第5章 选 择 器

第1～4章介绍了CSS的语法、使用方式和基本特征。本章将介绍CSS中的一款十分灵活的工具——选择器。

选择器通过一系列的规则指定应用当前样式规则的目标元素，也就是说，样式规则只对与选择器相匹配的元素生效，对不匹配的元素无效。

5.1 基本选择器

基本选择器包括以下4种类型。

1. 元素选择器

元素选择器使用元素标签作为选择器。p会匹配当前页面中的所有＜p＞元素。

```
/* 匹配当前页面中的所有 <p>元素 */
p {
    color: #333;
}
```

2. ID选择器

#target匹配id属性值为"target"的一个元素。

```
#header {
  margin-bottom: 24px;
}
```

一个页面中不能存在两个id属性值相同的元素。

3. 类选择器

.target匹配class属性值中包含"target"单词的元素。

```
.error {
  color: red;
}
```

HTML 元素的 class 属性可以包含多个值，这些值以列表的形式存在（在 DOM 中称为 classList[①]）。当列表中有一个值与选择器匹配时，样式就对该元素生效。

4. 全局选择器

* 可以匹配任意的 HTML 元素。

```
* {
  font-size: 14px;
}
```

不推荐使用全局选择器，选择器应当尽量明确目标，以缩小匹配范围。

5.2 属性选择器

属性选择器通过属性名和属性值匹配元素，有以下几种用法。

(1) [attr]：匹配所有包含 attr 属性的元素。

(2) [attr="value"]：匹配所有 attr 属性值为"value"的元素。

(3) [attr~="value"]：匹配所有 attr 属性值包含单词"value"的元素。

(4) [attr|="value"]：匹配所有 attr 属性值以"value"或"value-"作为开头的元素。

(5) [attr^="value"]：匹配所有 attr 属性值以"value"作为开头的元素。

(6) [attr$="value"]：匹配所有 attr 属性值以"value"作为结尾的元素。

(7) [attr*="value"]：匹配所有 attr 属性值包含字符串"value"的元素。表 5-1 展示了用法(3)和用法(7)的异同(√代表匹配，-代表不匹配)。

表 5-1 [attr~="cat"]与[attr*="cat"]的区别

元素属性	attr~="cat"	attr*="cat"
attr="cat"	√	√
attr="cat dog"	√	√
attr="catch"	—	√
attr="catch dog"	—	√

用法(3)要求属性值包含独立的单词，用法(7)则只要求属性值包含字符串。

5.3 伪类选择器

伪类是指一些添加到选择器的关键词，它们代表了元素的特殊状态或位置。选择器与关键词之间使用:分隔。

① classList：访问网址为 https://developer.mozilla.org/en-US/docs/Web/API/Element/classList。

5.3.1 a 元素专属的几种状态

(1) a:link 代表未被访问过的 a 元素。
(2) a:visited 代表已访问过的 a 元素。

5.3.2 :active 状态

target:active 代表处于激活状态(单击鼠标或点击触摸屏上的按钮)的 target 元素。

```
/* 单击鼠标时,设置 button 的背景色 */
button:active {
  background-color: #66F;
}
```

5.3.3 :hover 状态

target:hover 代表鼠标指针位于元素上时的状态。

```
/* 鼠标经过时,设置 div 的背景色 */
div:hover {
  background-color: #EEE;
}
```

5.3.4 :focus 状态

target:focus 代表 target 元素已获得焦点的状态。

```
/* button 获得焦点时,设置其轮廓 */
button:focus {
  outline: solid 2px #AAF;
}
```

5.3.5 :enabled 状态与:disabled 状态

target:enabled 代表表单元素和按钮未禁用的状态;target:disabled 代表元素已禁用的状态。

```
/* 当按钮被禁用时,设置其文本颜色 */
button:disabled {
  color: #CCC;
}
```

5.3.6 :checked 状态

target:checked 代表已选中的 checkbox 或 radio 元素。

5.3.7 :root 状态

target:root 代表根元素。在 HTML 中,根元素等同于 html 元素。

```
/* 设置 html 元素的字号 */
:root {
  font-size: 14px;
}
```

5.3.8 子元素位置

子元素位置有以下几种表示方法。

(1) target:first-child 代表作为外部元素第一个子元素的 target 元素。

(2) target:last-child 代表作为外部元素最后一个子元素的 target 元素。

(3) target:nth-child(n)代表作为外部元素第 n 个子元素的 target 元素。

(4) target:nth-last-child(n)代表作为外部元素倒数第 n 个子元素的 target 元素。

(5) target:only-child 代表作为外部元素唯一一个子元素的 target 元素。

其中,n 代表子元素的位置,可以有以下取值。

(1) 正整数(1, 2, 3,...)。

(2) odd: 代表奇数位置(1, 3, 5,...)。

(3) even: 代表偶数位置(2, 4, 6,...)。

(4) $ax+b$: a、b 为正整数,x 的最小值为 0(如 $2x+1$ 等同于 odd, $2x$ 等同于 even)。

```
/* 在列表项之间添加分隔线的一种方式 */

ul > li {
  /* 为每个列表项添加底部边框 */
  border-bottom: solid 1px #EEE;
}
```

```
ul > li:last-child {
  /* 取消最后一个列表项的底部边框 */
  border-bottom: none;
}
```

5.3.9 子元素类型

子元素有以下几种类型。

(1) target:first-of-type 代表作为外部元素内第一个 target 类型的子元素。

(2) target:last-of-type 代表作为外部元素内最后一个 target 类型的子元素。

(3) target:nth-of-type(n)代表作为外部元素内第 *n* 个 target 类型的子元素。

(4) target:nth-last-of-type(n)代表作为外部元素内倒数第 *n* 个 target 类型的子元素。

(5) target:only-of-type 代表作为外部元素内唯一一个 target 类型的子元素。

```
/* 为 div 内的第一个 p 元素添加边框 */
div > p:first-of-type {
  border: solid 1px #EEE;
}
```

5.3.10 :not(selector)

target:not(selector) 用来选择不匹配 selector 的 target 元素。

5.3.11 :fullscreen

target:fullscreen 用来匹配处于全屏显示模式的 target 元素,可以参考浏览器的 Fullscreen API[①]。

5.4 伪元素选择器

伪元素代表位于特殊位置的元素或内容。

为了区分伪类和伪元素,在 CSS 3 中推荐使用 :: 作为分隔符。如果需要兼容之前的浏览器,则可以使用下面的写法:

① Fullscreen API:访问网址为 https://developer.mozilla.org/en-US/docs/Web/API/Fullscreen_API。

```
div:after,
div::after {}
```

5.4.1 ::after

target::after 可以创建一个伪元素作为 target 元素的最后一个子元素，通常会配合 content 属性使用。

```
/* 在每个段落末尾插入一个元素 */
p::after {
  content:"△";  /* 设置该元素的内容为 "△" */
  color: red;    /* 设置其颜色为红色 */
}
```

可以继续给伪元素添加其他样式。

5.4.2 ::before

类似于 target::after，target::before 可以创建一个伪元素，作为 target 元素的第一个子元素。

5.4.3 ::first-letter

target::first-letter 代表元素内的首个字符。

```
p::first-letter {
  font-size: 2em;   /* 将段落首个字符的字号设置为两倍大小 */
}
```

5.4.4 ::first-line

target::first-line 代表元素内的首行字符。

5.4.5 ::selection

target::selection 代表已选中的字符(通常具有选中高亮的效果)。

注意：该伪类不支持 target:selection 的写法。

```
body::selection {  /* 设置选中文本的颜色和背景色 */
  color: #FFF;
  background-color: rgba(255, 64, 64, 0.5);
}
```

5.5 关系选择器

关系选择器通过元素之间的位置关系匹配目标元素。

以下内容中的“a”“b”“c”分别代表选择器。

1. 嵌套关系

a b 用于匹配 a 元素内的所有 b 元素(祖先元素与后代元素,包括父子关系)。

类似地,a b c 用于匹配 a 元素内的 b 元素内的所有 c 元素。通常不建议叠加超过三组选择器。

2. 父子关系

a ＞ b 用于匹配以 a 元素作为父元素的所有 b 元素(只能是父子关系)。

3. 相邻关系

a ＋ b 用于匹配紧跟在 a 元素后面的 b 元素(相邻关系)。

4. 兄弟关系

a ～ b 用于匹配和 a 元素具有相同父元素的所有 b 元素(兄弟关系)。

5.6 选择器组合

前面几节介绍了不同类型的选择器,这些选择器可以单独使用,也可以通过叠加和组合实现更复杂的匹配逻辑。

5.6.1 叠加

除了个别基本选择器,其他类型的选择器都可以相互叠加使用。

```
/* 基本选择器 +基本选择器 */
button.ok {}
/* 匹配 class 值包含 "ok" 的 button 元素 */
```

```
/* 基本选择器 +属性选择器 */
span[title="hello"] {}
/* 匹配 title 属性值为 "hello" 的 span 元素 */

/* 基本选择器 +伪元素选择器 */
button:focus {}
/* 匹配当前具有焦点的 button 元素 */

/* 基本选择器 +属性选择器 +伪元素选择器 */
input[type="checkbox"]:checked {}
/* 匹配已选中的 checkbox 表单元素 */
```

5.6.2 组合

a 和 b 分别代表不同的选择器，a，b 代表包含两组选择器所匹配元素的集合。

```
p {
  color: #333;
}
h2 {
  color: #333;
}

/* 等同于 */
p,
h2 {
  color: #333;
}
```

建议每组选择器独占一行。

5.7 选择器优先级

当一个元素对应多组样式规则时，浏览器通过选择器优先级决定将哪些样式应用到该元素。

读者在编写 CSS 代码时并不需要直接计算选择器的优先级，只需要判断两组选择器的优先级大小关系即可。

使用开发者工具可以直观方便地做出判断。如图 5-1 所示，位于上面的选择器的优先级更高。

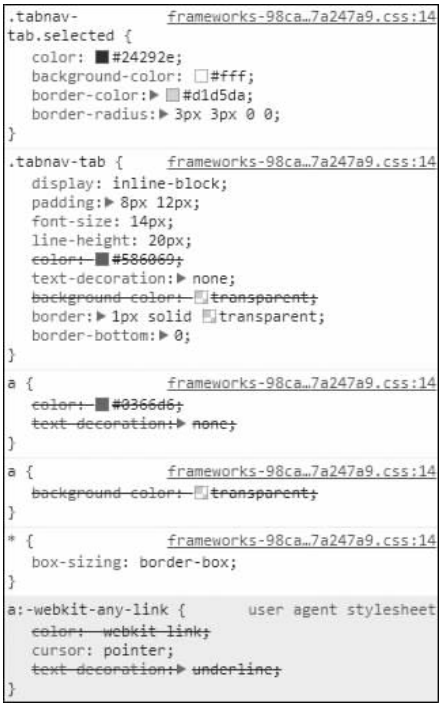

图 5-1　通过开发者工具判断选择器的优先级