

### 學習要點:

1. CSS3 前綴
2. 長度單位 rem

本章主要探討 HTML5 中 CSS 在發展中實行標準化的一些問題，重點探討 CSS3 中新屬性前綴問題和新的單位 rem。

#### 一、CSS3 前綴

在 CSS3 的很多新屬性推出時，這些屬性還處在不太穩定的階段，隨時可能被刪除。而此時的瀏覽器廠商為了實現這些屬性，採用前綴方法。各大廠商前綴列表如下：

瀏覽器	廠商前綴
Chrome、Safari	-webkit-
Opera	-o-
Firefox	-moz-
Internet Explorer	-ms-

我們之前學習過幾個 CSS3 的新屬性，比如: box-shadow、border-radius、opacity 等。這幾個屬性我們在前面的使用中，並沒有添加所謂的瀏覽器廠商前綴。那是因為，這些屬性已經在主流瀏覽器或版本成為了標準。具體進化步驟如下：

1. 屬性尚未提出，這個屬性所有瀏覽器不可用：
2. 屬性被提出，但未列入標準，瀏覽器廠商通過私有前綴來支持該屬性：
3. 屬性被列入標準，可以使用前綴或不使用前綴來實現屬性特性：
4. 屬性不需要再使用前綴，所有瀏覽器都穩定支持。

我們就拿 box-radius 舉例，它是 CSS3 的標準屬性。早期的時候處於實驗階段，尚未列入標準時，需要使用廠商前綴。具體瀏覽器支持度如下：

屬性	瀏覽器	帶前綴版本	不帶前綴版本	標準/實驗
border-radius	IE	不支持	9.0+	標準
	Firefox	3.0 需帶-moz-	4.0+	
	Safari	3.1 需帶-webkit-	5.1+	
	Chrome	4.0	5.0+	
	Opera	不支持	10.5+	

如果是手機等移動端一般採用的是 IOS 或安卓系統：那麼基本上它的引擎是 webkit，直接參考-webkit-即可。

在 CSS 手冊上，chrome 支持 border-radius 的版本為 13.0，而部分教材和文

章上寫到只要 5.0。當然，這裡可能表達的含義又可能不同，而截至到 2015 年四月份最新的 Chrome 版本已經到 14.0 了，所以，不管是 5.0 還是 13.0 都是老古董了，沒必要深究。Opera 支持 border-radius 版本為 11.5，而目前的版本是 28.0，也無傷大雅了。

而被列入標準的 box-shadow 和 opacity 基本與 border-radius 前綴版本一致。

```
//因為目前處在標準階段，沒必要寫前綴了
```

```
div {  
    border-radius: 50px;  
}
```

```
//實驗階段的寫法
```

```
div {  
    -webkit-border-radius: 50px;  
    -moz-border-radius: 50px;  
    border-radius: 50px;  
}
```

實驗階段的寫法有三句，分別解釋一下: -webkit-表示 Chrome 瀏覽器的私有屬性前綴，-moz-表示 Firefox 私有屬性前綴，如果是處於實驗階段的舊版本瀏覽器，那麼不支持 border-radius，從而通過廠商前綴的方式來支持。如果是新版瀏覽器，已經是處於標準階段，那麼又有兩種說法: 1. 如果新版瀏覽器廢棄了前綴，那麼前綴寫法就不支持了，僅支持標準寫法: 2. 如果新版瀏覽器沒有廢氣前綴，那麼兩種寫法都可以，但要注意順序，且屬性值要保持一致。

如果同時出現四個階段+一個標準寫法，四個前綴是當實驗階段時讓四種引擎的瀏覽器廠商支持自己的私有屬性，一個標準寫法表示當這個屬性列入標準後，使用新版瀏覽器運行時直接執行這個標準屬性。

```
//前綴寫法寫在標準後面，且值不一樣，就會出現問題
```

```
div {  
    border-radius: 50px;  
    -webkit-border-radius: 100px;  
}
```

特別注意: 1. IE 的前提-ms-，和 Opera 的前綴-o-，在 border-radius 中不存在: 2. 現在的 Opera 瀏覽器也支持-webkit-前綴，-webkit-border-radius 就能支持:

3. Safari for Windows 已被蘋果公司在 2012 年放棄，遺留版本為 5.1.7。

最後說明: W3C 官方的立場表示實驗階段的屬性僅為了測試，未來有可能修改或刪除。而大量的開發者也認為在實際項目中不應該使用前綴，而使用一種優雅降級保證一定的用戶體驗，而通過漸進增減的方式讓新版高級瀏覽器保證最高的結果。

## 二、長度單位 rem

CSS3 引入了一些新的尺寸單位，這裡重點推薦一個: rem 或者成為(根 em)。目前主流的現代瀏覽器都很穩定的支持。它和 em、百分比不同的是，它不是與父元素掛鉤，而是相對於根元素<html>的文本大小來計算的，這樣能更好的統一整體頁面的風格。

//首先，來一段 HTML

```
<h1>標題<em>小標題</em></h1>
```

```
<p>我是一個段落，我是一段<code>代碼</code></p>
```

//其次來一段 CSS

```
html {  
    font-size: 62.5%;  
}  
h1 {  
    font-size: 3em;  
}  
p {  
    font-size: 1.4em;  
}
```

這裡做幾個解釋，我們在之前的 web 設計中大量使用了 px 單位進行布局。因為，早期的固定布局使用 px 較為方便，逐漸養成了這種習慣。而使用 em 單位時更加靈活，尤其是在修改樣式時，只需要修改一下掛鉤元素的那個大小即可，無須每個元素一個個修改。

但就算是 em，還是有一定問題。網頁默認的字號大小為 16px，然後通過<html>設置 62.5%，將網頁基準設置為 10px。而<h1>設置為 3em，就是 10px 的 3 倍，即 30px: <p>設置為 1.4em，就是 10px 的 1.4 倍，即 14px。

現在問題來了，<code>裡面的文本想設置 11px，怎麼辦呢?設置 1.1em 嗎?不對，因為它掛鉤的父元素不是<html>而是<p>變成了 14px 的 1.1 倍了，而想設置 11px，則需要設置 0.786 倍才行。但是，這樣的計算量太大了。所以，

W3C 推出了直接基於根元素單位: rem 。

//直接基於<html>的單位

```
code {  
    font-size: 1.1rem;  
}
```

瀏覽器	rem 單位
Opera	11.6+
Firefox	3.6+
Safari	5.0+
Chrome	6.0+
IE	9.0+