1. 基本选择器
2. 通用选择器（通配符）

\*{margin：0； padding：0；}

2标签选择器

a{ text-decoration：none；}

3.id选择器

#box{ color：red；}

4.class选择器

.box{ color :red;}

5.群组选择器

ul，ol{ list-style：none；}

1. 层次选择器（关系选择器）
2. E F 包含选择器（后代选择器）

匹配E元素下的所有子元素F

1. E>F 子选择器

匹配E元素下第一级子元素。

1. E+F 相邻兄弟选择器

匹配E元素后紧邻的F元素，只能由一个。

1. E~F 通用兄弟选择器

匹配E元素后所有兄弟元素F，可以匹配多个。

♥伪类选择器

1. 动态伪类选择器

1.E:link 超链接的初始状态

2.E:visited 链接访问后的状态

**3.E:hover 鼠标划过（悬停）时的状态**

4.E:active 鼠标按下（链接激活）时的状态

5.E:focus 当获取到焦点时触发的状态。

四、结构性伪类选择器

1.  **:first-child 匹配父元素下的第一个子元素。**

备注： :first-child 等价于 :nth-child（1）

都是匹配父元素下的第一个子元素。

2. **:last-child 匹配父元素下最后一个子元素。**

备注： :last-child 等价于 :nth-last-child(1)

都是**匹配**父元素下最后一个子元素。

3. **:nth-child（n） 匹配父元素下的第n个子元素。**

4.  :nth-last-child(n) 匹配父元素从后往前找第n个子元素。

扩展：

:nth-child(2n) 等价于 :nth-child(even) 匹配父元素下的第偶数个子元素

:nth-child(2n+1) 等价于 :nth-child(odd) 匹配父元素第奇数个子元素

------------------------------------------------------------------------------------------

5. :first-of-type 匹配父元素下指定类型的第一个子元素

6. :last-of-type 匹配父元素下指定类型的最后一个子元素

7. :nth-of-type(n) 匹配父元素下指定类型的第n个子元素。

8. :nth-last-of-type(n) 匹配父元素下指定类型的倒数第n个子元素。

------------------------------------------------------------------------------------------

9. :only-child 匹配父元素下唯一一个子元素

10. :only-of-type 匹配父元素下指定类型的唯一一个子元素。

------------------------------------------------------------------------------------------

11. E:empty 匹配内容为空的E元素（空格 回车 换行符都不能有）

12. :root 匹配根元素 html

五、UI状态伪类选择器(主要针对form表单元素进行操作)

1. E:checked 匹配选中状态的单选或复选按钮

eg ：input：checked +label{ color：red； }

匹配选中状态下的单选或者复选按钮后面紧邻的label元素

2. E:enabled 匹配启用状态的表单元素

3. E：disabled 匹配禁用状态的表单元素

六、目标伪类选择器

语法： E:target 匹配锚点链接连接到的那个元素。

eg： <a href=”#news”>新闻资讯</a> <p id=”news”>…..</p>

p{display:none;} p:target{ display:block; }

当锚点链接连接到p元素时显示

七、语言伪类选择器

语法： E：lang（val） 匹配含有lang属性 ，并且值为val的属性

eg： html ：lang（en）{ font-family：“楷体”；}

八、否定伪类选择器

语法：E:not(f) 匹配不满足f的E元素

eg：li：not（[class=”lis”]）{ color:blue;}

匹配不满足条件的class=“lis”的li元素。

九、属性选择器

1.E[attr] 匹配含有attr属性的E元素

eg： a[title]{ color：red； }

**2.E[attr=”val”] 匹配含有title属性的E元素，并且属性值为val的E元素**

**eg： a[title=”教育”]{ font-size：30px；}**

3.E[attr^=”val”] 匹配含有attr属性，并且属性值是以val开头的E元素

eg ：a[href^=”http”]{ border: 1px solid red;}

4.E[attr$=”val”] 匹配含有attr属性 ，并且属性值**以val结尾**的E元素

eg： a[href$=”com”]{color :red;}

5.E[attr|=”val”] 匹配含有attr属性，并且属性值以val—开头或者就为val的E元素

eg： E[class|=”link”]{ color:red;}

6. E[attr~=”val”] 匹配含有attr属性，并且属性值列表中含有val这个词的E元素

eg：a[title~=“教育”]{ color：red；}

7.E[attr\*=”val”] 匹配含有attr属性，并且属性值列表中含有val字的E元素

eg：a[title\*=”教育”]{ color：red；}

十、渐进增强和优雅降级

1.渐进增强（由低到高）

刚开始针对低版本浏览器构建页面，完成基本功能，然后针对高版本浏览器添加交互效果，达到更好的用户体验

2.优雅降级（由高到低）

刚开始就以高版本浏览器完成网站的功能，让后针对个浏览器进行调试和修复

注：css3 属性针对不同浏览器内核兼容写法如下：

—webkit— webkit内核

—moz— 火狐内核

—ms— IE内核

—o— opera内核