**华山论剑：CSS选择器之基础选择器**

我们将介绍一下 CSS 里选择器的用法。CSS 选择器是用来指定该组 CSS 样式会对什么元素生效的，是连接 HTML 结构和 CSS 样式的桥梁。这部分内容总共分为三大块：

1. 首先会讲一下基础选择器。这一节里介绍的都是单一的某种选择器的用法，同时会对使用上给出一些建议。
2. 第二块将会讲解复合选择器。复合选择器是对基础选择器做各种方式的组合，从而实现更复杂的功能。因为基础选择器功能比较有限，所以在实际开发中复合选择器用的是最多的。
3. 第三块内容是伪类和伪元素选择器。这两种选择器也是对基础选择器功能的扩充，但不同的是它不是以选择器组合的形式来实现的。

这一节我们先进行基础选择器内容的介绍。**基础选择器包括 ID 选择器、类选择器、标签选择器、通配符选择器和属性选择器**这几种。

## **ID 选择器**

ID 选择器是用 “#” 号加 ID 名称 xxx 来表示，用来选择 HTML 中 id="xxx" 的 DOM 元素。我们以选择下面这个 ID 为 content 的元素为例：

<div id="content">我是id选择器需要选中的元素</div>

当我们想把样式应用到这个元素的时候，就可以用下面的 ID 选择器：

#content{

color: #fff;

background: #000;

}

这样 ID 为 content 的元素就会有一个黑底白字的效果了。在 ID 选择器中，有几点要注意。

****Tips：****  
1、ID 选择器只能对一个元素生效，同一个页面里不允许出现两个 ID 相同的元素。  
2、理论上 ID 选择器是效率最高的选择器。但是由于它只能选一个元素，特异性太高，在实际开发中也很少在 CSS 里使用 ID 选择器。  
3、也正是因为 ID 选择器特异性高，所以在 JS 里使用 ID 选择器的比较常见。

## **类选择器**

类选择器是用 “.” 加上 class 名称来表示，用来选择 HTML 中 class="xxx" 的 DOM 元素。我们以选择下面 class 名称为 list-item 的元素为例：

<ul>

<li class="list-item"></li>

<li class="list-item"></li>

<li class="list-item"></li>

<li class="list-item"></li>

</ul>

当我们想把样式应用到列表里每一条元素的时候，就可以用类选择器：

.list-item{

border-bottom: 1px solid #ccc;

}

这样列表里所有的项就都有一个宽 1px 灰色的下边框了。

在类选择器中，要注意以下几点：

****Tips：****  
1、类选择器的效率也是不错的，和 ID 选择器并不会有太大的差异。所以在写 CSS 的时候，比较推荐用类选择器。  
2、类选择器会选择到所有类名相同的 DOM 元素，没有数量限制。  
3、类选择器应该是样式开发中应用最多的选择器。

## **通配选择器**

通配选择器使用星号来选择到页面里所有元素。用法如下：

\*{

margin: 0;

padding: 0;

}

上面这个样式就是把所有元素的内外边距都归零。由于通配选择器要把样式覆盖到所有的元素上，可想而知它的效率并不会高，所以在实际开发中一般不建议使用通配选择器。如果只是做一个demo，你会发现好多人都在用通配符，因为本地运行速度是非常快的。实际开发中经常会直接对html、body等标签设置标签选择器样式：这也叫做浏览器CSS初始化。

## **标签选择器**

标签选择器的作用是选中 HTML 中某一种类的标签，它直接使用 HTML 中的标签名作为选择器的名称。比如我们需要把页面里所有大标题的字号都调成 20px，就可以用标签选择器来实现：

h1{

font-size: 20px;

}

****Tips：**** 标签选择器通常用来重置某些标签的样式，标签选择器的效率也不是很高，但要好过通配选择器。

## **属性选择器**

属性选择器比较好理解，就是通过 DOM 的属性来选择该 DOM 节点。属性选择器是用中括号 “[]” 包裹，比如选择所有带有 href 属性的标签，就可以使用这样的选择器:

a[href]{

color: red;

}

这条选择器就可以让所有带 href 属性的 a 标签字体都变成红色。

属性选择器有如下几种形式：

* ****[attr]****，用来选择带有 attr 属性的元素，如刚提到的 a [href]。

<!-- HTML： -->

<a href="/">返回主页</a>

// 下面的CSS会使所有带href的a标签字体变红色：

a[href]{

color: red;

}

* ****[attr=xxx]****，用来选择有 attr 属性且属性值等于 xxx 的元素，如选择所有文本类型的输入框，可以用 input [type=text]。

<!-- HTML： -->

<input type="text" value="大花碗里扣个大花活蛤蟆"/>

// CSS：

input[type=text]{

color: red;

}

这个选择器里面要注意，xxx 和 HTML 中的属性值必须完全相等才会生效。

<!-- HTML： -->

<input class="input text" type="text" value="大花碗里扣个大花活蛤蟆"/>

// CSS：

input[class=input]{

color: red;

}

上面例子中 input [class=input] 的选择器并不能选中 class=“input text” 的元素，如果非要用这种选择器，必须使用 input [class=“input text”] 才可以。

* ****[attr~=xxx]****，这个选择器中间用了～=，选择属性值中包含 xxx 的元素，但一定是逗号分隔的多个值中有一个能和 xxx 相等才行。

<!-- HTML： -->

<input class="input text" type="text" value="大花碗里扣个大花活蛤蟆"/>

// CSS：

input[class~=input]{

color: red;

}

在上面的例子中，使用 input [class~=input] 就可以选中 class=“input text” 的元素了。

* ****[attr|=xxx]****，这个选择器是用来选择属性值为 xxx 或 xxx- 开头的元素，比较常用的场景是选择某一类的属性。

<!-- HTML： -->

<div class="article">我是article</div>

<div class="article-title">我是article-title</div>

<div class="article-content">我是article-content</div>

<div class="article\_footer">我是article\_footer，不是以artical-开头的</div>

// CSS：

div[class|=article]{

color: red;

}

上面的选择器就可以对所有 article 开头的元素生效，包括 class=“article” 的元素。上面的例子中，选择器会对前三条都生效，但不会对第四条生效。

* ****[attr^=xxx]****，这个选择器会匹配以 xxx 开头的元素，实际上就是用正则去匹配属性值，只要是以 xxx 开头都可以。

<!-- HTML： -->

<div class="article">我是article</div>

<div class="article-title">我是article-title</div>

<div class="article-content">我是article-content</div>

<div class="article\_footer">我是article\_footer，不是以artical-开头的</div>

// CSS：

div[class^=article]{

color: red;

}

还是用刚才的例子，如果把选择器换成 div [class^=article]，那么上面四个 HTML 元素都会被选中了。

* ****[attr$=xxx]****，这个选择器和上一个相似，它是用正则匹配的方式来选择属性值以 xxx 结尾的元素。

<!-- HTML： -->

<button class="btn btn-disabled">禁用的按钮</button>

<select class="select select-disabled city-select"></select>

<input class="input input-disabled" value="禁用的输入框"/>

// CSS：

[class$=disabled]{

display: none;

}

上面的例子中，我们想把页面里有禁用标识的所有元素都隐藏掉，就可以使用 [class$=disabled] 来选择所有 class 里以 disabled 结尾的元素。这么用的前提是提前约定好，disabled 相关的类要放在最后，否则就像上面的 select 一样不会生效。

* ****[attr\*=xxx]****，最后一个，这个是用正则匹配的方式来选择属性值中包含 xxx 字符的所有元素。这个选择器的规则算是最宽泛的，只要 xxx 是元素属性值的子字符串，这个选择器就会生效。

<!-- HTML： -->

<button class="btn btn-disabled">禁用的按钮</button>

<select class="select select-disabled city-select"></select>

<input class="input input-disabled" value="禁用的输入框"/>

// CSS：

[class\*=disabled]{

display: none;

}

还是用刚才 disable 的例子，如果我们用 [class\*=disabled] 选择器来选择 disabled 元素，就可以不考虑 -disable 属性所在的位置了，它对所有带这个单词的属性都会生效，哪怕是 class=“i-am-a-disabled-element” 的元素都可以。

****Tips：****

1. 属性选择器要做文本的匹配，所以效率也不会高。  
2. 在使用属性选择器时，尽量要给它设置上生效的范围，如果只用了个 [href] 相当于要在所有元素里找带 href 的元素，效率会很低。如果用 a [href] 会好的多，如果用 .link [href] 就更好了。这种组合方式我们在下一节讲解。  
3. 属性选择器很灵活，如果能使用 CSS 代替 JS 解决一些需求，可以不用太纠结性能的问题，用 JS 实现也一样要耗费资源的。