

1 CSS 屬性與值

在 2018 年 6 月 1 日，Chrome 瀏覽器上任一元素的 style 物件，共有 **415** 個 CSS 獨特屬性。到了 12 月 21 日，這個數字變成了 **522**。僅僅 7 個月，Chrome 就新增了超過 100 個新屬性。隨著 CSS 規格不斷演進，這樣的現象會再發生許多次。

截至目前為止，你的瀏覽器裡面有多少個屬性呢？可以用下面這段簡單的 *JavaScript* 程式碼來驗證：

```
001 // 製作一個新的 HTML 文件
002 let element = document.createElement("div");
003
004 let p = 0; // 製作計數器
005 for (index in element.style)
006     p++;
007
008 // 截至 2018 年 12 月 21 號，在 Chrome 輸出 522
009 console.log( p );
```

圖 1 | 要列出瀏覽器中所有可用的 CSS 屬性，請執行以下 JavaScript 程式碼（*codepen.io* 是測試 CSS 和 *JavaScript* 的最快方法）。測試結果可能因瀏覽器和版本而異。

本書的編排方式是將所有屬性都依照主要類別（位置、尺寸、版面配置、**CSS 動畫**等）來分門別類。此外，我也針對所有對於呈現或改變視覺具有某種重要性的屬性，各別製作了圖表，並簡要說明了伴隨的屬性名稱和屬性值。

我們略過了許多很少使用（或者尚未獲得所有主要瀏覽器之全面支援的）CSS 屬性，因為它們只是製造不必要的雜亂而已。

```

001 /*200 梯度 (也稱為 gons 或 grades) */
002 transform: rotate(200grad);
003
004 /*1.4 弧度 */
005 transform: rotate(1.4rad);
006
007 /*0.5 轉或 180 度 (1 轉=360 度) */
008 transform: rotate(0.5turn);

```

圖 16 | 這裡我們使用了 grad (Gradians 梯度)、rad (Radians 弧度) 和 turn (Turns 旋轉) 來作為指定 HTML 元素旋轉角度的替代方案。

以不同的方法來指定值，在許多其他 CSS 屬性中並不罕見。例如 #F00、#FF0000、red、rgb(255, 255, 255) 和 rgba(255, 255, 255, 1.0) 指定的都是完全相同的顏色。

1.6 CSS 註釋

在程式碼中加入註釋時，CSS 僅支援「區塊註釋」語法。

它的寫法是在文字區塊的前後加上 /* 註釋 */ 符號。

```

001 /* 使用十六進制值將字體顏色設為白色 */
002 p { color: #FFFFFF; }
003
004 /* 使用十六進制值將字體顏色設為白色 */
005 p { color: #FFF; }
006
007 /* 使用顏色名將字體顏色設為白色 */
008 p { color: white; }
009
010 /* 使用 RGB 值將字體顏色設為白色 */
011 p { color: rgb(255,255,255); }
012
013 /* 製作 CSS 變數 --white-color (留意一下雙短槓) */
014 :root { --white-color: rgba(255, 255, 255, 1.0); }
015
016 /* 使用 CSS 變數將字體顏色設為白色 */
017 p { color: var(--white-color); }

```

圖 17 | 注意一下相同的 color 屬性採用了不同類型的值。使用 CSS 變數時，變數名稱前面要加上雙短槓。

3 偽選擇器

在 CSS 中，偽選擇器是任何以冒號 (:) 為開頭的選擇器，通常會附加到另一個元素名稱（通常是父容器）的末尾。它們也稱為「偽 class」。

偽選擇器 `:first-child` 和 `:last-child` 是用來選取子元素列表中的第一個和最後一個元素。

另一個偽選擇器的範例是 `:nth-child`，它是用來選取一組元素中或甚至 HTML 表格中之列或欄的一系列元素。

我們來看一些偽選擇器的使用案例。

它們要與其他元素選擇器一起使用才會有效果。只要看看偽選擇器如何影響 HTML 表格，就能輕鬆快速地了解偽選擇器的運作原理，因為表格中的子元素有兩個維度（列 x 欄）。

你可以使用 `table tr:first-child` 來選擇第一列的所有項目：




圖 28 | `table tr:first-child`



圖 43 | CSS 中沒有 margin-box，因為在定義下，margin 指的是內容周圍的空間。

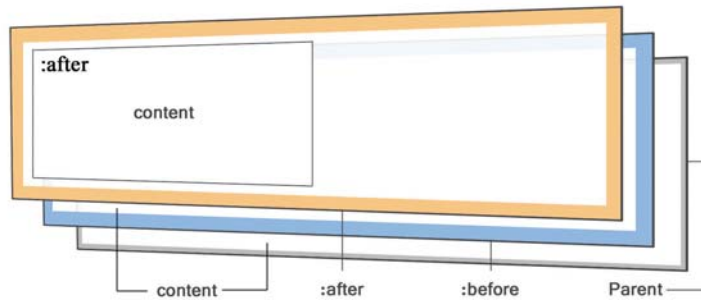


圖 44 | 一個 HTML 元素不僅是外表而已，還有很多細節。

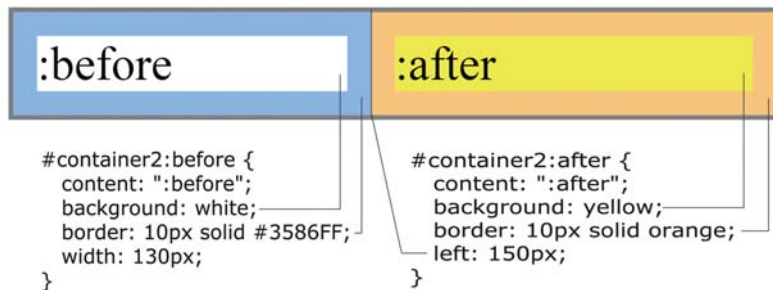


圖 45 | :before 和 :after 元素都是單一 HTML 元素的一部分。你甚至可以套用 position: absolute 而不必建立任何新元素就可以排列它們的位置！

7 邊距、圓角、陰影和 Z 索引

接下來我選擇了這幾個主題（無特定順序）來簡單示範常用的 CSS 屬性。

7.1 邊界半徑

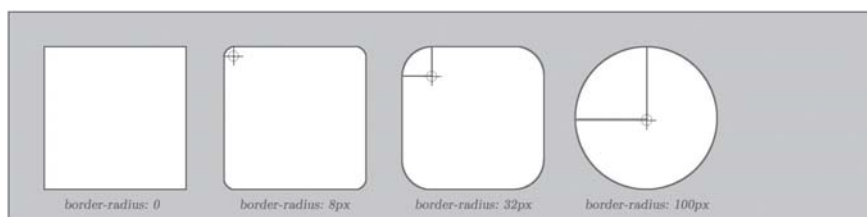


圖 97 | `border-radius` 屬性會在方形或矩形 HTML 元素上加上圓角。

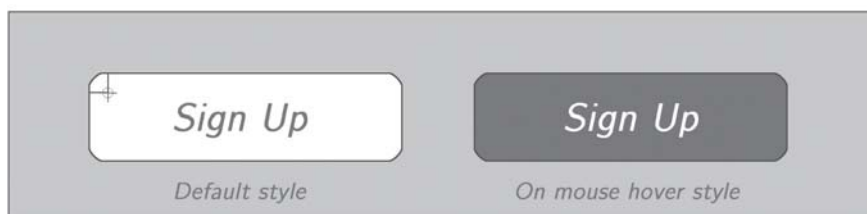
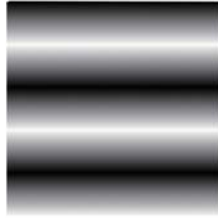
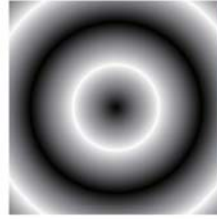


圖 98 | 使用 `:hover` 偽選擇器，你可以決定當滑鼠停在元素上方（進入其區域）時，應該發生什麼事。



repeating-linear-gradient
(white 100px,
black 200px,
white 300px);



repeating-radial-gradient
(white 100px,
black 200px,
white 300px);

圖 137 | 使用 `repeating-linear-gradient` 和 `repeating-radial-gradient` 可以產生重複的線性和放射狀圖案。你可以依據需要，一次設定數個重複的顏色值。別忘了用逗號隔開！



linear-gradient
hsl(0,100%,50%),
hsl(50,100%,50%),
hsl(100,100%,50%),
hsl(150,100%,50%),
hsl(200,100%,50%),
hsl(250,100%,50%),
hsl(300,100%,50%)



linear-gradient
hsl(0,100%,50%),
hsl(50,100%,50%),
hsl(300,100%,50%)

圖 138 | 最後是可以使用一系列 HSL 值建立的最進階漸層類型。HSL 值沒有名稱或 RGB 同等值，它們是從 0 到 300 的數字。請參見下面的說明。

19.3 Direction (方向)

你可以透過反轉來設定項目的走向。



圖 195 | `flex-direction: row-reverse` 改變了項目列表的方向。預設值為 `row`，代表它是從左到右，一如預期！

19.4 Wrap (換行)

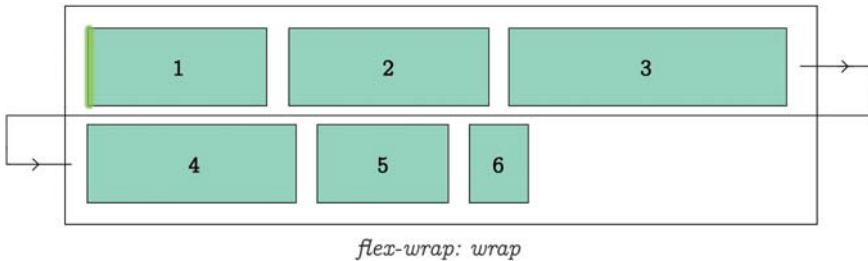


圖 196 | `flex-wrap: wrap` 決定了父容器空間不足時，項目該如何換行。

19.5 Flow (流向)

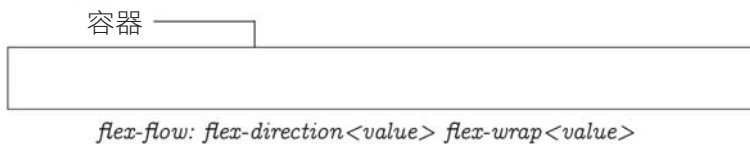


圖 197 | `flex-flow` 是 `flex-direction` 和 `flex-wrap` 的速記形式，讓你只使用一個屬性名稱就能指定兩者。

21 CSS 網格 —— 使用模版區域

先前的單元中曾討論過模版區域，但要如何利用它們來構建實際的版面配置呢？

請記得，為求創新，CSS Grid 的設計具有開放性，無需遵循任何公式或模式。

你會自然發現的一種常見技巧，是使用 `grid-template-areas` 將版面配置劃分為多個區域，並以字元串格式為每一列提供區域名稱，如下圖範例所示：

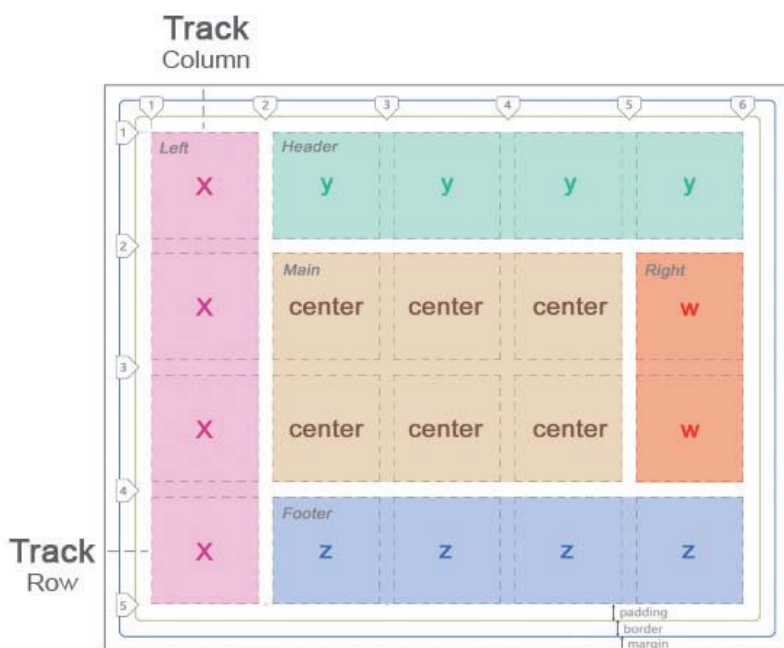


圖 216 | 你不能建立不規則形狀的區域。它們必須是正方形或矩形。

24 正向和逆向運動學

CSS 對於逆向運動學（Inverse Kinematics）並沒有開箱即用的支援。不過，透過 `transform:rotation (degree)` 和 `transform-origin` 屬性來指定父元素和子元素之間的樞軸點，我們可以模擬出它的效果。

正向（Forward）和逆向運動學指的是將旋轉角度平移到以樞軸點彼此相連的多個物件上。運動學通常用來模擬電子遊戲中的物理動作。但我們也可以使用相同原理來讓 2D 角色產生動作。

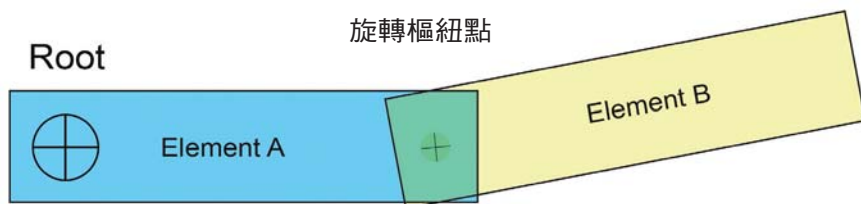


圖 274 | Root 點是主元素連接到另一個父元素或一個空間之虛構靜態點的位置。如果元素 A 移動，則必然會影響元素 B，因為它們在「旋轉樞軸點」處彼此相連。這代表它要使用三角公式來計算各種角度和長度。我們可以使用 JavaScript 或使用現有的向量／三角函式庫來達成。不過幸運的是，CSS 已經透過原生的 `transform-origin` 屬性為這些類型的元素動力學提供了支援。

「正向運動學」是當元素 A 移動時，元素 B 的運動也會受到影響（就像連鎖反應一樣），就好像它們在一個共享的樞軸點處彼此連接一樣。

code examples/sass/div-class-atom-15-times.png

```
001 | <!-- 重複這個元素 15 次 //-->  
002 | <div class = "atom"></div>
```

圖 313 | HTML 的部分。這是一個簡化的範例。確認做出 15 個 HTML 元素，class 為 "atom"。

最後，上述操作將產生以下動畫：

code examples/sass/sass-oscillator-image.png

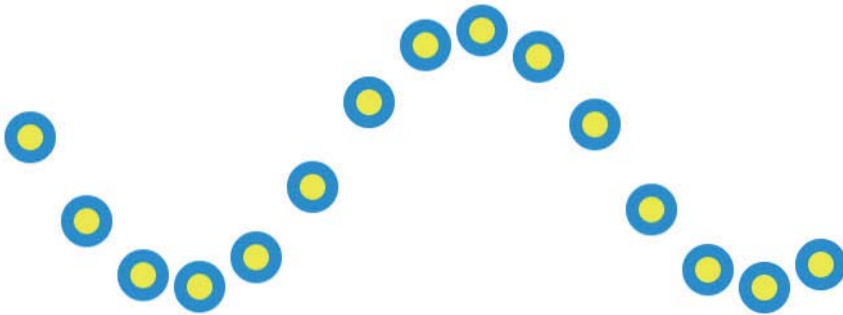


圖 314 | 上述程式碼的結果：動態的正弦波。

以上這些才用了不到 24 行的程式碼！



如果你使用的是 Vue 框架，你可以透過 v-for 指令使用此簡單的 for-loop 來運算全部 15 個元素—而不需手動輸入所有 15 個 HTML 元素，進一步省去麻煩。

26 特斯拉 CSS 藝術

儘管 CSS 語言主要是用來協助架設網站和 Web 應用軟體的版面配置，但一些才華橫溢的 UI 設計師將它推向了極限！有人認為這樣做並沒有實際用途，但不可抹滅的事實是…這些藝術家運用他們對 CSS 屬性和值的深入了解，做出了具有挑戰性的設計。

以下是「太空中的特斯拉」的 CSS 模型，是 Sasha Tran（Twitter@sa_sha26）專為本書設計的！

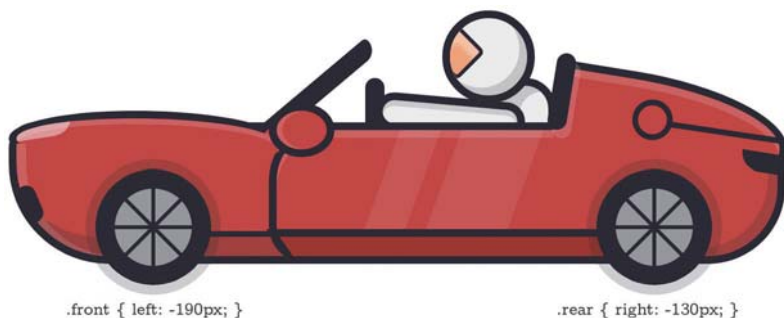


圖 317 | 太空中的特斯拉，由 Sasha Tran (@sa_sha26) 完全使用 CSS 設計而成，如果你想與才華橫溢的 UI 設計師保持聯繫，一定要追蹤她。

本書的最後這些頁面，將詳細描述汽車的每個單獨部分是如何創造的、使用了哪些 CSS 屬性等。

CSS 能耐的極限，我極力推薦她的作品網站（<http://diana-adrienne.com>），非常值得一看！網路上再也找不到其他這種類型的原創 CSS 藝術了。



圖 325 | Diana A Smith 的 CSS 畫作。你可以在以下網址欣賞到。網址為 <https://diana-adrienne.com/purecss-francine/>