red)
hex_number	规定颜色值为十六进制值的背景颜色(比如 #ff0000)
rgb_number	规定颜色值为 rgb 代码的背景颜色(比如 rgb(255,0,0))
transparent	默认。背景颜色为透明
inherit	规定应该从父元素继承 background-color 属性的设置

3) 用 CSS 设置多幅背景图像

CSS3 可将同一个元素设置多幅背景图像。多幅背景图像用逗号隔开,越靠前的图像层级越高。每一幅背景图像的写法都和以前单幅背景图像的写法相同,还可以指定图片的

平铺方式、位置等。如下面代码所示。

```
background:url("1.jpg") 0 0 no-repeat,
            url("2.jpg") 200px 0 no-repeat,
            url("3.jpg") 400px 201px no-repeat;
```

也可以这样写：

```
background-image:url("1.jpg"),url("2.jpg"),url("3.jpg");
background-repeat: no-repeat, no-repeat, no-repeat;
background-position: 0 0, 200px 0, 400px 201px;
```

下面通过一个案例展现 CSS3 的多背景效果。

demo3-5/multipleBg.htm:

```
<!doctype html >
<html >
<head >
<meta charset = "utf-8">
<title>multipleBg</title>
<style>
.demo {
    width: 340px;
    border: 20px solid rgba(104, 104, 142, 0.5);
    -moz-border-radius: 10px;
    -webkit-border-radius: 10px;
    border-radius: 10px;
    padding: 80px 60px;
    color: #f36;
    font-size: 25px;
    line-height: 1.5;
    text-align:center;
}
.multipleBg {
    background:url(lefttop.png) no-repeat left top, url(righttop.png)no-repeat right top,
    url(leftbottom.png)no-repeat left bottom,url(rightbottom.png) no-repeat right bottom,url
    (bg.jpg) repeat left top;
    /* 改变背景图片的 position 起始点,四朵花都是 border 边缘处起 */
    -webkit-background-origin: border-box, border-box, border-box, border-box, padding-box;
    -moz-background-origin: border-box, border-box, border-box, border-box, padding-box;
    -o-background-origin: border-box, border-box, border-box, border-box, padding-box;
    background-origin: border-box, border-box, border-box, border-box, border-box;
    /* 控制背景图片的显示区域,所有背景图片被裁剪到边框盒 */
    -moz-background-clip: border-box, border-box, border-box, border-box, padding-box;
    -webkit-background-clip: border-box, border-box, border-box, border-box, padding-box;
    -o-background-clip: border-box, border-box, border-box, border-box, padding-box;
    background-clip: border-box, border-box, border-box, border-box, padding-box;
}
</style>
</head>
<body >
<div class = "demo multipleBg">我使用了五张背景图片制作这样的效果</div>
```

```
</body>
</html>
```

其页面效果如图 3-5 所示。

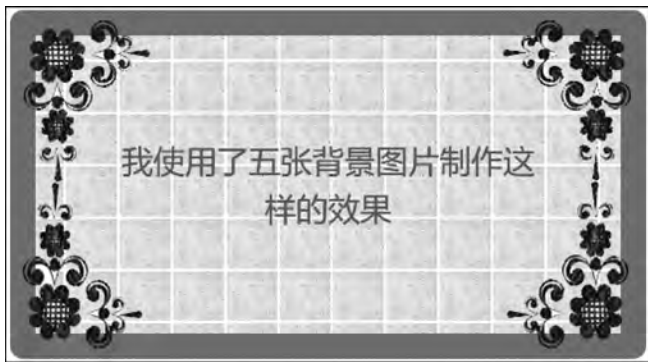


图 3-5 demo3-5 页面效果

课堂小实践

运用所学的 background 相关属性设置精美的网页背景。

3. CSS 圆角边框

1) CSS3 圆角的优点

在 CSS3 之前,对于网页中圆角背景的制作,大多采用的方法是在需要圆角的元素标签中加四个空标签,然后在每个空标签中设置圆角的背景图像,再对这几个应用了圆角的标签进行定位到相应的位置,具体实现过程相对烦琐。

而 CSS3 的 border-radius 出现后,制作圆角边框的方法较为简单,不需要设计圆角图像,也不需要考虑图像载入网页中的各种问题,减少了维护的工作量,提高了网页的性能,增加了视觉可靠性。

2) border-radius 属性

CSS3 圆角只需设置一个属性,即 border-radius(含义是“边框半径”)。为这个属性提供一个值,就能同时设置四个圆角的半径。所有合法的 CSS 度量值都可以使用 em、ex、pt、px、百分比等。

border-radius 语法规则如下。

```
border-radius: none | <length>{1,4} [ / <length>{1,4} ]?
```

页面中已有一个 div 蓝色区块,现设置它的圆角半径为 15px,即“border-radius:15px;”,则方形的蓝色区块变成如图 3-6 所示的圆角矩形。

border-radius 是一种缩写方法。如果“/”前后的值都存在,那么“/”前面的值设置其水平半径,“/”后面的值设置其垂直半径。如果没有“/”,则水平半径和垂直半径相等,图 3-6 就是水平半径与垂直半径相等时的效果。另外,其四个值是按照 top-left、top-right、bottom-right、bottom-left 的顺序来设置



图 3-6 圆角矩形效果



的,主要有下面几种情形。

- (1) 只有一个值,那么 top-left、top-right、bottom-right、bottom-left 四个值相等。
- (2) 有两个值,那么 top-left 等于 bottom-right,并且取第一个值; top-right 等于 bottom-left,并且取第二个值。
- (3) 有三个值,其中第一个值是设置 top-left,而第二个值是 top-right 和 bottom-left 并且它们会相等,第三个值是设置 bottom-right。
- (4) 有四个值,其中第一个值是设置 top-left,第二个值是 top-right,第三个值是 bottom-right,第四个值是 bottom-left。即依次对应左上角、右上角、右下角、左下角(顺时针顺序)。

不同 border-radius 属性值对应圆角矩形效果,如表 3-8 所示。

表 3-8 不同 border-radius 属性值对应圆角矩形效果

属 性 值	效 果
border-radius: 25px 10px;	
border-radius: 25px 10px 45px;	
border-radius: 35px 10px 20px 0px;	
border-radius: 35px/10px;	
border-radius: 10px 20px 30px 40px/40px 30px 20px 10px; (按顺时针的顺序,斜杠“/”左边是四个圆角的水平半径,右边是四个圆角的垂直半径)	

3) 单个圆角的设置

除了同时设置四个圆角以外,还可以单独对每个角进行设置。对应四个角,CSS3 提供四个单独的属性: `border-top-left-radius`、`border-top-right-radius`、`border-bottom-right-radius`、`border-bottom-left-radius`,这四个属性都可以同时设置一或两个值。如果设置一个值,表示水平半径与垂直半径相等。如果设置两个值,第一个值表示水平半径,第二个值表示垂直半径。

课堂小实践

题目 1: 按照图 3-7,写出对应代码。

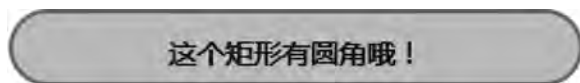
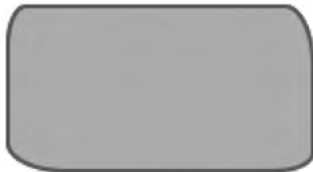


图 3-7 圆角边框效果

题目 2: 按照图 3-8,写出对应代码。

4. CSS 过渡

CSS3 的过渡就是平滑地改变一个元素的 CSS 值,使元素从一个样式逐渐过渡到另一个样式。



1) transition 语法规则

`transition: property duration timing-function delay;`

图 3-8 圆角边框效果

默认值分别为: `all 0 ease 0`。下面通过一个案例展现 CSS3 的 transition 效果。

demo3-6/transition.html:

```
<!doctype html>
<html>
<head>
<meta charset = "utf-8">
<title>transition</title>
<style type = "text/CSS">
div {
    width: 100px;
    height: 100px;
    background: red;
    -moz-transition: background 2s;           /* Firefox 4 */
    -webkit-transition: background 2s;       /* Safari and Chrome */
    -o-transition: background 2s;           /* Opera */
    transition: background 2s;
}
div:hover {
    background: green;
}
</style>
</head>
<body>
<div></div>
```

<p>当鼠标指针移动到红色 div 元素上,可以看到 div 元素在 2s 内由红变成绿的过渡效果。</p>

```
<p><b>注释: </b>本例在 Internet Explorer 中无效。</p>
</body>
</html>
```

注意：要实现 CSS 过渡的效果，必须规定两项内容：一是规定应用过渡的 CSS 属性名称；二是规定效果的时长。

2) transition 子属性

transition 属性是复合属性，包括 transition-property、transition-duration、transition-timing-function、transition-delay 几个子属性，其对应的描述如表 3-9 所示。

表 3-9 transition 子属性及其描述

子 属 性	描 述
transition-property	规定应用过渡的 CSS 属性的名称。 语法规则：transition-property: none all property; none, 没有属性会获得过渡效果; all, 所有属性都将获得过渡效果; 值为指定的 CSS 属性应用过渡效果, 多个属性间用逗号隔开
transition-duration	定义过渡效果花费的时间。 语法规则：transition-duration: time; 该属性主要用来设置一个属性过渡到另一个属性所需的时间, 即过渡持续时间
transition-timing-function	规定过渡效果的时间曲线。 语法规则：transition-timing-function: linear ease ease-in ease-out ease-in-out cubic-bezier(n,n,n,n); 该属性指的是过渡的“缓动函数”。主要用来指定浏览器的过渡速度, 以及过渡期间的操作进展情况。 注意: 值 cubic-bezier(n,n,n,n) 中可以定义自己的值, 如 cubic-bezier(0.42,0,0.58,1)
transition-delay	规定效果开始之前需要等待的时间, 为 time 值, 就是过渡效果开始前的延迟时间, 单位为秒或者毫秒

注意：若要改变多个 CSS 属性的过渡效果时，多个属性值对间用逗号隔开，比如：
a{transition: background 0.8s ease-in 0.3s,color 0.6s ease-out 0.3s;}，即在过渡效果发生时超链接的背景色和文字颜色同时发生变化。

3) transition 的局限

transition 的优点在于简单易用，但是它也有如下几个局限。

- (1) transition 需要事件触发，所以没法在网页加载时自动发生。
- (2) transition 是一次性的，不能重复发生，除非一再触发。
- (3) transition 只能定义开始状态和结束状态，不能定义中间状态，也就是说只有两个状态。

课堂小实践

页面中原本有一个方环，当鼠标停在方环上时，方环用 3s 的速度过渡到圆环。

5. 复合选择器

在 Web 页面设计中三种基本选择器远远不能满足要求，遇到更为复杂的情况时，该怎

么设计选择器呢? 以这三种选择器为基础,通过组合,可以产生更多种类的选择器,实现更方便、更强的选择功能。

复合选择器就是基本选择器通过不同的连接方式构成的,通过复合选择器可以精确匹配页面元素。主要包括包含选择器、子选择器、相邻选择器、属性选择器、集体声明等。

1) 包含选择器

包含选择器中前后两部分之间以空格隔开,根据左侧选择符指定的祖先元素,在该元素下寻找匹配右侧选择符的下级元素。定义包含选择器时,必须保证在 HTML 结构中第一个对象能够包含第二个对象,即 HTML 代码中的标签父辈元素包含子孙辈元素时,则设置相应的样式就有效。例如:

```
div p{color:red;font-size:20px;}
```

前面的 div 是包含选择符标识符,后面的 p 是被包含选择符名,这个语法就会选择从 div 元素继承的所有 p 元素(不论 p 的嵌套层次多深)将文字颜色变为红色、字体大小改成 20px。

定义包含选择器时,前后两个选择符可以是三种基本选择符中的任一种,但前一个选择符对应的标签必须能够包含后一个选择符对应的标签,即 HTML 代码中的标签父辈元素包含子孙辈元素时,则设置相应的样式就有效。

下面通过一个案例展现包含选择器效果。

demo3-7/包含选择器.html:

```
<!doctype html >
<html >
<head >
<meta charset = "utf - 8">
<title>包含选择器</title>
<style type = "text/CSS">
/* 定义类样式 */
.div1 {
    font-size: 14px;
    color: #FF0000;
}
/* 定义类样式下的标题元素 */
.div1 h1 {
    color: #FF00FF;
}
/* 定义类样式下的 span 元素 */
.div1 span {
    color: #0000FF;
}
</style>
</head>
<body>
<div class = "div1">这是类
    <h1>这是个标题<span>这是标题中的 span 元素</span></h1>
</div>
</body>
</html>
```

案例中,div 与 h1、div 与 span 都是包含关系,都能正确匹配,效果如图 3-9 所示。

这是类

这是个标题这是标题中的span元素

图 3-9 包含选择器案例效果

2) 子选择器

子选择器中前后部分之间用>隔开,前后两部分选择符在结构上属于父子关系。

子选择器是根据左侧选择符指定的父元素,之后在该父元素下寻找匹配右侧选择符的子元素,只有是父子关系才可受其 CSS 控制,祖父与孙子级别的元素则不受控制。例如:

```
div>p{color:red;font-size:20px;}
```

即表示所有父元素是 div 的 p 元素文字颜色变为红色、字体大小变为 20px。

demo3-7/子选择器.html:

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>子选择器</title>
<style type="text/CSS">
/* 设置父元素是 div 元素的 p 元素的样式 */
div>p {
    font-size: 12px;
    color: #0000FF;
}
/* 设置父元素是 p 元素的 a 元素的样式 */
p>a {
    font-size: 18px;
    color: #FF0000;
    text-decoration: none;
}
div>a {
    font-size: 24px;
    color: #00FF00;
    text-decoration: underline;
}
</style>
</head>
<body>
<div>
    <p>这是个段落<a href="#">这是超链接</a></p>
</div>
</body>
</html>
```

案例中,div 与 a 不是父子关系,不能正确匹配,所以 a 的样式不发生变化,效果如图 3-10 所示。

这是个段落 这是超链接

图 3-10 子选择器案例效果

课堂讨论

包含选择器和子选择器有何区别?

3) 相邻选择器

相邻选择器前后部分之间用一个加号(+)隔开,前后两部分选择符在结构上属于同级关系,即根据左侧选择符指定相邻元素,之后在该相邻元素后面寻找匹配右侧选择符的相邻元素。例如:

```
div + p{color:red;font-size:20px;}
```

即表示与 div 相邻的 p 元素文字颜色变为红色、字体大小变为 20px。

demo3-7/相邻选择器.html:

```
<!doctype html >
<html >
<head >
<meta charset = "utf - 8">
<title>相邻选择器</title>
<style type = "text/CSS">
# sub_wrap + p {
    font-size: 22px;
    color: red;
}
</style>
</head>
<body>
<div id = "wrap">
    <div id = "sub_wrap">
        <h2 class = "news">标题 2 </h2>
        <p class = "news">标题 2 下的段落</p>
        <span class = "news">标题 2 下的 span</span>
    </div>
    <p>相邻段落</p>
</div>
</body>
</html>
```

案例中,div #sub_wrap 是 p 的相邻元素,能正确匹配,所以 p 的文字大小和颜色都发生变化。

4) 属性选择器

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,是选择器的最高级形态,也是选择器发展的总趋势。

(1) 匹配属性。

若希望为所有包含 href 属性的超链接定义背景色,则用如下方法定义。

CSS:

```
a[href]{
    background: #cccccc;
}
```

HTML:

```
<h2>[attr]</h2>
<p><a name = "anchor">存在属性 href 才行</a>
<p><a href = "#">存在属性 href 才行</a>
```

(2) 匹配属性值。

若要为表单元素类型为 type 且值为 text 的元素设置背景色,则用如下方法定义。

CSS:

```
input[type = text]{
    background: #111111;
}
```

HTML:

```
<form>
    <input type = "text"/>属性值为 text 才行
    <br/>
    <input type = "button"/>属性值为 text 才行
</form>
```

(3) 匹配空白。

如果希望类别选择器中的值包含字母 first,若 class 内有多值,各个值之间用空格间隔设置背景色及文本颜色,则用如下方法定义。

CSS:

```
[class~ = first]{
    background: #030000;
    color: red;
}
```

HTML:

```
<h2>[attr~ = "value"]</h2>
<ul>
    <li class = "first">属性中存在或者含有 first,需要空格分隔</li>
    <li class = "second">属性中存在或者含有 first,需要空格分隔</li>
    <li class = "third">属性中存在或者含有 first,需要空格分隔</li>
    <li class = "first second">属性中存在或者含有 first,需要空格分隔</li>
    <li class = "first third">属性中存在或者含有 first,需要空格分隔</li>
    <li class = "second third">属性中存在或者含有 first,需要空格分隔</li>
    <li class = "first second third">属性中存在或者含有 first,需要空格分隔</li>
</ul>
```

(4) 匹配连字符。

与匹配空白一样,不同的是类别选择器中的值存在 first 值,若 class 内有多值,各个值之间用连字符间隔设置背景色和文本颜色,则用如下方法定义。

CSS:

```
[class| = "first"]{
```

```
background: # 030;
}
```

HTML:

```
<h2>[attr| = "value"]</h2>
<ul>
  <li class = "first">属性中存在或者含有 first,需要连字符分隔</li>
  <li class = "second">属性中存在或者含有 first,需要连字符分隔</li>
  <li class = "third">属性中存在或者含有 first,需要连字符分隔</li>
  <li class = "first - second">属性中存在或者含有 first,需要连字符分隔</li>
  <li class = "first - third">属性中存在或者含有 first,需要连字符分隔</li>
  <li class = "second - third">属性中存在或者含有 first,需要连字符分隔</li>
  <li class = "first - second - third">属性中存在或者含有 first,需要连字符分隔</li>
</ul>
```

(5) 匹配前缀。

若希望为段落提示属性 title 的值以 hello 开头的元素设置背景色和文本颜色,则用如下方法定义。

CSS:

```
[title^ = "hello"] {
  background: # 030000;
  color: red;
}
```

HTML:

```
<h2>[attr^ = "value"]</h2>
<p title = "good">属性的开头必须是 hello</p>
<p title = "hellom">属性的开头必须是 hello</p>
<p title = "Thehellom">属性的开头必须是 hello</p>
```

(6) 匹配后缀。

如果希望段落提示属性 title 的值以 bye 结尾设置背景色和文本颜色,则用如下方法定义。

CSS:

```
[title$ = "bye"] {
  background: # 030000;
  color: red;
}
```

HTML:

```
<h2>[attr$ = "value"]</h2>
<p title = "hello">属性的结尾必须是 bye</p>
<p title = "goodbye">属性的结尾必须是 bye</p>
<p title = "goodbye - 2">属性的结尾必须是 bye</p>
```

(7) 匹配字符串。

上述已给出 6 种属性选择器的使用方法,最后给出一种匹配子字符串,只要属性值包含指定的字符串,都可以定义样式,而不再受限制于匹配的位置和属性值的格式。例如:

CSS:

```
[title * = "good"] {
    background: # 030000;
    color: red;
}
```

HTML:

```
<h2>[attr * = "value"]</h2>
<p title = "hello">属性中含有 good 吗</p>
<p title = "good">属性中含有 good 吗</p>
<p title = "goodbook">属性中含有 good 吗</p>
```

其他部分复合选择器,如伪类选择器到后续课程中再讲解。

5) 集体声明

在定义 CSS 属性样式时,有些选择器定义的属性完全一致或部分一致,可以通过集体声明将一致的 CSS 属性放在一起,即将这些选择器用逗号连接,以减少代码的书写量。

下面是一个关于集体声明、包含选择器、子选择器、属性选择器、相邻选择器的综合案例。

demo3-7/集体声明.html:

```
<!doctype html >
<html >
<head >
<meta charset = "utf - 8">
<title>集体声明</title>
<style type = "text/CSS">
/* 六个标题文字集体声明 */
h1, h2, h3, h4, h5, h6 {
    background-color: # 99cc33;
    margin: 0;
    margin-bottom: 10px;
}
/* 与 h1 相邻的 h2、与 h3 相邻的 h4、与 h5 相邻的 h6 */
h1 + h2, h3 + h4, h5 + h6 {
    color: # 0099ff;
}
/* body 的子元素 h6、h1 的子元素 span、h4 的子元素 span */
body > h6, h1 > span, h4 > span {
    font-size: 40px;
}
/* h2 的后代元素 span, h3 的后代元素 span */
h2 span, h3 span {
    padding: 0 20px;
}
/* 拥有 class 属性且是 h5 后代元素的 span, 拥有 class 属性且是 h6 后代元素的 span */
h5 span[class], h6 span[class] {
    background-color: # cc0033;
}
</style>
</head>
<body>
<h1> h1 元素< span>这里是 span 元素</span></h1>
```



```

<h2> h2 元素< span>这里是 span 元素</span></h2>
<h3> h3 元素< span>这里是 span 元素</span></h3>
<h4> h4 元素< span>这里是 span 元素</span></h4>
<h5> h5 元素< span class = "s1">这里是 span 元素</span></h5>
<h6> h6 元素< span class = "s2">这里是 span 元素</span></h6>
</body>
</html>

```

3.1.5 知识点检测



3.1.6 项目实现

通过项目分析和知识点学习,相信读者已能逐步完成“盐城文化风情”网站 index. html 页面的代码编写。参考代码如下。

index. html:

```

<!DOCTYPE html>
<html lang = "en">
<head>
<meta charset = "utf - 8">
<meta name = "viewport" content = "width = device - width, initial - scale = 1.0, minimum - scale
= 1.0, maximum - scale = 1.0, user - scalable = no">
<meta name = "author" content = "helang.love@qq.com">
<title>盐城文化风情网欢迎您!</title>
<link rel = "stylesheet" href = "CSS/style.CSS">
<!-- 图标样式 -->
<link rel = "stylesheet" type = "text/CSS" href = "CSS/font - awesome.min.CSS">
</head>
<body>
<header id = "header">
    <nav class = "hl_nav">
        <ul class = "nav_list">
            <li class = "active"><a class = "nav_head" href = "#"><i class = "icon fa fa - home">
</i><span>网站首页</span></a></li>
            <li><a class = "nav_head" href = "#"><i class = "icon fa fa - star"></i><span>红色文
化</span></a></li>
            <li><a class = "nav_head" href = "#"><i class = "icon fa fa - diamond"></i><span>白
色文化</span></a></li>
            <li><a class = "nav_head" href = "#"><i class = "icon fa fa - tree"></i><span>绿色文
化</span></a></li>
            <li><a class = "nav_head" href = "#"><i class = "icon fa fa - delicious"></i><span>特
色美食</span></a></li>
            <li><a class = "nav_head" href = "#"><i class = "icon fa fa - bicycle"></i><span>特
色景点</span></a></li>
            <li><a class = "nav_head" href = "#"><i class = "icon fa fa - comments"></i><span>与
我联系</span></a></li>

```

```

        </ul>
    </nav>
</header>
<section id="fadeslider">
    <div id="bg1" class="bg"></div>
    <div id="bg2" class="bg fadein"></div>
    <div id="bg3" class="bg fadein"></div>
    <div id="bg4" class="bg fadein"></div>
</section>
<setion id="links">
    <article class="cultures news">
        <h3><i class="fa fa-flag fa-2x"></i>文化名篇赏析</h3>
        <ul>
            <li><a href="#">盐城红色文化 - 铁军精神</a></li>
            <li><a href="#">盐城白色文化 - 海盐文化</a></li>
            <li><a href="#">盐城绿色文化 - 湿地文化</a></li>
            <li><a href="#">盐城历史文化大全</a></li>
            <li><a href="#">作家邹凤岭畅谈盐城地域文化</a></li>
            <li><a href="#">盐文化与盐城精神</a></li>
            <li><a href="#">盐城大力传承红色文化 让铁军精神激励后人奋发有为</a></li>
        </ul>
    </article>
    <article class="tourism news">
        <h3><i class="fa fa-bicycle fa-2x"></i>走进旅游景点</h3>
        <ul>
            <li><a href="#">大丰麋鹿保护区</a></li>
            <li><a href="#">海盐历史文化景区</a></li>
            <li><a href="#">丹顶鹤湿地生态旅游区</a></li>
            <li><a href="#">荷兰花海</a></li>
            <li><a href="#">新四军纪念馆(国家 4A 级)</a></li>
            <li><a href="#">盐镇水街(国家 4A 级)</a></li>
            <li><a href="#">中国海盐博物馆</a></li>
            <li><a href="#">大纵湖景区(国家 4A 级)</a></li>
        </ul>
    </article>
    <article class="food news">
        <h3><i class="fa fa-delicious fa-2x"></i>美食速递</h3>
        <ul>
            <li><a href="#">伍佑醉螺</a></li>
        </ul>
        </article>
</setion>
<footer id="footer">
    <p>Copyright © 2019 小新工作室 All Rights Reserved. 盐城市开放大道 50 号</p>
</footer>
<script type="text/javascript">
//替换 class 达到淡入淡出的效果//
function fadeIn(e) {
    e.className = "bg fadein"
};
function fadeOut(e) {

```

```

        e.className = "bg"
    };
    //声明图片数组中当前的轮播图片
    cur_img = document.getElementById("fadeslider").children.length - 1;
    //图片轮播函数
    function turnImgs(fadeslider) {
        var imgs = document.getElementById("fadeslider").children;
        if (cur_img == 0) {
            fadeOut(imgs[cur_img]);
            cur_img = imgs.length - 1;
            fadeIn(imgs[cur_img]);
        } else {
            fadeOut(imgs[cur_img]);
            fadeIn(imgs[cur_img - 1]);
            cur_img--;
        }
    }
    //设置轮播间隔
    setInterval(turnImgs, 3000);
</script>
</body>
</html>

```

对应的 style. CSS 代码：

```

body {
    margin: 0;
    padding: 0;
    background-color: #F3F3F3;
    font-size: 14px;
    font-family: 'Microsoft YaHei', 'Times New Roman', Times, serif;
    letter-spacing: 0;
    min-width: 1200px;
    color: #333333;
}
ul {
    margin: 0;
    padding: 0;
}
/* 给 header 标签设置样式规则 */
body {
    margin: 0;
    padding: 0;
    background-color: #F3F3F3;
    font-size: 14px;
    font-family: 'Microsoft YaHei', 'Times New Roman', Times, serif;
    letter-spacing: 0;
    min-width: 1200px;
    color: #333333;
}
ul {

```

```

        margin: 0;
        padding: 0;
    }
    /* 给 header 标签设置样式规则 */
    .hl_nav {
        list-style: none;
        position: relative;
        min-width: 1200px;
    }
    .hl_nav a { /* 包含选择器,设置祖先元素类名为.hl_nav 的子元素 a 的样式规则 */
        display: block;
        text-decoration: none;
    }
    .hl_nav .nav_list {
        position: absolute;
        top: 0;
        left: 50%;
        margin: 0 0 0 -600px;
        z-index: 2;
        padding: 0;
        list-style: none;
        width: 1200px;
        height: 64px;
    }
    .hl_nav .nav_list > li { /* 子选择器,设置父元素为.nav_list 的子元素 li 的样式规则 */
        padding: 0;
        float: left;
        margin: 0;
        width: 150px;
        text-align: center;
    }
    .hl_nav .nav_head {
        line-height: 64px;
        font-size: 16px;
        color: #333333;
    }
    .hl_nav .nav_head .icon {
        font-size: 18px;
    }
    .hl_nav .nav_list > li:hover .nav_head, .hl_nav .nav_list > li.active .nav_head {
        background-color: rgba(255,157,0,1.00);
    }
    /* 给实现轮播图效果的 section 标签设置样式规则 */
    #fadeslider .bg {
        position: absolute;
        left: 0;
        top: 64px;
        width: 100%;
        height: 700px;
        -webkit-transition: opacity 2s linear;
        -moz-transition: opacity 2s linear;
    }

```

```

        - o - transition: opacity 2s linear;
        transition: opacity 2s linear;    /* 过渡,透明度在 2s 内实现由 0 线性过渡为 100% */
        opacity: 0;
    }
    #fadeslider #bg1 {
        background: url(.. /images/Index_img1.jpg) no-repeat;
        background-size: cover;    /* 背景图像的尺寸,背景图像会按照比例自适应铺满整个背景区域 */
    }
    #fadeslider #bg2 {
        background: url(.. /images/Index_img2.jpg) no-repeat;
        background-size: cover;
    }
    #fadeslider #bg3 {
        background: url(.. /images/Index_img3.jpg) no-repeat;
        background-size: cover;
    }
    #fadeslider #bg4 {
        background: url(.. /images/Index_img4.jpg) no-repeat;
        background-size: cover;
    }
    #fadeslider .fadein {
        opacity: 100;
        filter: alpha(opacity=100);
    }
    /* 给 id 名为 #links 的 section 标签设置样式规则 */
    #links {
        position: absolute;
        top: 764px;
        width: 1200px;
        left: 50%;
        margin: 0 0 0 -600px;
    }
    #links .news {
        float: left;
        width: 300px;
        height: 260px;
        margin: 25px;
        padding: 10px;
        border-radius: 10px;    /* 圆角半径为 10px */
        border: 1px dashed rgba(177,168,168,1.00);
    }
    #links .news h3 {
        margin: 10px 0 15px 20px;
        font-size: 1.5em;
        color: rgba(255,157,0,1.00);
    }
    #links .news ul {
        margin-left: 60px;
    }
    #links .news li {

```

```
        list-style: circle;
        line-height: 1.7em;
    }
    #links.news li a {
        text-decoration: none;
        color: #666;
    }
    #links.cultures h3 {
        color: red;
    }
    #links.tourism h3 {
        color: green;
    }
    #links.food img {
        width: 70%;
        margin-left: 45px;
        margin-top: 10px;
        border-radius: 6px;
    }
    /* 给 footer 标签设置样式规则 */
    #footer {
        clear: both;
        position: absolute;
        top: 1100px;
        width: 100%;
        height: 100px;
        background: #999;
        color: #fff;
    }
    #footer p {
        margin: 40px;
        text-align: center;
    }
}
```

3.1.7 项目总结

实施本项目的过程中,重点要练习的是使用图像标签,利用 CSS 背景、过渡、圆角边框美化页面,利用复合选择器精准定位到网页中的元素;要注意复合选择器的正确应用,以及过渡的语法规则;搜集、整理、分析、归纳家乡文化(如盐城)特色,进而了解家乡文化、热爱家乡文化,培养家国情怀、建设家乡的责任感和使命感。

项目 3.2 “图说我们盐城”页面设计

3.2.1 项目描述

“图说我们盐城”页面(Picture.html)主要通过红色文化、白色文化、绿色文化以及美食文化四个子模块展示盐城的文化特色,页面效果如图 3-11 所示。



图 3-11 “盐城文化风情”网站图片展示页面效果

3.2.2 项目分析

1. 页面整体结构

Picture.html 页面结构是典型的上中下结构,页面结构如图 3-12 所示。

2. 具体实现细节

页面结构简单,由< header >、< section >和< footer >组成,其中,< section >是图片主要展区,包含四个< article >,即四个图片展示模块。具体细节如下。

- (1) 页面头部< header >标签嵌套了< img >标签。
- (2) 每个图片展示模块中的背景实现了由透明到白色的渐变,标题部分由一幅标题图片和一个< span >组成。
- (3) 红色文化图片展示模块包含 6 幅图片,每幅图片都有阴影效果。
- (4) 白色文化图片展示模块包含 6 幅图片,每幅图片都设置了圆角矩形效果。
- (5) 绿色文化图片展示模块包含 8 幅图片,每幅图片都有对应的图片描述,当鼠标悬停

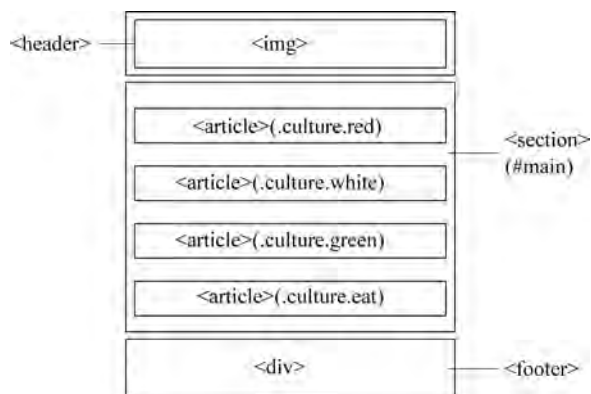


图 3-12 “盐城文化风情”网站图片展示页面结构

到每幅图片上时,会从图片底部浮出一个区块层,以对本幅图片进行描述。

(6) 美食文化图片展示模块还包含 8 幅图片,每幅图片都有对应的图片描述,当鼠标悬停到每幅图片上时,会显示出文字描述信息区块层,该层上应用了旋转和过渡的组合,形成了旋转动画效果。

(7) 页面尾部< footer >标签嵌套了< div >,用以放入尾部导航。

3.2.3 项目知识点分解

Picture.html 页面中实现了图片的阴影效果、图片的旋转动画效果、渐变背景效果等,所以该项目涉及的新知识点导图如图 3-13 所示。

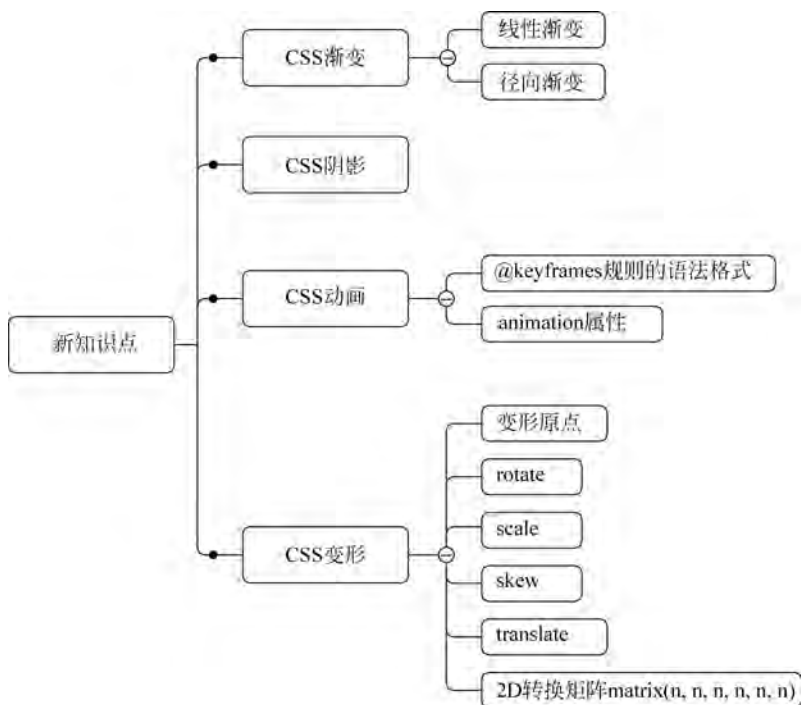


图 3-13 项目 3-2 知识点分解

3.2.4 知识点解析

1. CSS 渐变

渐变背景在 Web 页面中极为常见,但在 CSS3 之前,设计师都是通过图形软件设计渐变效果,之后以图片形式或者背景图片的形式运用到页面中。很多时候,还需进行切图,再通过样式应用到页面中,过程比较烦琐;而且在实际应用中可扩展性差,还直接影响页面性能。

自从 W3C 组织将渐变设计收入到 CSS3 标准中,渐变(gradients)很好地解决了上述问题。另外,渐变(gradient)是由浏览器生成的,放大时看起来效果更好。目前,越来越多的浏览器都支持这个属性了。

渐变在理论上可在任何使用 url() 值的地方采用,比如最常见的 background-image、list-style-type 以及前面介绍的 CSS3 的图像边框属性 border-image,但在背景图片中得到的支持较为完美。

CSS3 渐变(gradients)可以在两个或多个指定的颜色之间显示平稳过渡。CSS3 定义了线性渐变(linear gradients)和径向渐变(radial gradients)两种类型的渐变。

1) 线性渐变

为了创建一个线性渐变,必须至少定义两种颜色节点。颜色节点是想要呈现平稳过渡的颜色。同时,可以设置一个起点和一个方向(或一个角度)。语法规则如下。

```
background: linear-gradient(direction, color-stop1, color-stop2, ...);
```

(1) 线性渐变——从上到下(默认情况下)。

下面的代码演示了从上到下的线性渐变。起点是红色,慢慢过渡到蓝色。

CSS 部分:

```
#grad0 {
    height: 200px;
    background: -webkit-linear-gradient(red, blue); /* Safari 5.1 - 6.0 */
    background: -o-linear-gradient(red, blue); /* Opera 11.1 - 12.0 */
    background: -moz-linear-gradient(red, blue); /* Firefox 3.6 - 15 */
    background: linear-gradient(red, blue); /* 标准的语法(必须放在最后) */
}
```

HTML 部分:

```
<h3>线性渐变 - 默认情况下,从上到下</h3>
<p>从上边开始的线性渐变。起点是红色,慢慢过渡到蓝色:</p>
<div id="grad0"></div>
```

(2) 线性渐变——从左到右。

下面的代码演示了从左边开始的线性渐变。起点是红色,慢慢过渡到蓝色。

CSS 部分:

```
#grad1 {
    height: 200px;
    background: -webkit-linear-gradient(left, red, blue); /* Safari 5.1 - 6.0 */
```

```
background: -o-linear-gradient(left, red, blue);      /* Opera 11.1 - 12.0 */
background: -moz-linear-gradient(left, red, blue);    /* Firefox 3.6 - 15 */
background: linear-gradient(to right, red, blue);     /* 标准的语法(必须放在最后) */
}
```

HTML 部分:

```
<h3>线性渐变 - 从左到右</h3>
<p>从左边开始的线性渐变。起点是红色,慢慢过渡到蓝色:</p>
<div id="grad1"></div>
```

(3) 线性渐变——从右到左。

CSS 部分:

```
#grad2 {
    height: 200px;
    background: -webkit-linear-gradient(right, red, blue); /* Safari 5.1 - 6.0 */
    background: -o-linear-gradient(right, red, blue);    /* Opera 11.1 - 12.0 */
    background: -moz-linear-gradient(right, red, blue);   /* Firefox 3.6 - 15 */
    background: linear-gradient(to left, red, blue);      /* 标准的语法(必须放在最后) */
}
```

HTML 部分:

```
<h3>线性渐变 - 从右到左</h3>
<p>从右边开始的线性渐变。起点是红色,慢慢过渡到蓝色:</p>
<div id="grad2"></div>
```

(4) 线性渐变——对角。

可以通过指定水平和垂直的起始位置来制作一个对角渐变。

下面的实例演示了从左上角开始(到右下角)的线性渐变,起点是红色,慢慢过渡到蓝色。

CSS 部分:

```
#grad3 {
    height: 200px;
    background: -webkit-linear-gradient(top left, red, blue); /* Safari 5.1 - 6.0 */
    background: -o-linear-gradient(top left, red, blue);    /* Opera 11.1 - 12.0 */
    background: -moz-linear-gradient(top left, red, blue);   /* Firefox 3.6 - 15 */
    background: linear-gradient(to bottom right, red, blue); /* 标准的语法(必须放在最后) */
}
```

HTML 部分:

```
<h3>线性渐变 - 对角</h3>
<p>从左上角开始(到右下角)的线性渐变。起点是红色,慢慢过渡到蓝色:</p>
<div id="grad3"></div>
```

(5) 线性渐变——使用角度。

如果还想更多地控制渐变方向,可以改变渐变的角度,而不用预定义方向(to bottom、to top、to right、to left、to bottom right,等等)。

语法规则如下。

```
background: linear-gradient(angle, color-stop1, color-stop2);
```

角度是指水平线和渐变线之间的角度,逆时针方向计算。0deg 将创建一个从下到上的渐变,90deg 将创建一个从左到右的渐变,180deg 将创建一个从上到下的渐变,-90deg 将创建一个从右到左的渐变。

CSS 部分:

```
#grad4 {
    height: 100px;
    background: -webkit-linear-gradient(0deg, red, blue); /* Safari 5.1 - 6.0 */
    background: -o-linear-gradient(0deg, red, blue);      /* Opera 11.1 - 12.0 */
    background: -moz-linear-gradient(0deg, red, blue);    /* Firefox 3.6 - 15 */
    background: linear-gradient(0deg, red, blue);          /* 标准的语法(必须放在最后) */
}
#grad5 {
    height: 100px;
    background: -webkit-linear-gradient(90deg, red, blue); /* Safari 5.1 - 6.0 */
    background: -o-linear-gradient(90deg, red, blue);      /* Opera 11.1 - 12.0 */
    background: -moz-linear-gradient(90deg, red, blue);    /* Firefox 3.6 - 15 */
    background: linear-gradient(90deg, red, blue);          /* 标准的语法(必须放在最后) */
}
#grad6 {
    height: 100px;
    background: -webkit-linear-gradient(180deg, red, blue); /* Safari 5.1 - 6.0 */
    background: -o-linear-gradient(180deg, red, blue);      /* Opera 11.1 - 12.0 */
    background: -moz-linear-gradient(180deg, red, blue);    /* Firefox 3.6 - 15 */
    background: linear-gradient(180deg, red, blue);          /* 标准的语法(必须放在最后) */
}
#grad7 {
    height: 100px;
    background: -webkit-linear-gradient(-90deg, red, blue); /* Safari 5.1 - 6.0 */
    background: -o-linear-gradient(-90deg, red, blue);      /* Opera 11.1 - 12.0 */
    background: -moz-linear-gradient(-90deg, red, blue);    /* Firefox 3.6 - 15 */
    background: linear-gradient(-90deg, red, blue);          /* 标准的语法(必须放在最后) */
}
```

HTML 部分:

```
<h3>线性渐变 - 使用不同的角度</h3>
<div id="grad4" style="color:white;text-align:center;">0deg</div><br>
<div id="grad5" style="color:white;text-align:center;">90deg</div><br>
<div id="grad6" style="color:white;text-align:center;">180deg</div><br>
<div id="grad7" style="color:white;text-align:center;">-90deg</div>
```

其他角度的读者可以自己尝试一下。

(6) 线性渐变——使用多个颜色节点。

上述一直实现的是两个颜色节点的渐变,那如何实现多个颜色节点的渐变呢? 只需在参数中多加一个颜色节点即可。

CSS 部分:

```
#grad8 {
    height: 200px;
    background: -webkit-linear-gradient(red, green, blue); /* Safari 5.1 - 6.0 */
    background: -o-linear-gradient(red, green, blue);      /* Opera 11.1 - 12.0 */
    background: -moz-linear-gradient(red, green, blue);    /* Firefox 3.6 - 15 */
    background: linear-gradient(red, green, blue);         /* 标准的语法(必须放在最后) */
}
```

HTML 部分:

```
<h3>3 个颜色节点(均匀分布)</h3>
<div id="grad8"></div>
```

若要写出含 7 个颜色节点彩虹色块的效果,添加 7 个颜色节点即可,代码如下。

CSS 部分:

```
#grad9 {
    height: 200px;
    background: -webkit-linear-gradient(red, orange, yellow, green, blue, indigo,
violet);
    background: -o-linear-gradient(red, orange, yellow, green, blue, indigo, violet);
    background: -moz-linear-gradient(red, orange, yellow, green, blue, indigo, violet);
    background: linear-gradient(red, orange, yellow, green, blue, indigo, violet);
    /* 标准的语法(必须放在最后) */
}
```

HTML 部分:

```
<h3>7 个颜色节点(均匀分布)</h3>
<div id="grad9"></div>
```

再稍加修改下,添加一个方向 left,可以变成彩虹条的效果。

(7) 线性渐变——颜色分布不均匀。

上述颜色渐变都是均匀渐变,若颜色分布不均匀的渐变该如何设置呢? 下面的代码演示了不均匀渐变。

```
#grad10 {
    height: 200px;
    background: -webkit-linear-gradient(red 10%, green 85%, blue 90%);
    background: -o-linear-gradient(red 10%, green 85%, blue 90%);
    background: -moz-linear-gradient(red 10%, green 85%, blue 90%);
    background: linear-gradient(red 10%, green 85%, blue 90%);
    /* 标准的语法(必须放在最后) */
}
```

红色部分从顶端就开始着色,到 10%处开始渐变到绿色,渐变过渡区的占比为总的空间(高度或宽度)减去上下两个着色块空间占比剩下的空间。理解下面的绿色和蓝色开始着色及过渡的方法同上面的一致。

(8) 线性渐变——透明度变化。

CSS3 渐变也支持透明度(transparent),可用于创建透明度变化的效果。

要添加透明度,需使用 rgba()函数来定义颜色节点。rgba() 函数中的最后一个参数可以是从 0 到 1 的值,它定义了颜色的透明度:0 表示完全透明,1 表示完全不透明。具体实例如下,演示了从左边开始的线性渐变。起点是完全透明,慢慢过渡到完全不透明的红色。

CSS 部分:

```
#grad11 {
    height: 200px;
    background: -webkit-linear-gradient(to right, rgba(255,0,0,0), rgba(255,0,0,1));
                                /* Safari 5.1 - 6.0 */
    background: -o-linear-gradient(to right, rgba(255,0,0,0), rgba(255,0,0,1));
                                /* Opera 11.1 - 12.0 */
    background: -moz-linear-gradient(to right, rgba(255,0,0,0), rgba(255,0,0,1));
                                /* Firefox 3.6 - 15 */
    background: linear-gradient(to right, rgba(255,0,0,0), rgba(255,0,0,1));
                                /* 标准的语法(必须放在最后) */
}
```

HTML 部分:

<h3>线性渐变 - 透明度</h3>

<p>为了添加透明度,我们使用 rgba()函数来定义颜色节点。rgba() 函数中的最后一个参数可以是从 0 到 1 的值,它定义了颜色的透明度:0 表示完全透明,1 表示完全不透明。</p>

<div id="grad11"></div>

总的来看,上述线性渐变的效果主要取决于渐变的方向、颜色节点以及位置。

2) 径向渐变

径向渐变由它的中心定义渐变效果。为了创建一个径向渐变,必须至少定义两种颜色节点。颜色节点是想要呈现平稳过渡的颜色。同时,还可以指定渐变的中心、形状(圆形或椭圆形)、大小。默认情况下,渐变的中心是 center(表示在中心点),渐变的形状是 ellipse(表示椭圆形),渐变的大小是 farthest-corner(表示到最远的角落)。

语法规则如下。

```
background: radial-gradient(center, shape size, start-color, ..., last-color);
```

(1) 径向渐变——颜色节点均匀分布。

颜色节点均匀分布的径向渐变的代码如下。

CSS 部分:

```
#grad1 {
    height:150px;
    width:200px;
    background: -webkit-radial-gradient(green,purple,red); /* Safari 5.1 - 6.0 */
```

```

background: - o - radial - gradient(green, purple, red); /* Opera 11.6 - 12.0 */
background: - moz - radial - gradient(green, purple, red); /* Firefox 3.6 - 15 */
background: radial - gradient(green, purple, red); /* 标准的语法(必须放在最后) */
}

```

HTML 部分:

```

<h3>径向渐变 - 颜色节点均匀分布</h3>
<div id = "grad1"></div>

```

(2) 径向渐变——颜色节点不均匀分布。

颜色节点不均匀分布的径向渐变代码如下。

CSS 部分:

```

#grad2 {
    height:150px;
    width:200px;
    background: - webkit - radial - gradient(green 5%, purple 15%, red 60%);
                                                    /* Safari 5.1 - 6.0 */
    background: - o - radial - gradient(green 5%, purple 15%, red 60%);
                                                    /* Opera 11.6 - 12.0 */
    background: - moz - radial - gradient(green 5%, purple 15%, red 60%);
                                                    /* Firefox 3.6 - 15 */
    background: radial - gradient(green 5%, purple 15%, red 60%);
                                                    /* 标准的语法(必须放在最后) */
}

```

HTML 部分:

```

<h3>径向渐变 - 颜色节点不均匀分布</h3>
<div id = "grad2"></div>

```

(3) 径向渐变——形状变化。

径向渐变中还可以设置形状。shape 参数定义了形状。值为 circle 或 ellipse。其中, circle 表示圆形, ellipse 表示椭圆形。默认值是 ellipse。

椭圆形 Ellipse(默认)实例如下。

CSS 部分:

```

#grad3 {
    height:150px;
    width:200px;
    background: - webkit - radial - gradient(red, yellow, green); /* Safari 5.1 - 6.0 */
    background: - o - radial - gradient(red, yellow, green); /* Opera 11.6 - 12.0 */
    background: - moz - radial - gradient(red, yellow, green); /* Firefox 3.6 - 15 */
    background: radial - gradient(red, yellow, green); /* 标准的语法(必须放在最后) */
}

```

HTML 部分:

```

<p><strong>椭圆形 Ellipse(默认): </strong></p>
<div id = "grad3"></div>

```

圆形 Circle 实例如下。

CSS 部分：

```
#grad4 {
    height: 150px;
    width: 200px;
    background: -webkit-radial-gradient(circle, green, purple, red); /* Safari 5.1 - 6.0 */
    background: -o-radial-gradient(circle, green, purple, red); /* Opera 11.6 - 12.0 */
    background: -moz-radial-gradient(circle, green, purple, red); /* Firefox 3.6 - 15 */
    background: radial-gradient(circle, green, purple, red); /* 标准的语法(必须放在最后) */
}
```

HTML 部分：

```
<p><strong>圆形 Circle: </strong></p>
<div id="grad4"></div>
```

(4) 径向渐变——大小变化。

径向渐变中还可以设置 size,即定义渐变的大小,如表 3-10 所示。

表 3-10 径向渐变 size 值及其描述

size 值	描 述
farthest-corner (默认)	指定径向渐变的半径长度为从圆心到离圆心最远的角
closest-side	指定径向渐变的半径长度为从圆心到离圆心最近的边
closest-corner	指定径向渐变的半径长度为从圆心到离圆心最近的角
farthest-side	指定径向渐变的半径长度为从圆心到离圆心最远的边

利用不同尺寸大小关键字的实例如下。

CSS 部分：

```
#grad5 {
    height: 150px;
    width: 150px;
    background: -webkit-radial-gradient(60% 55%, closest-side, green, purple, red, black); /* Safari 5.1 - 6.0 */
    background: -o-radial-gradient(60% 55%, closest-side, green, purple, red, black); /* Opera 11.6 - 12.0 */
    background: -moz-radial-gradient(60% 55%, closest-side, green, purple, red, black); /* Firefox 3.6 - 15 */
    background: radial-gradient(60% 55%, closest-side, green, purple, red, black); /* 标准的语法(必须放在最后) */
}
#grad6 {
    height: 150px;
    width: 150px;
    background: -webkit-radial-gradient(60% 55%, farthest-side, green, purple, red, black); /* Safari 5.1 - 6.0 */
    background: -o-radial-gradient(60% 55%, farthest-side, green, purple, red, black); /* Opera 11.6 - 12.0 */
    background: -moz-radial-gradient(60% 55%, farthest-side, green, purple, red, black);
```

```

/* Firefox 3.6 - 15 */
background: radial-gradient(60% 55%, farthest-side, green, purple, red, black);
/* 标准的语法(必须放在最后) */
}
#grad7 {
    height: 150px;
    width: 150px;
    background: -webkit-radial-gradient(60% 55%, closest-corner, green, purple, red,
black);
/* Safari 5.1 - 6.0 */
    background: -o-radial-gradient(60% 55%, closest-corner, green, purple, red, black);
/* Opera 11.6 - 12.0 */
    background: -moz-radial-gradient(60% 55%, closest-corner, green, purple, red,
black);
/* Firefox 3.6 - 15 */
    background: radial-gradient(60% 55%, closest-corner, green, purple, red, black);
/* 标准的语法(必须放在最后) */
}
#grad8 {
    height: 150px;
    width: 150px;
    background: -webkit-radial-gradient(60% 55%, farthest-corner, green, purple, red,
black);
/* Safari 5.1 - 6.0 */
    background: -o-radial-gradient(60% 55%, farthest-corner, green, purple, red, black);
/* Opera 11.6 - 12.0 */
    background: -moz-radial-gradient(60% 55%, farthest-corner, green, purple, red,
black);
/* Firefox 3.6 - 15 */
    background: radial-gradient(60% 55%, farthest-corner, green, purple, red, black);
/* 标准的语法(必须放在最后) */
}

```

HTML 部分:

```

<p><strong>closest-side:</strong></p>
<div id="grad5"></div>
<p><strong>farthest-side:</strong></p>
<div id="grad6"></div>
<p><strong>closest-corner:</strong></p>
<div id="grad7"></div>
<p><strong>farthest-corner(默认):</strong></p>
<div id="grad8"></div>

```

课堂小实践

如果想重复一个渐变,可以使用 `repeating-linear-gradient` 和 `repeating-radial-gradient`。尝试做出如图 3-14 和图 3-15 所示效果图。

2. CSS 阴影

`box-shadow` 给元素添加一个或多个阴影。该属性是由逗号分隔的阴影列表,每个阴影由 2~4 个长度值、可选的颜色值以及可选的 `inset` 关键词来规定。省略长度的值是 0。语法规则如下。

```
box-shadow: h-shadow v-shadow blur spread color inset;
```




图 3-14 repeating-linear-gradient 效果



图 3-15 repeating-radial-gradient 效果

CSS3 阴影属性值及其描述如表 3-11 所示。

表 3-11 CSS3 阴影属性值及其描述

属 性 值	描 述
h-shadow	必需。水平阴影的位置。允许负值
v-shadow	必需。垂直阴影的位置。允许负值
blur	可选。模糊距离
spread	可选。阴影的尺寸
color	可选。阴影的颜色
inset	可选。将外部阴影(outset)改为内部阴影

box-shadow 案例如下。

demo3-9/box-shadow.html:

```
<!DOCTYPE html >
<html >
<head >
<title> box - shadow </title>
<style type = "text/CSS">
div {
width: 300px;
height: 200px;
```



```
background-color: #ff0099;
-moz-box-shadow: 20px 10px 5px #999999 inset; /* 老版 Firefox */
box-shadow: 20px 10px 5px #999999 inset; /* 添加阴影效果,表示左边框阴影为 20px(水平阴影位置)、右边框阴影为 10px(垂直阴影位置)阴影范围大小为 5px,阴影颜色为深灰色,为内阴影 */
}
</style>
</head>
<body>
<div></div>
</body>
</html>
```

3. CSS 动画

CSS 过渡也有缺陷,在其知识点讲解中已经做了描述,而 CSS 动画(Animation)正是为了解决这些缺陷而提出的。Animation 在不需要触发任何事件的情况下,也可以显式地随时间变化来改变元素 CSS 属性,达到一种动画的效果;并且动画可以设置多个帧,增加动画的复杂程度。

一个完整的 CSS 动画由两个部分组成:一组定义的动画关键帧和描述该动画的 CSS 声明。

1) @keyframes 规则的语法格式

使用@keyframes 规则来创建动画,keyframes 可以设置多个关键帧,每个关键帧表示动画过程中的一个状态,多个关键帧就可以使动画十分多彩。

@keyframes 规则的语法格式如下。

```
@keyframes animationname {
  keyframes-selector{CSS-styles;}
}
```

CSS3 动画@keyframes 属性值及其描述如表 3-12 所示。

表 3-12 CSS3 动画@keyframes 属性值及其描述

属 性 值	描 述
animationname	表示当前动画的名称,它将作为引用时的唯一标识,因此不能为空
keyframes-selector	是关键帧选择器,即指定当前关键帧要应用到整个动画过程中的位置值可以是一个百分比、from 或者 to。其中,from 和 0 效果相同,表示动画的开始;to 和 100%效果相同,表示动画的结束
CSS-styles	定义执行到当前关键帧时对应的动画状态

注意,以上三个属性都是必需,缺一不可。

2) animation 属性

animation 属性将动画与元素绑定,用于描述动画的 CSS 声明,包括指定具体动画以及动画时长等行为。

animation 属性的基本语法如下。

```
animation: name duration timing-function delay iteration-count direction fill-mode play-state;
```

以上参数分别对应 8 个子属性,如表 3-13 所示。

表 3-13 CSS3 动画 animation 子属性及其描述

属性值	描 述
animation-name	指定要绑定到选择器的关键帧的名称
animation-duration	动画指定需要多少秒或毫秒完成
animation-timing-function	设置动画将如何完成一个周期,使用名为三次贝塞尔(Cubic Bezier)函数的数学函数,来生成速度曲线。在该函数中使用自己的值,也可以使用预定义的值。 linear: 动画从头到尾的速度是相同的。 ease: 默认,动画以低速开始,然后加快,在结束前变慢。 ease-in: 动画以低速开始。 ease-out: 动画以低速结束。 ease-in-out: 动画以低速开始和结束。 cubic-bezier(n,n,n,n): 在 cubic-bezier 函数中自己的值。可能的值是从 0 到 1 的数值
animation-delay	设置动画在启动前的延迟间隔
animation-iteration-count	定义动画的播放次数,若值为 infinite,则规定动画应该无限次播放
animation-direction	指定是否应该轮流反向播放动画,取值如下。 normal: 默认值,动画应该正常播放。 alternate: 动画应该轮流反向播放
animation-fill-mode	规定当动画不播放时(当动画完成时,或当动画有一个延迟未开始播放时),要应用到元素的样式。动画按执行时间来划分,分为三个过程,或者说一次动画过程可以将元素划分为三个状态:动画等待、动画进行和动画结束。默认情况下,只有在动画进行状态,才会应用@keyframes 所声明的动画;而在动画等待和动画结束状态,对元素样式并不会产生任何的影响。取值如下。 none: 默认值。动画不会对动画等待和动画完成的元素样式产生改变。 forwards: 告诉浏览器动画结束后,元素的样式将设置为动画的最后一帧的样式。 backwards: 在动画等待的那段时间内,元素的样式将设置为动画第一帧的样式。 both: 相当于同时配置了 backwards 和 forwards,意味着在动画等待和动画结束状态,元素将分别应用动画第一帧和最后一帧的样式
animation-play-state	指定动画是否正在运行或已暂停,取值如下。 paused: 规定动画已暂停 running: 规定动画正在播放
initial	设置属性为其默认值
inherit	从父元素继承属性

下面通过一个案例演示 CSS 动画使用方法。

```
<!DOCTYPE html >
<html >
<head >
<meta charset = "utf - 8">
<title> CSS animation </title>
```

```
<style>
#myanimation {
    background: red;
    animation: mymove 5s infinite;
    /* Safari 和 Chrome */
    -webkit-animation: mymove 5s infinite;
}
@keyframes mymove {
0 % {
    width:300px;
    height: 300px;
}
50 % {
    width:100px;
    height: 100px;
}
100 % {
    width:300px;
    height: 300px;
}
}
/* Safari 和 Chrome */
@-webkit-keyframes mymove {
0 % {
    width:300px;
    height: 300px;
}
50 % {
    width:100px;
    height: 100px;
}
100 % {
    width:300px;
    height: 300px;
}
}
</style>
</head>
<body>
<p>div 区块逐渐由小变大再逐渐由大变小</p>
<div id="myanimation"></div>
</body>
```

在这个实例中,用@keyframes 规则定义了一个名称为 mymove 的动画,在该动画中定义了三个关键帧:第一个关键帧,即动画开始时,区块宽度和高度都为 300px;第二个关键帧,即 50%处,区块高度和宽度都为 100px;第三个关键帧,即动画结束时,区块大小恢复至 300px。使用 animation 属性调用了这个动画效果并应用到了 div 区块上,形成了区块由大到小再由小到大的动画效果。如果只设立两帧,要形成实例中的效果,该如何设置呢?

课堂小实践

题目 1: 在网页中对一个 div 元素制作出从绿色变为蓝色,再从蓝色变为绿色的效果。

题目 2: 下载齿轮图片,运用 CSS 动画制作成齿轮转动的效果。

4. CSS 变形

CSS transform 允许对元素在二维或三维空间进行旋转、缩放、移动或倾斜。3D 变形与 2D 变形的最大不同就在于其参考的坐标轴不同,2D 变形的坐标轴是平面的,只存在 X 轴和 Y 轴;而 3D 变形的坐标轴则是 X、Y、Z 三条轴组成的立体空间,X 轴正向是朝右,Y 轴正向是朝下,Z 轴正向是朝屏幕外,如图 3-16 所示。

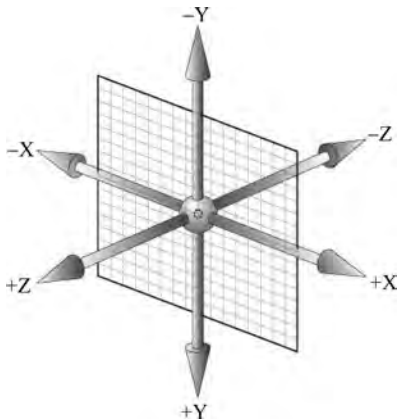


图 3-16 三维坐标系示意

本教材中主要讲解二维空间下的几种变形方式。

transform 的语法规则如下。

```
transform:none | <transform-function> [ <transform-function> ] *
```

也就是:

```
transform: rotate | scale | skew | translate | matrix;
```

其中,none 表示不进行变换;<transform-function>表示一个或多个变换函数,即可以同时 对一个元素进行 transform 的多种属性操作,但需注意的,以往叠加效果是用逗号(“,”) 隔开,但 transform 中使用多个属性时却需要用空格隔开。

1) 变形原点

transform-origin 是变形原点,也就是元素围绕着变形或旋转的那个点。此属性只有在 设置了 transform 属性的时候才起作用;元素默认基点就是其中心位置,但若需要在不同的 位置对元素进行这些操作,就需使用 transform-origin 来对元素进行基点位置改变,使元素 基点不再是中心位置,以满足多种需求。

语法规则如下。

```
transform-origin: x-axis y-axis z-axis;
```

transform-origin 最多接受三个值,分别是 X 轴、Y 轴和 Z 轴的偏移量,如表 3-14 所示。



表 3-14 transform-origin 属性值及其描述

属 性 值	描 述
x-axis	X 轴的偏移量。可能的值： left center right length %
y-axis	Y 轴的偏移量。可能的值： top center bottom length %
z-axis	Z 轴的偏移量。可能的值： length

其中, left、center、right 是水平方向的取值, 对应的百分值为 left=0%; center=50%; right=100%, 而 top、center、bottom 是垂直方向的取值, 对应的百分值为 top=0%; center=50%; bottom=100%, 如果只取一个值, 表示垂直方向值不变。

需要注意的是, transform-origin 并不是 transform 中的属性值, 它有自己的语法, 但是要结合 transform 才能起作用。

2) rotate

rotate()方法通过角度参数值来设定旋转的幅度。语法规则如下。

```
transform: rotate(<angle>);
```

若角度值为正数, 元素将顺时针旋转; 若为负数, 元素将逆时针旋转。
还可以使用 rotateX()与 rotateY()方法将元素只在水平方向或垂直方向旋转。

3) scale

scale()方法用于将元素缩放。在二维空间中, 可以分别指定元素水平方向的缩放倍率与垂直方向的缩放倍率, 语法规则如下。

```
transform: scale(X, Y);
```

其中, X 与 Y 的值设置为 0.01~0.99 的任何值时, 使一个元素缩小; 而设置为任何大于或等于 1 的值时, 使一个元素放大; Y 是一个可选参数, 如果没有设置 Y 值, 则表示 X 和 Y 两个方向的缩放倍数是一样的。

还可以使用 scaleX()与 scaleY()方法将元素只在水平方向或垂直方向缩放。

4) skew

skew()方法用于将元素倾斜显示, 可以将一个对象以其中心位置围绕着 X 轴和 Y 轴按照一定的角度倾斜。这与 rotate()方法的旋转不同, rotate()方法只是旋转, 而不会改变元素的形状。语法规则如下。

```
transform: skew(x-angel, y-angel);
```

其中,x-angel 用来指定元素水平方向(X 轴方向)倾斜的角度;y-angel 用来指定元素垂直方向(Y 轴方向)倾斜的角度,如果未显式地设置这个值,其默认值为 0。

还可以使用 skewX()和 skewY()方法将元素只在水平方向或垂直方向倾斜。

在默认情况下,skew()函数都是以元素的原中心点对元素进行倾斜变形,当然也可以根据 transform-origin 属性重新设置元素基点对元素进行倾斜变形。

5) translate

translate()方法用于将元素向指定的方向移动。语法规则如下。

```
transform: translate(tx,ty);
```

其中,translate()方法可以取一个值,即 ty 没有显式设置时,相当于 ty=0;若同时取两个值 tx 和 ty,tx 是一个代表 X 轴(横坐标)移动的向量长度,当其值为正值时,元素向 X 轴右方向移动,反之其值为负值时,元素向 X 轴左方向移动;ty 是一个代表 Y 轴(纵坐标)移动的向量长度,当其值为正值时,元素向 Y 轴下方向移动,反之其值为负值时,元素向 Y 轴上方向移动。

变形综合案例如下。

demo3-11/transform.html:

```
<!DOCTYPE html >
<html lang="en">
  <head>
    <meta charset="utf-8">
    <title>transform</title>
    <style>
      .box {
        background-color: #cccccc;
        width: 100px;
        height: 100px;
        transition: 2s ease;
        margin: 20px;
        display: inline-block;
      }
      /* 旋转,顺时针旋转一圈 */
      #box:hover {
        transform: rotate(360deg);
      }
      /* 倾斜转换,沿着 X 轴倾斜 45° */
      #box2:hover {
        transform: skew(45deg, 0deg);
      }
      /* 缩放,放大至 3 倍 */
      #box3:hover {
        transform: scale(3, 3);
      }
      /* 移动,向 X 轴右方向移动 100px,向 Y 轴下方向移动 100px */
      #box4:hover {
```

```

        transform: translateX(100px) translateY(100px);
    }
</style>
</head>
<body>
<div class = "box" id = "box"></div>
<div class = "box" id = "box2"></div>
<div class = "box" id = "box3"></div>
<div class = "box" id = "box4"></div>
</body>
</html>

```

6) 2D 转换矩阵 matrix(n,n,n,n,n,n)

matrix() 方法把所有 2D 转换方法组合在一起。matrix() 方法共需要六个参数, 包含数学函数, 允许旋转、缩放、移动以及倾斜元素。

demo3-10/matrix.html:

```

<!doctype html>
<html>
<head>
<meta charset = "utf - 8">
<title> 2D 转换矩阵 matrix</title>
<style>
div {
    width: 100px;
    height: 75px;
    background-color: blue;
    border: 1px solid gray;
}
div#div2 {
    transform: matrix(0.9, 0.56, -0.6, 0.7, 0, 0);
    -ms-transform: matrix(0.9, 0.56, -0.6, 0.7, 0, 0);           /* IE 9 */
    -moz-transform: matrix(0.9, 0.56, -0.6, 0.7, 0, 0);         /* Firefox */
    -webkit-transform: matrix(0.9, 0.56, -0.6, 0.7, 0, 0);      /* Safari and Chrome */
    -o-transform: matrix(0.9, 0.56, -0.6, 0.7, 0, 0);           /* Opera */
}
</style>
</head>
<body>
<div>这是一个 div 元素。</div>
<div id = "div2">这是一个 div 元素。</div>
</body>
</html>

```

3.2.5 知识点检测



3.2.6 项目实施

通过项目分析和 CSS 渐变、变形、动画、阴影等知识点学习,读者定能逐步理解并完成“图说我们盐城”页面(Picture.html)的代码编写。参考代码如下。

Picture.html:

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<link rel="stylesheet" type="text/CSS" href="CSS/picture.CSS">
<title>图说盐城</title>
</head>
<body>
<header>
    
</header>
<section class="main">
    <article class="culture red">
        <div class="title">
            
            <span class="more">更多<</span>
        </div>
        <div class="pic1">
            
            
            
            
            
            
        </div>
    </article>
    <article class="culture white">
        <div class="title">
            
            <span class="more">更多<</span>
        </div>
        <div class="pic2">
            
            
            
            
            
            
        </div>
    </article>
    <article class="culture green">
        <div class="title">
            
            <span class="more">更多<</span>
        </div>
    </article>
```

```
</div>
<div class="item1">
  <div class="lipic">
    
  </div>
  <div class="desc">
    <div class="detail">大纵湖旅游景区</div>
  </div>
</div>
<div class="item1">
  <div class="lipic">
    
  </div>
  <div class="desc">
    <div class="detail">大丰麋鹿园</div>
  </div>
</div>
<div class="item1">
  <div class="lipic">
    
  </div>
  <div class="desc">
    <div class="detail">金沙湖旅游度假区</div>
  </div>
</div>
<div class="item1">
  <div class="lipic">
    
  </div>
  <div class="desc">
    <div class="detail">大丰荷兰花海</div>
  </div>
</div>
<div class="item1">
  <div class="lipic">
    
  </div>
  <div class="desc">
    <div class="detail">大洋湾生态运动公园</div>
  </div>
</div>
<div class="item1">
  <div class="lipic">
    
  </div>
  <div class="desc">
    <div class="detail">大洋湾湿地公园</div>
  </div>
</div>
<div class="item1">
  <div class="lipic">
```

```

        
    </div>
    <div class="desc">
        <div class="detail">大丰东方湿地公园</div>
    </div>
</div>
<div class="item1">
    <div class="lipic">
        
    </div>
    <div class="desc">
        <div class="detail">盐渎公园</div>
    </div>
</div>
</article>
<article class="culture eat">
    <div class="title">
        
        <span class="more"><a href="#">更多</a></span>
    </div>
    <div class="item2">
        
        <div class="hover">
            <h3>建湖藕粉圆子</h3>
            <p>是盐城非常著名的美食之一,经常被用作招待客人</p>
        </div>
    </div>
    <div class="item2">
        
        <div class="hover">
            <h3>蟹黄包</h3>
            <p>蟹黄和猪肉、面粉等做成的,其味鲜美</p>
        </div>
    </div>
    <div class="item2">
        
        <div class="hover">
            <h3>大纵湖醉蟹</h3>
            <p>蟹个大黄多,吃起来醉味浓郁</p>
        </div>
    </div>
    <div class="item2">
        
        <div class="hover">
            <h3>伍佑醉螺</h3>
            <p>选用海滩中的泥螺腌制而成</p>
        </div>
    </div>
    <div class="item2">
        
        <div class="hover">

```

```

        <h3>阜宁大糕</h3>
        <p>选用优质糯米、纯净白糖、精制油脂及高级蜜饯,按其比例加工而成</p>
    </div>
</div>
<div class="item2">
    
    <div class="hover">
        <h3>四鳃鲈鱼</h3>
        <p>肉质洁白似雪,肥嫩鲜美</p>
    </div>
</div>
<div class="item2">
    
    <div class="hover">
        <h3>东台鱼汤面</h3>
        <p>汤稠如乳,点滴成珠,面白细匀,鲜而不腻</p>
    </div>
</div>
<div class="item2">
    
    <div class="hover">
        <h3>建阳米饭饼</h3>
        <p>米饼外焦里嫩,绵软喷香,入口即化,食多不腻</p>
    </div>
</div>
</article>
</section>
<footer>
    <div class="services">关于我们 | 联系我们 | 网站地图</div>
</footer>
</body>
</html>

```

对应的部分样式规则:

```

body {
    margin: 0;
    padding: 0;
    background: url(../images/Picture_img_bg.jpg);
    height: 100%;
}
header {
    text-align: center;
}
section {
    width: 1000px;
    margin: 0 auto;
}
.culture {
    width: 980px;
    border: 1px solid #ccc;
}

```

```

padding: 10px 10px;
background: -webkit-linear-gradient(0deg, #ffffff, rgba(255, 255, 255, 0.2));
/* Safari 5.1 - 6.0 */
background: -o-linear-gradient(0deg, #ffffff, rgba(255, 255, 255, 0.2));
/* Opera 11.1 - 12.0 */
background: -moz-linear-gradient(0deg, #ffffff, rgba(255, 255, 255, 0.2));
/* Firefox 3.6 - 15 */
background: linear-gradient(0deg, #ffffff, rgba(255, 255, 255, 0.2));
/* 标准的语法(必须放在最后),设置线性渐变的背景 */

margin: 10px;
}
.culture .title {
border-bottom: 1px solid #f8d361;
height: 50px;
line-height: 55px;
overflow: hidden; /* 溢出时内容会被修剪,并且其余内容不可见 */
}
.more {
float: right;
margin: 5px;
font-weight: normal;
font-size: 14px;
}

.pic1 img, .pic2 img {
width: 300px;
margin: 10px;
}
.pic1 img{
box-shadow: 5px 5px 5px #999999;
}
.pic2 img{
border-radius: 8px;
}
.green, .eat{
overflow: auto;
}
.item1{
width: 215px;
height: 145px;
text-align: center;
margin-left: 10px;
margin-top: 10px;
padding: 10px;
background-color: #FFF;
float: left;
position: relative;
top: 5px;
overflow: hidden;
transition: all .5s; /* 过渡,all(默认值)指所有属性改变,整个转换过程在 0.5s 内完成。 */
border-radius: 5px; /* 盒阴影: 向下偏移 5px 模糊值 5px 颜色为 #41a8ff */

```

```

    }
    .item1 .lipic {
        margin: 0px;
    }
    .item1 .lipic img{
        width: 100%;
    }
    .item1 .desc {
        position: absolute; /* 绝对定位 */
        bottom: -80px;
        width: 100%; /* 宽度是父元素宽度的 100% */
        height: 80px;
        transition: all .5s;
        background-color: rgba(36,158,88,0.6);
    }
    /* 当鼠标悬停在该元素时,该元素绝对定位在父元素顶部 - 5px 的位置,并且盒阴影为模糊度 15px 的 #AAA 色 */
    .item1: hover {
        top: -5px;
        box-shadow: 0 0 15px #AAA;
    }
    /* 当鼠标悬停在类名为 item 的元素上时,该元素的类名为 .desc 的子元素绝对定位,其底部与父元素底部对齐 */
    .item1: hover .desc {
        bottom: 0;
    }
    .item1 .detail{
        font-weight: bold;
        font-size: 20px;
        margin-top: 30px;
    }
    .item2 {
        width: 225px;
        overflow: hidden;
        position: relative;
        margin: 10px;
        float: left;
    }
    .item2 .hover {
        width: 225px;
        background: rgba(228,169,170,0.6);
        position: absolute;
        top: 0px;
        left: 0;
        text-align: center;
        color: #fff;
        transform: rotate(55deg); /* 变形: 旋转 55° */
        -webkit-transform: rotate(55deg);
        transition: all 0.5s; /* 过渡: 所有属性都改变,时长 0.5s */
        -webkit-transition: all 0.5s;
    }

```

```

        overflow: hidden;
        height: 0;
        z-index: 4000;      /* 设置元素的堆叠顺序,属性值越大,该元素层离用户越近 */
    }
    .item2 .hover h3 {
        color: #fff;
        border-bottom: 2px solid rgba(76, 179, 77, 0.5);
        padding-bottom: 10px;
    }
    .item2: hover .hover {
        height: 200px;
        transform: rotate(0deg);
        -webkit-transform: rotate(0deg);
    }

    footer {
        clear: both;
        width: 880px;
        margin: 0 auto;
    }
    footer > p {
        font-family: 'STXingkai';
        font-size: 35px;
        color: #AAE6DA;
        line-height: 20px;
        padding: 20px;
        text-align: center;
    }
    .services {
        font-family: 'Microsoft Yahei';
        font-size: 15px;
        color: #374136;
        padding-bottom: 50px;
        text-align: center;
    }
}

```

3.2.7 项目总结

项目实施过程中,主要是练习 CSS 渐变、阴影、过渡、变形、复合选择器的使用;要特别掌握综合使用 CSS 渐变、阴影、过渡、变形、复合选择器创建精美炫酷的动态效果;通过搜集、整理、分析、归纳文化(如盐城)特色,进而了解家乡文化、热爱家乡文化,培养家国情怀、建设家乡的责任感和使命感;同时通过对表现家乡文化图片的搜集、选取及优化处理,提升审美能力和艺术修养。

3.2.8 能力拓展

搜集、整理、分析、归纳网站的“红色文化”及“特色美食”两个模块的资源,并进行优化处理,参考 index.html 和 Picture.html 两个页面模仿设计 redculture.html 和 delicious.html 两个页面。