第 2 章 CSS3 基础

学习目标

- 熟悉 CSS3 的基础知识。
- 理解盒子模型。
- 掌握 CSS 中选择器的使用方法。
- 掌握背景重复、背景定位的使用方法。

2.1 CSS3 基础知识

2.1.1 CSS 的含义及其语法

CSS（cascading style sheets，层叠样式表），用于设置 HTML 标签样式的一种标记语言。通过使用 CSS 可以使网站外观更加美观，结构更加简化。

CSS 的语法格式：

```
选择符{属性:属性值}
```

属性必须要包含在“{}”中，属性与属性值之间用“:”分隔。当有多个属性时，用“;”进行区分。示例代码：

```
/* div 选择符
width、height 样式属性 480px、320px 样式属性值
* /
div {
    width:480px;
    height:320px;
}
```

2.1.2 CSS 的引入方式

引入 CSS 主要有四种方式，即行内样式、内嵌样式、外链样式、导入式。

1. 行内样式

行内样式是指将 style 作为一个标签的属性，然后通过它的值来设置样式。示例代码：

```
<body>
    <div style="width:400px; color:red"></div>
<body>
```

2. 内嵌样式

内嵌样式是指将 style 作为一个标签，然后在标签内通过样式选择符来设置样式。示例代码：

```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>
<style>
    div {
        width:400px;
        color:red;
    }
</style>

<body>
    <div></div>
</body>

</html>
```

3. 外链样式

外链样式是指将样式单独放置在一个 CSS 文件中，然后在页面中通过<link>标签的 href 属性引入该样式文件。这样做的目的是抽离 CSS，便于维护和管理。示例代码：

```
    <!DOCTYPE html>
    <html lang="en">

    <head>
        <meta charset="UTF-8">
        <meta http-equiv="X-UA-Compatible" content="IE=edge">
        <meta name="viewport" content="width=device-width, initial-
scale=1.0">
        <!--外联样式 -->
        <title>Document</title>
        <link rel="stylesheet" href="./index.css">
    </head>
    <style></style>

    <body>
        <div></div>
    </body>

    </html>

        /* 外部 CSS 样式部分 */
        div {
            width:400px;
            color:red;
        }
```

4. 导入式

该方法是在<style>标签里或者 CSS 文件中通过@ import 来导入外部 CSS 文件，这点和通过<link>标签导入外部样式是一样的，但其他方面却与之有很大不同。平时开发很少用到这种方法，简单了解即可。示例代码：

```
    <!DOCTYPE html>
    <html lang="en">
    <head>
        <meta charset="UTF-8">
        <meta http-equiv="X-UA-Compatible" content="IE=edge">
```

```
        <meta name="viewport" content="width=device-width, initial-
scale=1.0">
        <title>Document</title>
    </head>
    <style>
        /* 导入式 */
        @ import url('./index.css')
    </style>
    <body>
        <div></div>
    </body>
    </html>
```

2.1.3 盒子模型

盒子模型是将网页布局中的元素（行内元素、行内块元素）进行拟物化的一种比喻。

CSS 盒子模型的类型主要有以下两种：

IE 浏览器盒子模型：改变盒子的宽度或者高度，盒子总体大小都不会改变。核心代码：

```
box-sizing:border-box;
```

标准（W3C 以及其他主流浏览器）盒子模型（默认）：改变盒子的宽度或者高度，盒子总体大小会随之改变。核心代码：

```
box-sizing:content-box;
```

1. 盒子的组成

一个盒子由外到内可以分成四个部分：margin（外边距）、border（边框）、padding（内边距）、content（内容），如图 2-1 所示。通过调整 CSS 中的 margin、border 和 padding 属性，可以控制盒子在布局中的位置和样式，而 content 部分由 HTML 元素的标签和其包含的内容组成。这样的盒子模型可帮助我们可以更好地控制网页的布局和外观。

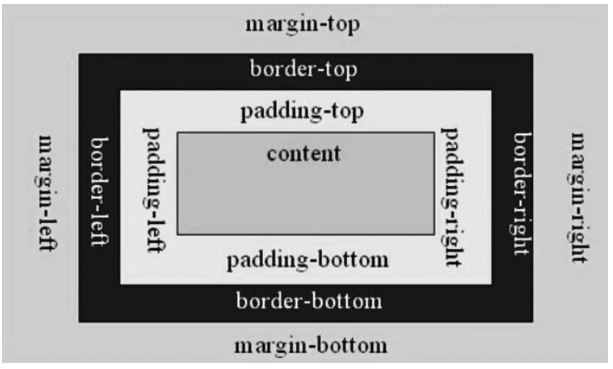

图 2-1 盒子模型

2. 盒子的大小

盒子的大小指的是盒子的宽度和高度。大多数初学者容易将宽度和高度误解为 width 和 height 属性，然而默认情况下 width 和 height 属性只是设置 content 部分的宽和高。盒子真正的宽和高按下式计算：

盒子的宽度=内容宽度+左填充+右填充+左边框+右边框+左边距+右边距

盒子的高度=内容高度+上填充+下填充+上边框+下边框+上边距+下边距

3. 盒子的圆角（border-radius 属性）

利用 border-radius 属性可以设置边框圆角和元素的边界。该属性虽然带了一个 border 字样，但它和“border”并无太大的关系，它是对元素的“左上”“右上”“右下”和“左下”四个角的“圆度”进行设置。

border-radius 实现的原理是，根据该属性所设置的值，以该圆的边缘形状来设置边角的圆度。可以参考图 2-2。

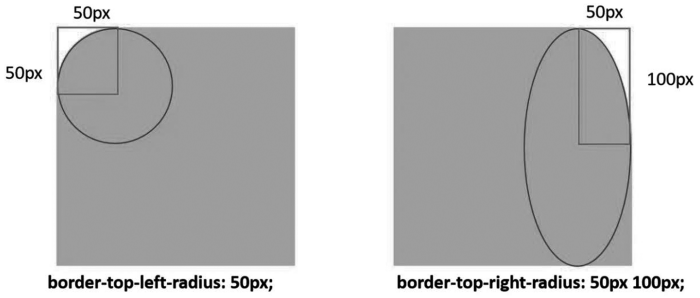

图 2-2　border-radius 实现的原理

border-radius 单位可以是像素（px），也可以是百分比，如果采用百分比如 50%，并且元素宽高相等，则会呈现一个正圆，否则会呈现出一个椭圆。使用百分比时，其上限值为 50%，超过 50%的值并不会出现任何效果。

设值时，可以设置两个值，用“/”进行分割，如 border-radius：100px/120px，第一个值表示水平方向圆的半径，第二个值表示垂直方向圆的半径。

一个（椭）圆与边框的交集形成圆角效果，如图 2-3 所示。

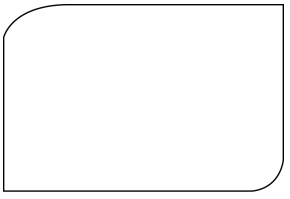

图 2-3　圆与边框的交集形成圆角效果

示例代码：（圆角边框）

```
<html lang="en">
<head>
```

```
    <meta charset="UTF-8">
    <title>Document</title>
    <style>
    div {
        height:100px;
        width:150px;
        border:1px solid blue;
        border-top-left-radius:40px 20px;
        border-bottom-right-radius:20px;
    }
    </style>

</head>
<body>
    <div></div>
</body>
</html>
```

4. 元素阴影（box-shadow 属性）

box-shadow 属性能够让元素获得一个或多个阴影效果，因为可设置不同的颜色，所以又称作发光效果。box-shadow 属性没有分支属性，它的值是以组合值的形式进行设置的，最多允许六个值的组合。其语法格式：

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

注意 box-shadow 属性把一个或多个下拉阴影添加到框上。该属性是一个用逗号分隔阴影的列表，每个阴影由 2～4 个长度值、一个可选的颜色值和一个可选的 inset 关键字来规定。省略长度的值是 0。

h-shadow：阴影在水平方向的偏移，负数是向左偏移，正数是向右偏移，单位为 px。

v-shadow：阴影在垂直方向的偏移，负数是向上偏移，正数是向下偏移，单位为 px。

blur：阴影的“模糊距离”或“模糊程度”，单位为 px。

spread：阴影的扩展范围，若将“blur”设为“0”，该值则相当于一个可以进行位置偏移但不具备“outline-offset”的“outline”，单位为 px。

color：阴影的颜色，支持 Web 技术中的常用颜色模式：“颜色英文单词”“HEX”“RGBa”“HSLa”。

inset：将默认向外的阴影方向改为向内。

示例代码：（盒子阴影）

```
<html lang="en">
<head>
```

```
    <meta charset="UTF-8">
    <title>Document</title>
    <style>
        div
        {
            width:300px;
            height:100px;
            background-color:#f90;
            box-shadow:10px 10px 5px #888;
        }
    </style>

</head>
<body>
    <div></div>
</body>
</html>
```

盒子阴影完成的效果如图 2-4 所示。

图 2-4　盒子阴影完成效果

2.1.4　CSS 选择器

在 CSS 中，选择器（Selector）是一种模式，用于选择需要添加样式的元素。

1. 标签选择器

一个完整的 HTML 页面是由很多不同的标签组成的，而标签选择器，则是决定哪些标签采用相应的 CSS 样式。示例代码：

```
div {width:300px; height:300px; background-color:red;}
p {text-indent:2em; color:blue;}
span {letter-spacing:5px; font-size:20px;}
```

2. 类名选择器

类名选择器（又称 class 选择器）根据类名来选择元素。虽然这个类名是自定义的，但我们在定义此类名时也应该尽量反映被设置元素的实际功能。同一个类名的选择器，理

论上可以被任意多的标签元素使用。在 CSS 中，定义类名选择器应该以“.”作为开头，否则浏览器将视为自定义的标签名。示例代码：

```
.box {width:300px; height:300px; background-color:red;}
p.des {text-indent:2em; color:blue;}
```

3. ID 选择器

ID 选择器，即根据元素的 ID 属性值来选取元素，这和类名选择器类似。但值得注意的是，ID 表示唯一标识符，即同一个页面只能出现一个 ID。定义一个 ID 选择器以“#”开头。示例代码：

```
#box {width:300px; height:300px; background-color:red;}
#des {text-indent:2em; color:blue;}
```

4. 通用选择器

通用选择器使用“*”作为开头，它的作用是选择页面中所有的标签元素。一般用于：

```
* { magrin:0; padding:0;}
```

5. 后代选择器

后代选择器用于对某元素所嵌套的指定元素进行选择，每个选择符之间用空格进行分割，多个嵌套层次应该以多个空格进行分割。示例代码：

HTML 代码：

```
<div class="container">
    <article>
        <h1>Napoléon Bonaparte</h1>
        <p>Adversity is the midwife of genius.</p>
    </article>
</div>
```

CSS 代码：

```
.container article { text-align:center;}
.container h1 { color:#000000;}
.container p { color:#008800;}
```

6. 子类选择器

子选择器与后代选择器的区别：后代选择器可以选择嵌套在标签内部任意层级的标签元素，而子选择器只能选择当前标签往内一层的元素，即直接子元素。每个选择符之间用“>”进行分割。示例代码：

HTML 代码：

```
<header>
    <img src="images/logo.png" alt="logo">
    <nav>
        <ul class="menu-list">
            <li><a href="javascript:;">首页</a></li>
            <li><a href="javascript:;">新闻</a></li>
            <li><a href="javascript:;">科技</a></li>
            <li><a href="javascript:;">社会</a></li>
        </ul>
    </nav>
</header>
```

CSS 代码：

```
/* index.css */
header > img {width:80px; height:30px;}
header > nav > ul.menu-list { list-style:none;}
```

7. 伪类选择器

伪类选择器和其他选择器有所不同，它是通过触发一定的事件来实现效果，也就是说，如果不进行任何操作是看不到该选择器所设置的 CSS 样式的。以 Google Chrome 浏览器开发者工具为例，要想看到所设置的伪类选择器样式需通过单击 Element 选项栏下 Style 选项栏中的 hov 按钮，然后勾选需要查看的操作事件查看样式。目前支持的操作事件有 hover、active、visited 和 focus。

(1):hover：鼠标悬浮于该元素上设置的样式。

(2):active：鼠标悬点击时该元素上设置的样式。

(3):visited：鼠标悬点击后该元素上设置的样式（了解）。

(4):focus：表单元素获得焦点后设置的样式。

示例代码：

```
a { text-decoration:none;}
a:hover { text-decoration:underline;}
```

8. 群组选择器

群组选择器也是一种 CSS 选择器，它允许将多个选择器组合在一起，从而为这些选择器应用相同的样式或一组样式。这样可以让 CSS 代码更加简洁、高效，并提高代码的可读性和维护性。示例代码：

HTML 代码：

```
<a href="javascript:;">超链接</a>
<div>布局标签</div>
```

```
<span>文本标签</span>
<p>段落标签</p>
```

CSS 代码：

```
/* index.css */
a, div, span, p { font-size:20px;}
div, p { margin:0; padding:0;}
```

9. 同级选择器

该选择器能选定与当前选择器同级的其他指定选择器，平时虽使用得不多，但配合伪类选择器可以做出一些很有新意的效果，也能减少对 JavaScript 的依赖。同级元素有两种：

（1）+：选择该选择器相邻的下一个选择器。

（2）~：选择该选择器后的所有同级选择器。

2.2 Web 前端布局的基本美化

文本是 HTML 页面中最基本的表现形式，通过文本能的有效而详细地说明网页中的内容。但若不对页面中的文本做任何处理，用户的浏览体验会不好。通过 CSS 对文本进行设置后，不仅能让用户的浏览体验更佳，也会让页面的美观度得到提升。

2.2.1 文本设置

1. 文本对齐方式（text-align 属性）

text-align 属性用于控制行内块元素、块元素或行内元素（文本元素）的居中方式。

text-align 属性有以下值。

（1）left：局左对齐（默认）。

（2）right：居右对齐。

（3）center：居中对齐。

（4）justify：两端对齐。

2. 段落首行缩进（text-indent 属性）

text-indent 属性用于设置每个段落首行缩进数量值。

CSS 字体大小（font-size）可以设置的数值和单位在该属性的值中都可以使用（除了百分比）。如果是用于中文布局，一般会使用“2 em”（2 字符）来为每个段落的首行缩进两个字符。

3. 文本装饰线（text-decoration 属性）

text-decoration 属性用于为文本添加一条装饰线。带“href”属性的<a>标签默认会带有一条下画线，就是由该属性的值“underline”设置的。

text-decoration 属性有以下值。

（1）none（默认）：不显示任何装饰线。

（2）underline：在文本下方显示装饰线。

（3）overline：在文本上方显示装饰线。

（4）line-through：在文本中间显示装饰线，相当于删除线。

4. 大小写转换（text-transform 属性）

text-transform 属性能对行内元素中的英文文本进行大小写转换，以满足网站对规范性的要求。

text-transform 属性有以下值。

（1）none（默认）：保持文本中英文单词的默认大小写。

（2）capitalize：每个英文单词首字母为大写字母，其他为小写字母。

（3）uppercase：将所有英文单词转换为大写字母。

（4）lowercase：将所有英文单词转换为小写字母。

5. 文本阴影（text-shadow 属性）

text-shadow 属性为文本添加阴影效果。目前，除了 IE9 及之前版本不支持该属性外，其他主流浏览器均支持该属性。text-shadow 属性有以下值。

（1）H：水平偏移，“0”表示维持原位，正数为向右偏移，负数为向左偏移，单位为 px。

（2）V：垂直偏移，“0”表示维持原位，正数为向下偏移，负数为向上偏移，单位为 px。

（3）blur：模糊距离，用正数表示阴影模糊的单位距离，距离越大模糊程度越高，单位为 px。

（4）color：阴影颜色，支持 Web 技术中的常用颜色模式，如“颜色英文单词”（red、blue、green 等）、“HEX”、“RGBa”、“HSLa”。

示例代码：（文字阴影）

```
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Document</title>
    <style>
        h1 {
            text-shadow:2px 2px #FF0000;
        }
    </style>

</head>
<body>
    <h1>Web design</h1>
</body>
</html>
```

6. 文本行高（line-height 属性）

line-height 属性用于设置行内元素中文本元素在一行中所占据的高度，可以使用的值和字体大小（font-size）属性的一样，而且同样能够使用百分比，或不带单位的浮点数（“1”表示“100%”，“1.5”表示“150%”，以此类推）。该属性除字面上的意思外，很多时候还有一个小妙用，就是当文本元素只有一行时，可以将该行的文本行高设为和父容器元素的高度一致，以此达到文本垂直居中的效果。使用场景有表格、导航按钮、自定义样式按钮、标题栏等。

7. 单词间距（word-spacing 属性）

word-spacing 属性用于设置英文单词之间的间距和设置中文文本中空格的距离，单位为 Web 技术的常用度量单位，如像素（px）、字符（em）、点（pt）等，可以为负数。

8. 字符间距（letter-spacing 属性）

letter-spacing 属性用于控制字符间的间距。无论单词或词语中是否含有空格（该属性对空格字符无效），该属性都会生效，单位同样为 Web 技术的常用度量单位，同样为可以为负数。

9. 文本裁切（text-overflow 属性）

text-overflow 属性用于在文本溢出包含元素时指定如何处理文本内容。

text-overflow 属性主要有两个值：

（1）clip：裁切文本。

（2）ellipsis：显示省略符号来代表被裁剪的文本。

10. 文本自动换行（word-wrap 属性）

word-wrap 属性允许长的内容可以自动换行。

语法格式：

```
word-wrap:normal/break-word;
```

word-wrap 属性有两个值：

（1）normal：只在允许的断字点换行（浏览器保持默认处理）。

（2）break-word：在长单词或 URL 地址内部进行换行。

示例代码：（word-wrap 应用）

```
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Document</title>
    <style>
    p.wdrp
    {
        width:8em;
        border:1px solid #333;
        word-wrap:break-word;
```

```
    }
    </style>
</head>
<body>
    <p class="wdrp">这是最长的英文单词,
pneumonoultramicroscopicsilicovolcanoconiosis.</p>
</html>
```

文本自动换行的完成效果如图 2-5 所示。

图 2-5　完成效果

2.2.2　字体设置

1. font-face 属性和@font-face 规则

（1）font-face 属性

font-face 是 CSS3 中允许使用自定义字体的一个模块，主要是把定义的 Web 字体嵌入网页中。font-face 属性语法结构如图 2-6 所示。如需将字体用于 HTML 元素，需通过 font-family 属性引用字体的名称，其语法格式：

```
<font face="font-family">
```

图 2-6　font-face 属性语法结构说明

其属性值 font-family 用于规定文本的字体。如需规定若干字体的优先表，请使用逗号分隔字体名称，如

```
<font face="verdana, rial, sansserif">
```

（2）@font-face 规则

利用@font-face 规则，定义 Web 字体，并引用需要字体的文件格式（见表 2-1 所列），确保能在主流浏览器中正常显示该字体。

表 2-1 浏览器支持的字体

字体文件后缀	适用的浏览器
.ttf 或 .otf	Firefox、Safari、Opera、chrome
.eot	Internet Explorer 4.0+
.svg	Chrome、IPhone
.woff	Chrome、Firefox

注：Interret Evoloret 8 及更早版本不支持@font-face 规则。

2. 字体颜色（font-color 属性）

在目前的浏览器标准中，要想改变浏览器默认字体的颜色（#000000），唯一的途径是通过 CSS 的 color 属性进行设置，有以下颜色模式。

（1）颜色英文单词*。

（2）HEX（16 进制颜色）*。

（3）RGBa/RGB（Alpha 的三原色）。

（4）HSLa（Alpha 的 Hue、Saturation、Lightness）。

（5）Transparent（透明）*。

（6）inherit（继承父级）。

3. 字体样式（font-style 属性）

font-style 属性用于设置字体类型，有以下值。

（1）普通字体。

（2）italic：斜体（<i>）*。

（3）oblique：倾斜字体。

4. 字体粗细（font-weight 属性）

font-weight 属性用于设置字体粗细，有以下值。

（1）normal：正常粗细。

（2）bold：粗体（<b>）。

（3）bolder：更粗的字体。

（4）lighter：更细的字体。

（5）100～900：步长为 100 等同于 lighter，400 等同于 normal，而 700 等同于 bold。

5. 字体大小（font-size 属性）

font-size 属性用于设置字体大小，可以是任意正整数和浮点数，但考虑到 Chrome 等

主流浏览器不支持 12 像素以下的字体，若字号以像素为单位，或通过设置其他单位转换为的像素值，不应该小于 12 像素。常用的字号单位有：

（1）px：像素。

（2）em：和父元素成比例关系，比如父元素 16 像素，设置子元素为 2 em，就表示子元素字体大小为 32 px。

（3）rem：类似于 em，但是 rem 是相对于根元素而言的。

2.2.3 显示与定位

在 HTML 中，元素会按照标准的文档流布局方式进行布局，即“从左到右，从上到下”的方式进行布局，而通过 CSS 里面的部分定位和显示方式的设置可以使元素脱离文档流，采用特殊的布局方式进行布局，或者在页面中进行“隐藏”。而“隐藏”在 CSS 中又有两种定义方式：一种可以脱离文档流，另一种仍然存在于文档流的布局中。

1. 元素显示方式（display 属性）

display 属性用于规定元素的显示方式。元素类型可分为块级元素、行内元素以及行内块元素，各类型的元素在呈现方式上也不尽相同。HTML 文档中自带的标签元素的“显示类型”已经被定义，如果考虑到布局需要，有的时候我们会强制将“显示类型”进行转换，在进行转换后，该元素的功能和特性（如将 标签转换为“inline-block”之后，也不能嵌套在<p>标签里和转换为“block”的标签里）也不会产生变化。

display 属性有表 2-2 所列的值。

表 2-2 display 的属性值

属性值	描述
none	将元素设定为不显示，使元素完全地脱离“文档流”（主要）
inline	将元素设定为行内元素（主要）
inline-block	将元素设定为行内块元素（主要）
block	将元素设定为块元素（主要）
list-item	将元素设置为列表项
run-in	根据上下文作为行内块元素或块元素显示
table	将元素作为 table 显示
inline-table	将元素作为行内的 table 显示
table-row-group	将元素作为一个 tbody 元素显示
table-header-group	将元素作为一个 thead 元素显示
table-footer-group	将元素作为一个 tfoot 元素显示

续表

属性值	描述
table-row	将元素作为一个 tr 元素显示
table-column-group	将元素作为一个 colgroup 元素显示
table-column	将元素作为一个 col 元素显示
table-cell	将元素作为一个 table 单元格元素显示
table-caption	将元素作为一个 table 标题显示
flex	弹性布局（主要）
grid	网格布局（主要）

2. 元素的可见性（visibility 属性）

visibility 属性用于设置元素是否可见。即使该属性的值是元素不可见，元素也不会脱离文档流，仍会占据空间。该属性可以设置以下值。

（1）visible：默认，元素可见。

（2）hidden：元素不可见，仍会占据空间。

3. 元素不透明度（opacity 属性）

opacity 属性用于设置元素的不透明度。该属性的值的范围为 0～1，可以保留两位小数，设值时可以省略 0，如 .5 或 .75。0（相当于 visibility:hidden;）表示完全透明，1（默认）表示完全不透明。

4. 鼠标指针类型（cursor 属性）

cursor 属性用于规定要显示的鼠标指针的类型（形状）。该属性具备表 2-3 所列的值。

表 2-3 cursor 属性的属性值

属性值	描述
url	需使用的自定义光标的 URL。需注意的是，请在此列表的末端始终定义一种普通的光标，以防没有由 URL 定义的可用光标，如 cursor:url("../imgs/custorm.jpeg"), auto; *
default	默认光标（通常是一个箭头）
auto	默认，浏览器设置的光标
crosshair	光标呈现为十字线
pointer	光标呈现为指示链接的指针（一只手） *
move	此光标指示某对象可被移动 *
e-resize	此光标指示矩形框的边缘可被向右（东）移动
ne-resize	此光标指示矩形框的边缘可被向上及向右（北、东）移动

续表

属性值	描述
nw-resize	此光标指示矩形框的边缘可被向上及向左（北、西）移动
n-resize	此光标指示矩形框的边缘可被向上（北）移动
se-resize	此光标指示矩形框的边缘可被向下及向右（南、东）移动
sw-resize	此光标指示矩形框的边缘可被向下及向左（南、西）移动
s-resize	此光标指示矩形框的边缘可被向下（南）移动
w-resize	此光标指示矩形框的边缘可被向左（西）移动
text	此光标指示文本 *
wait	此光标指示程序正忙（通常是一只表或沙漏） *
help	此光标指示可用的帮助（通常是一个问号或一个气球） *

5. 浮动（float 属性）

float 属性可以使元素脱离文档流，在父容器中进行浮动，停靠到父元素的内容边界或其他浮动元素的边框。浮动的元素会忽略元素间的空格，让同样具有该属性的元素“紧密”地排列在一起。该属性有以下三个允许的值。

（1）none：默认，元素不进行浮动。

（2）left：元素从左到右进行浮动。

（3）right：元素从右到左进行浮动。

6. 定位（position 属性）

在 HTML 中，元素的布局可以通过 CSS 的浮动属性 float 和外间距属性 margin，甚至是内间距属性 padding 来实现，但是这些布局方式都存在着各种各样的局限性。position 属性是对元素布局的一个非常好的补充，而且该属性对所有显示类型的元素都适用。该属性的值主要有：static（默认值，采用元素默认的定位方式）、relative（相对定位）、absolute（绝对定位）、fixed（固定定位）。

（1）相对定位

当一个元素的 position 属性的值为 relative 时，元素对其原始位置进行相对定位，但元素在文档流中所占据的位置仍然保留。

注意：

①开启元素的相对定位后，如果不设置偏移量，元素不会发生任何变化。

②相对定位元素相对于其自身在文档流中的位置来定位。

③相对定位的元素不会脱离文档流。

④相对定位不会改变元素的性质，块元素还是块元素，内联元素还是内联元素。

⑤相对定位的元素会提升一个层级。

（2）绝对定位

当一个元素的 position 属性的值为 absolute 时，元素对其原始位置进行绝对定位。在

CSS 中，绝对定位是使元素的位置与文档流无关的一种定位方式。设置为绝对定位的元素会从文档流完全删除，并相对于其包含块定位；默认情况下，绝对定位的位置是相对于浏览器而言的，配合 top、right、bottom、left 属性进行定位。

（3）固定定位

当一个元素的 position 属性的值为 fixed 时，那它会相对于浏览器边界进行定位；并且当页面存在滚动条时，页面的滚动也不会对该元素和浏览器边界之间的相对位置产生影响，就像是“漂浮”在一个固定的地方一样。所以该定位方式也会脱离文档流，它和文档内的元素不会有任何布局上的互相干扰。

2.2.4 元素背景

1. 背景颜色（background-color 属性）

background-color 属性主要用于设置元素的背景颜色。可以使用的颜色为 Web 技术中的常用颜色模式：“颜色英文单词”“HEX”“RGBa”“HSLa”。

2. 背景图片（background-image 属性）

background-image 属性用于设置背景图片。在实际开发中需要注意的是，background-image 是可以和 background-color（背景色）共存的。

3. 背景重复（background-repeat 属性）

background-repeat 属性用于设置背景图片的重复方式。当一张背景图片的宽或高小于其元素容器，或用 background-size（背景尺寸）属性设置的宽、高小于元素容器的宽、高时，背景图片默认会以平铺的方式排满整个元素的背景。该属性有以下值。

（1）repeat：默认，以平铺的方式排列图片。

（2）repeat-x：只在水平方向（X 轴方向）进行重复。

（3）repeat-y：只在垂直方向（Y 轴方向）进行重复。

（4）no-repeat：使背景图片不重复。

4. 背景尺寸（background-size 属性）

background-size 属性用于设置背景图片的大小，主要可以通过以下四种类型的单位进行设置。

（1）像素：可以通过像素精确地设置背景图片的大小。需要设置两个值：第一个值表示背景图的宽度，第二个值表示背景图的高度。也可以通过设置宽度或高度的其中一个，将另外一个值设置成 auto，达到保持背景图原始图像比例的效果。如“240px 120px”“180px auto”或“auto（相当于还原默认的大小）”。

（2）百分比：用百分比作为值，结构和用像素作单位一致，也是两个值：第一个值是相对于元素宽度的百分比值，第二个值是相对于元素高度的百分比值。如“100% 80%”“100% auto”或“auto 50%”。

（3）Contain：自动将背景图像填满元素的其中一边，图像的比例保持不变。

（4）Cover：在保持图像原始比例的情况下，将元素的背景区域完全覆盖，超出元素

宽高的部分会自动被裁剪，这是一种比较智能的背景图片大小设置方式。通过配合 background-position（背景定位）属性使用可以将开发者认为背景图像中更重要的部分通过定位显示出来。

5. 背景定位（background-position 属性）

background-position 属性用于设置背景图片在元素内出现的位置。该属性的值有以下三种。

（1）方位英文单词："left" "right" "top" "bottom" 和 "center"。用法如 "left top"（默认），设置背景图在元素内的 "左上方"；"right bottom"，设置背景图在元素内的 "右下方"；"center center"，设置背景图在元素的 "中心"。

（2）百分比：用法如 "0% 50%"，设置背景图在元素内 "水平方向" 的 "左方"，垂直方向的 "中心"；"50% 50%"，设置背景图在元素内的 "中心"；"100% 100%"，设置背景图在元素内的 "右下方"。

（3）像素：背景图的 "左上角" 相对于元素 "左上角" 偏移的距离，如 "10px 20px"，设置背景图 "水平向右" 偏移 10 像素，"垂直向下" 偏移 20 像素。

6. 背景固定（background-attachment 属性）

background-attachment 属性用于固定背景图。该属性主要有两个值：

（1）fixed：即使页面出现滚动条，背景图也会固定在原来的位置不会跟随页面滚动。

（2）scroll：默认值，当页面滚动的时候，背景图也跟随页面同步滚动。

7. 背景组合值

在实际的开发过程中为了节省代码量，减小维护难度，一般采用组合值的写法去设置背景，通常是下面这种模式：

```
background:color image repeat attachment position/size;
```

以上介绍的属性也可以省略，只设置其中一个，其中 background-clip、background-size 及 background-orign 需单独设置。在实际开发中，只需要注意 position 属性和 size 属性的固定写法，其余的属性可以任意交换顺序。

2.2.5 Flex 布局

Flex 是 Flexible Box 的缩写，Flex 布局意为 "弹性布局"，用来为盒子模型提供最大的灵活性。采用 Flex 布局的元素，称为 Flex 容器（flex container），简称容器。它的所有子元素自动成为容器成员，称为 Flex 项目（flex item），简称项目。容器可设置以下六个属性。

（1）flex-direction 属性：用于定义主轴的方向（即子元素的排列方向）。

（2）flex-wrap 属性：默认情况下，子元素都排在一条线（又称轴线）上。flex-wrap 属性用于定义如果一条轴线排不下如何换行。

（3）justify-content 属性：用于定义子元素在主轴上的对齐方式。

（4）align-items 属性：用于定义项目在交叉轴上如何对齐。

（5）align-content 属性：用于定义多条轴线的对齐方式。如果项目只有一条轴线，该

属性不起作用。

以上属性的具体值见表 2-4 所列。

表 2-4 容器的属性

属性	值	描述
flex-direction	row	默认值，主轴为水平方向，起点在左端
	row-reverse	主轴为水平方向，起点在右端
	column	主轴为垂直方向，起点在上
	column-reverse	主轴为垂直方向，起点在下
flex-wrap	nowrap	默认值，不换行
	wrap	换行，第一行在上方
	wrap-reverse	换行，第一行在下方
justify-content	flex-start	默认值，左对齐
	flex-end	右对齐
	center	居中
	space-between	两端对齐，子元素之间的间隔都相等
	space-around	每个子元素两侧的间隔相等，所以子元素之间的间隔比项目与边框的间隔大一倍
align-items	flex-start	交叉轴的起点对齐
	flex-end	交叉轴的终点对齐
	center	交叉轴的中点对齐
	baseline	子元素的第一行文字的基线对齐
	stretch	默认值，如果子元素未设置高度或设为 auto，将占满整个容器的高度
align-content	flex-start	与交叉轴的起点对齐
	flex-end	与交叉轴的终点对齐
	center	与交叉轴的中点对齐
	space-between	与交叉轴两端对齐，轴线之间的间隔平均分布
	space-around	每根轴线两侧的间隔都相等，所以轴线之间的间隔比轴线与边框的间隔大一倍
	stretch	默认值，轴线占满整个交叉轴（需设置高度为 auto）

2.2.6 过渡特效

在 CSS 中，用于设置过渡特效的属性是 transition。该属性允许 CSS 的属性值在一定的时间区间内平滑地过渡。该属性包含多个分支属性，主要有过渡 CSS 属性名称、过渡执行

时间、过渡时间速率曲线和过渡的延时执行四个主要内容。下面我们对它的各分支属性进行详细的学习。

1. transition-property（过渡属性）

该属性是用于规定应用过滤效果的 CSS 属性的名称。该属性有三种类型的值，见表 2-5 所列。

表 2-5　transition-property 的属性值

属性值	描述
none	将过渡效果设置为无过渡效果，或停止当前过渡效果
all	默认值，为所支持的所有 CSS 属性在值变化时执行动画过渡效果
property	定义应用过渡效果的 CSS 属性名称列表，列表以逗号分隔

2. transition-duration（过渡持续时间）

该属性用于设定一个属性的值完成过渡需要经历的时间，单位为秒（s），如“0.3 s”或“.3 s”。它的默认值为“0”，单位不能省略，否则过渡动画无法执行。

3. transition-timing-function（过渡线性规律/时间曲线）

该属性用于规定过渡效果随着时间改变的速度。目前，它能够设置六种类型的值，见表 2-6 所列。

表 2-6　transition-timing-function 的属性值

属性值	描述
cubic-bezier (x1,x2,y1,y2)	定义一个时间函数(贝塞尔函数),可以为其配置 4 个参数,前两个参数 x1 和 x2,定义“开始控制点”,后两个参数 y1 和 y2 定义“结束控制点”
linear	规定以相同速度开始至结束的过渡效果[等于 cubic-bezier(0,0,1,1)]
ease	默认值,规定慢速开始,然后变快,然后慢速结束的过渡效果[cubic-bezier(0.25,0.1,0.25,1)]
ease-in	规定以慢速开始的过渡效果[等于 cubic-bezier(0.42,0,1,1)]
ease-out	规定以慢速结束的过渡效果[等于 cubic-bezier(0,0,0.58,1)]
ease-in-out	规定以慢速开始和结束的过渡效果[等于 cubic-bezier(0.42,0,0.58,1)]

4. transition-delay（过渡延迟）

该属性规定在用户进行操作后多久开始执行动画，也就是延迟执行过渡动画的时间，单位是秒（s），写法与 transition-duration 属性一致，默认值也为“0”。

示例代码：

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>CSS3 transition-timing-function 属性</title>
```

```
    <style type="text/css">
        div
        {
            width:100px;
            height:50px;
            text-align:center;
            line-height:50px;
            margin-top:10px;
            border-radius:0;
            background-color:#14C7F3;
            transition-property:width;
            transition-duration:2s ;
            transition-delay:0;
        }

        #div1{transition-timing-function:linear;}
        #div2{transition-timing-function:ease;}
        #div3{transition-timing-function:ease-in;}
        #div4{transition-timing-function:ease-out;}
        #div5{transition-timing-function:ease-in-out}
        div:hover
        {
            width:300px;
        }
    </style>
</head>
<body>
    <div id="div1">linear</div>
    <div id="div2">ease</div>
    <div id="div3">ease-in</div>
    <div id="div4">ease-out</div>
    <div id="div5">ease-in-out</div>
</body>
</html>
```

5. transition（过渡动画组合）

其语法格式：

```
transition:property duration timing-function delay;
```

2.2.7 变形转化

在 CSS 中，执行变形转化的属性是 transform，能够执行缩放、倾斜、翻转、透视等变形转化的设置。

1. translate

该属性可以让元素从当前位置根据“left（X 轴）”参数和“top（Y 轴）”参数的设置在水平和垂直方向进行移动。该属性可设置四种类型的值，见表 2-7 所列。

表 2-7 translate 的属性值

属性值	描述
translateX(x)	设置元素在 X 轴方向的偏移
translateY(y)	设置元素在 Y 轴方向的偏移
translateZ(z)	设置元素在 Z 轴方向的偏移
translate(x, y)	同时设置元素在 X 轴和 Y 轴方向的偏移

2. rotate *

该属性用于规定元素以当前元素的中心（X=width/2，Y=height/2）进行旋转（若不对 transform-origin 属性进行设置），旋转的角度为参数所设定的值，正值表示顺时针，负值表示是逆时针，单位为度（deg）。该属性有三种类型的值，见表 2-8 所列。

表 2-8 rotate 的属性值

属性值	描述
rotateX(angle)	设置元素在 X 轴方向的旋转
rotateY(angle)	设置元素在 Y 轴方向的旋转
rotateZ(angle)	设置元素在 Z 轴方向的旋转

3. scale *

该属性用于规定元素以当前元素的中心进行缩放，参数的值为一个整数或浮点数，如“1”（默认值）不进行缩放，“0.8”缩小为原来的 80%，“1.5”扩大到原来的 150%，参数不需要单位。该属性值有三种类型的值，见表 2-9 所列。

表 2-9 scale 的属性值

属性值	描述
sclaeX（x）	设置元素在 X 轴方向的缩放
sclaeY（y）	设置元素在 Y 轴方向的缩放
sclae（x，y）	设置元素在 X 轴及 Y 轴方向的缩放

4. skew

该属性用于规定元素根据水平（X 轴）和垂直（Y 轴）线参数设定倾斜角度。这里的 X 轴和 Y 轴与我们平时认为的坐标系的 X 轴、Y 轴正好相反；且 X 轴正值是逆时针变形，而 Y 轴正值是顺时针变形。基点默认为中心点。该属性有三种类型的值，见表 2-10 所列。

表 2-10 skew 的属性值

属性值	描述
skewX(angle)	设置元素在水平轴方向的倾斜
skewY(angle)	设置元素在垂直轴方向的倾斜
skew(x-angle, y-angle)	设置元素在水平轴及垂直轴方向的倾斜

5. transform 组合值

和其他具有分支属性的 CSS 属性一样，transform 属性也能采用多值组合的方式来实现复杂的 CSS 效果。需要特别注意的是，若该属性配合 transition 属性使用，值的先后顺序不一样，很多时候效果也是不一样的，需要根据需要来调整值的顺序。

6. transform-origin *

该属性用于设置旋转元素的原点（基点）位置，默认为元素的中点。该属性有三种类型的值：方位英文单词、Web 常用长度单位和百分数。

（1）方位英文单词：有“top”“right”“bottom”“left”和“center”五个值可以使用，如“top left”表示左上角，“right bottom”表示右下角，“center center”表示默认的中点。

（2）Web 常用长度单位：如最常用的像素（px），字符（em），点（pt）等，如“16px 20px”表示旋转的轴在 X 轴的 16 像素，Y 轴（向下为正，向上为负）的 20 像素的位置。

（3）百分数：第一个百分数表示相对于元素宽度的百分比位置，第二个百分数表示相对于元素高度的百分比位置，如“0% 0%”相当于左上角，“50% 50%”相当于默认的中点，“100% 100%”相当于元素的右下角。

2.2.8 动画

在 CSS 中，执行动画的属性是 animation。该属性可以让元素随着时间的推进，产生位置、形状、颜色、大小、透明度等动画效果。

1. animation-name

该属性用于定义动画的名称，以供需要使用该动画的 animation 属性调用。虽说该名称完全是由用户自定义的，但也应该语义化一些，方便调用者能直观地了解此处定义动画的实际作用。

2. animation-duration

该属性用于定义动画执行的时间，即一段动画从开始到结束所经历的时间，单位为秒（s）或毫秒（ms）。该属性的默认值为“0”，即不执行任何动画，所以在定义设置一个动

画的时候始终都要设置属性值，并给定一个大于“0”的时间。

3. animation-timing-function

该属性定义元素随着时间的推进执行动画的速率变化（线性规律）。该属性的值见表2-11所列。

表 2-11　animation-timing-function 的属性值

值	描述
linear	动画从头到尾的速度是相同的
ease	默认值，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
steps(int,start/end)	指定时间函数中的间隔数量（步长）。有两个参数：第一个参数 int 指定函数的间隔数，是一个正整数；第二个参数 start/end 是可选的，表示动画是从时间段的开头连续还是末尾连续，其中 start 表示直接开始，end 的默认值，表示戛然而止
cubic-bezier(x1,x2,y1,y2)	定义一个时间函数（贝塞尔函数），可以为其配置四个参数，前两个参数 x1 和 x2，定义“开始控制点”，后两个参数 y1 和 y2 定义“结束控制点”

4. animation-delay

该属性用于定义动画延迟时间，即动画什么时候开始，单位为秒（s）或毫秒（ms）。它可以设置三种类型的值。

（1）0（默认值）：立即执行动画。

（2）正值：延迟指定时间后，开始执行动画。

（3）负值：立即执行，但跳过指定时间后进入动画。

5. animation-iteration-count

该属性用于设置动画播放的次数，它可以设置三种类型的值。

（1）1（默认值）：表示在执行某事件后只执行一次动画。

（2）[number]：任意正整数，表示在执行某事件后只执行[number]次动画。

（3）infinite：表示在执行某事件后无限次执行动画。

6. animation-direction

该属性用于设置元素动画是否能够周期性地逆向播放，逆向动画播放的进行时间和“正向播放”一致，时间速度曲线会按照“100%（to）”到“0%（from）”的方向进行。

（1）normal：默认值。

（2）alternate：逆向执行。

7. animation-play-state

该属性用于设置动画播放状态，其值有两种类型。

（1）running：播放动画。

（2）paused：暂停动画。

8. animation-fill-mode

若不将动画的 animation-iteration-count 属性设置为 infinite，动画在播放完成后会还原到元素没有“挂载”动画播放效果之前的状态，在有的应用场景里这样似乎没有问题，但在有些应用场景下，这样的设定是不被允许的。animation-fill-mode 属性的出现克服了这个问题，它可以预设动画播放前的第一帧和保留动画播放完成后的最后一帧，可以通过以下值进行设置：

（1）backwards：让元素保持动画第一帧定义里所设置的 CSS 属性，直到动画开始执行。

（2）forwards：让元素保持动画播放结束后最后一帧定义里所设置的 CSS 属性。

（3）both：让元素保持动画第一帧里定义的 CSS 属性，直到动画开始，动画播放完成后又保持动画最后一帧的属性。

9. animation

动画组合值写法，其语法格式：

```
    animation:name duration timing-function delay ineration-count diec-
tion play-state fill-mode;
```

2.3 实战与提高

运用 animation 属性来实现进度条。

相关要求

1. 了解动画标签。
2. 了解结构布局。

任务分析

最后实现效果如图 2-7 所示。

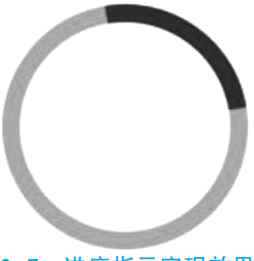

图 2-7 进度指示实现效果图

1. 该任务分为两部分：黄色区域和紫色区域。
2. 每隔一定的时间紫色区域会转动。
3. 这个动画效果是永久性的。

核心代码

```
<div class="box"></div>
.box {
    width:120px;
    height:120px;
    margin:100px auto;
    border:10px solid orange;
    border-left-color:purple;
    border-radius:50%;
    animation:loading 1.2s linear infinite;
    -webkit-animation:loading 1.2s linear infinite;
}
@ -webkit-keyframes loading {
    from {
        -webkit-transform:rotate(0deg);
    }
    to {
        -webkit-transform:rotate(360deg);
    }
}
@ keyframes loading {
    from {
        transform:rotate(0deg);
    }
    to {
        transform:rotate(360deg);
    }
}
```

经验分析

1. 这道题应该先实现静态图。
2. 题目要求是环形进度条的实现，需要运用到圆角部分的知识。
3. 实现动态的时候，注意黄色区域是不动的，只是紫色区域在移动。
4. 该动画的播放次数是永久性的。

课后习题

1. 创建一个名为“mycss1”的样式文件，该样式定义字体为华文仿宋、幼圆和宋体，字号为12pt，颜色为黄色，背景为蓝色，并在一个HTML文件中链接该样式文件。

2. 设置盒子模型实现购物网站商品橱窗展示，效果如图2-8所示。

运动裤
￥99.00

运动鞋
￥159.00

图2-8 完成效果图