第八章 CSS3简介及新增选择器

8.1 CSS3简介

CSS3是css（层叠样式表）技术的升级版本，于1999年开始制定，2001年5月23日W3C完成了CSS3的工作草案，主要包括盒子模型、列表模型、超链接方式、语言模块、背景和边框、文字特效、多栏布局等模块。

CSS演进的一个主要变化就是W3C决定将CSS3分成一系列模块。浏览器厂商按CSS节奏快速创新，因此通过采用模块方法，CSS3规范里的元素能以不同速度向前发展，因为不同的浏览器厂商只支持给定特性。但不同浏览器在不同时间支持不同特性，这也让跨浏览器开发变得复杂 。

8.1.1 发展历程

早在2001年W3C就完成了CSS3的草案规范。CSS3规范的一个新特点是被分为若干个相互独立的模块。一方面分成若干较小的模块较利于规范及时更新和发布，及时调整模块的内容，这些模块独立实现和发布，也为日后CSS的扩展奠定了基础。另外一方面，由于受支持设备和浏览器厂商的限制，设备或者厂商可以有选择的支持一部分模块，支持CSS3的一个子集，这样有利于CSS3的推广。

1. CSS1.0：网页基本样式
2. CSS2.0：DIV（块）+ CSS，提出HTML与CSS结构分离的思想，网页变得简单，利于SEO
3. CSS2.1：浮动，定位
4. CSS3.0：圆角，阴影，动画...浏览器兼容性

8.1.2 新增特性

- 边框特性
- 多背景图
- 颜色与透明度
- 多列布局与弹性盒模型布局
- 盒子变形
- 过渡与动画
- 选择器
- Web字体
- 媒体查询
- 阴影
- 兼容问题

8.1.3 优势

1. 减少开发成本和维护成本
2. 提高页面性能

8.2 CSS 选择器

8.2.1 什么是选择器

在 CSS 中，选择器是选取需设置样式的元素的模式。

8.2.2 常用的选择器

1. ID选择器

   id选择器：给标签指定id，并对已指定id的标签设置样式。通过#号来定义

   注意:**不允许多个id选择器同名，id名唯一；**

```
<style>
    #pName{
    color: #fd0000;
}
</style>
<p id="pName">趁着万家灯火时爬上那山坡，再唱起那首歌好吗</p>
```

2. 类选择器

   类选择器：对已指定class的选择器进行样式更改。

   注意：

   - **一个标签可以同时设置多个class，用空格隔开即可。**
   - **允许多个class名字相同；**

```
<style>
    /*定义类选择器*/
    .oneclass{
     width:800px;
    }
</style>
<h2 class="oneclass towclass">你好</h2>
```

3. 标签（元素）选择器

   标签选择器：又叫元素选择器，可以批量选择同名的标签进行样式更改。

   注意：是选择所有同名的标签，而不是一个。

```
<style>
    span{
      color: red;
    }
</style>
<p>学完了<span>前端</span>，继续学Java</p>
```

4. 全局（通配符）选择器

   全局选择器：又叫通配符选择器，可以与任何元素匹配，一般只做初始化用，影响范围最大所以优先级越低

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

5. 基础选择器之间的优先级

   各选择器之间存在优先级关系，一般来说，标签的影响范围越大，优先级越低。

   - 优先级：**行内选择器>id选择器>类选择器>标签选择器>全局选择器**

6. 并集选择器(群组选择器)

并集选择器：使用**逗号**，来选择需要**设置相同样式**的标签

```
<style>
  li,p,span,h5{
  color: red;
}
</style>
<ul>
  <li>111</li>
  <li>222</li>
</ul>
<p>相邻选择器</p>
<h5>hello html</h5>
<span>hello java</span>
```

7. 交集选择器

不使用任何分隔符，选择同时满足多个条件的元素，只有同时匹配两个选择器的元素才会被选中。

```
<!--只有同时具有"box"和"content"两个class的按钮元素才会被选中，并应用红色字体颜色。-->
<style>
  box.content{
    color:red;
  }
}
</style>
<span class="box content">hello java</span>
```

8. 后代选择器

后代选择器：使用空格，选择所有被父元素包含的后代元素。

```
<style>
    ul li{
    color: red;
}
</style>
<ul>
    <li>111</li>
    <li>222</li>
    <div>
        <ol>
            <li>333</li>
        </ol>
    </div>
</ul>
```

9. 子代选择器

子代选择器：使用 >，选择所有被父元素包含的**直接子元素**，更深一层的元素不起作用。

```
<style>
    ol>li{
    color: blue;
}
```

```
</style>o
<ul>
    <li>111</li>
    <li>222</li>
    <div>
        <ol>
            <li>333</li>
        </ol>
    </div>
</ul>
```

10. 兄弟选择器

    兄弟选择器：使用 ~，在同一父元素下选择在指定元素之后的所有兄弟元素，

```
<style>
    p~p{
     color: blue;
    }
</style>
<p>这是第一个段落</p>
<div>这是一个 div 元素</div>
<p>这是第二个段落</p>
<p>这是第三个段落</p>
```

8.2.3 css选择器特性

1. 折叠性（Collapsing）

> 折叠性是指当多个CSS属性应用到同一个元素上时，它们如何相互作用以确定最终的样式。当不同的规则决定同一个属性的值时，折叠性规定了哪个规则的值会被应用。

折叠性的规则如下：

- 样式冲突，遵循的原则是就近原则，哪个样式离结构近，就执行哪个样式
- 样式不冲突，不会层叠

```
<!DOCTYPE html>
<html>
<head>
  <style>
    p {
      color: red;
    }

    p {
      color: blue;
    }
  </style>
</head>
<body>
  <p>This is a paragraph.</p>
</body>
</html>
<!--两个p选择器都定义了不同的颜色属性，一个是红色，一个是蓝色。根据折叠性规则，后面的规则会覆盖前面的规则。因此，最终应用到段落的颜色将是蓝色。-->
```

2. 继承性（Inheritance）

   继承性是指某些CSS属性的值能够被其子元素继承。当一个元素的某个属性没有显式地定义时，它将从父元素继承该属性的值。

   - 例如，如果父元素的颜色属性设置为红色，子元素的颜色属性通常也会继承为红色，除非在子元素上显式地设置了不同的值。
   - 并非所有的CSS属性都具有继承性，只有特定的属性才会被继承。
   - 子元素可以继承父元素的样式（text-,front-,line-这些元素开头的可以继承，以及color属性）
   - 行高的继承性

     ```
     <!DOCTYPE html>
     <html>
     <head>
       <style>
         .parent {
           font-size: 20px;
         }
       </style>
     </head>
     <body>
       <div class="parent">
         <p>This is a paragraph with inherited font size.</p>
       </div>
     </body>
     </html>
     ```

     在上面的示例中，父元素的字体大小设置为20像素，子元素的字体大小将继承为20像素。

3. 优先级（Specificity）

   CSS中的优先级是用于确定哪个规则将应用到特定元素的机制。它基于不同选择器和声明之间的特定性和重要性。

   - 权重优先级

     ```
     继承或 * ：0

     元素选择器：1

     类选择器，伪类选择器：10

     id选择器：100

     行内样式 style=“” ：1000

     !important ：无穷大
     ```

   - 权重由4组数字组成，但是**不会有进位**。可以理解为**类选择器永远大于元素选择器，id选择器永远大于类选择器，以此类推**
   - 权重叠加：如果是复合选择器，**权重会叠加**到一起，再计算
   - 优先级相同，则最后执行的样式生效

选择器	选择器权重
继承或者*	0,0,0,0
元素选择器	0,0,0,1
类选择器，伪类选择器	0,0,1,0
ID选择器	0,1,0,0
行内样式	1,0,0,0
!important	无穷大

8.3 CSS3新增选择器

8.3.1 后代级别选择器

1. :only-child 选择器

   :only-child 选择器 匹配属于父元素中 唯一子元素的 元素。

   注意：only-child 的冒号的前面的这个元素，一定要有一个父元素 对其进行包裹

```
<style>
        /* :only-child 选择器 匹配属于父元素中 唯一子元素的 元素。 */
            .con p:only-child{
                 background-color: red;
             }
</style>
<body>
         <div class="con">
             <div><p>小汤</p><p>小马</p></div>
             <div><p>小余</p></div>
             <div><span>小汤</span><p>小汤</p><span>小汤</span></div>
             <p><b>小汤</b></p>
             <div><p>小马过河</p></div>
         </div>
</body>
```

2. 结构伪类选择器(上)

   首先是第一类常用的结构伪类选择器，这类选择器常用于根据父级选择器来选择里面的子元素。

   基础语法如下表：

语法	说明
E:first-child	选择父元素中的第一个子元素，若该元素为E，则选中，否则选择器不生效
E:last-child	选择父元素中的最后一个子元素，若该元素为E，则选中，否则选择器不生效
E:nth-child(n)	根据n来选择父元素中的子元素，若选中的子元素为E，则选中，否则选择器不生效

   注意，这类选择器的**选择步骤**如下：

   - 先给所有子元素**从1开始**进行编号；

- 根据选择器来进行选择。如：E:first-child就选择第一个子元素，如果这个子元素是E的话，那么就选中了；但如果第一个子元素不是E的话，那么这类选择器就不会生效。

总之，这类选择器是“先编号，再选择，选择的元素为E，则选中”。

实例：

```
① ul li:first-child ：选择 ul 下的第1个子元素，若该元素为li，则选中该元素，否则不生效；

② ul li:nth-child(6) ：选择 ul 下的第6个子元素，若该元素为li，则选中该元素，否则不生效；

③ ul li:nth-child(2n) ：选择 ul 下的所有第偶数个子元素（2n即为偶数），若其为 li 则选中。
```

接着，这里还要对**E:nth-child(n)**这一基本语法按照 n 的分类进行详细的说明：

E:nth-child(n)中 n 的类型	**说明（选中的子元素为E则生效）**
数字	选择第n个子元素
关键字"even"	选择所有第偶数个子元素
关键字"odd"	选择所有第奇数个子元素
公式	根据公式进行选择。注意：公式中的 n **从0开始计算**，但子元素是**从1开始计数**的，因此第0个子元素是不存在的

当E:nth-child(n)中 n 的类型为“公式”时，提供以下实例供理解（**n 从0开始计算**）：

选择器	**说明（选中的子元素为E则生效）**
E:nth-child(n)	选择从第0个开始的所有子元素
E:nth-child(n+3)	选择从第3个开始的所有子元素
E:nth-child(2n)	选择所有第偶数个子元素
E:nth-child(2n+1)	选择所有第奇数个子元素
E:nth-child(-n+3)	选择前3个子元素

实际上，在日常使用中，由于nth-child(n)会给所有子元素进行编号（不管是不是E），因此E:nth-child(n) 中 n 的值和 E 作为子元素的位置往往是对应的（这样选择器才会生效）。

3. 结构伪类选择器（下）

接着是第二类常用的结构伪类选择器，这类选择器也用于根据父级选择器来选择里面的子元素，但和第一类有些差别。

基础语法如下表：

语法	说明
E:first-of-type	选择父元素中的第一个子元素E
E:last-of-type	选择父元素中的最后一个子元素E
E:nth-of-type(n)	根据n来选择父元素中的子元素E

注意，这类选择器的**选择步骤**如下：

（1）先给所有子元素E**从1开始**进行编号；

（2）根据选择器来进行选择。如：E:first-of-type就选择第一个子元素E。

实例：

```
① div p:first-of-type ：选择 div 下的第1个子元素p；

② div p:nth-of-type(2) ：选择 div 下的第2个子元素p。
```

结构伪类选择器的比较:

- 不同点：
  - :first-child、:last-child、:nth-child(n) 先给所有子元素编号，再选择，选中若是E则选择器有效，否则无效
  - :first-of-type、:last-of-type、:nth-of-type(n) 先给所有子元素E编号，再选择，选中一定是E
- 相同点：
  - 结构伪类选择器（如: first-child）的权重是0,0,1,0 E:first-child的权重是 E 的权重 + 0,0,1,0

4. :root 选择器

:root选择器 用匹配文档的根元素。

在HTML中根元素始终是HTML元素(HTML标签)。

```
<style>
        :root{
            /* 因为css2.1中的继承性---给自身设置样式，其后代元素会继承使用 */
            font-size: 30px;
            color: blue;
            background-color: pink;
        }
    </style>
```

因为给根元素HTML标签，设置了css样式，其后代元素会继承使用。

5. :empty 选择器

:empty选择器 选择每个**没有任何子级**的元素（包括文本节点）。

没有任何子级 指的是 有空标签都不行！！！

```
  <style>
        .con p:empty{
            background-color: blue;
            height: 3px;
        }
    </style>
```

```
<div class="con">
        <p>段落</p>
        <p></p>
        <p>段落</p>
        <p></p>
        <p>段落</p>
        <p><span></span></p>
        <p>段落</p>
</div>
```

8.3.2 伪元素选择器

- **伪类选择器：用来选择处于特定状态或位置的元素的选择器。**

1.超链接状态

超链接伪类一共有四个状态，如果要**同时设置**这四个状态，一定要**按顺序依次设置**，否则不起作用。

选择器	作用
:link	访问前
:visited	访问后
:hover	鼠标悬停
:active	点击时(激活)

```
<style>
        a:link{
            color: red;
        }
        a:visited{
            color: aqua;
        }
        a:hover{
            color: saddlebrown;
        }
        a:active{
            color: cornflowerblue;
        }
    </style>
 <a href="#">boss直聘</a>
```

2.focus获取焦点

focus：选择当前**获取焦点**的元素

```
<style>
    input:focus{
        background-color: skyblue;
    }
</style>
<input type="text">
```

3.伪元素选择器 before&after

伪元素选择器：用于在元素内容的前面和后面插入生成的内容，不需要修改 HTML 结构，为元素添加自定义的样式和装饰效果。

- 注意：
  - **必须要有content属性，要来设置伪元素的内容。如果没有则引号留空即可；**
  - **伪元素默认是行内显示模式；**
  - **优先级和标签选择器相同。**

选择器	说明
E::before	在E元素里面最前面添加一个伪元素
E::after	在E元素里面最后面添加一个伪元素

::before： before选择器用于给选定的元素之前插入内容

::after： after选择器一般给选定的元素之后插入内容

```
<style>
    span::before{
    content: "before";
    color: red;
  }
    span::after{
    content: "after";
    color: red;
  }
</style>
<span>伪元素选择器</span>
```

4.checked伪类选择器

用于选择在表单中**已被勾选**元素，注意**只能用于表单元素**，且仅用于那些可以被选中或取消选中的元素，如单选框和复选框。

```
<style>
    input[type="radio"]:checked{
       background-color: skyblue;
    }
    input[type="checkbox"]:checked{
       background-color: skyblue;
       font-size: 20px;
    }
</style>
<input type="radio" checked>男
<input type="checkbox" checked>同意
```

8.3.3 属性选择器

属性选择器是根据元素属性来选择元素，使用 `[]` 来包裹

语法	说明
标签[属性] {}	选择有目标属性的标签
标签[属性="value"] {}	选择有目标属性且属性值为"value"的标签
标签[属性^="x"] {}	选择有目标属性且属性值以"x"开头的标签
标签[属性$="x"] {}	选择有目标属性且属性值以"x"结尾的标签
标签[属性*="x"] {}	选择有目标属性且属性值包含"x"的标签

```
/* 选择所有带有 target 属性的 <a> 元素*/
a[target] {
  background-color: yellow;
}

/* 选取所有带有 target="_blank" 属性的 <a> 元素 */
a[target="_blank"] {
  background-color: yellow;
}

/* 选取 title 属性包含 "flower" 单词的所有元素 */
[title~="flower"] {
  border: 5px solid yellow;
}

/* 选取 class 属性以 "top" 开头的所有元素（值必须是完整） */
[class|="top"] {
  background: yellow;
}

/* 选取 class 属性以 "top" 开头的所有元素（值不一定是完整） */
[class^="top"] {
  background: yellow;
}
/* 选取 class 属性以 "test" 结尾的所有元素（值不一定是完整）*/
[class$="test"] {
  background: yellow;
}
/* 选取 class 属性包含 "my" 的所有元素（值不一定是完整） */
[class*="my"] {
  background: yellow;
}
```