

CSS伪元素::before和::after的应用

Alex Sun

2014-10-10

1. 伪元素与伪类的区别

在CSS中，有伪类和伪元素，CSS3为了区分这两者，规定伪类是以一个冒号开始，而伪元素是两个冒号开始。常见的伪类有 `:hover`, `:focus`, `:first-child`, `:last-child` 等，伪元素有 `::before`, `::after`, `::first-letter`, `::first-line` 等。两者比较相似，有时候比较难以区分。其实最关键的区别就是伪类通常表示一种状态，例如鼠标移到控件上之后会触发 `:hover`，移开之后该状态就消失。而伪元素则相当于是创造了一个新的元素，例如 `::before`，相当于是文档中插入了一个新的节点进去。

不过，既然是伪元素，可见于真实的元素还是有所区别的。其中最关键的就是伪元素只在渲染的时候展示出来，是无法通过操作DOM文档获取到的。

2. ::before和::after

本文中重点介绍`::before`和`::after`这两个伪元素。首先看下面的例子：

```
<!DOCTYPE html>
<html>
<head>
<title>Examples</title>
<style type="text/css">
p::before{
    content:"Welcome! ";
}
p::after{
    content: " Thank you!";
}
</style>
</head>
<body>
    <p>Alex</p>
</body>
```

`</html>`

通过运行我们可以发现，插入的内容并不是在段落外的前后，而是在段落之中内容的前后。然后我们也可以对伪元素进行一些样式上的设定，例如 `display`, `width`, `height`, `border`, `background` 等等。但是需要注意的是，即使我们不需要插入任何文本内容，也要写上 `content: ""`，如果没有 `content`，那么该伪元素就不会渲染出来。

3. 实例分析1

了解了上面的内容后，我们来分析一个例子。如图所示是一种常见的导航栏样式：



如果不使用背景图片而纯粹用CSS来实现其样式似乎是比较困难的，这个时候，我们就可以借助伪元素了。DOM结构如下：

```
<body>
  <ul class="nav">
    <li>Home</li>
    <li>Alex</li>
    <li>Download</li>
    <li>css</li>
  </ul>
</body>
```

CSS样式如下：

```
@charset "UTF-8";

body{
  font-family:Arial, "Times new Roman";
}

ul.nav{
  list-style:none;
}
li{
  position: relative;
  float:left;
  height:40px;
```

```

    line-height:40px;
    background:#3498db;
    color:#fff;
    padding-right:10px;
    margin-right:22px;
    font-size:12px;
    cursor: pointer;
}

li::before,
li::after{
    content:"";
    position: absolute;
    top:0;
    width:0;
    height:0;
    border-width: 20px 10px;
    border-style: solid;
}
li::before{
    left:-20px;
    border-color:#3498db;
    border-left-color: transparent;
}
li::after{
    right:-20px;
    border-color:transparent;
    border-left-color: #3498db;
}

li:first-child{
    padding-left:10px;
    border-radius:4px 0 0 4px;
}
li:last-child{
    border-radius: 0 4px 4px 0;
}
li:first-child::before,
li:last-child::after{
    border:none;
}

li:hover{
    background: #1abc9c;
}
li:hover::before{
    border-color:#1abc9c;
    border-left-color: transparent;
}

```

```
li:hover::after{
    border-color:transparent;
    border-left-color: #1abc9c;
}
```

这里我们是在每个li元素前后都加上了一个伪元素，并设置其width和height均为0，然后设置border，通过四个border的颜色不同来形成前后图形。

4. 实例分析2

在某些前端库中，往往会包含一些图标控件。同样，我们也可以通过使用伪元素来实现这一功能，例如我们可以定义一个icon-open的类，然后使用：

```
.icon-open::before{
    content:url("open.png");
}
```

这样之后我们只要给需要图标的控件加上icon-open的类就可以了。

同样有些网页中会有一些文件的链接，例如doc、pdf等，有些时候我们也希望给这些链接一个图标，我们可以这样做：

```
a[href$=".pdf"]::after{
    content:url("pdf.png");
}
```

这样我们便可以非常简单地给链接添加图标了。

5. 再谈content

在前面的例子中，我们知道了content的内容可以是字符串，也可以是一个url，不过，content可接受的内容并不限于这些，还可以接受一下方式：

(1) 多个片段，例如：

```
p::before{
    content: "hello" "world";
}
```

```
p::before{
    content: "hello" url("pdf.png");
}
```

多个片段之间有没有空格隔开都可以。

(2) attr获取的属性，例如：

```
a::before{
    content: attr(href);
}
```

即使是自定义的属性也是可以的。

(3) 使用自定义quote，例如：

```
p{
    quotes: "{" " "}";
}
p::before{
    content:open-quote;
}
p::after{
    content:close-quote;
}
```

首先要定义quotes，quotes必须是成对出现的，然后通过下面的语句即可实现花括号括住段落的效果。

(4) 使用counter，例如：

```
<!DOCTYPE html>
<html>
  <head>
    <title>Examples</title>
    <style type="text/css">
      body {
        counter-reset: chapter section subsec;
      }

      h1 {
        font-size: 28pt;
        font-weight: bold;
        color: red;
        counter-reset: section subsec;
```

```

        counter-increment: chapter;
    }

    h1::before {
        content: counter(chapter,upper-alpha) ".";
    }

    h2 {
        font-size: 24pt;
        font-weight: bold;
        color: green;
        counter-reset: subsec;
        counter-increment: section;
    }

    h2::before {
        content: counter(chapter,upper-alpha) "." counter(section) ".";
    }

    h3 {
        font-size: 22pt;
        font-weight: bold;
        color: blue;
        counter-increment: subsec;
    }

    h3::before {
        content: counter(chapter,upper-
alpha) "." counter(section) "." counter(subsec,lower-roman) ".";
    }
</style>
</head>
<body>
    <h1>This is H1 element</h1>
    <h2>This is H2 element</h2>
    <h3>This is H3 element</h3>
    <h3>This is H3 element</h3>
    <h2>This is H2 element</h2>
    <h3>This is H3 element</h3>
    <h3>This is H3 element</h3>
    <h1>This is H1 element</h1>
    <h2>This is H2 element</h2>
    <h3>This is H3 element</h3>
    <h1>This is H1 element</h1>
    <h2>This is H2 element</h2>
    <h3>This is H3 element</h3>
</body>
</html>

```

`counter-reset`设置计数器的起点，默认为0，可以为负数。`counter-increment`设置计数器递增量，默认为1，可以为负数。这样通过`counter`就可以为章节添加序号了。

6. 更多例子

以下链接有一些应用伪元素的不错的例子，可以作为参考：

[基于单个 Div 的 CSS 绘图](#)

[Creative Link Effects](#)

[Understanding Pseudo-Element :Before And :After](#)