

1 CSS

CSS（Cascading Style Sheets 层叠样式表）是一种向网页添加样式的机制，主要用作页面美化和布局控制！

层叠：多个样式可以作用在同一个html的元素上，同时生效。

学习网站：<https://www.w3school.com.cn/css/index.asp>

1.1 CSS概述

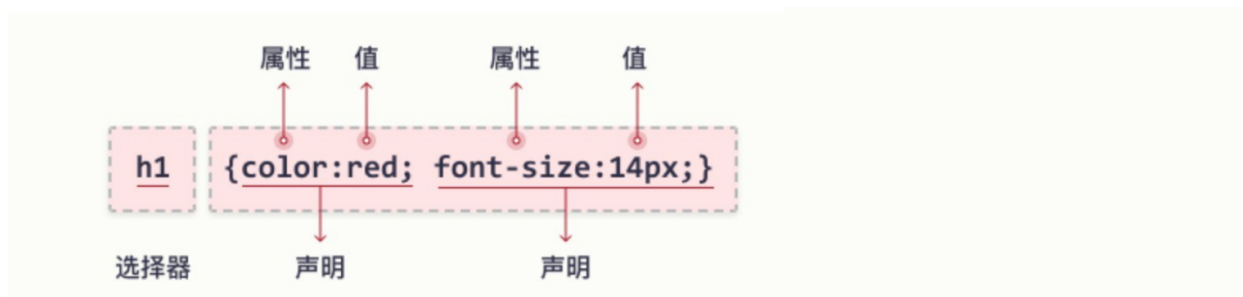
CSS全称为Cascading Style Sheets，即层叠样式表，是一种用来表现HTML或XML等文件样式的计算机语言，其是控制网页样式和布局的首选语言。

CSS的主要作用是将样式应用于HTML文档中的元素。通过CSS，可以改变元素的字体、颜色、大小、布局、背景等属性，从而实现对网页或应用程序的外观和样式的控制，CSS 文件扩展名为 **.css**。

CSS优势：样式与内容分离、灵活的样式控制、广泛的选择器、响应式设计、丰富的样式属性和浏览器兼容性等。

CSS语法：

```
选择器 {  
    /* 这是个注释 */  
    属性名：值；  
    属性名：值；  
    属性名：值1 值2 ...；  
    ...  
}
```



注意：

- CSS属性和值之间用 `:` 分隔
- CSS属性之间用 `;` 分隔
- CSS的值有多个的时候使用空格分隔

例如： `padding: 5px 10px 10px 5px;`

例如， `div.css`

```
div{  
    /* 字体大小 */  
    font-size: 30px;  
    /* 颜色 */  
    color: red;  
}
```

1.2 CSS引入

CSS通常和HTML结合在一起使用。

HTML文件引入CSS文件的方式有三种：**内联样式**、**内部嵌入**、**外部引用**。

1) 内联样式

在标签内使用**style属性**指定css代码。

每个HTML标签都包含有一个style属性，使用该属性可以指定样式。

新建 01-css引入.html

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>01-css引入</title>
</head>
<body>
  <!-- 内联样式 -->
  <div style="font-size: 30px; color: red;">
    hello world
  </div>
</body>
</html>
```

2) 嵌入式

把CSS样式直接嵌入到HTML页面中。

在head标签内，定义style标签，style标签的标签体内容就是css代码。

注意：style的type属性必须为text/css。

修改 01-css引入.html 如下：

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>01-css引入</title>
  <style type="text/css">
    div{
      /* 注释写法, 字体大小 */
      font-size: 30px;
      color: red;
    }
  </style>
```

```
</head>
<body>

    <div>hello world</div>

</body>
</html>
```

3) 外部引用

- 先定义css资源文件
- 再在head标签内，定义link标签，引入外部的资源文件

新增 `css/div.css` 文件：

```
div{
    font-size: 30px;
    color: red;
}
```

修改 `01-css引入.html` 如下：

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>01-css引入</title>
    <!-- 引入css，注意路径不要出问题 -->
    <link rel="stylesheet" href="css/test.css" type="text/css">
</head>
<body>

    <div>hello world</div>

</body>
```

```
</html>
```

CSS引入小结:

- 1、2、3种方式，css作用范围越来越大
- 方式1不常用，后期常用方式2、3
- 第3种格式也可以写为（了解）

```
<!-- 单独添加style标签 -->  
<style type="text/css">  
    @import "css/test.css";  
</style>
```

1.3 基础选择器

CSS选择器，其实只是CSS的一种语法，通过它可以筛选具有相似特征的元素。

选择器可以分为2大类：

1) 基础选择器

- id选择器
- 元素选择器
- 类选择器

id选择器优先级 > 类选择器 > 元素选择器

2) 扩展选择器

- 通配符选择器
- 组合选择器

- 关联选择器
- 属性选择器
- 伪类选择器

基础选择器介绍如下：

1) 标签选择器

通过HTML元素的标签名，来选择指定的元素

语法：

```
标签名 {  
    /* CSS样式 */  
    属性名： 值；  
    ...  
}
```

2) ID选择器

通过HTML元素的id属性值，来选择指定的元素

语法：

```
#id属性值 {  
    /* CSS样式 */  
    属性名： 值；  
    ...  
}
```

3) 类选择器

通过HTML元素的class属性值，来选择指定的元素

语法：

```
.class属性值{
    /* CSS样式 */
    属性名: 值;
    ...
}
```

案例展示:

新建 02-基础选择器.html，测试上述3种选择器，代码如下：

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>02-基础选择器</title>

    <style type="text/css">
        /* 标签选择器 */
        div{
            font-size: 20px;
            color: red;
        }

        /* id选择器 */
        #div02{
            font-size: 30px;
            color: blue;
        }

        /* class选择器 */
        .sp{
            color: rgb(0, 255, 195);
        }
    </style>
</head>
<body>
```

```
    </style>
</head>
<body>
    <div>hello myDiv01</div>
    <div id="div02">hello myDiv02</div>

    <div>
        <table>
            <tr>
                <td><span class="sp">class mySpan01</span></td>
            </tr>
            <tr>
                <td><span class="sp">class mySpan02</span></td>
            </tr>
        </table>
    </div>

</body>
</html>
```

1.4 扩展选择器

1) 通配符选择器

通配所有元素。

语法：

```
*{
    /* CSS样式 */
    属性名：值；
    ...
}
```


2) 组合选择器

使用逗号 , 隔开多个选择器, 此时多个选择器同时生效

语法:

```
选择器1, 选择器2, ...{  
    /* css样式 */  
    属性名: 值;  
    ...  
}
```

案例测试:

新增 03-扩展选择器.html 代码如下

```
<!DOCTYPE html>  
<html lang="en">  
<head>  
    <meta charset="UTF-8">  
    <meta name="viewport" content="width=device-width, initial-scale=1.0">  
    <title>03-扩展选择器</title>  
  
    <style type="text/css">  
        /* 通配符选择器 */  
        *{  
            font-size: 20px;  
        }  
  
        /* 组合选择器 */  
        #div1, .sc{  
            color: red;  
        }  
    </style>  
</head>  
<body>
```

```
<div id="div1">this a first div</div>
<div id="div2">this a second div</div>
<span class="sc">this is a span</span> <br>
</body>
</html>
```

3) 关联选择器

形式1：选中标签1中包含的所有标签2元素

```
标签名1 标签名2{
    /* css样式 */
    属性名：值；
    ...
}
```

例如：

```
/* 选中div中包含的所有span元素 */
div span{
    ...
}
```

形式2：选中标签1中所有子标签2

```
标签名1 > 标签名2{
    /* css样式 */
    属性名：值；
    ...
}
```

例如：

```
/* 选择父元素是<div>的所有子元素<span> */
div > span{
    ...
}
```

案例展示:

修改 03-扩展选择器.html 代码如下:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <title>03-扩展选择器</title>

    <style type="text/css">
        /* 通配符选择器 */
        *{
            font-size: 20px;
        }

        /* 组合选择器 */
        #div1, .sc{
            color: red;
        }

        /* 组合选择器形式1: 选择器1 选择器2
           例: div span 会选中div中所有span元素*/
        div span {
            color: ;
        }

        /* 组合选择器形式2: 选择器1 > 选择器2
           例: div > span 选择父元素是<div>的所有<span>元素 */
```

```

        /* div > span {
            color: red;
        } */

</style>
</head>
<body>
    <div id="div1">this a first div</div>
    <div id="div2">this a second div</div>
    <span class="sc">this is a span</span> <br>

    <hr>

    <!-- 父元素 -->
    <div>
        <!-- 子元素1 -->
        <table>
            <tr>
                <td>
                    <span>div内部的span01</span>
                </td>
                <td>
                    <span>div内部的span02</span>
                </td>
            </tr>
        </table>

        <!-- 子元素2 -->
        <!-- div中子元素span -->
        <span>div子元素span03</span>
    </div>

</body>
</html>

```

代码中注释 `div > span` , 运行效果:

this a first div
this a second div
this is a span

div内部的span01 div内部的span02
div子元素span03

← div span 会选中div中包含的所有span

取消注释的 `div > span` , 效果如下:

this a first div
this a second div
this is a span

div内部的span01 div内部的span02
div子元素span03

← div span 选中div的子标签span, 设置color: red

4) 伪类选择器

CSS伪类选择器是用于选择元素的特定状态或位置的选择器。

其以冒号 (:) 开头, 并附加在选择器的末尾。

格式:

```
元素:状态 {  
    /* CSS样式 */  
    属性名: 值;  
    ...  
}
```

常见伪类选择器:

- `:hover`: 选择鼠标悬停在元素上的状态
- `:active`: 选择被点击的元素的状态
- `:focus`: 选择获得焦点的元素的状态, 比如表单元素
- `:visited`: 选择已访问链接的状态

- `:first-child`: 选择作为其父元素的第一个子元素的元素
- `:last-child`: 选择作为其父元素的最后一个子元素的元素

案例展示:

在html文件body标签的最后添加下面代码

```
71      <!-- 超链接 -->
72      <div><a href="#" target="_blank">百度一下</a></div>
73      <div>I am the last div in body!</div>
74 </body>
75 </html>
```

```
<!-- 超链接 -->
<div><a href="#" target="_blank">百度一下</a>
<div>I am the last div in body!</div>
```

style标签中添加伪类选择器:

```
/* 伪类选择器 */
/* 修改悬浮时颜色 */
a:hover{
    color: yellow;
}
/* 修改body最后一个div元素样式 */
body div:last-child{
    font-size: 40px;
    color: green;
    font-style: italic;
}
```

运行效果:

this a first div
this a second div
this is a span

div内部的span01 div内部的span02

div子元素span03

百度一下

I am the last div in body!

伪类选择器应用

03-扩展选择器.html 完整代码:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <title>03-扩展选择器</title>

  <style type="text/css">
    /* 通配符选择器 */
    *{
      font-size: 20px;
    }

    /* 组合选择器 */
    #div1, .sc{
      color: red;
    }

    /* 组合选择器形式1: 选择器1 选择器2
      例: div span 会选中div中所有span元素*/
    div span {
      color: blue;
    }

    /* 组合选择器形式1: 选择器1 > 选择器2
      例: div > span 选择父元素是<div>的所有<span>元素 */
    div > span {
```

```

        color: red;
    }

    /* 伪类选择器 */
    /* 修改悬浮时颜色 */
    a:hover{
        color: yellow;
    }
    /* 修改body最后一个div元素样式 */
    body div:last-child{
        font-size: 40px;
        color: green;
        font-style: italic;
    }

</style>
</head>
<body>
    <div id="div1">this a first div</div>
    <div id="div2">this a second div</div>
    <span class="sc">this is a span</span> <br>

    <hr>

    <!-- 父元素 -->
    <div>
        <!-- 子元素1 -->
        <table>
            <tr>
                <td>
                    <span>div内部的span01</span>
                </td>
                <td>
                    <span>div内部的span02</span>
                </td>
            </tr>
        </table>
    </div>

```



```

    <!-- 子元素2 -->
    <!-- div中子元素span -->
    <span>div子元素span03</span>
</div>

<!-- 超链接 -->
<div><a href="#" target="_blank">百度一下</a></div>
<div>I am the last div in body!</div>
</body>
</html>

```

1.4 常见属性

CSS样式属性有很多，我们不可能也没有必要去强制记忆，大家应该掌握查阅CSS在线文档的方式，我们以查找 **background** 为例：

CSS在线教程

W3school

HTML/CSS Browser Side Server Side Programming XML Web Building Reference

HTML 教程
HTML
HTML5
XHTML
CSS
CSS3
Bootstrap 5
TCP/IP
浏览器脚本
JavaScript

领先的 Web 技术教程 - 全部免费

在 W3School，你可以找到你所需要的所有的网站建设教程。

从基础的 HTML 到 CSS，乃至进阶的 XML、SQL、JS、PHP 和 ASP.NET。

从左侧的菜单选择你需要的教程！

参考手册
HTML 标签
HTML 颜色
CSS
JavaScript
HTML DOM
jQuery
jQuery Mobile
VBScript
ASP
...

课程表

CSS 参考手册
CSS 参考手册
CSS 浏览器支持
CSS 选择器
CSS 函数
CSS 动画相关属性
CSS 网络安全字体
CSS 字体回退
CSS 单位
CSS 颜色
CSS 颜色值

CSS 参考手册

← 上一节 下一节 →

CSS 属性 按照字典顺序

A 查找有困惑的属性

align-content	规定弹性容器内的行之间的对齐方式，当项目不使用所有可用空间时。
align-items	规定弹性容器内项目的对齐方式。
align-self	规定弹性容器内所选项目的对齐方式。

工具箱

参考书
实验室
小测验
必修课



学会了如何查阅CSS文档，接下来我们来学习CSS常见样式属性和值。

1) 尺寸属性

- width 设置元素的宽度
- height 设置元素高度
- max-height 设置元素的最大高度
- max-width 设置元素的最大宽度
- min-height 设置元素的最小高度
- min-width 设置元素的最小宽度

2) 字体属性

- font-size 字体大小
- font-style 字体风格
normal正常、italic斜体、oblique倾斜
- font-weight 字体加粗
normal正常、bold粗体、bolder更粗、lighter更细
- font-family 字体族科
微软雅黑 Microsoft YaHei
- text-decoration 规定添加到文本的修饰

none	默认，没有线条
underline	规定在文本下方显示线条
overline	规定在文本上方显示线条
line-through	规定显示横穿文本的线条

例如：

```
<style type="text/css">
  #div1{
    font-size: 20px;
    font-style: italic;
    font-weight: bold;
    text-decoration: underline;
  }
</style>
```

3) 颜色

- color 设定文本的颜色
- opacity 设置透明度

所有浏览器都支持 opacity 属性，IE8以及更早的版本支持替代的 filter 属性

例如， `filter:Alpha(opacity=50)`

一般两个属性一起写，保证兼容性

```
<style type="text/css">
  #div1{
    opacity: 0.5;
    filter: Alpha(opacity=50);
  }
</style>
```

4) 背景

- background-color 设置元素的背景颜色

- background 复合属性

简写属性在一个声明中设置所有的背景属性

- background-image

设置元素的背景图像

- background-repeat

设置是否及如何重复背景图像

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

- background-position

设置背景图像的开始位置

案例：

```
<style type="text/css">
  #div1{
    background: url("image/01.png") repeat-x center;
  }
</style>
```

5) 边框属性

- border 复合属性

设置的属性（按顺序）：`border-width`、`border-style`和`border-color`

例：`border: 1px solid #ff0000;`

- border-width

设置4个边框的宽度

- border-style

设置4个边框的样式: **solid**实线 **dotted**点状 **double**双线 **dashed**虚线

如果4个值都给定了, 分别应用于上, 右, 下, 左边

如果给定1个值, 应用于4条边

如果给定2个值, 第1个值应用于上下边, 第2个值应用于左右边

- border-color

设置边框颜色

案例:

```
<style type="text/css">
  #div1{
    border: 2px dotted red;
  }
</style>
```

6) 鼠标光标属性

cursor, 属性规定要显示的光标的类型 (形状)

- none 无
- auto 默认, 浏览器设置的光标
- pointer 光标呈现为指示链接的指针 (一只手)
- wait 此光标指示程序正忙 (通常是一只表或沙漏)
- move 此光标指示某对象可被移动

```
<style type="text/css">
  span{
    /* cursor: move; */
    cursor: wait;
  }
</style>
```

7) 表格属性

- color
- font-size
- text-align 文字对齐
- background 背景
- border 边框, 只能用于table、th、td
- width 宽, 只能用于table、th、td
- height 高, 只能用于table、th、td

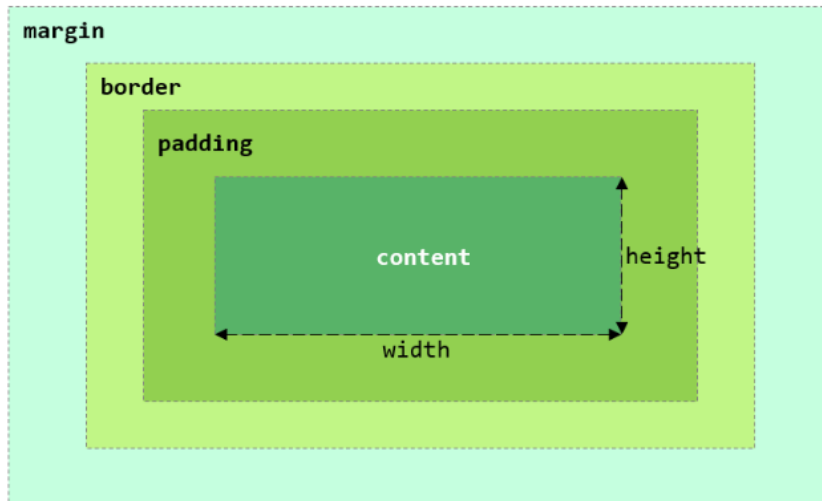
案例

```
<style>
  td,th{
    text-align: center;
    border: 1px dotted blue;
  }
</style>
```

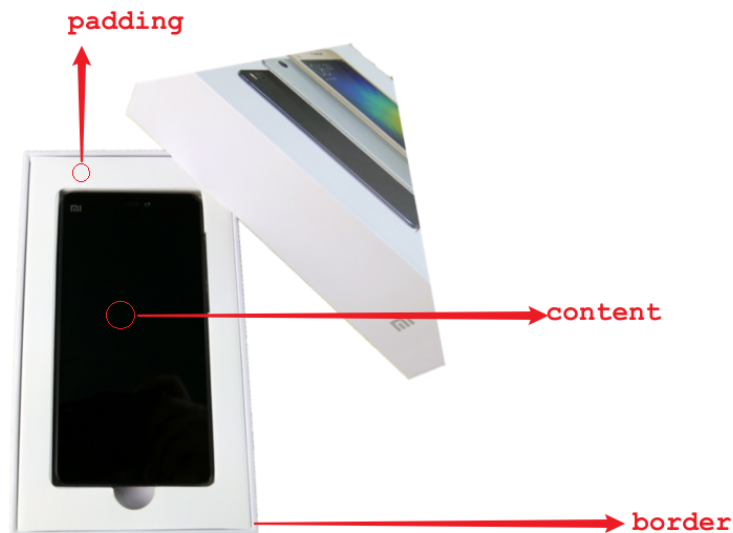
1.5 盒子模型

CSS 会把所有的 HTML 元素都看成一个盒子，所有的样式也都是基于这个盒子。

- margin (外边距) : 盒子与外界的距离
- border (边框) : 盒子的边框
- padding (内边距) : 紧贴内容的补白区域
- content (内容区域) : 元素中的文本或后代元素都是它的内容



CSS盒子模型和日常生活中的包装盒非常类似，比如：



盒子的大小，其实就包括三个部分：border边框、padding内边距、content内容。

盒子的大小 = content + 左右 padding + 左右 border;

注意：margin外边距不包括在盒子之内。

相关属性:

- border: 边框, 复合属性: border-width、border-style和border-color

例: border: 1px solid red;

- padding: 内边距

例: padding: 5px 10px 10px 5px; /* top right bottom left */

- margin: 外边距

例: margin: 10px 20px; /* top-bottom left-right */

补充内容: float属性, 表示浮动效果

float是CSS的一种属性, 主要属性值为: left (左浮动)、none (不浮动)、right (右浮动)、inherit (继承父元素浮动), 多用于网页排版。

案例展示: 05-float浮动.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>float浮动效果</title>

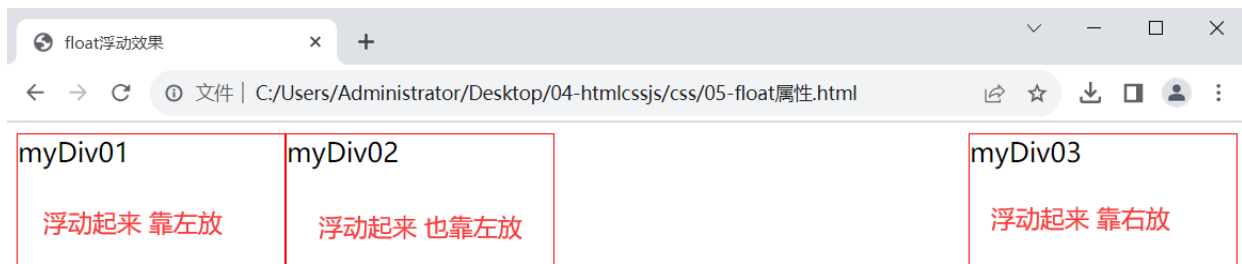
  <style>
    div{
      font-size: 20px;
      width: 200px;
      height: 100px;
      border: 1px solid red;
    }
  </style>
</head>
<body>
  <div>
    1
  </div>
</body>
</html>
```



```
#d1{
    float: left;
}
#d2{
    float: left;
}
#d3{
    float: right;
}

</style>
</head>
<body>
    <div id="d1">myDiv01</div>
    <div id="d2">myDiv02</div>
    <div id="d3">myDiv03</div>
</body>
</html>
```

效果：



正常情况下每个div都要占一整行，但飘浮起来，3个div就可以放到1行显示。

2 综合案例

📁 案例2_注册页面(css)

2023/11/22 8:07

文件夹

📁 案例3 图书商城(css)

2023/11/22 8:07

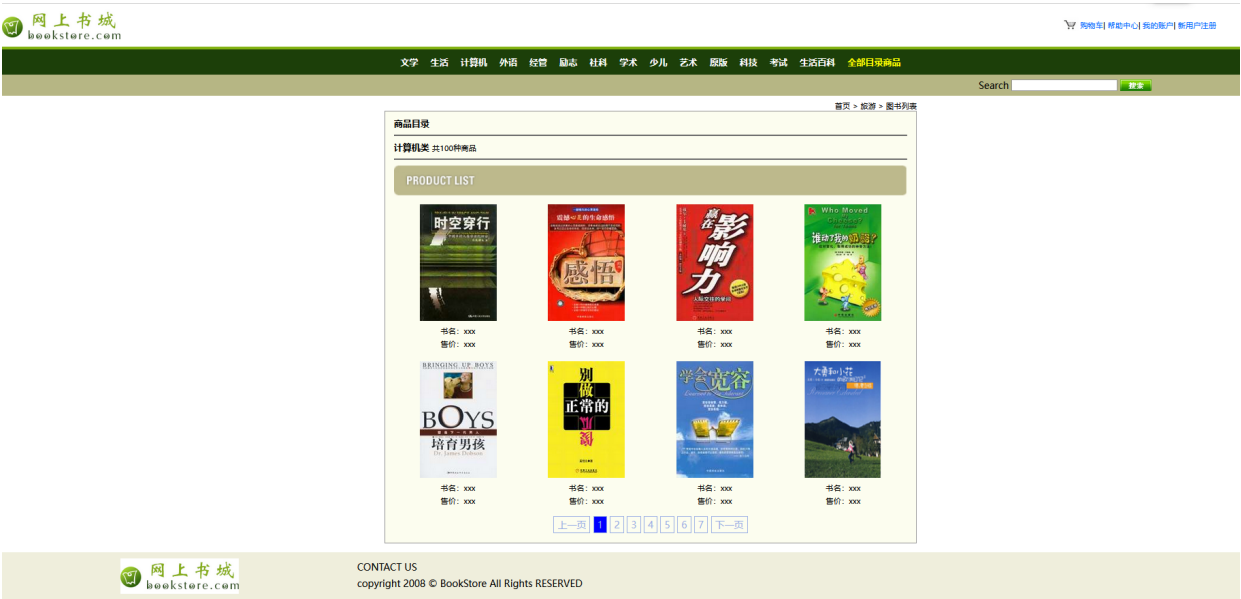
文件夹

阅读授课资料中提供的上述案例源码，熟悉掌握CSS的用法。

案例2效果：



案例3效果：



3 文档查阅

文档地址: <https://www.w3school.com.cn/index.html>

以padding属性为例:

