你也许已经掌握了id、class、后台选择器这些基本的css选择器。但这远远不是css的全部。下面向大家系统的解析css中30个最常用的选择器，包括我们最头痛的浏览器兼容性问题。掌握了它们，才能真正领略css的巨大灵活性。

**1. \***

1. \* {      margin: 0;      padding: 0;     }

复制代码

星状选择符会在页面上的每一个元素上起作用。web设计者经常用它将页面中所有元素的margin和padding设置为0。 \*选择符也可以在子选择器中使用。

1. #container \* { border: 1px solid black; }

复制代码

上面的代码中会应用于id为container元素的所有子元素中。 除非必要，我不建议在页面中过的的使用星状选择符，因为他的作用域太大，相当耗浏览器资源。 兼容浏览器：IE6+、Firefox、Chrome、Safari、Opera  
  
**2. #X**

1. #container { width: 960px; margin: auto; }

复制代码

井号作用域有相应id的元素。id是我们最常用的css选择器之一。id选择器的优势是精准，高优先级(优先级基数为100，远高于class的10)， 作为javascript脚本钩子的不二选择，同样缺点也很明显优先级过高，重用性差，所以在使用id选择器前，我们最好问下自己，真的到了非用id选择 器的地步? 兼容浏览器：IE6+、Firefox、Chrome、Safari、Opera  
  
**3. .X**

1. .error { color: red; }

复制代码

这是一个class(类)选择器。class选择器与id选择器的不同是class选择器能作用于期望样式化的一组元素。 兼容浏览器：IE6+、Firefox、Chrome、Safari、Opera  
  
**4. X Y**

1. li a { text-decoration: none; }

复制代码

这也是我们最常用的一种选择器——后代选择器。用于选取X元素下子元素Y，要留意的点是，这种方式的选择器将选取其下所有匹配的子元素，无视层级，所以有 的情况是不宜使用的，比如上述的代码去掉li下的所有a的下划线，但li里面还有个ul，我不希望ul下的li的a去掉下划线。使用此后代选择器的时候要 考虑是否希望某样式对所有子孙元素都起作用。这种后代选择器还有个作用，就是创建类似命名空间的作用。比如上述代码样式的作用域明显为li。 兼容浏览器：IE6+、Firefox、Chrome、Safari、Opera  
  
**5. X**

1. a { color: red; }
2. ul { margin-left: 0; }

复制代码

标签选择器。使用标签选择器作用于作用域范围内的所有对应标签。优先级仅仅比\*高。 兼容浏览器：IE6+、Firefox、Chrome、Safari、Opera  
  
**6. X:visited和X:link**

1. a:link { color: red; }
2. a:visted { color: purple; }

复制代码

使用:link伪类作用于未点击过的链接标签。:hover伪类作用于点击过的链接。 兼容浏览器：IE7+、Firefox、Chrome、Safari、Opera  
  
**7. X+Y**

1. ul + p { color: red; }

复制代码

相邻选择器，上述代码中就会匹配在ul后面的第一个p，将段落内的文字颜色设置为红色。(只匹配第一个元素) 兼容浏览器：IE7+、Firefox、Chrome、Safari、Opera  
  
**8. X>Y**

1. div#container > ul {border: 1px solid black;}
2. <div id="container">
3. <ul>
4. <li> List Item
5. <ul>
6. <li> Child </li>
7. </ul>
8. </li>
9. <li> List Item </li>
10. <li> List Item </li>
11. <li> List Item </li>
12. </ul>
13. </div>

复制代码

子选择器。与后代选择器X Y不同的是，子选择器只对X下的直接子级Y起作用。在上面的css和html例子中，div#container>ul仅对container中最近一级的ul起作用。从理论上来讲X > Y是值得提倡选择器，可惜IE6不支持。 兼容浏览器：IE7+、Firefox、Chrome、Safari、Opera  
  
**9. X ~ Y**

1. ul ~ p {color: red;}

复制代码

相邻选择器，与前面提到的X+Y不同的是，X~Y匹配与X相同级别的所有Y元素，而X+Y只匹配第一个。 兼容浏览器：IE7+、Firefox、Chrome、Safari、Opera  
  
**10. X[title]**

1. a[title] {color: green;}

复制代码

属性选择器。比如上述代码匹配的是带有title属性的链接元素。  
兼容浏览器：IE7+、Firefox、Chrome、Safari、Opera  
  
**11. X[title="foo"]**

1. a[href="http://www.w3cfuns.com"] {color: #1f6053;}

复制代码

属性选择器。 上面的代码匹配所有拥有href属性，且href为[http://www.w3cfuns.com](http://www.w3cfuns.com/)的所有链接。  
这个功能很好，但是多少又有些局限。如果我们希望匹配href包含css9.net的所有链接该怎么做呢?看下一个选择器。 兼容浏览器：IE7+、Firefox、Chrome、Safari、Opera  
  
**12. X[title\*="css9.net"]**

1. a[href\*="w3cfuns.com"] {color: #1f6053;}

复制代码

属性选择器。正如我们想要的，上面代码匹配的是href中包含"w3cfusn.com"的所有链接。  
兼容浏览器：IE7+、Firefox、Chrome、Safari、Opera  
  
**13. X[href^="http"]**

1. a[href$=".jpg"], a[href$=".jpeg"], a[href$=".png"], a[href$=".gif"] {
2. color: red;
3. }

复制代码

看上去比较麻烦。另一个解决办法是为所有的图片链接加一个特定的属性，例如‘data-file’  
html代码

1. <a href="path/to/image.jpg" data-filetype="image"> 图片链接 </a>
2. css代码如下： a[data-filetype="image"] { color: red;}

复制代码

这样所有链接到图片的链接字体颜色为红色。  
兼容浏览器：IE7+、Firefox、Chrome、Safari、Opera  
  
  
**15. X[foo~="bar"]**  
属性选择器。属性选择器中的波浪线符号可以让我们匹配属性值中用空格分隔的多个值中的一个。看下面例子：  
html代码

1. <a href="path/to/image.jpg" data-info="external image"> Click Me, Fool </a>

复制代码

css代码

1. a[data-info~="external"] {
2. color: red;
3. }
4. a[data-info~="image"] {
5. border: 1px solid black;
6. }

复制代码

在上面例子中，匹配data-info属性中包含“external”链接的字体颜色为红色。匹配data-info属性中包含“image”的链接设置黑色边框。  
兼容浏览器：IE7+、Firefox、Chrome、Safari、Opera  
  
**17. X:checked**  
checked伪类用来匹配处于选定状态的界面元素，如radio、checkbox。

1. input[type=radio]:checked {
2. border: 1px solid black;
3. }

复制代码

上面代码中匹配的是所有处于选定状态的单选radio，设置1px的黑色边框。  
兼容浏览器：IE9+、Firefox、Chrome、Safari、Opera  
  
**18. X:after和X:before**  
这两个伪类与content结合用于在元素的前面或者后面追加内容，看一个简单的例子：

1. h1:after {content:url(/i/logo.gif)}

复制代码

上面的代码实现了在h1标题的后面显示一张图片。  
我们也经常用它来实现清除浮动，写法如下：

1. .clearfix:after {
2. content: "";
3. display: block;
4. clear: both;
5. visibility: hidden;
6. font-size: 0;
7. height: 0;
8. }
9. .clearfix {
10. \*display: inline-block;
11. \_height: 1%;
12. }

复制代码

**19. X:hover**

1. div:hover {
2. background: #e3e3e3;
3. }

复制代码

:hover伪类设定当鼠标划过时元素的样式。上面代码中设定了div划过时的背景色。  
需要注意的是，在ie 6中，:hover只能用于链接元素。  
这里分享一个经验，在设定链接划过时出现下滑线时，使用border-bottom会比text-decoration显得更漂亮些。代码如下：

1. a:hover {
2. border-bottom: 1px solid black;
3. }

复制代码

兼容浏览器：IE6+、Firefox、Chrome、Safari、Opera  
  
**20. X:not(selector)**

1. p::first-letter {
2. float: left;
3. font-size: 2em;
4. font-weight: bold;
5. font-family: cursive;
6. padding-right: 2px;
7. }

复制代码

下面的代码中设定了段落中第一行的样式：

1. p::first-line {
2. font-weight: bold;
3. font-size: 1.2em;
4. }

复制代码

兼容浏览器：IE6+、Firefox、Chrome、Safari、Opera  
(IE6竟然支持，有些意外啊。)  
  
**22. X:nth-child(n)**

1. li:nth-child(3) {
2. color: red;
3. }

复制代码

这个伪类用于设定一个序列元素(比如li、tr)中的第n个元素(从1开始算起)的样式。在上面例子中，设定第三个列表元素li的字体颜色为红色。  
看一个更灵活的用法，在下面例子中设定第偶数个元素的样式，可以用它来实现隔行换色：

1. tr:nth-child(2n) {
2. background-color: gray;
3. }

复制代码

兼容浏览器：IE9+、Firefox、Chrome、Safari  
  
**23. X:nth-last-child(n)**

1. li:nth-last-child(2) {
2. color: red;
3. }

复制代码

与X:nth-child(n)功能类似，不同的是它从一个序列的最后一个元素开始算起。上面例子中设定倒数第二个列表元素的字体颜色。  
兼容浏览器：IE9+、Firefox、Chrome、Safari、Opera  
  
**24. X:nth-of-type(n)**

1. ul:nth-of-type(3) {
2. border: 1px solid black;
3. }

复制代码

与X:nth-child(n)功能类似，不同的是它匹配的不是某个序列元素，而是元素类型。例如上面的代码设置页面中出现的第三个无序列表ul的边框。  
兼容浏览器：IE9+、Firefox、Chrome、Safari  
  
**25. X:nth-last-of-type(n)**

1. ul:nth-last-of-type(3) { border: 1px solid black; }

复制代码

与X:nth-of-type(n)功能类似，不同的是它从元素最后一次出现开始算起。上面例子中设定倒数第三个无序列表的边框  
兼容浏览器：IE9+、Firefox、Chrome、Safari、Opera  
  
**26. X:first-child**  
:first-child伪类用于匹配一个序列的第一个元素。我们经常用它来实现一个序列的第一个元素或最后一个元素的上(下)边框，如：

1. ul:nth-last-of-type(3) {
2. border: 1px solid black;
3. }

复制代码

兼容浏览器：IE7+、Firefox、Chrome、Safari、Opera  
  
**27. X:last-child**

1. ul > li:last-child {
2. border-bottom:none;
3. }

复制代码

与:first-child类似，它匹配的是序列中的最后一个元素。  
兼容浏览器：IE9+、Firefox、Chrome、Safari、Opera  
  
**28. X:only-child**

1. div p:only-child {
2. color: red;
3. }

复制代码

这个伪类用的比较少。在上面例子中匹配的是div下有且仅有一个的p，也就是说，如果div内有多个p，将不匹配。

1. <div>
2. <p> My paragraph here. </p>
3. </div>
4. <div>
5. <p> Two paragraphs total. </p>
6. <p> Two paragraphs total. </p>
7. </div>

复制代码

在上面代码中第一个div中的段落p将会被匹配，而第二个div中的p则不会。  
兼容浏览器：IE9+、Firefox、Chrome、Safari、Opera  
  
**29. X:only-of-type**

1. li:only-of-type {
2. font-weight: bold;
3. }

复制代码

这个伪类匹配的是，在它上级容器下只有它一个子元素，它没有邻居元素。例如上面代码匹配仅有一个列表项的列表元素。  
兼容浏览器：IE9+、Firefox、Chrome、Safari、Opera  
  
**30. X:first-of-type**  
:first-of-type伪类与:nth-of-type(1)效果相同，匹配出现的第一个元素。我们来看个例子：

1. ul:first-of-type > li:nth-child(2) {
2. font-weight: bold;
3. }

复制代码

方案二：

1. p + ul li:last-child {
2. font-weight: bold;
3. }

复制代码

方案三：

1. ul:first-of-type li:nth-last-child(1) {
2. font-weight: bold;
3. }

复制代码

兼容浏览器：IE9+、Firefox、Chrome、Safari、Opera。