

css选择器

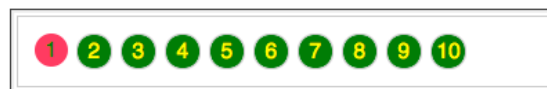
1. 相邻兄弟选择器

七、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，而且他们具有一个相同的父元素，换句话说，EF两元素具有一个相同的父元素，而且F元素在E元素后面，而且相邻，这样我们就可以使用相邻兄弟元素选择器来选择F元素。

```
li + li {background: green;color: yellow; border: 1px solid #ccc;}
```

上面代码表示选择li的相邻元素li，我们这里一共有十个li，那么上面的代码选择了从第2个li到10个li，一共九个，请看效果：



因为上面的li+li其中第二li是第一li的相邻元素，第三个又是第二个相邻元素，因此第三个也被选择，依此类推，所以后面九个li都被选中了，如果我们换过一种方式来看，可能会更好的理解一点：

```
.active + li {background: green;color: yellow; border: 1px solid #ccc;}
```

按照前面所讲的知识，这句代码很明显选择了li.active后面相邻的li元素，注意了和li.active后面相邻的元素仅只有一个的。如图：



IE6不支持这个选择器

2. 通用兄弟选择器

八、通用兄弟选择器 (E ~ F)

通用兄弟元素选择器是CSS3新增加一种选择器，这种选择器将选择某元素后面的所有兄弟元素，他们也和相邻兄弟元素类似，需要在同一个父元素之中，换句话说，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F元素。比如下面的代码：

```
.active ~ li {background: green;color: yellow; border: 1px solid #ccc;}
```

上面的代码所表示的是，选择中了li.active 元素后面的所有兄弟元素li,如图所示：



3. 属性选择器

- **E[attr]** : 只使用属性名, 但没有确定任何属性值; E[attr="value"]能帮我们定位不同类型的元素, 特别是表单的form元素的操作, 比如input[type="text"]
- **E[attr="value"]** : 指定属性名, 并指定了该属性的属性值; 多属性选择器: E[attr="value1"] [attr="value2"]
- **E[attr~="value"]** : 指定属性名, 并且具有属性值, 此属性值是一个词列表, 并且以空格隔开, 其中词列表中包含了一个value词, 而且等号前面的“~”不能不写;
- **E[attr^="value"]** : 指定了属性名, 并且有属性值, 属性值是以value开头的; 能在网站中帮助我们匹配不同类型的文件, 可以给".doc", ".pdf", ".png" ".ppt"配置不同的icon(字体图标)
- **E[attr\$="value"]** : 指定了属性名, 并且有属性值, 而且属性值是以value结束的;
- **E[attr*="value"]** : 指定了属性名, 并且有属性值, 而且属性值中包含value;
- **E[atr|= "value"]** : 指定了属性名, 并且属性值是value或者以“value-”开头的值 (比如说zh-cn);

对于E[attr="value"]这种属性值选择器有一点需要注意: 属性和属性值必须完全匹配, 特别是对于属性值是词列表的形式时, 如:

```
<a href="" class="links item" title="open the website">?</a>
```

例如上面的代码, 如果你写成:

```
.demo a[class="links"] {color:red}; /*这是一种写法不能和上面的html所匹配*/
```

上面的属性选择器并不会和上在的html匹配, 因为他们的属性和属性值没有完全匹配, 需要改成如下所示的代码, 才能正确匹配:

```
.demo a[class="links item"] {color:red}; /*这样才是匹配的, 记得中间的空格不能少的哟*/
```

三、E[attr~="value"]

如果你想根据属性值中的词列表的某个词来进行选择元素, 那么就需要使用这种属性选择器: E[attr~="value"], 这种属性选择器是属性值是一个或多个词列表, 如果是列表时, 他们需要用空格隔开, 只要属性值中有一个value相匹配就可以选中该元素, 而我们前面所讲的E[attr="value"]是属性值需要完全匹配才会被选中, 他们两者区别就是一个有“~”号, 一个没有“~”号。我们来看一个这方面的实例:

```
.demo a[title~="website"] {background:orange;color:green;}
```

上面代码表示的是, div.demo下的a元素的title属性中, 只要其属性值中含有“website”这个词就会被选择, 回头看看我们的html, 不难发现所有a元素中“2, 6, 7, 8”这四个a元素的title中都含有, 所以被选中, 请看效果:



如果我们在上面的代码中, 把那个“~”号省去, 大家看看他们不同之处:

```
.demo a[title="website"] {background:orange;color:green;}
```

这样将不会选中任何元素, 因为在所有a元素中无法找到完全匹配的“title='website'”, 换句话说就没有选中任何元素, 效果如下:



这个实例再次证明了E[attr="value"]和E[attr~="value"]之间的区别, 和其中“~”所取的作用, 我总结了一句话: **属性选择器中有波浪(~)时属性值有value时就相匹配, 没有波浪(~)时属性值要完全是value时才匹配。**

IE6不支持E[attr~="value"]属性选择器。

七、E[attr]="value"]

E[attr]="value"]是属性选择器中的最后一种，在说这个选择器使用之前先提醒大家attr后面的是一个竖线“|”而不是!，小心搞错了。E[attr]="value"]被称作为特定属性选择器。这个选择器会选择attr属性值等于value或以value-开头的所有元素。我们来看个实例：

```
.demo a[lang|="zh"] {background:gray;color:yellow;}
```

上面的代码会选中了div.demo中lang属性等于zh或以zh-开头的所有a元素，大家可以对照前面的html代码，其中“2,3,4,6”被选中，因为他们都有一个lang属性，并且他们的属性值都符合以“zh”或“zh-”开始的元素。具体效果如下：



所以这种属性选择器用来匹配以“value-1”,“value-2”的属性是很方便的，比如说你页面中有很多图片，图片文件名都是以“figure-1”,“figure-2”这样的方式来命名的，那么使用这种选择器选中图片就方便了，大家可以在本地尝试一下，这种属性选择器最常常用的地方是如上面的示例用来匹配语言。

4. 伪类选择器

二、UI元素状态伪类

我们把“:enabled”,“:disabled”,“:checked”伪类称为UI元素状态伪类，这些主要是针对于HTML中的Form元素操作，最常见的比如我们“type=“text”有enable和disabled两种状态，前者为可写状态后者为不可状态；另

外“type=“radio”和“type=“checkbox”有“checked”和“unchecked”两种状态。来看两个实例，比如说你想将“disabled”的文本框与别的文本框区别出来，你就可以这样应用

```
input[type="text"]:disabled {border:1px solid #999;background-color: #fefefe;}
```

伪元素选择器：

::first-line选择元素的第一行，比如说改变每个段落的第一行文本的样式，我们就可以使用这个

```
p::first-line {font-weight:bold;}
```

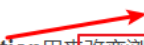
::first-letter选择文本块的第一个字母，除非在同一行里面包含一些其它元素，不过这个主要运用于段落排版上多，比如说首字下沉，

```
p::first-letter {font-size: 56px;float:left;margin-right:3px;}
```

::before和**::after**这两个主要用来给元素的前面或后面插入内容，这两个常用"content"配合使用，见过最多的就是清除浮动，

```
.clearfix:before,
.clearfix:after {
  content: ".";
  display: block;
  height: 0;
  visibility: hidden;
}
.clearfix:after {clear: both;}
.clearfix {zoom: 1;}
```

当然可以利用他们制作出其他更好的效果，比如本站首面的那个三角纸张效果，也是通过这个来实现的。

 **::selection**用来改变浏览网页选中文字的默认效果，有关于“::selection”使用，大家可以点击[《CSS ::Selection》](#)。