

第十一讲：选择器

学时计划：2 学时 理论，0 学时 实验

（无实验教学内容）

教学大纲：

- 1、认识选择器
- 2、基本选择器
- 3、属性选择器
- 4、结构性伪类选择器
- 5、UI 元素状态伪类选择器
- 6、案例：数据表格
- 7、讨论与思考

选择器是 CSS 的重要内容。通过选择器的使用，在标签中指定特定的 HTML 元素而不必使用多余的 class、ID，从而实现轻量级的标签书写以及结构与表现更好的分离。CSS 中的选择器大部分并不是 CSS 3 中新添加的，而是没有被得到应有的广泛应用。

本讲详细介绍 CSS 的选择器，并通过案例展示选择器的使用方法以及通过选择器如何提高页面的简洁性，从而让读者体会到选择器的重要价值和意义。

一、认识选择器

选择器是 CSS 的重要内容，使用选择器可以大幅度的提高开发人员在书写或修改页面样式是效率，并使得页面更加简洁和易于阅读。

选择器主要是用来确定 HTML 的树形结构中的 DOM 元素节点。在 CSS 3 中提倡使用选择器来讲样式与元素直接绑定起来，使得样式表中的样式定义和页面中的元素相匹配起来。通过选择器，还可以实现各种复杂的指定，同时能大量减少样式表的代码书写量。

CSS 选择器分为三部分：

第一部分是常用的部分，叫做基本选择器；

第二部分是属性选择器；
第三部分是伪类选择器。

二、基本选择器

2.1 通配符选择器

通配符选择器是用来选择所有元素，也可以选择某个元素下的所有元素。

案例：

示例 11-1：通配符选择器

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>通配符选择器</title>
<style type="text/css">
* {
  margin:3px;
  padding:0px;
  border:1px solid #F00;
}
</style>
</head>

<body>
<div>
  <ul>
    <li>1</li>
    <li>2</li>
    <li>3</li>
    <li>4</li>
    <li>5</li>
    <li>6</li>
    <li>7</li>
    <li>8</li>
  </ul>
</div>
</body>
</html>
```

2.2 元素选择器(E)

元素选择器，是 CSS 选择器中最常见而且最基本的选择器。元素选择器其实就是对文档的元素进行样式定义，如 html, body, p, div 等定义样式。

案例：

示例 11-2：元素选择器

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>元素选择器</title>
<style type="text/css">
li {
margin:10px;
padding:5px;
width:18px;
font-size:13px;
color:#FFF;
text-align:center;
border-radius:5px;
border:1px solid #F00;
background-color: #F90;
float:left;
list-style:none;
}
</style>
</head>

<body>
<div>
<ul>
<li>1</li>
<li>2</li>
<li>3</li>
<li>4</li>
<li>5</li>
<li>6</li>
<li>7</li>
<li>8</li>
</ul>
</div>
</body>
</html>
```

2.3 类选择器 (.className)

类选择器是以一独立于文档元素的方式来指定样式，使用类选择器之前需要在 html 元素上定义类名，换句话说需要保证类名在 html 标记中存在，这样才能选择类。

类选择器还可以结合元素选择器来使用，比如说，文档中有多个元素使用了类名“items”，但是只需要在使用了类型“items”的 p 元素上修改样式，则可以将类名称定义为“p.items”。

也可以为一个元素定义多个名称，从而实现多个类共同对单一元素起作用。

案例：

示例 11-3：类选择器

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>类选择器</title>
<style type="text/css">
li {
margin:10px;
padding:5px;
width:18px;
font-size:13px;
color:#FFF;
text-align:center;
border-radius:5px;
border:1px solid #F00;
background-color: #F90;
float:left;
list-style:none;
}
.ButtonNow {
font-weight:bold;
background-color:#FF0;
border:1px solid #F00;
color:#F00;
}
</style>
</head>

<body>
```

```
<div>
  <ul>
    <li>1</li>
    <li>2</li>
    <li class="ButtonNow">3</li>
    <li>4</li>
    <li>5</li>
    <li>6</li>
    <li>7</li>
    <li>8</li>
  </ul>
</div>
</body>
</html>
```

2.4 ID 选择器 (#ID)

ID 选择器和类选择器是很相似的，在使用 ID 选择器之前也需要先在 html 文档中加注 ID 名称，这样在样式选择器中才能找到相对应的元素。与类选择器不同的是，ID 选择器是一个页面中唯一的值。

在定义类时在相对应的类名前加上一个“.”号 (.className)，而在定义 ID 选择器时则是在名称前使用“#”号 (#id)。

案例：

示例 11-4：ID 选择器

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>ID 选择器</title>
<style type="text/css">
li {
  margin:10px;
  padding:5px;
  width:18px;
  font-size:13px;
  color:#FFF;
  text-align:center;
  border-radius:5px;
  border:1px solid #F00;
  background-color: #F90;
  float:left;
  list-style:none;
```

```
}
#ButtonNow {
  font-weight:bold;
  background-color:#FF0;
  border:1px solid #F00;
  color:#F00;
}
</style>
</head>

<body>
<div>
  <ul>
    <li>1</li>
    <li>2</li>
    <li id="ButtonNow">3</li>
    <li>4</li>
    <li>5</li>
    <li>6</li>
    <li>7</li>
    <li>8</li>
  </ul>
</div>
</body>
</html>
```

2.5 后代选择器 (E F)

后代选择器也被称作包含选择器，就是可以选择某元素的后代元素。

比如说 E F，前面 E 为祖先元素，F 为后代元素。所表达的意思就是选择了 E 元素的所有后代 F 元素。请注意祖先元素和后代元素之间使用一个空格隔开。这里 F 不管是 E 元素的子元素或者是孙元素或者是更深层次的关系，都将被选中。也就是说，E F 将递归选中 E 元素中的所有 F 元素。

2.6 子元素选择器 (E>F)

子元素选择器只能选择某元素的子元素。

比如说 E>F，其中 E 为父元素，而 F 为子元素，其中 E>F 所表示的是选择了 E 元素下的所有子元素 F。和后代选择器 (E F) 不同，

E>F 仅选择了 E 元素的 F 子元素，而不是递归查询。

案例：

示例 11-5：后代选择器与子选择器

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>后代选择器与子选择器</title>
<style type="text/css">
nav>li {
  border:1px dashed #333333;
}
nav>p {
  font-size:16px;
  color:#F00;
}
div li {
  border:1px dashed #333333;
}
</style>
</head>

<body>
<nav>
  <ul>
    <li>1</li>
    <li>2</li>
    <li>3</li>
    <li>4</li>
    <li>5</li>
    <li>6</li>
    <li>7</li>
    <li>8</li>
  </ul>
</nav>
<nav>
  <p>网站首页</p>
  <p>联系我们</p>
</nav>
<div>
  <ul>
    <li>1</li>
    <li>2</li>
    <li>3</li>
  </ul>
</div>
```

```
<li>4</li>
<li>5</li>
<li>6</li>
<li>7</li>
<li>8</li>
</ul>
</div>
</body>
</html>
```

2.7 相邻兄弟元素选择器 (E+F)

相邻兄弟选择器可以选择紧接在另一元素后的元素，而且具有一个相同的父元素。

比如 E+F，E 和 F 元素具有一个相同的父元素，而且 F 元素在 E 元素后面且紧相邻，这样就可以使用相邻兄弟元素选择器来选择 F 元素。

2.8 通用兄弟选择器 (E~F)

通用兄弟元素选择器是 CSS 3 新增的选择器，将选择某元素后面的所有兄弟元素，通用兄弟元素需要在同一个父元素之中。

比如 E~F，E 和 F 元素是属于同一父元素之内，并且 F 元素在 E 元素之后，那么 E~F 选择器将选择中所有 E 元素后面的所有 F 元素。

通用兄弟选择器和相邻兄弟选择器极其相似，不同的是，相邻兄弟选择器仅选择是元素的仅与其相邻的后面元素（选中的仅一个元素）；而通用兄弟元素选择器，选中的是元素后面的所有兄弟元素。

案例：

示例 11-6：兄弟选择器

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>兄弟选择器选择器</title>
<style type="text/css">
li {
margin:10px;
font-size:12px;
width:24px;
```



```
height:18px;
padding-top:5px;
text-align:center;
border:1px solid #999;
background-color:#CCC;
border-radius:15px;
list-style:none;
float:left;
}
nav{ width:408px;}
.First + li {
border:1px solid #F00;
background-color:#FC0;
color:#F00;
font-weight:bold;
}
.Second ~ li {
border:1px solid #F00;
background-color:#FC0;
color:#F00;
font-weight:bold;
}

</style>
</head>

<body>
<nav>
<ul>
<li>1</li>
<li class="First">2</li>
<li>3</li>
<li>4</li>
<li>5</li>
<li>6</li>
<li>7</li>
<li>8</li>
</ul>
</nav>
<nav>
<ul>
<li>1</li>
<li class="Second">2</li>
<li>3</li>
<li>4</li>
<li>5</li>
```

```
<li>6</li>
<li>7</li>
<li>8</li>
</ul>
</nav>
</body>
</html>
```

2.9 群组选择器 (selector1,selector2,...,selectorN)

群组选择器是将具有相同样式的元素分组在一起，每个选择器之间使用逗号“,”隔开。

三、属性选择器

在 HTML 中，通过各种各样的属性，可以给元素增加许多附加信息。这样就可以通过属性的不同，指定具有特定属性的元素。

属性选择器在 CSS 2 中就被引入，其主要作用是对带有指定属性的 HTML 元素设置样式。使用 CSS 3 属性选择器，可以只指定元素的某个属性，也可以同时指定元素的某个属性和其对应的属性值。

3.1E[attr]

E[attr]属性选择器是 CSS 3 属性选择器中最简单的一种。

如果希望选择有某个属性的元素，而不论这个属性值是什么，就可以使用此属性选择器。

属性选择器可以指定一个属性，也可以指定多个属性。

案例：

示例 11-7：不指定属性值的选择器

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>不指定属性值的属性选择器</title>
<style type="text/css">
[id] {
font-size:12px;
color:#C00;
}
</style>
```

```
</head>

<body>
<header>
  <div id="News1">这是第一条新闻的标题</div>
  <div id="News1-c">这是第一条新闻的简介，主要说明了新闻内容。</div>
  <div id="News2">这是第二条新闻的标题</div>
  <div id="News2-c">这是第二条新闻的简介，主要说明了新闻内容。</div>
  <div id="News3">这是第三条新闻的标题</div>
  <div id="News3-c">这是第三条新闻的简介，主要说明了新闻内容。</div>
  <div id="News4">这是第四条新闻的标题</div>
  <div id="News4-c">这是第四条新闻的简介，主要说明了新闻内容。</div>
</header>
</body>
</html>
```

3. 2E[attr="value"]

E[attr="value"]选择器和 E[attr]选择器，从字面上就能很清楚的理解出来其区别。E[attr="value"]是指定了属性值“value”，而 E[attr]只是选择了有对应的属性，并没有明确指其对应的属性值“value”，这也是这两种选择器的最大区别之处。

E[attr="value"]选择器要求属性和属性值必须完全匹配。

案例：

示例 11-8：指定属性值的选择器

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>指定属性值的属性选择器</title>
<style type="text/css">
[id] {
  font-size:12px;
  color:#C00;
}
[id="News1"] {
  font-size:14px;
  color:#000;
}
</style>
</head>
```

```
<body>
<header>
  <div id="News1">这是第一条新闻的标题</div>
  <div id="News1-c">这是第一条新闻的简介，主要说明了新闻内容。</div>
  <div id="News2">这是第二条新闻的标题</div>
  <div id="News2-c">这是第二条新闻的简介，主要说明了新闻内容。</div>
  <div id="News3">这是第三条新闻的标题</div>
  <div id="News3-c">这是第三条新闻的简介，主要说明了新闻内容。</div>
  <div id="News4">这是第四条新闻的标题</div>
  <div id="News4-c">这是第四条新闻的简介，主要说明了新闻内容。</div>
</header>
</body>
</html>
```

3. 3E[attr~="value"]

E[attr~="value"]根据属性值中的词列表的某个词来进行选择元素。

此属性选择器要求属性值是一个或多个词列表，如果是列表时，多个词需要用空格隔开，只要元素的属性值中有一个词和 value 相匹配就可以选中该元素。

案例：

示例 11-9：根据属性值单词的选择器

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>根据属性值单词的选择器</title>
<style type="text/css">
[id] {
  font-size:12px;
  color:#C00;
}
[id="News1"] {
  font-size:14px;
  color:#000;
}
[id~="Content"] {
  color:#090;
}
</style>
```

```
</head>

<body>
<header>
  <div id="News1">这是第一条新闻的标题</div>
  <div id="News1-c">这是第一条新闻的简介，主要说明了新闻内容。</div>
  <div id="News2">这是第二条新闻的标题</div>
  <div id="News2-c News2C Content">这是第二条新闻的简介，主要说明了新闻内容。</div>
  <div id="News3">这是第三条新闻的标题</div>
  <div id="News3-c">这是第三条新闻的简介，主要说明了新闻内容。</div>
  <div id="News4">这是第四条新闻的标题</div>
  <div id="News4-c">这是第四条新闻的简介，主要说明了新闻内容。</div>
</header>
</body>
</html>
```

3.4 E[attr^="value"]

E[attr^="value"] 属性选择器，指的是选择 attr 属性值以“value”开头的元素，换句话说，选择的属性其以对应的属性值是以“value”开始的。

3.5 E[attr\$="value"]

E[attr\$="value"] 属性选择器，表示的是选择 attr 属性值以“value”结尾的所有元素，换句话说，选择的属性其以对应的属性值是以“value”结尾的。

3.6 E[attr*="value"]

E[attr*="value"] 属性选择器，表示的是选择 attr 属性值中包含子串“value”的所有元素。也就是说，只要所选择的属性，其属性值中有这个“value”值都将被选中。

案例：

示例 11-10：匹配属性值的选择器

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
```

```
<title>根据属性值单词的选择器</title>
<style type="text/css">
[id^="News"] {
  font-size:14px;
  color:#F30;
  line-height:180%
}
[id$="c"] {
  font-size:12px;
  color:#999;
}
[id*="3"] {
  text-decoration:underline;
}
</style>
</head>

<body>
<header>
  <div id="News1">这是第一条新闻的标题</div>
  <div id="News1-c">这是第一条新闻的简介，主要说明了新闻内容。</div>
  <div id="News2">这是第二条新闻的标题</div>
  <div id="News2-c">这是第二条新闻的简介，主要说明了新闻内容。</div>
  <div id="News3">这是第三条新闻的标题</div>
  <div id="News3-c">这是第三条新闻的简介，主要说明了新闻内容。</div>
  <div id="News4">这是第四条新闻的标题</div>
  <div id="News4-c">这是第四条新闻的简介，主要说明了新闻内容。</div>
</header>
</body>
</html>
```

3.7 E[attr|"value"]

E[attr|"value"]属性选择器，表示的是选择 attr 属性值等于“value”或者是“value-”开头的元素。

注意，attr 后面的是竖线“|”。

四、结构性伪类选择器

4.1 伪类选择器

在 CSS 中，可以使用类选择器把相同的元素定义成不同的样式。在前文的讲解中已经进行了介绍。

除了前文中介绍的类选择器之外，CSS 中还有伪类选择器。伪类选择器与类选择器的区别是，类选择器可以自由命名，而伪类选择器是 CSS 中已经定义好的选择器，不能够随便命名和定义。

最为常见的伪类选择器，是定义在 a 元素上的选择器，例如：
a:link、a:visited、a:hover、a:active。

案例：

示例 11-11：超链接的案例

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>超链接的案例</title>
<style type="text/css">
* {
margin:0px;
padding:0px;
}
ul {
margin:20px 0px;
height:30px;
background-color:#F09;
text-align:center;
}
li {
padding:10px 5px 2px 5px;
font-size:13px;
color:#FFF;
float:left;
list-style:none;
}
a:link,a:visited {
text-decoration:none;
color:#FFF;
}
a:hover,a:active {
text-decoration:none;
font-size:16px;
color:#FF9;
}
#TitleInfo {
padding-left:50px;
}
}
```

```

</style>
</head>

<body>
<nav>
  <ul>
    <li id="TitleInfo">请选择访问的网站: </li>
    <li><a href="http://www.sina.com.cn" target="_blank"> 新 浪 网
</a></li>
    <li><a href="http://www.sohu.com" target="_blank">搜狐</a></li>
    <li><a href="http://www.163.com" target="_blank">163</a></li>
    <li><a href="http://www.qq.com" target="_blank">QQ</a></li>
    <li><a href="http://www.baidu.com" target="_blank">百度</a></li>
  </ul>
</nav>
</body>
</html>

```

4.2 伪元素选择器

伪元素选择器，是指并不是针对真正的元素使用的选择器，而是针对 CSS 中已经定义好的伪元素使用的选择器。

其具体的用法是：

选择器:伪元素{属性:值}

伪元素选择器也可以与类配合使用，具体用法如下：

选择器 类名称:伪元素{属性:值}

在 CSS 中，主要有四个伪元素选择器。

(1) first-line

first-line 伪元素选择器用于为某个元素中的第一行文字使用样式。

案例：

示例 11-12: first-line 选择器

```

<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>first-line 选择器</title>
<style type="text/css">

```



```

p:first-line {
  font-size:16px;
  color:#F00;
}
</style>
</head>

<body>
<article>
  <h2>李克强抵灾区遇余震 今晚将挨家挨户敲门</h2>
  <p>据人民网芦山 4 月 20 日电 今天早上，四川省雅安市芦山县发生 7. 0
级地震后，中共中央政治局常委、国务院总理李克强等乘专机紧急飞赴灾区。
</p>
  <p>李克强飞抵四川后，转乘直升机飞赴灾区。李克强来到芦山县龙门乡政
府，在院子里现场办公了解救灾情况。刚刚摊开地图，就发生了一次余震。李
克强马上询问： " 这样的余震大概属于几级？震后发生的次数是否频繁？ "
</p>
  <p>据了解，由于这次地震是浅源地震，对房屋造成了结构性损坏，很多房
屋虽然没有倒塌，但已成为危房。李克强在现场办公时要求军队和武警部队今
晚进行 " 拉网式 " 排查，尽最大能力保证人员安全。 </p>
</article>
</body>
</html>

```

(2) first-letter

first-letter 伪元素选择器用于为某个元素中的文字的首字母使用样式。在英文中，首字母是第一个英文字符，在中文或日文中，首字母是第一个汉字。

案例：

示例 11-13: first-letter 选择器

```

<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>first-letter 选择器</title>
<style type="text/css">
p:first-letter {
  font-size:40px;
  color:#F00;
}
</style>
</head>

```

```

<body>
<article>
  <h2>李克强抵灾区遇余震 今晚将挨家挨户敲门</h2>
  <p>据人民网芦山 4 月 20 日电 今天早上，四川省雅安市芦山县发生 7. 0 级地震后，中共中央政治局常委、国务院总理李克强等乘专机紧急飞赴灾区。</p>
  <p>李克强飞抵四川后，转乘直升机飞赴灾区。李克强来到芦山县龙门乡政府，在院子里现场办公了解救灾情况。刚刚摊开地图，就发生了一次余震。李克强马上询问：" 这样的余震大概属于几级？震后发生的次数是否频繁？ " </p>
  <p>据了解，由于这次地震是浅源地震，对房屋造成了结构性损坏，很多房屋虽然没有倒塌，但已成为危房。李克强在现场办公时要求军队和武警部队今晚进行 " 拉网式 " 排查，尽最大能力保证人员安全。</p>
</article>
</body>
</html>

```

(3) before

before 伪元素选择器用于在某个元素之前插入一些内容。

案例：

示例 11-14: before 选择器

```

<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>before 选择器</title>
<style type="text/css">
li {
  height:20px;
  list-style:none;
  font-size:13.5px;
  vertical-align:middle;
  border-bottom:1px dashed #CCCCCC;
  padding-top:8px;
}
li:before {
  content: url(images/News-Icon.png);
  margin-right:10px;
}
</style>
</head>

```

```

<body>
<aside>
<h3>今日最热新闻</h3>
<ul>
  <li>伤亡 | 5000 余人伤 部分遇难者名单
  <li>灾情 | 宝兴食品仅够维持 2 天 进出道路坍塌
  <li>灾区 | 芦山县急需帐篷饮水 县城恢复电力
  <li>感人 | 儿抱父 1 小时苦候手术 姐为救弟受重伤
  <li>应对 | 李克强抵灾区遇余震 今晚将挨户敲门
  <li>现场 | 母亲救儿顶起百斤预制板 一车 4 人被埋
  <li>细节 | 老板娘免费发矿泉水 主播穿婚纱播报
  <li>目击 | 学生跳楼重伤 有人全裸逃跑
</ul>
</aside>
</body>
</html>

```

(4) after

after 伪元素选择器用于在某个元素之后插入一些内容。

案例：

示例 11-15：after 选择器

```

<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>before 选择器</title>
<style type="text/css">
li {
  height:20px;
  list-style:none;
  font-size:13.5px;
  vertical-align:middle;
  border-bottom:1px dashed #CCCCCC;
  padding-top:8px;
}
li:before {
  content: url(images/News-lcon.png);
  margin-right:10px;
}
li.NewContent:after {
  content:"(最新发布)";
  color:#900;
  font-size:12px;
}

```

```
padding-left:5px;
}
</style>
</head>

<body>
<aside>
<h3>今日最热新闻</h3>
<ul>
<li>伤亡 | 5000 余人伤 部分遇难者名单
<li>灾情 | 宝兴食品仅够维持 2 天 进出道路坍塌
<li>灾区 | 芦山县急需帐篷饮水 县城恢复电力
<li class="NewContent">感人 | 儿抱父 1 小时苦候手术 姐为救弟受重伤
<li>应对 | 李克强抵灾区遇余震 今晚将挨户敲门
<li>现场 | 母亲救儿顶起百斤预制板 一车 4 人被埋
<li>细节 | 老板娘免费发矿泉水 主播穿婚纱播报
<li>目击 | 学生跳楼重伤 有人全裸逃跑
</ul>
</aside>
</body>
</html>
```

4.3 结构性伪类选择器：root、not、empty、target

(1) root

root 选择器用于指定页面根元素的样式。所谓根元素，是指位于文档树中最顶层结构的元素，在 HTML 文档中就是指包含着整个页面的<html>部分。

案例：

示例 11-16：root 选择器

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>root 选择器</title>
<style type="text/css">
:root {
background-color:#B5F7E7;
}
body {
background-color:#FC3;
}
</style>
```

```
</style>
</head>

<body>
<h1>经典微博</h1>
<p>感谢小学同学、初中同学、高中同学、大学同学，感谢初中室友、高中室友、大学室友，是你们对我的关心和帮助，是你们对我的手下留情，让我依然活到现在。感谢你们，感谢你们的不杀之恩。</p>
<p>你所浪费的今天，是昨天死去的人奢望的明天。你所厌恶的现在，是未来的你回不去的曾经。</p>
<p>鸡蛋从外打破，是食物；从内打破，是生命。人生，从外打破，是压力；从内打破，是成长。</p>
<p>别人想什么，我们控制不了；别人做什么，我们也强求不了。唯一可以做的，就是尽心尽力做好自己的事，走自己的路，按自己的原则，好好生活。即使有人亏待了你，时间也不会亏待你，人生更加不会亏待你！</p>
</body>
</html>
```

(2) not

如果需要对某个结构元素使用样式，但是需要排除这个结构元素下的特定子结构元素，可以使用 not 选择器。

案例：

示例 11-17：not 选择器

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>not 选择器</title>
<style type="text/css">
:root {
  background-color:#B5F7E7;
}
body *:not(h1){
  background-color:#FC3;
}
</style>
</head>

<body>
<h1>经典微博</h1>
<p>感谢小学同学、初中同学、高中同学、大学同学，感谢初中室友、高中室友、大学室友，是你们对我的关心和帮助，是你们对我的手下留情，让我依然
```

```
活到现在。感谢你们，感谢你们的不杀之恩。</p>
<p>你所浪费的今天，是昨天死去的人奢望的明天。你所厌恶的现在，是未来的你回不去的曾经。</p>
<p>鸡蛋从外打破，是食物；从内打破，是生命。人生，从外打破，是压力；从内打破，是成长。</p>
<p>别人想什么，我们控制不了；别人做什么，我们也强求不了。唯一可以做的，就是尽心尽力做好自己的事，走自己的路，按自己的原则，好好生活。即使有人亏待了你，时间也不会亏待你，人生更加不会亏待你！</p>
</body>
</html>
```

(3) empty

empty 选择器用来指定当元素内容为空时使用的样式。

案例：

示例 11-18: empty 选择器

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>empty 选择器</title>
<style type="text/css">
:root {
  background-color:#B5F7E7;
}
:empty {
  height:30px;
  background-color:#FC3;
}
</style>
</head>

<body>
<h1>经典微博</h1>
<p>感谢小学同学、初中同学、高中同学、大学同学，感谢初中室友、高中室友、大学室友，是你们对我的关心和帮助，是你们对我的手下留情，让我依然活到现在。感谢你们，感谢你们的不杀之恩。</p>
<p></p>
<p>你所浪费的今天，是昨天死去的人奢望的明天。你所厌恶的现在，是未来的你回不去的曾经。</p>
<p>鸡蛋从外打破，是食物；从内打破，是生命。人生，从外打破，是压力；从内打破，是成长。</p>
<p>别人想什么，我们控制不了；别人做什么，我们也强求不了。唯一可以做的，就是尽心尽力做好自己的事，走自己的路，按自己的原则，好好生活。即
```

```
使有人亏待了你，时间也不会亏待你，人生更加不会亏待你！ </p>
</body>
</html>
```

(4) target

target 选择器用来指定对页面中某个 target 指定样式，该选择器定义的样式在用户点击页面中的超链接并且跳转后方起作用。

案例：

示例 11-19：target 选择器

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>target 选择器</title>
<style type="text/css">
div {
margin:5px;
width:300px;
height:300px;
background-color:#CCC;
float:left;
}
:target {
background-color:#F00;
}
</style>
</head>

<body>
<nav>
<ul>
<li id="TitleInfo">请选择访问的网站： </li>
<li><a href="#PartA">区块 A</a></li>
<li><a href="#PartB">区块 B</a></li>
<li><a href="#PartC">区块 C</a></li>
</ul>
</nav>
<div id="PartA"></div>
<div id="PartB"></div>
<div id="PartC"></div>
</body>
</html>
```

4.4 CSS 3 的 nth 选择器

nth 选择器是 CSS 3 的最新内容，也被称为 CSS 3 结构类。

(1) :first-child

:first-child 是用来选择某个元素的第一个子元素。

(2) :last-child

:last-child 选择器与 :first-child 选择器的作用类似，:last-child 选择的是元素的最后一个子元素。

案例：

示例 11-20：改变列表第一项和最后一项

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>改变列表第一项和最后一项</title>
<style type="text/css">
li {
  height:20px;
  list-style:none;
  font-size:13.5px;
  vertical-align:middle;
  border-bottom:1px dashed #CCCCCC;
  padding-top:8px;
}
li:before {
  content: url(images/News-Icon.png);
  margin-right:10px;
}
li:first-child {
  color:#090;
}
li:last-child {
  color:#06F;
}
</style>
</head>

<body>
<aside>
<h3>今日最热新闻</h3>
<ul>
  <li>伤亡 | 5000 余人伤 部分遇难者名单
```



```
<li>灾情 | 宝兴食品仅够维持 2 天 进出道路坍塌
<li>灾区 | 芦山县急需帐篷饮水 县城恢复电力
<li>感人 | 儿抱父 1 小时苦候手术 姐为救弟受重伤
<li>应对 | 李克强抵灾区遇余震 今晚将挨户敲门
<li>现场 | 母亲救儿顶起百斤预制板 一车 4 人被埋
<li>细节 | 老板娘免费发矿泉水 主播穿婚纱播报
<li>目击 | 学生跳楼重伤 有人全裸逃跑
</ul>
</aside>
</body>
</html>
```

(3) :nth-child()

:nth-child() 可以选择某个的一个或多个特定的子元素。其常用的形式如下:

:nth-child(length)	参数是具体数字
:nth-child(n)	参数是 n,n 从 0 开始计算
:nth-child(n*length)	n 的倍数选择, n 从 0 开始计算
:nth-child(n+length)	选择大于 length 后面的元素
:nth-child(-n+length)	选择小于 length 前面的元素
:nth-child(n*length+1)	表示隔几选一

:nth-child() 可以定义值, 值可以是整数, 也可以是表达式。其中 n 字母是固定的, 如果换做其他字母, 就没有效果。

案例:

示例 11-21: 选择偶数项

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>选择偶数项</title>
<style type="text/css">
li {
  height:20px;
  list-style:none;
  font-size:13.5px;
  vertical-align:middle;
  border-bottom:1px dashed #CCCCCC;
  padding-top:8px;
}
li:before {
  content: url(images/News-lcon.png);
```

```
margin-right:10px;
}
li:nth-child(2n) {
  color:#039;
}
</style>
</head>

<body>
<aside>
<h3>今日最热新闻</h3>
<ul>
  <li>伤亡 | 5000 余人伤 部分遇难者名单
  <li>灾情 | 宝兴食品仅够维持 2 天 进出道路坍塌
  <li>灾区 | 芦山县急需帐篷饮水 县城恢复电力
  <li>感人 | 儿抱父 1 小时苦候手术 姐为救弟受重伤
  <li>应对 | 李克强抵灾区遇余震 今晚将挨户敲门
  <li>现场 | 母亲救儿顶起百斤预制板 一车 4 人被埋
  <li>细节 | 老板娘免费发矿泉水 主播穿婚纱播报
  <li>目击 | 学生跳楼重伤 有人全裸逃跑
</ul>
</aside>
</body>
</html>
```

(4) :nth-last-child()

:nth-last-child() 选择器和前面的:nth-child() 相似，该选择器在进行选择时是从最后一个元素开始计算来选择特定元素。

案例：

示例 11-22：每隔 2 项选择一项

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>每个 2 项选择一项</title>
<style type="text/css">
li {
  height:20px;
  list-style:none;
  font-size:13.5px;
  vertical-align:middle;
  border-bottom:1px dashed #CCCCCC;
```

```
padding-top:8px;
}
li:before {
  content: url(images/News-lcon.png);
  margin-right:10px;
}
li:nth-last-child(3n+1) {
  color:#039;
}
</style>
</head>

<body>
<aside>
<h3>今日最热新闻</h3>
<ul>
  <li>伤亡 | 5000 余人伤 部分遇难者名单
  <li>灾情 | 宝兴食品仅够维持 2 天 进出道路坍塌
  <li>灾区 | 芦山县急需帐篷饮水 县城恢复电力
  <li>感人 | 儿抱父 1 小时苦候手术 姐为救弟受重伤
  <li>应对 | 李克强抵灾区遇余震 今晚将挨户敲门
  <li>现场 | 母亲救儿顶起百斤预制板 一车 4 人被埋
  <li>细节 | 老板娘免费发矿泉水 主播穿婚纱播报
  <li>目击 | 学生跳楼重伤 有人全裸逃跑
</ul>
</aside>
</body>
</html>
```

(5) :nth-of-type()

:nth-of-type() 选择器类似于:nth-child() 选择器,不同的是仅计算选择器中指定的元素类型。

(6) :nth-last-of-type()

:nth-last-of-type() 选择器和:nth-of-type() 相似,该选择器在进行选择时是从最后一个元素开始计算来选择特定元素。

案例:

示例 11-23: 改变新闻标题的样式

```
<!doctype html>
<html>
<head>
```

```

<meta charset="utf-8">
<title>每个 2 项选择一项</title>
<style type="text/css">
li {
  height:20px;
  list-style:none;
  font-size:13.5px;
  vertical-align:middle;
  border-bottom:1px dashed #CCCCCC;
  padding-top:8px;
}
li:before {
  content: url(images/News-Icon.png);
  margin-right:10px;
}
/*
h3:nth-child(2n) {
  color:#F00;
}
h3:nth-child(2n+1) {
  color:#F0F;
}
*/
h3:nth-of-type(odd) {
  color:#F00;
}
h3:nth-of-type(even) {
  color:#F0F;
}

</style>
</head>

<body>
<aside>
  <h3>国内新闻</h3>
  <ul>
    <li>中国慈善联合会成立 民政部长当选为首任会长
    <li>三大医保今年收入或超万亿元 人社部卫计委争夺
    <li>广州一村支书被曝身家超 10 亿元 全家移民加拿大
  </ul>
  <h3>国际新闻</h3>
  <ul>
    <li>波士顿爆炸案嫌犯在逃期间发推特：大家保重
    <li>俄罗斯议员要求除吉尔吉斯斯坦以湖抵债

```

```
<li>美国得州化肥厂爆炸后 60 人仍下落不明 200 人受伤
<li>埃及穆兄会支持者与反对者爆发冲突致 60 人受伤
</ul>
<h3>社会新闻</h3>
<ul>
<li>老人家中灌装煤气泄漏 四扇窗户被炸飞三扇
<li>8 名女子组团背孩子偷衣服 毯子里塞进 40 件衬衫
<li>朱令铨中毒案匿名黑客：希望推动该案调查重启
</ul>
<h3>军事新闻</h3>
<ul>
<li>成都军区 2000 人和 2 架直升机已赶往雅安抗震
<li>朝鲜称核武器为朝韩共同资产 对韩态度转变
<li>解放军苏 30 撞鸟发动机爆炸 飞行员勉强着陆
<li>日敲定自卫队法修正案 允许海外动用装甲车
</ul>
</aside>
</body>
</html>
```

(7) :first-of-type 和 :last-of-type

:first-of-type 和 :last-of-type 选择器和 :first-child 和 :last-child 选择器相似,不同之处在于指定了元素类型进行计算。

(8) :only-child 和 :only-of-type

:only-child 选择器用来指定一个元素是它的父元素的唯一元素。

:only-of-type 选择器用来指定指定类型的一个元素是它的父元素的唯一元素。

案例:

示例 11-24: 唯一元素的样式变化

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>唯一元素的样式变化</title>
<style type="text/css">
li {
height:20px;
list-style:none;
font-size:13.5px;
```

```
vertical-align:middle;
border-bottom:1px dashed #CCCCCC;
padding-top:8px;
}
li:before {
content: url(images/News-Icon.png);
margin-right:10px;
}
li:only-child {
color:#0C0;
}
</style>
</head>

<body>
<aside>
<h3>国内新闻</h3>
<ul>
<li>中国慈善联合会成立 民政部长当选为首任会长
<li>三大医保今年收入或超万亿元 人社部卫计委争夺
<li>广州一村支书被曝身家超 10 亿元 全家移民加拿大
</ul>
<h3>国际新闻</h3>
<ul>
<li>波士顿爆炸案嫌犯在逃期间发推特：大家保重
</ul>
<h3>社会新闻</h3>
<ul>
<li>老人家中灌装煤气泄漏 四扇窗户被炸飞三扇
</ul>
<h3>军事新闻</h3>
<ul>
<li>成都军区 2000 人和 2 架直升机已赶往雅安抗震
<li>日敲定自卫队法修正案 允许海外动用装甲车
</ul>
</aside>
</body>
</html>
```

五、UI 元素状态伪类选择器

在 CSS 3 的选择器中，除了结构性伪类选择器外，还有 UI 元素状态伪类选择器。这类选择器指定的样式只有当元素处于某种状态下

才起作用，在默认状态下是不起作用的。

在CSS 3中，共有11种UI元素状态伪类选择器，分别是E:hover、E:active、E:focus、E:enabled、E:disabled、E:read-only、E:read-write、E:checked、E:default、E:indeterminate、E::selection。

(1) E:hover、E:active、E:focus

E:hover 选择器用来指定当鼠标指针移动到元素上面时元素所使用的样式。

E:active 选择器用来指定当元素被激活的样式，例如鼠标在元素上按下且没有松开时。

E:focus 选择器用来指定元素获得光标焦点时的样式，主要是在文本框控件获得焦点并进行文字输入的使用使用。

案例：

示例 11-25：有提示效果的输入框

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>有提示效果的输入框</title>
<style type="text/css">
body {
  margin:50px;
  font-size:12px;
}
input {
  width:200px;
  height:22px;
  border:1px solid #CCC;
}
input:focus {
  border:1px solid #F00;
  background-color:#FFC;
  color:#F00;
}
</style>
</head>

<body>
<form method="post" action="11-form.php" name="UserLogin">
```

```
<div><label>用户名: </label><input type="text" id="UserName">
</div>
</form>
</body>
</html>
```

(2) E:enabled、E:disabled

E:enabled 选择器用来指定当元素处于可用状态时的样式。

E:disabled 选择器用来指定当元素处于不可用状态时的样式。

(3) E:read-only、E:read-write

E:read-only 选择器用来指定当元素处于只读状态时的样式。

E:read-write 选择器用来指定当元素处于非只读状态时的样式。

(4) E:checked、E:default、E:indeterminate

E:checked 选择器用来指定当表单中的 radio 元素或 checkbox 元素处于选取状态下的样式。

E:default 选择器用来指定当页面打开时默认处于选取状态的单选框或复选框控件的元素样式。

E:indeterminate 选择器用来指定当页面打开时，如果一组单选框中任何一个单选框都没有被选中时，整组单选框的统一样式。

(5) E::selection

E::selection 选择器用来指定当元素处于选中状态时的样式。

六、案例：数据表格

案例：

示例 11-26：数据表格

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>数据表格</title>
<style type="text/css">
body {
```



```

margin:30px auto;
text-align:center;
font-size:13px;
color:#666;
}
h1 {
margin:10px;
font-size:18px;
color:#F30;
font-family:微软雅黑, 黑体;
letter-spacing:3px;
font-weight:600;
}
table {
margin:0px auto;
border-collapse:collapse;
}
tr {
height:30px;
}
tr:nth-of-type(odd) {
background-color:#FFF;
}
tr:nth-of-type(even) {
background-color:#F2F2F2;
}
tr:hover {
color:#F30;
background-color:#EFFEED;
}
tr:hover td:nth-child(1){
background:url(images/tr-hover-bg.png) no-repeat 3px 8px;
}
th {
border-bottom:1px solid #CCC;
border-top:1px solid #CCC;
color:#333;
background-color:#F8F4DA;
}
td {
border-bottom:1px solid #CCC;
}
/*定义列的宽度*/
[id^="Student"] {
width:100px;
}

```

```

[id="StudentAddress"] {
  width:180px;
}
[id="StudentName"],[id="StudentSex"] {
  width:60px;
}
[id="StuAddress"] {
  text-align:left;
  text-indent:3px;
}
</style>
</head>

<body>
<h1>学生基本信息表</h1>
<table>
  <tr>
    <th id="StudentID">学号</th>
    <th id="StudentName">姓名</th>
    <th id="StudentSex">性别</th>
    <th id="StudentBirthday">出生日期</th>
    <th id="StudentAddress">籍贯</th>
    <th id="StudentPhone">电话</th>
  </tr>
  <tr>
    <td>2012001001</td>
    <td>张德文</td>
    <td>男</td>
    <td>1992-10-18</td>
    <td id="StuAddress">河南省洛阳市伊川县</td>
    <td>18601010001</td>
  </tr>
  <tr>
    <td>2012001002</td>
    <td>李国强</td>
    <td>男</td>
    <td>1991-03-12</td>
    <td id="StuAddress">河南省郑州市</td>
    <td>18601010002</td>
  </tr>
  <tr>
    <td>2012001003</td>
    <td>许盈盈</td>
    <td>女</td>
    <td>1992-12-25</td>
    <td id="StuAddress">河南省南阳市</td>
  </tr>

```

```

        <td>18601010001</td>
</tr>
<tr>
    <td>2012001004</td>
    <td>刘冬冬</td>
    <td>男</td>
    <td>1991-04-09</td>
    <td id="StuAddress">河南省焦作市沁阳市</td>
    <td>18601010004</td>
</tr>
<tr>
    <td>2012001005</td>
    <td>潘靓颖</td>
    <td>女</td>
    <td>1991-10-22</td>
    <td id="StuAddress">河南省新乡市</td>
    <td>18601010005</td>
</tr>
<tr>
    <td>2012001006</td>
    <td>朱德刚</td>
    <td>男</td>
    <td>1992-01-16</td>
    <td id="StuAddress">河南省许昌市许昌县</td>
    <td>18601010041</td>
</tr>
<tr>
    <td>2012001007</td>
    <td>刘娟</td>
    <td>女</td>
    <td>1991-03-06</td>
    <td id="StuAddress">河南省信阳市</td>
    <td>18601010012</td>
</tr>
<tr>
    <td>2012001008</td>
    <td>刘哲</td>
    <td>男</td>
    <td>1991-09-13</td>
    <td id="StuAddress">河南省驻马店市</td>
    <td>18601010123</td>
</tr>
<tr>
    <td>2012001009</td>
    <td>赵晓飞</td>
    <td>男</td>

```

```
<td>1992-07-11</td>
<td id="StuAddress">河南省许昌市襄城县</td>
<td>18601011232</td>
</tr>
<tr>
<td>2012001010</td>
<td>冯娜娜</td>
<td>女</td>
<td>1991-12-28</td>
<td id="StuAddress">河南省焦作市</td>
<td>18601017878</td>
</tr>
</table>
</body>
</html>
```

七、讨论与思考

7.1 选择器的浏览器支持

(1) 选择器的浏览器支持情况如何？

选择器的支持情况：<http://kimblim.dk/css-tests/selectors/>

(2) 如何提高选择器的浏览器兼容性？

7.2 几个使用选择器可以实现的效果

(1) 浏览器中文字选中时反显得颜色为红色。

(2) 鼠标指向可以变换变换颜色的导航按钮。

(3) 不同状态的分页按钮。

7.3 选择器的优化

(1) 浏览器如何识别选择器？

(2) CSS 选择器的效率如何？不同类型的选择器，效率的排序是什么？

(3) 如何书写高效率的选择器？有哪些应该遵循的原则？