

# 专题二: CSS选择器世界深度剖析

精通CSS选择符-众多高级选择器技术的核心

选择器的权重和优先级

选择器的命名

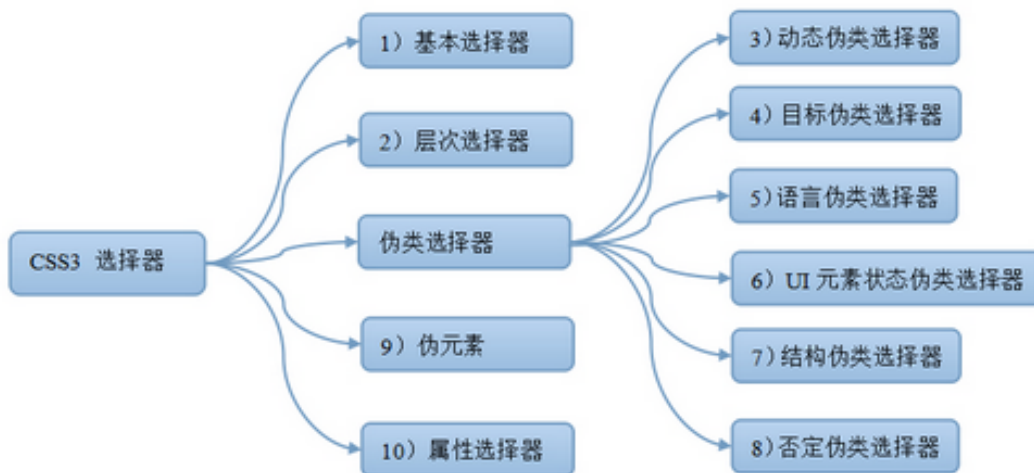
选择器中的正则表达式

AMCSS开发模式简介

全面解读伪类和伪元素

## 大纲目录:

- 精通CSS选择符-众多高级选择器技术的核心
- 选择器的权重和优先级
- 选择器的命名
- 选择器中的正则表达式
- AMCSS开发模式简介
- 全面解读伪类和伪元素



## 精通CSS选择符-众多高级选择器技术的核心

[https://developer.mozilla.org/zh-CN/docs/Web/Guide/CSS/Getting\\_Started/Selectors](https://developer.mozilla.org/zh-CN/docs/Web/Guide/CSS/Getting_Started/Selectors)

<http://css.cuishifeng.cn/>



## 基本选择器

选择器	类型	功能描述
*	通配选择器	选择文档中所有HTML元素
E	元素选择器	选择指定类型的HTML元素
#id	ID选择器	选择指定ID属性值为“id”的任意类型元素
.class	类选择器	选择指定class属性值为“class”的任意类型的任意多个元素
selector1,selectorN	群组选择器	将每一个选择器匹配的元素集合并

## 层次(关系)选择器

选择器	类型	功能描述
E F	后代选择器（包含选择器）	选择匹配的F元素，且匹配的F元素被包含在匹配的E元素内
E>F	子选择器	选择匹配的F元素，且匹配的F元素所匹配的E元素的子元素
E+F	相邻兄弟选择器	选择匹配的F元素，且匹配的F元素紧位于匹配的E元素的后面
E~F	通用选择器	选择匹配的F元素，且位于匹配的E元素后的所有匹配的F元素

## 动态伪类选择器

选择器	类型	功能描述
E:link	链接伪类选择器	选择匹配的E元素，而且匹配元素被定义了超链接并未被访问过。常用于链接描点上
E:visited	链接伪类选择器	选择匹配的E元素，而且匹配元素被定义了超链接并已被访问过。常用于链接描点上

E:active	用户行为选择器	选择匹配的E元素，且匹配元素被激活。常用于链接描点和按钮上
E:hover	用户行为选择器	选择匹配的E元素，且用户鼠标停留在元素E上。IE6及以下浏览器仅支持a:hover
E:focus	用户行为选择器	选择匹配的E元素，而且匹配元素获取焦点

### 目标伪类选择器

选择器	功能描述
E:target	选择匹配E的所有元素，且匹配元素被相关URL指向

### UI元素状态伪类选择器

选择器	类型	功能描述
E:checked	选中状态伪类选择器	匹配选中的复选按钮或者单选按钮表单元素
E:enabled	启用状态伪类选择器	匹配所有启用的表单元素
E:disabled	不可用状态伪类选择器	匹配所有禁用的表单元素

### 结构伪类选择器

选择器	功能描述
E:first-child	作为父元素的第一个子元素的元素E。与E:nth-child(1)等同
E:last-child	作为父元素的最后一个子元素的元素E。与E:nth-last-child(1)等同
E:root	选择匹配元素E所在文档的根元素。在HTML文档中，根元素始终是html，此时该选择器与html类型选择器匹配的内容相同
E F:nth-child(n)	选择父元素E的第n个子元素F。其中n可以是整数（1，2，3）、关键字（even，odd）、可以是公式（2n+1），而且n值起始值为1，而不是0。
E F:nth-last-child(n)	选择父元素E的倒数第n个子元素F。此选择器与E:nth-child(n)选择器计算顺序刚好相反，但使用方法都是一样的，其中：nth-last-child(1)始终匹配最后一个元素，与last-child等同
E:nth-of-type(n)	选择父元素内具有指定类型的第n个E元素
E:nth-last-of-type(n)	选择父元素内具有指定类型的倒数第n个E元素
E:first-of-type	选择父元素内具有指定类型的第一个E元素，与E:nth-of-type(1)等同
E:last-of-type	选择父元素内具有指定类型的最后一个E元素，与E:nth-last-of-type(1)等同

E:only-child	选择父元素只包含一个子元素，且该子元素匹配E元素
E:only-of-type	选择父元素只包含一个同类型子元素，且该子元素匹配E元素
E:empty	选择没有子元素的元素，而且该元素也不包含任何文本节点

### 否定伪类选择器

选择器	功能描述
E:not(F)	匹配所有除元素F外的E元素

### 伪元素选择器

选择器	功能描述
::first-letter	设置对象内的第一个字符的样式
::first-line	设置对象内的第一行的样式
::before	设置在对象前（依据对象树的逻辑结构）发生的内容。用来和content属性一起使用，并且必须定义content属性
::after	设置在对象后（依据对象树的逻辑结构）发生的内容。用来和content属性一起使用，并且必须定义content属性
::placeholder	设置对象文字占位符的样式
::selection	设置对象被选择时的样式,::selection只能定义被选择时的background-color及color。

## 选择器的权重和优先级

CSS三大特性：继承性、优先级和层叠性；

### 优先级的等级：

- 0级(0)：通配选择器（\*）、选择符（+、>、~、空格、||）
- 1级(1)：元素、关系、伪元素
- 2级(10)：类选择器、属性选择器、伪类
- 3级(100)：ID选择器
- 4级(1000)：style内联选择器
- 5级(10000)：!important

选择器	权值	权级
!important	10000	5级
内联样式	1000	4级
id选择器	100	3级
class选择器	10	2级
元素选择器	1	1级
通配符选择器	0	0级

#### 优先级的计算规则：

- 权重顺序 !important>行内样式>id选择器>类选择器>标签选择器>通配符>继承>浏览器默认

#### 选择器权重：

- !important 优先级最高
- 元素属性 优先级高
- 相同权重 后写的生效

```

1 <ul class="shopping-list" id="awesome">
2     <li><span>Milk</span></li>
3     <li class="favorite" id="must-buy"><span class="highlight">
      Sausage</span></li>
4 </ul>
5
6 // 情况一：Blue(蓝色)
7 这是一个很基础的试题，考的就是选择器的权重问题，上面两个选择器都是标签选择器，按照选择器的权重等级之分，
8 他们都是“0，0，0，1”。但同一个元素有两个或多个冲突的属性声明，那么有最高特殊性的声明就会胜出。
9 如果同一个元素具有相同的权重，处在后面的样式就会比前面的声明重
10 ul {color:red;}
11 li {color:blue;}
12
13 // 情况二：Blue(蓝色)
14 在选择器权重当中，一个ID的权重是“0,1,0,0”，而一个标签元素是0,0,0,1。
15 针对这个示例来说，前一个选择器权重是“0,0,0,2”；后者的权重是“0,1,0,0”。因此，在这里他的颜色是蓝色。
16 ul li {color: red;/*0,0,0,2*/}
17 #must-buy{color: blue;/*0,1,0,0*/}
18
19 // 情况三：Blue(蓝色)

```

```

20 .shopping-list .favorite {color: red;/*0,0,2,0*/}
21 #must-buy {color: blue;/*0,1,0,0*/}
22
23 // 情况四: Blue(蓝色)
24 ul#awesome {color: red;/*0,1,0,1*/}
25 ul.shopping-list li.favorite span {color: blue;/*0,0,2,3*/}
26
27 // 情况五: Blue(蓝色)
28 ul#awesome #must-buy {color: red;/*0,2,0,1*/}
29 .favorite span {color: blue!important;/*1,0,0,1,1*/}
30
31 // 情况六: Blue(蓝色)
32 ul.shopping-list li .highlight {color: red;/*0,0,2,2*/}
33 ul.shopping-list li .highlight:nth-of-type(odd){color: blue;/*0,
    0,3,2*/}
34
35 // 情况七: Red(红色)
36 #awesome .favorite:not(#awesome) .highlight {color: red;/*0,2,2,0
    */}
37 #awesome .highlight:nth-of-type(1):nth-last-of-type(1) {color: bl
    ue;/*0,1,3,0*/}
38
39 对于选择器中给定的各个ID属性值, 加“0,1,0,0”。
40 对于选择器中给定的各个类属性值, 属性选择或伪类, 加“0,0,1,0”。
41 对于选择器中给定的各个元素为伪元素, 加“0,0,0,1”。
42 结合符和通配符“*”以及“:not()”没有任何的加分权。
43 对于行内样式, 加为“1,0,0,0”
44 对于“!important”加分高于上面的一切, 将变成“1,0,0,0,0”

```

## 选择器的命名

### 骆驼命名法

第一个字母要小写, 后面的词的第一个字母就要用大写, 例如: studentInfo、navMenuRedButton

### 帕斯卡命名法

这种命名法同样也是大小写字母混编而成, 和骆驼命名法很像, 但和骆驼命名法有一点区别, 就是所有单词的首字母都要大写, 当然也包括第一个单词; 例如: StudentInfo、NavMenuRedButton

### 匈牙利命名法

在名称前面加上一个或多个小写字母作为前缀，来让名称更加好认，更容易理解，例如：  
head\_navigation、  
red\_navMenuButton；

页面模块的常用命名：

头：header	导航：nav	菜单：menu	友情链接：friendlink
页面外围包裹：wrapper	子导航：subnav	子菜单：submenu	下载：download
页面主体：main	广告：banner	侧栏：sidebar	小技巧：tips
内容：content	标志：logo	栏目：column	滚动：scroll
页脚：footer	搜索：search	热点：hot	上一个：prev
版权：copyright	登录条：loginbar	新闻：news	下一个：next

BEM命名法

案例参考：[https://y.qq.com/n/yqq/song/0022O8fw36UsfQ.html?ADTAG=h5\\_playsong&no\\_redirect=1](https://y.qq.com/n/yqq/song/0022O8fw36UsfQ.html?ADTAG=h5_playsong&no_redirect=1)

BEM的意思就是块（block）、元素（element）、修饰符（modifier），是由Yandex团队提出的一种前端命名方法论。这种巧妙的命名方法让你的CSS类对其他开发者来说更加透明而且更有意义。BEM命名约定更加严格，而且包含更多的信息，它们用于一个团队开发一个耗时的大项目。

- BEM：块（block）、元素（element）、修饰符（modifier）
- block 代表了更高级别的抽象或组件。
- block\_\_element 代表.block的后代，用于形成一个完整的.block的整体。
- block--modifier代表.block的不同状态或不同版本，用于修饰。

```
1 air-table{} /* 块 */
2 air-table__footer{} /* 元素 */
3 air-table--full{} /* 修饰符 */
4
5 // vue组件下使用
6 <template>
7     // wrapper主要用于sass嵌套，以免父（子）组件里的css冲突
8     <div class="air-table(组件名)-wrapper">
9         <el-table class="air-table"></el-table>
10        <div class="air-table__footer air-table__footer--full">
11            <button class="air-table__footer--prev">上</button>
12            <button class="air-table__footer--next">下</button>
13        </div>
14    </div>
```

```

15 </template>
16 <style lang="scss" scoped>
17 .air-table(组件名)-wrapper {
18     .air-table {}
19     .air-table__footer {
20         .air-table__footer-prev {}
21
22         .air-table__footer-bext {}
23
24         &.air-table__footer--full {}
25     }
26 }
27 </style>

```

## 选择器中的正则表达式

<https://www.zhangxinxu.com/wordpress/2016/08/regular-expression-in-css-selector/>

正则表达式字符含义:

- 字符 `^` 表示字符串开始位置匹配;
- 字符 `$` 表示字符串结束位置匹配;
- 字符 `*` 表示字符串任意位置匹配;
- 字符 `i` 表示字符串匹配不区分大小写;
- 字符 `g` 表示字符串全局匹配;

CSS属性选择器版本:

- CSS2.1属性选择器:
  - 直接匹配: `[attr]`、`[attr="val"]`、`[attr~="val"]`、`[attr|="val"]`;
- CSS3属性选择器:
  - 正则匹配: `[attr^='val']`、`[attr$='val']`、`[attr*='val']`,
- CSS4属性选择器
  - 忽略大小写匹配: `[attr='val' i]`

属性选择器

选择器	功能描述
<code>[attribute]</code>	用于选取带有指定属性的元素。
<code>[attribute=value]</code>	用于选取带有指定属性和值的元素。
<code>[attribute~=value]</code>	用于选取属性值中包含指定词汇的元素。



[ <i>attribute</i>  = <i>value</i> ]	用于选取带有以指定值开头的属性值的元素，该值必须是整个单词。
[ <i>attribute</i> ^= <i>value</i> ]	选择匹配元素E，且E元素定义了属性attr，其属性值以val开头的任何字符串。
[ <i>attribute</i> \$= <i>value</i> ]	选择匹配元素E，且E元素定义了属性attr，其属性值以val结束的任何字符串。
[ <i>attribute</i> *= <i>value</i> ]	选择匹配元素E，且E元素定义了属性attr，其属性值任意位置包含了val。

### CSS属性选择器分解：

[attr]：

- 只要元素有attr这个属性就可以：
- <div attr="val"></div> 对
- <div attr="text val"></div> 对
- <div attr="value"></div> 对
- <div attr="val-ue"></div> 对

[attr="val"]：

- 元素的属性名是"attr",值必须是'val'
- <div attr="val"></div> 对
- <div attr="text val"></div> 错
- <div attr="value"></div> 错
- <div attr="val-ue"></div> 错

[attr~="val"]：

- "attr"值需含有单词"val"
- <div attr="val"></div> 对
- <div attr="text val"></div> 对
- <div attr="value"></div> 错
- <div attr="val-ue"></div> 错

[attr|="bar"]：

- "attr" 属性值开头必须是bar的单词，或者开头是bar-
- <div attr="bar"></div> 对
- <div attr="bar-val"></div> 对
- <div attr="barval"></div> 错
- <div attr="bar val"></div> 错

[attr^='val']：

- 属性值开头三个字符需要是val
- <div attr="val"></div> 对
- <div attr="text val"></div> 错

- `<div attr="value"></div>` 对
- `<div attr="val-ue"></div>` 对

[attr\$='val'] :

- 属性值最后三个字符需要是val
- `<div attr="val"></div>` 对
- `<div attr="text val"></div>` 对
- `<div attr="value"></div>` 错
- `<div attr="val-ue"></div>` 错

[attr\*='val']:

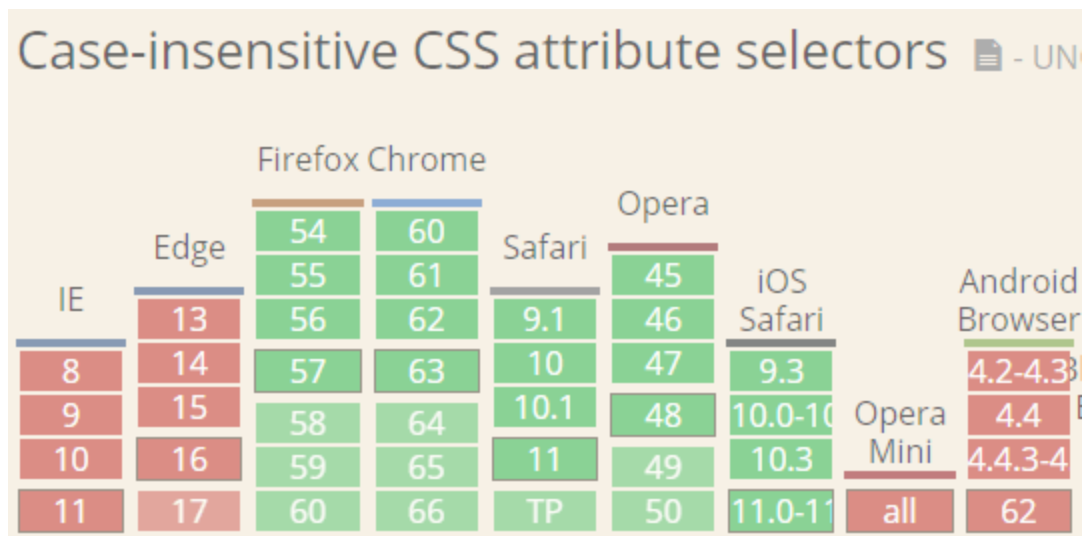
- 属性值任意位置包含val这三个字符就可以
- `<div attr="val"></div>` 对
- `<div attr="text val"></div>` 对
- `<div attr="value"></div>` 对
- `<div attr="val-ue"></div>` 对

[attr\*='val' i]:

- 就和正则表达式中的作i用一样，忽略大小写
- `<div attr="VAL"></div>` 对
- `<div attr="Text val"></div>` 对
- `<div attr="Value"></div>` 对
- `<div attr="Val-ue"></div>` 对

兼容性:

- 目前Chrome, FireFox, Safari已支持*i*正则，但是，IE惊喜过后又打回了原形，包括到IE14都还没支持不区分大小写；
- 估计很快，实际项目中就可以应用此特性了。



## AMCSS开发模式简介

<https://www.zhangxinxu.com/wordpress/2018/03/amcss-attribute-modules-css/>  
<https://css-tricks.com/methods-organize-css/>

```
<div am-size="large" am-disabled></div>
```

### AMCSS的含义:

- 它是Attribute Modules for CSS的缩写, 表示借助HTML属性来进行CSS相关开发;

### 传统类名和AMCSS写法对比:

- 传统写法:
  - `div class="button button-large button-blue">Button</div>`
- AMCSS写法:
  - `<div button="large blue">Button</div>`
  - `<div am-button="large blue">Button</div>`(避免属性名称冲突)

### AMCSS的官网:

- AMCSS有专门的介绍网站: <http://amcss.github.io/>
- 优点: 每个属性有效地声明一个单独的命名空间, 用于封装样式信息, 从而产生更易于阅读和维护的HTML和CSS;

```
1 // 目前主流类名选择器: 通过多个类名进行控制
2 .button {...}
3 .button-large {...}
4 .button-blue {...}
5
6 // AMCSS则是基于属性控制
7 [am-button] {...}
8 [am-button~="large"] {...}
9 [am-button~="blue"] {...}
```

## 使用案例1: Bootstrap Buttons

```
1 <style>
2 .btn{ /* Default button styles */ }
3 .btn-primary{ /* Primary colours */ }
4 .btn-large{ /* Large sizing */ }
5
6 [am-Button] { /* Default button styles */ }
7 [am-Button~="primary"] { /* Primary colours */ }
8 [am-Button~="large"] { /* Large sizing */ }
9 </style>
10
11 // Large primary button
12 <a class="btn btn-primary btn-lg">Large primary button</a>
13 <a am-Button="primary large">Large primary button</a>
14
15 // Default button
16 <a class="btn btn-default">Default button</a>
17 <a am-Button>Default button</a>
18
19 // Small info button
20 <a class="btn btn-info btn-sm">Small info button</a>
21 <a am-Button="info small">Small info button</a>
22
23 // Extra-small danger button
24 <a class="btn btn-danger btn-xs">Extra-small danger button</a>
25 <a am-Button="danger extra-small">Extra-small danger button</a>
```

## 使用案例2: Flexbox Grid

```
1 <div class="row reverse">
2   <div class="column-12--hand column-8--lap">
3     <div class="box">Responsive</div>
4   </div>
5 </div>
6
7 <div am-Grid-Row="reverse">
8   <div am-Grid-Col="12 lap:8">
```

```
9     <div am-Demo="box">Responsive</div>
10 </div>
11 </div>
```

### 使用案例3: Traits

```
1 <div class="u-posAbsoluteCenter">
2   <div class="u-textTruncate u-textCenter">
3     Very centered text.
4   </div>
5 </div>
6
7 <div am-position="absolute center">
8   <div am-text="truncate center">
9     Very centered text.
10  </div>
11 </div>
```

## 全面解读伪类和伪元素

[https://developer.mozilla.org/en-US/docs/Learn/CSS/Building\\_blocks/Selectors/Pseudo-classes\\_and\\_pseudo-elements](https://developer.mozilla.org/en-US/docs/Learn/CSS/Building_blocks/Selectors/Pseudo-classes_and_pseudo-elements)

### 伪类:

- 用于当已有元素处于的某个状态时, 为其添加对应的样式, 这个状态是根据用户行为而动态变化的。比如说, 当用户悬停在指定的元素时, 我们可以通过: hover来描述这个元素的状态。虽然它和普通的css类相似, 可以为已有的元素添加样式, 但是它只有处于dom树无法描述的状态下才能为元素添加样式, 所以将其称为伪类。
- 伪类存在的意义是为了通过选择器, 格式化DOM树以外的信息以及不能被常规CSS选择器获取到的信息。

### 伪元素:

- 用于创建一些不在文档树中的元素, 并为其添加样式。比如说, 我们可以通过: before来在一个元素前增加一些文本, 并为这些文本添加样式。虽然用户可以看到这些文本, 但是这些文本实际上不在文档树中。
- 伪元素可以创建一些文档语言无法创建的虚拟元素。比如: 文档语言没有一种机制可以描述元素内容的第一个字母或第一行, 但伪元素可以做到(::first-letter、::first-line)。同时, 伪元素还可以创建源文档不存在的内容, 比如使用 ::before 或 ::after

## 区别：

伪类的操作对象是文档树中已有的元素，而伪元素则创建了一个文档数外的元素。因此，伪类与伪元素的区别在于：有没有创建一个文档树之外的元素

- 伪类本质上是为了弥补常规CSS选择器的不足，以便获取到更多信息；
- 伪元素本质上是创建了一个有内容的虚拟容器；
- CSS3中伪类和伪元素的语法不同；
- 在CSS3中，已经明确规定了伪类用一个冒号来表示，而伪元素则用两个冒号来表示。
- 可以同时使用多个伪类，而只能同时使用一个伪元素；

