

# CSS3介绍

---

CSS3在CSS2基础上，**增强或新增**了许多特性，弥补了CSS2的众多不足之处，使得Web开发变得更为高效和便捷。

## CSS3的现状

- 浏览器支持程度不够好，有些需要添加**私有前缀**
- 移动端支持优于PC端
- 不断改进中
- 应用相对广泛

## 应对的策略：渐进增强

- （1）坚持**渐进增强**的原则：**让低版本浏览器能正常访问页面，高版本的浏览器用户体验更好。**  
【重要】

比如说，同样是一个头像，可能在低版本的浏览器中，头像方的；在高版本的浏览器中，头像是圆的。

- （2）考虑用户群体。
- （3）遵照产品的方案。

参考链接：

- [渐进增强 VS 优雅降级 | 简书](#)
- [渐进增强和优雅降级之间的不同（面试题目）](#)

## 浏览器的版本问题

由于CSS3普遍存在兼容性问题，为了避免因兼容性带来的干扰，浏览器的建议版本为：

- Chrome浏览器 version 46+
- Firefox浏览器 firefox 42+

## 如何使用手册

CSS参考手册的网址：<http://css.doyoe.com/>

CSS参考手册的下载链接：<http://download.csdn.net/download/smyhvae/10243974>

在查看[CSS参考手册](#)时，需要注意以下符号：

- `[]` 表示可选项 如下图
- `||` 表示或者 `padding: [ <length> | <percentage> ]{1,4}`
- `|` 表示多选一 `border: <line-width> || <line-style> || <color>`
- `?` 0个或1个 `box-shadow: none | <shadow> [ , <shadow> ]*`
- `*` 表示0个或多个
- `{}` 表示范围

比如说, `{1,4}` 表示可以设置一至四个参数。

下面讲CSS3的基础知识。本文讲一下 CSS3 选择器的内容。

## CSS3 选择器

我们之前学过 CSS 的选择器, 比如:

```

1      div  标签选择器
2
3      .box  类名选择器
4
5      #box  id选择器
6
7      div p  后代选择器
8
9      div.box  交集选择器
10
11     div,p,span  并集选择器
12
13     div>p  子代选择器
14
15     * : 通配符
16
17     div+p: 选中div后面相邻的第一个p
18
19     div~p: 选中的div后面所有的p
20

```

CSS3新增了许多灵活查找元素的方法, 极大的提高了查找元素的效率和精准度。CSS3选择器与 jQuery 中所提供的绝大部分选择器兼容。

## 属性选择器

属性选择器的标志性符号是 `[]`。

匹配含义:

```

1 | ^: 开头  $: 结尾  *: 包含

```

格式:

- `E[title]` 选中页面的E元素，并且E存在 title 属性即可。
- `E[title="abc"]` 选中页面的E元素，并且E需要带有title属性，且属性值**完全等于**abc。
- `E[attr~=val]` 选择具有 attr 属性且属性值为：用空格分隔的字词列表，其中一个等于 val 的E元素。
- `E[attr|=val]` 表示要么是一个单独的属性值，要么这个属性值是以“-”分隔的。
- `E[title^="abc"]` 选中页面的E元素，并且E需要带有 title 属性,属性值以 abc 开头。
- `E[title$="abc"]` 选中页面的E元素，并且E需要带有 title 属性,属性值以 abc 结尾。
- `E[title*="abc"]` 选中页面的E元素，并且E需要带有 title 属性,属性值任意位置包含abc。

比如说，我们用属性选择器去匹配标签的className，是非常方便的。

这里我们用class属性来举例。代码举例：

```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>选择器 - 属性</title>
6      <style>
7          body {
8              margin: 0;
9              padding: 0;
10             font-family: '微软雅黑';
11             background-color: #F7F7F7;
12         }
13
14         .wrapper {
15             width: 1024px;
16             margin: 0 auto;
17         }
18
19         .wrapper > section {
20             min-height: 300px;
21             margin-bottom: 30px;
22             box-shadow: 1px 1px 4px #DDD;
23             background-color: #FFF;
24         }
25
26         .wrapper > header {
27             text-align: center;
28             line-height: 1;
29             padding: 20px;
30             margin-bottom: 10px;
31             font-size: 30px;
32         }
33
34         .wrapper section > header {
35             line-height: 1;
36             padding: 10px;
37             font-size: 22px;
38             color: #333;
39             background-color: #EEE;
40         }

```

```
41
42     .wrapper .wrap-box {
43         padding: 20px;
44     }
45
46     form {
47         width: 300px;
48         height: 300px;
49         margin: 0 auto;
50     }
51
52     form input[type="text"] {
53         width: 200px;
54         height: 30px;
55     }
56
57     form input[type="password"] {
58         width: 200px;
59         height: 30px;
60     }
61
62     .attr1 {
63
64     }
65
66     .download {
67     }
68
69     .attr1 a[href="/a.rmvb"] {
70         color: red;
71     }
72
73     .attr1 a[href="/b.rmvb"] {
74         color: pink;
75     }
76
77     /* E[attr~=val] 表示的一个单独的属性值 这个属性值是以空格分隔的*/
78     .attr2 a[class~= "download"] {
79         color: red;
80     }
81
82     /* E[attr|=val] 表示的要么一个单独的属性值 要么这个属性值是以 "-" 分隔的*/
83     .attr3 a[class|= "download"] {
84         color: red;
85     }
86
87     /* E[attr*=val] 表示的属性值里包含val字符并且在“任意”位置 */
88     .attr4 a[class*= "download"] {
89         color: red;
90     }
91
92     /* E[attr^=val] 表示的属性值里包含val字符并且在“开始”位置 */
93     .attr5 a[class^= "download"] {
94         color: red;
95     }
96
```

```

97      /* E[attr$=val] 表示的属性值里包含val字符并且在“结束”位置 */
98      .attr6 a[class$="download"] {
99          color: red;
100      }
101  </style>
102 </head>
103 <body>
104 <div class="wrapper">
105     <header>CSS3-属性选择器</header>
106     <section>
107         <header>简介</header>
108         <div class="wrap-box">
109             <form action="">
110
111                 <ul>
112                     <li>
113                         姓名: <input type="text">
114                     </li>
115                     <li>
116                         密码: <input type="password">
117                     </li>
118
119                     <li>
120                         性别: <input type="radio">男
121                             <input type="radio"> 女
122                     </li>
123                     <li>
124                         兴趣: <input type="checkbox" name="" id="">写代码
125                     </li>
126                     <li>
127                         <input type="submit" value="提交">
128                     </li>
129                 </ul>
130             </form>
131         </div>
132     </section>
133     <section class="attr1">
134         <header>E[attr]</header>
135         <div class="wrap-box">
136             <a href="./a.rmvb" class="download download-movie">下载</a>
137             <a href="./b.rmvb" class="download download-movie">下载</a>
138             <a href="./a.mp3" class="download download-music">下载</a>
139         </div>
140     </section>
141     <section class="attr2">
142         <header>E[attr~=attr]</header>
143         <div class="wrap-box">
144             <a href="./a.rmvb" class="download download-movie">下载</a>
145             <a href="./b.rmvb" class="download download-movie">下载</a>
146             <a href="./a.mp3" class="download download-music">下载</a>
147         </div>
148     </section>
149     <section class="attr3">
150         <header>E[attr|=attr]</header>
151         <div class="wrap-box">
152             <a href="./a.rmvb" class="download">下载</a>

```

```

153         <a href="./b.rmvb" class="download-movie">下载</a>
154         <a href="./a.mp3" class="download-music">下载</a>
155     </div>
156 </section>
157 <section class="attr4">
158     <header>E[attr*=val]</header>
159     <div class="wrap-box">
160         <a href="./a.rmvb" class="download">下载</a>
161         <a href="./b.rmvb" class="moviedownload">下载</a>
162         <a href="./a.mp3" class="downloadmusic">下载</a>
163     </div>
164 </section>
165 <section class="attr5">
166     <header>E[attr^=val]</header>
167     <div class="wrap-box">
168         <a href="./a.rmvb" class="download">下载</a>
169         <a href="./b.rmvb" class="moviedownload">下载</a>
170         <a href="./a.mp3" class="downloadmusic">下载</a>
171     </div>
172 </section>
173 <section class="attr6">
174     <header>E[attr$=val]</header>
175     <div class="wrap-box">
176         <a href="./a.rmvb" class="download">下载</a>
177         <a href="./b.rmvb" class="moviedownload">下载</a>
178         <a href="./a.mp3" class="downloadmusic">下载</a>
179     </div>
180 </section>
181 </div>
182 </body>
183 </html>

```

最后来张表格：

属性选择符 Attribute Selectors

选择符	版本	描述
E[att]	CSS2	选择具有att属性的E元素。
E[att="val"]	CSS2	选择具有att属性且属性值等于val的E元素。
E[att~="val"]	CSS2	选择具有att属性且属性值为一用空格分隔的单词列表，其中一个等于val的E元素。
E[att^="val"]	CSS3	选择具有att属性且属性值为以val开头的字符串的E元素。
E[att\$="val"]	CSS3	选择具有att属性且属性值为以val结尾的字符串的E元素。
E[att*="val"]	CSS3	选择具有att属性且属性值为包含val的字符串的E元素。
E[att ="val"]	CSS2	选择具有att属性且属性值为以val开头并用连接符"- "分隔的字符串的E元素，如果属性值仅为val，也将被选择。

紫色 链接表示该属性在CSS3有修改或者增加了新的属性值；橙色 链接表示该属性是CSS3新增属性

## 结构伪类选择器

伪类选择器的标志性符号是 `:`。

CSS中有一些伪类选择器，比如 `:link`、`:visited`、`:hover`、`:active`、`:focus`。

CSS3又新增了其它的伪类选择器。这一小段，我们来学习CSS3中的**结构伪类选择器**：即通过**结构**来进行筛选。

## 1、格式：（第一部分）（重要）

- `E:first-child` 匹配父元素的第一个子元素E。
- `E:last-child` 匹配父元素的最后一个子元素E。
- `E:nth-child(n)` 匹配父元素的第n个子元素E。**注意**，盒子的编号是从1开始算起，不是从0开始算起。
- `E:nth-child(odd)` 匹配奇数
- `E:nth-child(even)` 匹配偶数
- `E:nth-last-child(n)` 匹配父元素的倒数第n个子元素E。

理解：

(1) 这里我们要好好理解**父元素**的含义，它指的是：以 E 元素的父元素为参考。

(2) 注意：以上选择器中所选到的元素的类型，必须是指定的类型E，如果选不中，则无效。这个要好好理解，具体可以看CSS参考手册中的 `E:nth-child(n)` 的示例。我们可以理解成：**先根据选择器找到选中的全部位置，如果发现某个位置不是类型E，则该位置失效。**

(3) 另外，`E:nth-child(n)` 这个属性也很有意思。比如，针对下面这样一组标签：

```
1      <ul>
2          <li>1</li>
3          <li>2</li>
4          <li>3</li>
5          <li>4</li>
6          <li>5</li>
7          <li>6</li>
8          <li>7</li>
9          <li>8</li>
10         <li>9</li>
11         <li>10</li>
12     </ul>
```

上方代码中：

- 如果选择器写成 `li:nth-child(2)`，则表示第2个 `li`。
- 如果选择器写成 `li:nth-child(n)`，则表示**所有的** `li`。因为此时的 `n` 表示 0,1,2,3,4,5,6,7,8.....  
(当n小于1时无效，因为n = 0 也是不会选中的)
- 如果选择器写成 `li:nth-child(2n)`，则表示所有的**第偶数个** `li`。
- 如果选择器写成 `li:nth-child(2n+1)`，则表示所有的**第奇数个** `li`。
- 如果选择器写成 `li:nth-child(-n+5)`，则表示**前5个** `li`。
- 如果选择器写成 `li:nth-last-child(-n+5)`，则表示**最后5个** `li`。
- 如果选择器写成 `li:nth-child(7n)`，则表示选中7的倍数。。

上面列举的选择器中，我们只要记住：`n` 表示 0,1,2,3,4,5,6,7,8.....就很容易明白了。

## 2、格式：（第二部分）

- `E:first-of-type` 匹配同类型中的第一个同级兄弟元素E。
- `E:last-of-type` 匹配同类型中的最后一个同级兄弟元素E。
- `E:nth-of-type(n)` 匹配同类型中的第n个同级兄弟元素E。

- `E:nth-last-of-type(n)` 匹配同类型中的倒数第n个同级兄弟元素E。

既然上面这几个选择器带有 `type`，我们可以这样理解：先在同级里找到所有的E类型，然后根据 n 进行匹配。

### 3、格式：（第三部分）

- `E:empty` 匹配没有任何子节点（包括空格等text节点）的元素E。
- `E:target` 匹配相关URL指向的E元素。要配合锚点使用。

举例：

我们可以把多个伪类选择器结合起来使用，比如：

一	二	三	四	五	六	日
30 十九	1 艾滋病日	2 廿一	3 廿二	4 廿三	5 廿四	6 廿五
7 廿六	8 廿七	9 廿八	10 廿九	11 初一	12 初二	13 初三
14 初四	15 初五	16 初六	17 初七	18 初八	19 初九	20 初十
21 十一	22 十二	23 十三	24 十四	25 十五	26 十六	27 十七
28 十八	29 十九	30 二十	31 廿一	1 元旦	2 廿三	3 廿四

如果想把上图中，第一行的前三个 span 标红，我们可以这样使用结构伪类选择器：

```

1 dt:first-child span:nth-of-type(-n+3) {
2     color: red;
3 }
```

最后来张表格：



选择符	版本	描述
<a href="#">E:link</a>	CSS1	设置超链接a在未被访问前的样式。
<a href="#">E:visited</a>	CSS1	设置超链接a在其链接地址已被访问过时的样式。
<a href="#">E:hover</a>	CSS1/2	设置元素在其鼠标悬停时的样式。
<a href="#">E:active</a>	CSS1/2	设置元素在被用户激活（在鼠标点击与释放之间发生的事件）时的样式。
<a href="#">E:focus</a>	CSS1/2	设置元素在成为输入焦点（该元素的onfocus事件发生）时的样式。
<a href="#">E:lang(fr)</a>	CSS2	匹配使用特殊语言的E元素。
<a href="#">E:not(s)</a>	CSS3	匹配不含有s选择符的元素E。
<a href="#">E:root</a>	CSS3	匹配E元素在文档的根元素。
<a href="#">E:first-child</a>	CSS2	匹配父元素的第一个子元素E。
<a href="#">E:last-child</a>	CSS3	匹配父元素的最后一个子元素E。
<a href="#">E:only-child</a>	CSS3	匹配父元素仅有的一个子元素E。
<a href="#">E:nth-child(n)</a>	CSS3	匹配父元素的第n个子元素E。
<a href="#">E:nth-last-child(n)</a>	CSS3	匹配父元素的倒数第n个子元素E。
<a href="#">E:first-of-type</a>	CSS3	匹配同类型中的第一个同级兄弟元素E。
<a href="#">E:last-of-type</a>	CSS3	匹配同类型中的最后一个同级兄弟元素E。
<a href="#">E:only-of-type</a>	CSS3	匹配同类型中的唯一的一个同级兄弟元素E。
<a href="#">E:nth-of-type(n)</a>	CSS3	匹配同类型中的第n个同级兄弟元素E。
<a href="#">E:nth-last-of-type(n)</a>	CSS3	匹配同类型中的倒数第n个同级兄弟元素E。
<a href="#">E:empty</a>	CSS3	匹配没有任何子元素（包括text节点）的元素E。
<a href="#">E:checked</a>	CSS3	匹配用户界面上处于选中状态的元素E。（用于input type为radio与checkbox时）
<a href="#">E:enabled</a>	CSS3	匹配用户界面上处于可用状态的元素E。
<a href="#">E:disabled</a>	CSS3	匹配用户界面上处于禁用状态的元素E。
<a href="#">E:target</a>	CSS3	匹配相关URL指向的E元素。
<a href="#">@page:first</a>	CSS2	设置页面容器第一页使用的样式。仅用于 <a href="#">@page</a> 规则
<a href="#">@page:left</a>	CSS2	设置页面容器位于装订线左边的所有页面使用的样式。仅用于 <a href="#">@page</a> 规则
<a href="#">@page:right</a>	CSS2	设置页面容器位于装订线右边的所有页面使用的样式。仅用于 <a href="#">@page</a> 规则

[紫色](#) 链接表示该属性在CSS3有修改或者增加了新的属性值；[橙色](#) 链接表示该属性是CSS3新增属性

## 伪元素选择器

伪元素选择器的标志性符号是 `::`。

### 1、格式：（第一部分）

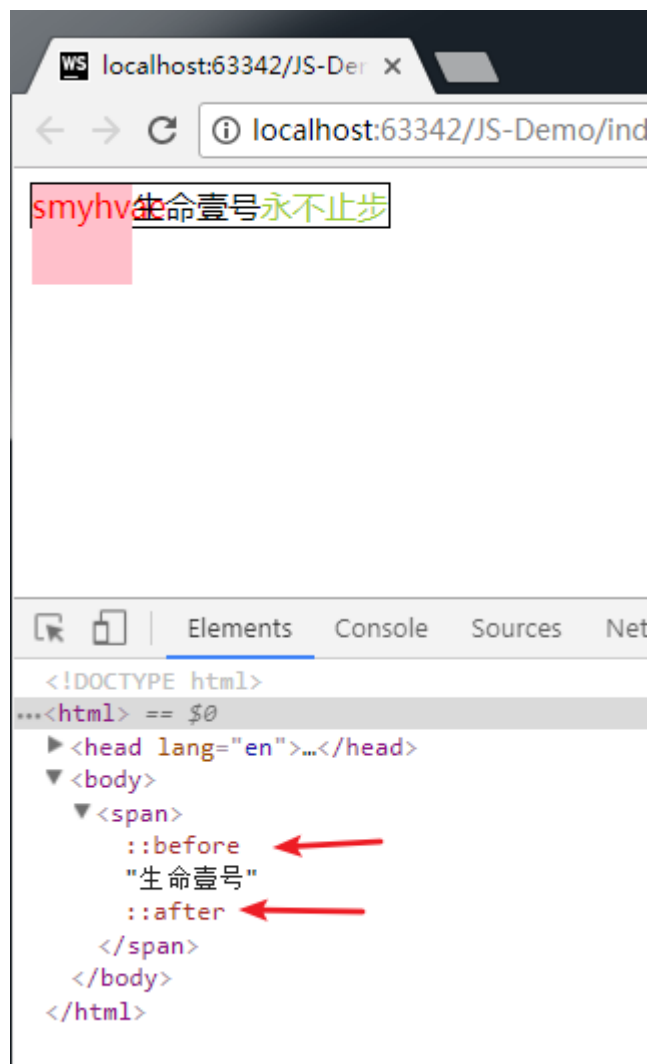
- `E::before` 设置在 元素E 前面（依据对象树的逻辑结构）的内容，配合content属性一起使用。
- `E::after` 设置在 元素E 后面（依据对象树的逻辑结构）的内容，配合content属性一起使用。

E:after、E:before 在旧版本里是伪类，在 CSS3 这个新版本里是伪元素。新版本里，E:after、E:before 会被自动识别为 E::after、E::before，按伪元素来对待，这样做的目的是用来做兼容处理。

举例：

```
1  <!DOCTYPE html>
2  <html>
3  <head lang="en">
4      <meta charset="UTF-8">
5      <title></title>
6      <style>
7
8          /*::before 和::after 是通过 css 模拟出的html标签的效果*/
9          span::before{
10              content:"smyhvae";
11              color:red;
12              background-color: pink;
13              width: 50px;
14              height: 50px;
15              display: inline-block;
16          }
17
18          span::after{
19              content:"永不止步";
20              color:yellowgreen;
21          }
22
23          /*给原本的span标签设置一个默认的属性*/
24          span{
25              border: 1px solid #000;
26          }
27      </style>
28  </head>
29  <body>
30
31  <span>生命壹号</span>
32  </body>
33  </html>
34
```

效果如下：



上图可以看出：

- 通过伪元素选择器，就可以添加出类似于span标签的效果（记得要结合 content 属性使用）。
- 通过这两个属性添加的伪元素，是行内元素，需要转换成块元素才能设置宽高。

## 2、格式：（第二部分）

- `E::first-letter` 设置元素 E 里面的**第一个字符**的样式。
- `E::first-line` 设置元素 E 里面的**第一行**的样式。
- `E::selection` 设置元素 E 里面被鼠标选中的区域的样式（一般设置颜色和背景色）。

`E::first-letter` 的举例：



E::first-line 的举例：

```
1 <!DOCTYPE html>
2 <html>
3 <head lang="en">
4 <meta charset="UTF-8">
5 <title></title>
6 <style>
7     div{
8         width: 500px;
9         border: 1px solid #000;
10        height: 300px;
11    }
12    /* 伪元素选择器，设置 p 标签里的第一行内容的样式 */
13    p::first-line{
14        color:red;
15    }
16 </style>
17 </head>
18 <body>
19 <div>
20 <p>渐进增强（Progressive Enhancement）：一开始就针对低版本浏览器进行构建页面，完成基本的功能，然后再针对高级浏览器进行效果、交互、追加功能达到更好的体验。
21 <p>优雅降级（Graceful Degradation）：一开始就构建站点的完整功能，然后针对浏览器测试和修复。比如一开始使用 CSS3 的特性构建了一个应用，然后逐步针对各大浏览器进行 hack 使其可以在低版本浏览器上正常浏览。
22 <p>其实渐进增强和优雅降级并非什么新概念，只是旧的概念换了一个新的说法。在传统软件开发中，经常会提到向上兼容和向下兼容的概念。渐进增强相当于向上兼容，而优雅降级相当于向下兼容。
23 </div>
24 </body>
25 </html>
```

localhost:63342/JS-Demo/index.html?\_ijt=o1cpd951sn5ulhh0

渐进增强（Progressive Enhancement）：一开始就针对低版本浏览器进行构建页面，完成基本的功能，然后再针对高级浏览器进行效果、交互、追加功能达到更好的体验。

优雅降级（Graceful Degradation）：一开始就构建站点的完整功能，然后针对浏览器测试和修复。比如一开始使用 CSS3 的特性构建了一个应用，然后逐步针对各大浏览器进行 hack 使其可以在低版本浏览器上正常浏览。

其实渐进增强和优雅降级并非什么新概念，只是旧的概念换了一个新的说法。在传统软件开发中，经常会提到向上兼容和向下兼容的概念。渐进增强相当于向上兼容，而优雅降级相当于向下兼容。

最后来张表格：

伪对象选择符 Pseudo-Element Selectors		
选择符	版本	描述
E::first-letter/E::first-letter	CSS1/3	设置对象内的第一个字符的样式。
E::first-line/E::first-line	CSS1/3	设置对象内的第一行的样式。
E::before/E::before	CSS2/3	设置在对象前（依据对象树的逻辑结构）发生的内容。用来和content属性一起使用
E::after/E::after	CSS2/3	设置在对象后（依据对象树的逻辑结构）发生的内容。用来和content属性一起使用
E::placeholder	CSS3	设置对象文字占位符的样式。
E::selection	CSS3	设置对象被选择时的颜色。

\* CSS3将伪对象选择符(Pseudo-Element Selectors)前面的单个冒号(:)修改为双冒号(::)用以区别伪类选择符(Pseudo-Classes Selectors)，但以前的写法仍然有效。

紫色 链接表示该属性在CSS3有修改或者增加了新的属性值；橙色 链接表示该属性是CSS3新增属性

## 我的公众号

想学习更多技能？不妨关注我的微信公众号：千古壹号（id：qianguyihao）。

扫一扫，你将发现另一个全新的世界，而这将是一场美丽的意外：

