

CSS 入門

CSS 為 Cascading Style Sheet 的縮寫，有人將其譯為串接樣式表。透過調整 HTML 元素的 CSS 屬性值，可以用來調整 HTML 文件的外觀樣式，設計得當的話，亦可以統一整個網站的顯示風格。

在正式進入 CSS 語法前，我們先來提一下如何調整我們有興趣的 HTML 網頁元素的外觀。

- 特定單一元素

以下兩種方式任選一種皆可：

- 直接寫進欲調整的 HTML 標記的 **style** 屬性中

```
<p>這是很<span style="color: red; font-weight: bold; font-size: larger">重要</span>的內容</p>
```

這是很**重要**的內容

- 若該標記有指定 **id** 屬性，則可透過 CSS 語法設定此 id 的 CSS 屬性值

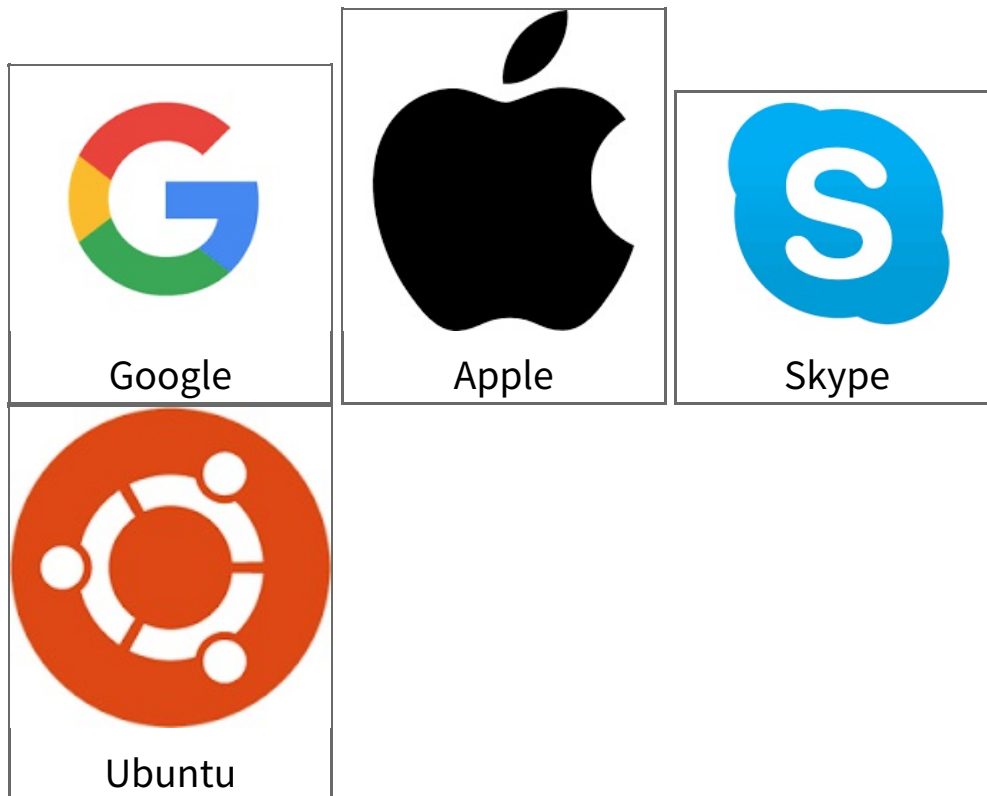
```
<div id="header">
  <div id="site-name">網站名稱</div>
</div>
<style>
#header {background: url(img/logo.jpeg) no-repeat; background-size: contain; padding-left: 60px; border: 1px solid gray}
#site-name {font-size: 24pt;}
</style>
```

網站名稱

- 一群相類似的元素

將這些相關的標記皆指定相同的 **class** 屬性，再透過 CSS 語法設定該 class 的 CSS 屬性值

```
<div id="logo-list">
  <div class="logo-item">
    
    <div class="logo-name">Google</div>
  </div>
  <div class="logo-item">
    
    <div class="logo-name">Apple</div>
  </div>
  <div class="logo-item">
    
    <div class="logo-name">Skype</div>
  </div>
  <div class="logo-item">
    
    <div class="logo-name">Ubuntu</div>
  </div>
</div>
<style>
.logo-item {display: inline-block; width: 120px; text-align: center; border: 1px solid dimgray}
.logo-img {max-width: 100%}
</style>
```



CSS 語法

CSS 是由多組規則組成，每組規則由選擇器(Selector)加上多組屬性(Property)以及值(Value)的配對所組成：

- 選擇器: 多個選擇器可以用半形逗號(,)隔開
- 屬性: CSS1~CSS3定義了許多屬性，用來控制選擇器的樣式
- 值: 該屬性所接受的設定值，若該屬性設定值同時分成好幾段，各段則以空格隔開

一組 CSS 規則大概看起來會像下面這樣：

```
選擇器 {  
    屬性1: 值11 值12 值13;  
    屬性2: 值2;  
    屬性3: 值31 值32;  
    ...餘類推...  
}
```

寫在哪裡

CSS 規則組要寫在哪?

- 當成 `<style></style>` 標記的內容

```
<style>
.logo-item {display: inline-block; width: 120px; text-align: center; border: 1px solid dimgray}
.logo-img {max-width: 100%}
</style>
```

- 寫成獨立的外部檔案，再透過 `<link />` 標記載入

```
<link type="text/css" rel="stylesheet"
href="css/layout.css" />
```

如果在開發網站的話，為了維持視覺設計的一致性，建議採用第二種方式，除了可以縮小 HTML 檔案本身的資料量之外，由於網站各頁面皆載入相同的 css 檔案，因為瀏覽器的快取(cache)機制，css 檔案只有第一次需要透過網路取得，之後皆可透過瀏覽器的快取檔案取得，可減少網路頻寬的使用。

選擇器

基本選擇器

- 類型選擇器(HTML 標籤，如 h1, p, a 等)

```
element
```

- 類別選擇器(以class屬性選擇)

```
.class
```

- ID選擇器(以id屬性選擇)

```
#id
```

注意：同一個頁面上的 id 應該為**唯一**且**不可重複**的。

- 萬用選擇器

```
*
```

- 屬性選擇器

```
[attr]
```

屬性選擇器允許用戶自訂屬性名稱，而不僅僅限於id，class屬性。

選擇器	說明
<code>[attr]</code>	元素有 <code>attr</code> 這個屬性
<code>[attr=value]</code>	<code>attr</code> 的屬性值為 <code>value</code>
<code>[attr~=value]</code>	<code>attr</code> 屬性值中以空白分隔的字串中其中一個是 <code>value</code>
<code>[attr =value]</code>	<code>attr</code> 屬性值以 <code>value</code> 或 <code>value-</code> 開頭
<code>[attr^=value]</code>	<code>attr</code> 屬性值以 <code>value</code> 開頭
<code>[attr\$=value]</code>	<code>attr</code> 屬性值以 <code>value</code> 結尾
<code>[attr*=value]</code>	<code>attr</code> 屬性值包含 <code>value</code> 字串

假設有一段 HTML 碼如下：

```
<ul class="filelist">
  <li><a id="f1" href="file/201701081109.doc">檔案1</a>
</li>
  <li><a id="f2" href="file/201701091047.pdf">檔案2</a>
</li>
```

```
<li><a id="f3" href="file/201701091052.pdf">檔案3</a>
</li>
<li><a id="f4" href="file/201702010909.xls">檔案4</a>
</li>
</div>
```

舉例來說，下面這個選擇器的寫法：

```
a[href $= ".pdf"]
```

會篩選出網頁中所有目標網址以 **.pdf** 結尾的 **a** 元素，換句話說，會選取到所有連結目標為 pdf 檔案的超連結元素。以上面的例子來說，會選取到 **id** 為 **f2** 及 **f3** 兩個 **a** 元素。

虛擬元素選擇器(pseudo-element)

虛擬元素可以選擇不在 HTML 結構中的資訊，或元素的特定部份。以下列出幾個較常用的虛擬元素選擇器：

虛擬元素	說明	範例
::after	元素之後	p::after
::before	元素之前	p::before
::first-letter	文字的第一個字元	p::first-letter
::first-line	第一行文字	p::first-line

虛擬類別選擇器(pseudo-class)

虛擬類別指的是元素的特定狀態，以下列出幾個較常使用的虛擬類別選擇器：

虛擬類別	說明	範例
:link	超連結未被存取過	a:link

:visted	超連結被存取過	a:visited
:active	超連結正被按下	a:active
:hover	滑鼠正移動到元素上	.menu-item:hover
:empty	該元素無內容	p:empty
:first-child	該元素為其父項的第一個子項	p:first-child
:first-of-type	該元素為其父項的第一個同類子項	p:first-of-type
:last-child	該元素為其父項的最後一個子項	p:last-child
:last-of-type	該元素為其父項的最後一個同類子項	p:last-of-type
:nth-child(n)	該元素為其父項的第n個子項	p:nth-child(2)
:nth-last-child(n)	該元素為其父項的倒數第n個子項	p:nth-last-child(3)
:nth-of-type(n)	該元素為其父項的第n個同類子項	p:nth-of-type(2)
:nth-last-of-type(n)	該元素為其父項的倒數第n個同類子項	p:nth-last-of-type(2)

要特別說明的是 **:nth-** 開頭的這4組虛擬類別選擇器，它們在圓括弧裡可以額外接受參數，參數可以是以下幾種樣態：

- 特定數值，例 **:nth-child(3)**
只會選到該特定數值序位的元素
- **odd** 或 **even**，例 **:nth-child(odd)**
odd 會選到第1, 3, 5, 7 ... 等奇數序位的元素，而 **even** 則會選到第2, 4, 6, 8 ... 等偶數序位的元素
- **An+B**，其中 A, B 皆為整數，例 **:nth-child(3n+1)**
每 A 個為一個循環，會選出所有循環中第 B 個元素。如 **3n+1** 則會選出第 1, 4, 7, 10 ... 個元素。

選擇組合符號(Combinator)

前面所述的選擇器可以透過下列 4 個組合符號來達到更精確的篩選：

,	或	<code>h1, p</code>	篩選出所有 <code>h1</code> 或 <code>p</code> 元素
<code>></code>	下一層元素	<code>.logo-list > div</code>	篩選出 <code>logo-list</code> 類別下一層所有的 <code>div</code> 元素
<code>~</code>	有相同父項的元素	<code>.logo-img ~ div</code>	篩選出與 <code>.logo-img</code> 類別同一層的所有 <code>div</code> 元素
	(空格)後代元素	<code>.content p</code>	篩選出 <code>.content</code> 類別內所有的 <code>p</code> 元素(不論層數)

常用的 CSS 屬性

顏色

在 CSS 中顏色屬性值的設法有以下幾種寫法：

- 顏色名稱

HTML 與 CSS 規範中預先定義了約 140 個顏色名稱，請參見[w3school: HTML Color Names](https://www.w3schools.com/html/html_color_names.asp)

例：`steelblue` 、`firebrick` 

- 16進位色碼 `#rrggbb`

以 # 開頭，後面跟著 6 個 16 進位數字，兩兩一組，第 1 組代表紅色強度，第 2 組代表綠色強度，第 3 組代表藍色強度，每組強度由 00(0) ~ ff(255)

例：`#87cefa` 、`#808000` 

- RGB `rgb(R, G, B)`


R, G, B 分別代表紅色、綠色、藍色的強度，數值範圍皆為 0~255。

例：`rgb(208, 124, 97)` 

- RGBA `rgba(R, G, B, A)`

R, G, B 同前項說明

A 則代表透明度，範圍為 0~1 之間的一個小數，0 表示完全透明(看不到)，1 為完全不透明

例：`rgba(208, 124, 97, 0.45)` 

另外還有 **HSL**、**HSLA** 等表示法，但較少用。

與顏色相關的屬性有：

color	文字的前景顏色
background-color	元素的背景顏色

如下例為黑底白字：

```
#site-name {color: white; background-color: #000000}
```

背景

- **background-color** 背景顏色
- **background-image** 背景圖片
其圖片網址的寫法為 **url(URL)**，例
url(https://zh.wikipedia.org/static/images/project-logos/zhwiki.png)
url(img/bg1.png)
若要同時套疊多張背景圖片，則以半形逗號(,)隔開，最後面的圖片會被放在最底層，最前面的會被疊在最上層。

```
.content {background-color: url(img/bg-tree.png),  
url(img/bg-flower.png);}
```

- **background-position** 背景圖片位置
設定背景圖片的起始位置，預設為元素的左上角，水平與垂直重複展開。其屬性值的設定方式有以下幾種：
 - 位置名稱

屬性值	起始位置	屬性值	起始位置	屬性值	起始位置
left top	左上角	center top	中上方	right top	右上角

left center	左中 央	center center	正中 央	right center	右中央
left bottom	左下 角	center bottom	中下 方	right bottom	右下角

註：若僅填寫一個位置名稱，則另一個缺少的位置會自動補為 center。

```
.logo {
  background-image: url(img/site-logo.png);
  background-position: right center;
}
```

- **xlength ylength**

分別代表水平位置與垂直位置，以 CSS 的長度表示法之，可使用 %、px、pt、em 等長度單位。

```
.logo {
  background-image: url(img/site-logo.png);
  background-position: 20% 15pt;
}
```

- **background-repeat** 背景圖片重覆填滿方式

屬性值	填滿方式
repeat	水平及垂直方向皆重覆填滿
repeat-x	僅水平方向重覆填滿
repeat-y	僅垂直方向重覆填滿
no-repeat	不重覆

- **background-attachment** 背景圖片捲動方式

屬性值	捲動方式
-----	------

fixed	固定不捲動
scroll	背景與元素一起捲動
local	背景跟著元素內容捲動

另外 **background** 屬性可以一次將上述幾個屬性同時設定，語法如下：

```
background: bg-color bg-image bg-position bg-repeat bg-attachment;
```

例：

```
.logo {
  background: white url(img/site-logo.png) left center no-repeat;
}
```

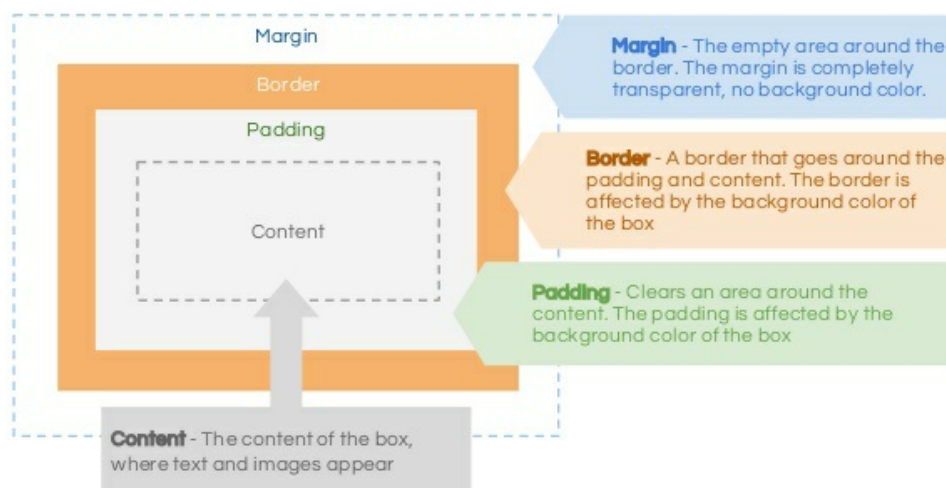
border 邊框

CSS 在處理 HTML 元素時使用一種名為 Box Model 的概念模型，如下圖：

CSS for Web Designer

by Nicuza, nicha.inth

CSS Box Model



Margin(外邊界)為與其他元素的間距，從 Border(邊框)開始往內都算是元素自身的範圍，其下還有 Padding(內邊界) 以及最裡層的 Content(元素內容)。

外邊界(margin)、邊框(border)以及內邊界(padding)都有上、右、下、左四個方向，因此若要設定這三個類型的相關屬性值時，可指定 1 ~ 4 個值，其套用對象如下：

屬性值個數	套用方向
A B C D	上方值為A, 右方為B, 下方為C, 左方為D
A B C	上方值為A, 左右為B, 下方為C
A B	上下方值為A, 左右方為B
A	上下左右四方值皆為A

- **boder-top-width, boder-right-width, border-bottom-width, border-left-width**

分別為上邊框、右邊框、下邊框、左邊框的寬度，屬性值設定方式如下：

medium	中等寬度框線
thin	細框線
thick	粗框線
X%, Xpx, Xpt, Xem	指定 X 單位寬度的框線，X為數值

另外可以用 **border-width** 屬性同時設定四個方向的邊框寬度，請參考邊框與邊界屬性值個數與套用對象表格之說明。例：

```
.border-ex { border-width: thick 5pt 1px thin }
```

- **border-top-color, border-right-color, border-bottom-color, border-left-color**

分別為上邊框、右邊框、下邊框、左邊框的顏色，屬性值設定請參考 **color**

屬性。123 **border-color** 屬性可同時設定四個方向的邊框顏色，例：

```
.border-ex { border-color: red yellow } /* 上下紅，左右黃 */
```

- **border-top-style, border-right-style, border-bottom-style, border-left-style**

分別為上邊框、右邊框、下邊框、左邊框的樣式，常用屬性值如下：

屬性值	框線樣式
none	無框
dotted	點點線框
dashed	短虛線框
solid	單線框
double	雙層線框

border-style 屬性可用時設定四個方向的邊框樣式。

- **border-top, border-right, border-bottom, border-left**

分別為上邊框、右邊框、下邊框、左邊框的寬度、樣式與顏色，屬性值設定方式如下：

寬度 樣式 顏色，例：

```
border-ex { border-left: 2px solid #306af9 }
```

- **border** 屬性可直接同時設定四個方向框線的寬度、樣式與顏色，屬性值設定方式如下：

border-width border-style border-color，例：

```
.border-ex { border: 1px solid steelblue }
```

- **border-radius** (CSS3) 屬性可設定左上、右上、右下、左下四個角落的圓角

半徑

```
<div id="border-radius-ex">圓角測試</div>
<style>
#border-radius-ex {
  border: 1px solid red;
  border-radius: 2pt 3pt 4pt 5pt;
}
</style>
```

圓角測試

margin 外邊界

- **margin-top, margin-right, margin-bottom, margin-left**
分別為上外邊界、右外邊界、下外邊界、左外邊界的寬度，屬性值設定方式如下：

X%, Xpx,	指定 X 單位的寬度，X為數值。若單位為 %，則計算依據
Xpt, Xem	為該元素上層元素之寬度

auto	由瀏覽器自動計算
-------------	----------

- **margin** 屬性可直接設定四個方向的外邊界寬度，例：

```
.border-ex { margin: 0 2px 0 3px }
```

padding 內邊界

- **padding-top, padding-right, padding-bottom, padding-left**
分別為上內邊界、右內邊界、下內邊界、左內邊界的寬度，屬性值設定方式同外邊界之說明。
- **padding** 屬性可直接設定四個方向的內邊界寬度

box 盒模型相關

- **position** 指定元素的定位參考對象，其常用屬性值如下：

屬性值	說明
static	元素依其在文件流內的順序定位，此為預設值
fixed	定位方式為與瀏覽器視窗的相對位置
relative	定位方式為與其正常位置的相對位置，例： left: 10px 表示元素最左側額外偏移10px
absolute	定位方式為與其第一個已定位(非static)的前代元素的相對位置

- **left, top, right, bottom**
分別為該元素最左邊、最上面、最右邊及最下面相對於定位參考對象的偏移量。若元素的 **position** 屬性值為 **static**，則此四個屬性的設定值無效。
- **display** 設定元素的顯示方式，常用屬性值如下：

屬性值	說明
inline	以行內元素方式顯示(後方元素不換行，像或<a>)，不可指定寬度與高度
block	以區塊元素方式顯示(後方元素會自動換行，像<p>或<h1>)，可指定寬度與高度
inline-block	兼具行內與區塊元素之特性，可指定寬高，但後方元素不換行
none	不顯示，頁面上也不會保留該元素的空間

- **width, height** 指定元素的寬度與高度
屬性值設定方式同外邊界之說明
- **float** 將元素設定為浮動元素

說明

屬性值

left	往左浮動
------	------

right	往右浮動
-------	------

none	依其原本在文字流中的位置不浮動，為預設值
------	----------------------

直接看範例比較好懂：

```
<div id="float-ex" class="div-ex">
  Mozilla Firefox (簡稱Firefox) ，中文俗稱火狐（中國大陸官方有使用但未註冊為
商標），是一個自由及開放原始碼的網頁瀏覽器[18]，由Mozilla基金
會及其子公司Mozilla公司開發。Firefox支援Windows、OS X及
Linux，其行動版支援Android及Firefox OS，這些版本的Firefox使
用Gecko來排版網頁，Gecko是一個執行當前與預期之網頁標準的排版引
擎，而在2015年釋出的Firefox for iOS則非使用Gecko。
</div>
<div class="div-ex">
  Mozilla Firefox (簡稱Firefox) ，中文俗稱火狐（中國大陸官方有使用但未註冊為
商標），是一個自由及開放原始碼的網頁瀏覽器[18]，由Mozilla基金
會及其子公司Mozilla公司開發。Firefox支援Windows、OS X及
Linux，其行動版支援Android及Firefox OS，這些版本的Firefox使
用Gecko來排版網頁，Gecko是一個執行當前與預期之網頁標準的排版引
擎，而在2015年釋出的Firefox for iOS則非使用Gecko。
</div>

<style>
#float-ex>img {float: right}
.div-ex>img {width: 120px}
</style>
```

Mozilla Firefox（簡稱Firefox），中文俗稱火狐（中國大陸官方有使用但未

註冊為商標），是一個自由及開放原始碼的網頁瀏覽器[18]，由Mozilla基金會及其子公司Mozilla公司開發。Firefox支援Windows、OS X及Linux，其行動版支援Android及Firefox OS，這些版本的Firefox使用Gecko來排版網頁，Gecko是一個執行當前與預期之網頁標準的排版引擎，而在2015年釋出的Firefox for iOS則非使用Gecko。



Mozilla Firefox（簡稱Firefox），中文俗稱火狐（中國大陸官方有使用但未註冊為商標），是一個自由及開放原始碼的網頁瀏覽器[18]，由Mozilla基金會及其子公司Mozilla公司開發。Firefox支援Windows、OS X及Linux，其行動版支援Android及Firefox OS，這些版本的Firefox使用Gecko來排版網頁，Gecko是一個執行當前與預期之網頁標準的排版引擎，而在2015年釋出的Firefox for iOS則非使用Gecko。

- **clear** 消除浮動元素

當某元素被設定為浮動元素後，從該元素之後的文字皆會環繞在該元素周圍，有時可能會影響到其它不相關的段落文字，此時便可透過 **clear** 屬性來解決這個問題。

none 預設值，允許左右兩側有浮動元素

left 禁止左側有浮動元素

right 禁止右側有浮動元素

both 左右兩側皆不允許有浮動元素

同樣直接看範例比較清楚：

```
<div id="clear-none-ex" class="div-ex">
  Mozilla Firefox（簡稱Firefox），中文俗稱火狐（中國大陸官方有使用但未註冊為
  商標），是一個自由及開放原始碼的網頁瀏覽器[18]，由Mozilla基金
```

會及其子公司Mozilla公司開發。

Firefox支援Windows、OS X及Linux，其行動版支援Android及Firefox OS，這些版本的Firefox使用Gecko來排版網頁，Gecko是一個執行當前與預期之網頁標準的排版引擎，而在2015年釋出的Firefox for iOS則非使用Gecko。

</div>

<div id="clear-left-ex" class="div-ex">

Mozilla Firefox（簡稱Firefox），中文俗稱火狐（中國大陸官方有使用但未註冊為商標），是一個自由及開放原始碼的網頁瀏覽器[18]，由Mozilla基金會及其子公司Mozilla公司開發。

Firefox支援Windows、OS X及Linux，其行動版支援Android及Firefox OS，這些版本的Firefox使用Gecko來排版網頁，Gecko是一個執行當前與預期之網頁標準的排版引擎，而在2015年釋出的Firefox for iOS則非使用Gecko。

</div>

<div id="clear-right-ex" class="div-ex">

Mozilla Firefox（簡稱Firefox），中文俗稱火狐（中國大陸官方有使用但未註冊為商標），是一個自由及開放原始碼的網頁瀏覽器[18]，由Mozilla基金會及其子公司Mozilla公司開發。

Firefox支援Windows、OS X及Linux，其行動版支援Android及Firefox OS，這些版本的Firefox使用Gecko來排版網頁，Gecko是一個執行當前與預期之網頁標準的排版引擎，而在2015年釋出的Firefox for iOS則非使用Gecko。

</div>

<div id="clear-both-ex" class="div-ex">

Mozilla Firefox（簡稱Firefox），中文俗稱火狐（中國大陸官方有使用但未註冊為商標），是一個自由及開放原始碼的網頁瀏覽器[18]，由Mozilla基金會及其子公司Mozilla公司開發。

Firefox支援Windows、OS X及Linux，其行動版支援Android及Firefox OS，這些版本的Firefox使用Gecko來排版網頁，Gecko是一個執行當前與預期之網頁標準的排版引擎，而在2015年釋出的Firefox

```
for iOS則非使用Gecko。
</div>
<style>
div[id]="clear">img {float:right}
#clear-left-ex>br {clear:left}
#clear-right-ex>br {clear:right}
#clear-both-ex>br {clear:both}
</style>
```

Mozilla Firefox（簡稱Firefox），中文俗稱火狐（中國大陸官方有使用但未註冊為商標），是一個自由及開放原始碼的網頁瀏覽器[18]，由Mozilla基金會及其子公司Mozilla公司開發。



Firefox支援Windows、OS X及Linux，其行動版支援Android及Firefox OS，這些版本的Firefox使用Gecko來排版網頁，Gecko是一個執行當前與預期之網頁標準的排版引擎，而在2015年釋出的Firefox for iOS則非使用Gecko。

Mozilla Firefox（簡稱Firefox），中文俗稱火狐（中國大陸官方有使用但未註冊為商標），是一個自由及開放原始碼的網頁瀏覽器[18]，由Mozilla基金會及其子公司Mozilla公司開發。



Firefox支援Windows、OS X及Linux，其行動版支援Android及Firefox OS，這些版本的Firefox使用Gecko來排版網頁，Gecko是一個執行當前與預期之網頁標準的排版引擎，而在2015年釋出的Firefox for iOS則非使用Gecko。

Mozilla Firefox（簡稱Firefox），中文俗稱火狐（中國大陸官方有使用但未註冊為商標），是一個自由及開放原始碼的網頁瀏覽器[18]，由Mozilla基金會及其子公司Mozilla公司開發。



Firefox支援Windows、OS X及Linux，其行動版支援Android及Firefox OS，這些版本的Firefox使用Gecko來排版網頁，Gecko是一個執行當前與預期之網頁標準的排版引擎，而在2015年釋出的Firefox for iOS則非使用Gecko。

Mozilla Firefox（簡稱Firefox），中文俗稱火狐（中國大陸官方有使用但未註冊為商標），是一個自由及開放原始碼的網頁瀏覽器[18]，由Mozilla基金會及其子公司Mozilla公司開發。



Firefox支援Windows、OS X及Linux，其行動版支援Android及Firefox OS，這些版本的Firefox使用Gecko來排版網頁，Gecko是一個執行當前與預期之網頁標準的排版引擎，而在2015年釋出的Firefox for iOS則非使用Gecko。

- **visibility** 設定元素是否可見，常用屬性值如下：

屬性值	說明
visible	可見，此為預設值
hidden	隱藏元素，但仍會保留元素佔用空間

text 文字屬性

- **line-height** 設定文字列高，常用屬性值如下：

屬性值	說明
normal	普通行高，此為預設值
N	N為正整數，表示要將行高設為目前字型大小的N倍
X%, Xpx, Xpt, Xem	指定 X 單位的高度，X為數值。若單位為 %，則計算依據為目前字型大小

- **text-align** 設定文字對齊方式，常用屬性值如下：

屬性值	說明
left	靠左對齊
right	靠右對齊
center	置中對齊
justify	分散對齊，將元素內每行的頭尾元素對齊

- **text-indent** 首行文字縮排，常用屬性值如下：

屬性值	說明
X%, Xpx, Xpt, Xem	指定 X 單位的高度，X為數值。若單位為 % ，則計算依據為父項元素的寬度

- **text-decoration** 設定文字的附加裝飾，常用屬性值如下：

屬性值	說明
none	正常文字無裝飾，此為預設值
underline	附加底線
overline	附加頂線
line-through	附加刪除線

font 字型

- **font-family** 設定字型族系

可指定一個備用字型清單，如果第一個字型瀏覽器無法使用，會試著使用清單中的下一個備用字型，直到找到可用字型為止。

```
body {
  font-family: "Heiti TC", "微軟正黑體", "Microsoft Sans
  Serif", Helvetica, Geneva, sans-serif, serif;
}
```

- **font-size** 設定字型尺寸，常用屬性值如下：

屬性值	說明
xx-small 、 x-small 、 small medium large 、 x-large 、 xx-large	預先定義的字型級距
smaller 、 larger	比父項元素字型尺寸稍小或稍大

X%	指定為父項元素字型尺寸的 X%，X為數值
Xpx 、 Xpt 、 Xem	直接指定字型尺寸

例：

```
h1 {font-size: 28px}
h2 {font-size: 24px}
.item-title {font-size: larger}
.item-desc {font-size: 70%}
```

- **font-weight** 設定字型粗細，常用設定值如下：

屬性值	說明
normal	正常粗細
bold	粗體
bolder	稍粗，但實測看不太出來與 bold 的差別
lighter	稍細，但實測看不太出來與 normal 的差別
100 、 200 、 300	由最細至最粗
400 、 500 、 600	400 即為 normal ，700 等同於 bold
700 、 800 、 900	但實測僅有 normal 與 bold 兩個級別

- **font-style** 設定字型樣式(是否斜體)，常用設定值如下：

屬性值	說明
normal	正體
italic 、 oblique	斜體

- **font** 同時設定是否斜體、粗細、尺寸以及字型族系，例：

```
.site-name {font: bold 32px "Heiti TC", "微軟正黑體", "Microsoft Sans Serif", Helvetica, Geneva, sans-serif, serif }
```

list 清單

- **list-style-type** 設定清單項目符號標記的樣態，常用屬性值如下：

屬性值	說明
none	不使用項目標記符號
disc	● 實心圓點
circle	○ 空心圓點
square	■ 實心方塊
cjk-ideographic	一. 第一項 二. 第二項 三. 第三項
decimal	1. 第一項 2. 第二項 3. 第三項
decimal-leading-zero	01. 第一項 02. 第二項 03. 第三項
lower-alpha	a. 第一項 b. 第二項 c. 第三項
lower-roman	i. 第一項 ii. 第二項 iii. 第三項

upper-alpha	A. 第一項 B. 第二項 C. 第三項
upper-roman	I. 第一項 II. 第二項 III. 第三項
hiragana	あ. 第一項 い. 第二項 う. 第三項
hiragana-iroha	い. 第一項 ろ. 第二項 は. 第三項
katakana	ア. 第一項 イ. 第二項 ウ. 第三項
katakana-iroha	イ. 第一項 ロ. 第二項 ハ. 第三項

- **list-style-position** 指定項目符號標記的位置，常用屬性值如下：

屬性值	說明
inside	在元素內部
outside	在元素外部，此為預設值

- **list-style-image** 自訂項目符號標記圖片，常用屬性值如下：

屬性值	說明
none	不使用圖片做為項目符號標記，此為預設值。 會由 list-style-type 屬性值會決定項目符號標記的呈現樣態

url(URL)



指定來源為 URL 圖片當做項目符號標記。

例：

```
.pin-item {list-style-image: url(img/pin.png) }
```

- **list-style** 同時指定 **list-style-type** 、 **list-style-position** 、 **list-style-image** 屬性值，例：

```
.insie-pin-item {list-style: inside url(img/pin.png) }
```

table 表格

- **boder-collapse** 合併表格欄位框線，常用設定值用下：

屬性值	說明
separate	儲存格邊框彼此分開，此為預設值
collapse	相鄰儲存格邊框合併為單一邊框

效果如下：

border-collapse: separate

1	2
3	4

border-collapse: collapse

12
34

例：

```
table.collapse {border-collapse: collapse}
```

- **border-spacing** 設定表格相鄰儲存格的間距
僅在 **border-collapse** 為 **separate** 時有效，可接受 2 個值分別代表水平間距與垂直間距，可採用 **Xpx**、**Xpt**、**Xcm** 等格式指定間距寬度。例：

```
table.test-format {  
    border-collapse: separate;  
    border-spacing: 2px 5px;  
}
```

參考資料

- [維基百科階層式樣式表](#)
- [w3school: CSS Reference](#)