文章编号：2096-1472(2018)-03-34-03
DOI:10.19644/j.cnki.issn2096-1472.2018.03.009

深入理解CSS3结构伪类选择器

黄志刚
(湖南科技职业学院软件学院，湖南 长沙 410118)

摘　要：本文从结构伪类与类选择器的比较、结构伪类与文档树的关系、结构伪类的功能与应用等三个层面，深入阐述了结构伪类的作用，结构伪类所表示的文档树结构，结构伪类通过位置索引定位子元素的机制。通过将定位子元素的结构伪类分为带参数和不带参数两类，深入论述了带参数结构伪类的参数模式、典型应用及与不带参数结构伪类的对应关系。

关键词：选择器；结构伪类；文档树；位置索引
中图分类号：TP391　　**文献标识码**：A

Deep Understanding of CSS3 Structural Pseudo-Classes Selectors

HUANG Zhigang
(*College of Software*,*Hunan Vocational College of Science and Technology*,*Changsha* 410118,*China*)

Abstract: The article describes deeply the function of the structural pseudo-classes,the document tree structure represented by structural pseudo-classes and structural pseudo-classes' mechanism of positioning sub-element by means of position index from the aspects of the comparison of the structural pseudo-classes and class selectors,the relationship of the structural pseudo-classes and the document tree as well as the functions and applications of the structural pseudo-classes.The article discusses in detail the parameter model and typical application with parameter structural pseudo-class,the corresponding relations between structural pseudo-class with and without parameters by dividing the structural pseudo-class of positioning sub-element into two classes with and without parameters.

Keywords:selectors;structural Pseudo-classes;document tree;position index

1 引言(Introduction)

CSS(Cascading Style Sheets)即层叠样式表，是用于控制网页显示效果的技术。选择器是匹配网页元素的模式[1]，它作为样式表的基本组成部分，用于匹配网页中的元素。CSS3是CSS的第三个升级版本，它是一系列规范的集合。选择器规范是CSS3系列规范中的一个，它的W3C(World Wide Web Consortium，万维网联盟)推荐标准是“Selectors Level 3(选择器第三级)”。结构伪类(Structural Pseudo-classes)选择器(规范中简称结构伪类)是CSS3选择器规范新引入的一类选择器，它基于文档树的结构信息匹配元素[1]，功能强大、最具特色，但也较难理解和使用。

2 结构伪类与类选择器(Structural pseudo-classes and class selectors)

在CSS3之前，为给网页元素设置样式，Web开发人员一般是在HTML代码中手动给元素添加类名，然后使用CSS的类选择器匹配元素。这样无节制使用类名的做法，总是导致网页文档中类名泛滥，不仅使编码耗时费力，而且代码不整洁、维护困难。为改变这种状况，HTML5规范建议，要避免为了给元素定义样式而使用类名。基于这种理念，CSS3引入了结构伪类，有效的减少类名的使用。以一个无序列表为例，对类选择器和结构伪类的应用做一个比较。

2.1 使用类选择器匹配元素

为选择第一个列表项和最后一个列表项，首先须在文档源代码中给它们添加类名，然后在CSS代码中用类选择器匹配元素。示例代码如下：

HTML代码：

```
<ul>
<li class="first">1</li>
<li>2</li>
<li class="last">3</li>
</ul>
```

CSS代码：

```
/*通过类选择器匹配第一个列表项*/
li.first {background-color:red;}
/*通过类选择器匹配最后一个列表项*/
li.last {background-color:red;}
```

如果插入或追加列表项，则第一个或最后一个列表项会发生改变，所以，类名也要作相应变动，这给代码维护带来了极大的困难。

2.2 使用结构伪类匹配元素

结构伪类在插入或追加列表项时，文档源代码无须做任何改变，它也总是可以匹配到发生改变后的第一个和最后一个列表项。示例代码如下：

HTML代码：

```
<ul>
<li>1</li>
<li>2</li>
<li>3</li>
</ul>
```

CSS代码：

```
/*通过结构伪类匹配第一个列表项*/
li:first -child{background-color:red;}
/*通过结构伪类匹配最后一个列表项*/
li:last-child {background-color:red;}
```

从以上比较可知，使用类名会改变文档源代码，而且，当项目变化时，还须手动维护类名，即元素的类名的获得是静态分配的；但使用结构伪类，不仅无须改变文档源代码，而且自动跟踪项目系列的序号变化，使元素定位更为准确、高效[2]。也就是说元素的结构伪类可以动态的获取和失去。

3 结构伪类与文档树(Structural pseudo-classes & document tree)

所有的结构伪类都基于HTML文档树(Document Tree)的结构信息，文档树也称为文档对象模型(Document Object Model,DOM)。当创建一个HTML文档(图1)并用Web浏览器查看时，浏览器把文档中元素间的嵌套关系映射为一个树形节点层次结构，即文档树。图1所示文档可以表示为如图2所示的文档树。

```
<html>
  <body>
    <div>
      <ul>
        <li>1</li>
        <li>2</li>
        <li>3</li>
      </ul>
      <span>span-one</span>
      <p>para-one</p>
      <span>span-two</span>
      <p>para-two</p>
      <p></p>
      <img src="h5.png" alt="h5 图标">
    </div>
  </body>
</html>
```

图1 HTML文档

Fig.1 HTML document

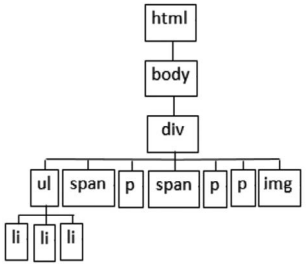

图2 文档树

Fig.2 Document tree

3.1 文档树结构

文档树由节点构成，主要有元素节点、属性节点和文本节点，树中的每个节点都处于文档整体结构的某个层级(本文所说元素均指元素节点)。文档树中位于同一个分支的元素间的关系，有祖先后代关系、父子关系和兄弟关系。在不同的上下文中，对同一个元素的称谓可能是父元素、子元素、祖先元素、后代元素和兄弟元素。

3.1.1 父子关系、父元素和子元素

在文档树的层级结构中，如果从一个元素到另一个元素的路径，处于两个相邻层级，那么，这两个元素是父子关系。位于上面层级的是父元素，位于下面层级是子元素。例如：在图2中，<body>元素是 元素的父元素，反之， 元素是<body>元素的子元素。

3.1.2 祖先后代关系、祖先元素和后代元素

在文档树的层级结构中，如果从一个元素到另一个元素的路径，跨越两个或两个以上的层级，那么，这两个元素是祖先后代关系，在一个元素之下的所有层级的一组元素是其后代元素。一个元素的任何父元素、祖父元素、及其上层的所有元素是其祖先元素。例如：在图2中，<body>元素是<ul>元素的祖先元素，反之，<ul>元素是<body>元素的后代元素。<html>元素是所有元素的祖先元素，是HTML文档的根元素。

3.1.3 兄弟关系和兄弟元素

存在于同一个层级中，具有相同父元素的元素是兄弟关系，彼此互为兄弟元素。例如：图1中，<ul>元素、元素、<p>元素、<img>元素都是 元素的子元素，是兄弟关系，互为兄弟元素。

3.2 结构伪类中的结构信息

CSS3定义了12个结构伪类，它们分别表示文档树的三种结构，定位三种元素。

(1):root pseudo-class(根伪类)，表示文档树的根节点，用来定位根元素。在HTML文档中，html元素就是根元素，用来表示整个网页。

(2):empty pseudo-class(空伪类),表示文档树的空节点，用来定位空元素。空节点是指既没有文本子节点，也没有元素子节点的节点，例如，在图2中，第三个<p>元素和<img>元素就是空元素

(3):nth-child()等其余10个结构伪类，表示文档树父子关系中的子元素节点，通过子元素的位置信息定位子元素。例如：在图2中， 元素

有7个子元素，p:nth-child(3)表示匹配第3个子元素，而且，这个元素必须是<p>元素。

4 结构伪类功能及应用(Functions and applications of the structural pseudo-classes)

CSS3中的12个结构伪类，分别定位文档的根元素、空元素和子元素。根伪类和空伪类的应用较为简单，其余定位子元素的结构伪类的应用较为复杂，尤其是带参数结构伪类的应用灵活多变、较难理解。

4.1 定位根元素

根伪类(:root pseudo-class)用来定位文档树的根元素。例如，要设置整个页面的背景颜色为蓝色，则CSS代码可写成：

:root {background-color：blue;} 或写成

html:root {background-color：blue;}

4.2 定位空元素

空伪类(:empty pseudo-class)用来定位文档树中的空元素。如果要匹配图2中的空段落，为其设置背景颜色，则CSS代码可写成：

p:empty{background-color：blue;}

如果要匹配图2中的所有空元素(一个<p>元素和一个<img>元素)，为它们设置背景颜色，则CSS代码可写成：

:empty{background-color：blue;}

4.3 定位子元素

匹配子元素的结构伪类共10个，可分为带参数和不带参数两类。

4.3.1 带参数结构伪类

带参数结构伪类有四个，名称均以“:nth-”为前缀，最后跟一对圆括号，可匹配一个或多个子元素。带参数结构伪类通过建立子元素的位置索引来定位元素。下面主要以:nth-child()的应用为例，阐述带参数结构伪类的工作机制。

4.3.1.1 位置索引

系统在匹配子元素时，按照子元素在兄弟元素中的相对位置，给每个子元素都设置了一个位置索引，起始值为1，需注意的是，独立文本和非元素节点不参与位置索引的计数。不同的结构伪类，位置索引的编排规则不同。

用:nth-child()伪类匹配子元素时，子元素位置索引的编排规则是，对所有兄弟元素从上到下顺序索引，第1个子元素的位置索引为1。例如在图2中， 元素有7个子元素，<ul>元素的位置索引为1，<img>元素的位置索引为7。

4.3.1.2 伪类参数

参数可以是表达式，也可以关键字。

(1)表达式

表达式的一般形式为$an+b$，其中，a和b是任意整型常量，n是大于等于零的整型变量，要特别注意的是，在CSS代码中表达式中的变量一定要用字母n表示，否则无效。当$n=0,1,2,\cdots$时，$an+b$的值构成了一个集合，其中正整数才表示一个子元素的位置索引，负整数和0是无效值，可直接忽略。当位置索引的值大于页面中子元素的数量时，就没有元素可选，其值也可忽略。

例如：选择器span:nth-child(3n-5)中，表达式$3n-5$的取值为{-5,-2,1,4,7,10,…}，位置索引不能为零和负值，所以-5、-2无效，这个选择器匹配的子元素的位置索引为1、4、7、10…，而且必须是元素。在图2中，位置索引为1的是<ul>元素，位置索引为7的是<img>元素，位置索引10超出了子元素的数量范围，所以，这个选择器只匹配到位置索引为4的元素。

表达式$an+b$匹配元素的模式灵活多变，当a、b取特殊值时，出现了一些有趣的应用。可以分$a=0$和$a\neq0$两种情况。

①$a=0$

当$a=0$，$b>1$时，$an+b$的值是一个常数，表示匹配位置索引的值为b的子元素。例如：p:nth-child(3)，表达式中，$a=0$，$b=3$，表示匹配位置索引为3而且是<p>元素的子元素。

②$a\neq0$

a.当$a>0$且$b>0$时

表达式表示将兄弟元素分组，每组a个元素(最后一组为剩余元素)，每组中的第b个元素被选中[1]，或者说表达式表示在兄弟元素列表中，从第b个元素开始，每隔a个元素选中一个[2]。

例如，假定图2中<ul>元素扩展到10个列表项，则选择器li:nth-child($3n+2$)，匹配的列表项的位置索引为{2,5,8,11，…}，可以理解为将兄弟元素分成每3个一组，10个元素可分成4组，最后一组，只有一个元素，然后选中每组中的第2个元素；也可以表述为，从第2个元素开始，每隔3个元素选中一个；所以，选择器匹配位置索引为2、5、8的列表项。

如果要循环选中每组中的所有列表项，则选择器可写为：

li:nth-child(3n+1)，匹配每组中的第一个子元素；

li:nth-child(3n+2)，匹配每组中的第二个子元素；

li:nth-child(3n+3)，匹配每组中的第三个子元素。

b.当a=1，b>0

假设$b=6$，则表达式为$n+6$，取值集合为{6,7,8,9，…}，所以，选择器li:nth-child($n+6$)将匹配位置索引大于等于6的所有元素。

c.当$a=-1$，$b>0$

假设b=6，则表达式为$-n+6$，取值集合为{6,5,4,3，…}，所以，选择器li:nth-child($-n+6$)将匹配位置索引小于等于6的所有元素。

(2)关键字

关键字有“odd”和“even”，“odd”表示匹配位置索引为奇数的子元素，“even”表示匹配位置索引为偶数的子元素。例如，在图2中，li:nth-child(even)将匹配到所有偶数列表项，相当于li:nth-child(2n)；li:nth-child(odd)将匹配到所有奇数列表项，相当于li:nth-child(2n+1)；所以，“关键词”是表达式为“2n”和“2n+1”的语义化表示。

4.3.2 其他带参数结构伪类

带参数的结构伪类除:nth-child()外，还有:nth-last-child()、:nth-of-type()、:nth-last-of-type()，它们除位置索引的编排规则与:nth-child()不同之外，用法和:nth-child()相同。

(1):nth-last-child()伪类

:nth-last-child()伪类的位置索引的编排规则是，从倒数第一个元素开始，对所有兄弟元素建立索引。例如，在图2中，选择器p:nth-last-child(5)，表示在所有兄弟元素中，选

(下转第30页)