

灝邦集團美編人員課程：「網頁設計與平面設計」講義

關於這份教材

城邦集團於 2009/10 與 Jedi 聯繫開始規劃這項課程，原訂開課日期為 2010 年春季，後來城邦集團如期收到初期教材後，卻於開課日期前 3 個工作天提出課程延期；稍後城邦集團負責此項課程的人員離職，改由人資部門主管直接聯繫，但最後在多次延期之後，此課程一直沒有展開。

目前為止，城邦集團除了支付因課程規劃改變所造成的交通住宿損失（就是退票、退訂房間所造成的費用差額）外，並沒有支付任何款項，對此處所提供的課程教材亦無持有任何權利。

簡單來說，因為城邦集團想開這個課程，所以 Jedi 編撰了這份教材；然後因為城邦集團一直沒有展開課程，而且因為一開始沒有先簽約（因為是熟人牽線，等於是賣人情……連報價都是人情價），Jedi 又不希望這些辛苦寫出來的教材遭到埋沒，所以在這邊公開提供給大眾使用。

最後致城邦集團：下一次你們來找我，可就沒有人情價這回事了喔。

本系列的課程，係設計給從平面設計轉戰網頁設計的同儕，目標是在兩個月內，讓從未接觸過網頁設計的美編人員，也能著手製作正確的網頁。這一系列課程會兼重理論與實務，該講授的重要觀念不會少，該實際演練的時候也不會省，因為觀念正確了，做起網頁來纔不會心虛；親手實作驗證理論過，印象纔會深刻。

在第一堂課中，所要講述的是「網頁」這種媒體的特性——此處並不是要講述抽象的資訊社會學或資訊傳播學，而是要以「平面設計」的觀點為基礎，來強調該注意、該突破、該轉變的重要細節，以及將來開始著手撰寫網頁時，所會用到的資訊細節。

畫布特性

當網頁設計超越了科學家撰寫論文的需求後，參與網頁設計技術發展的人當中，除了數學家跟資訊科學家外，還包括許多原本從事平面設計的人，因此網頁設計在許多方面都受到平面設計的影響；即使到了今天，我們仍然可以從網頁設計的專業術語當中，看到平面設計遺留的影子。原本從事平面設計的設計師，要轉戰網頁設計的領域時，如果能夠掌握這些設計概念與技術的轉變，就能夠更迅速地理解網頁設計與平面設計這兩者間的異同。

在平面設計的世界當中，印製用的紙張有著既有的尺寸規格：全開、菊全開、對開、菊對開、四開、菊四開、八開、**D** 八開、十六開、菊十六開、三十二開、菊三十二開、六十四開，要用哪一種規格的紙張來印刷，是平面設計師在一開始就決定好了的；一本菊十六開大小的書，傳遞到讀者手上時，讀者看到的就會是菊十六開的大小，就算有印刷裁切上的誤差，也會是在可容許的微小差距內。當設計師著手進行設計時，就會根據目標成品的尺寸，選擇對應的畫布尺寸，或者說是版面尺寸，以便進行「所見即所得」的視覺設計。

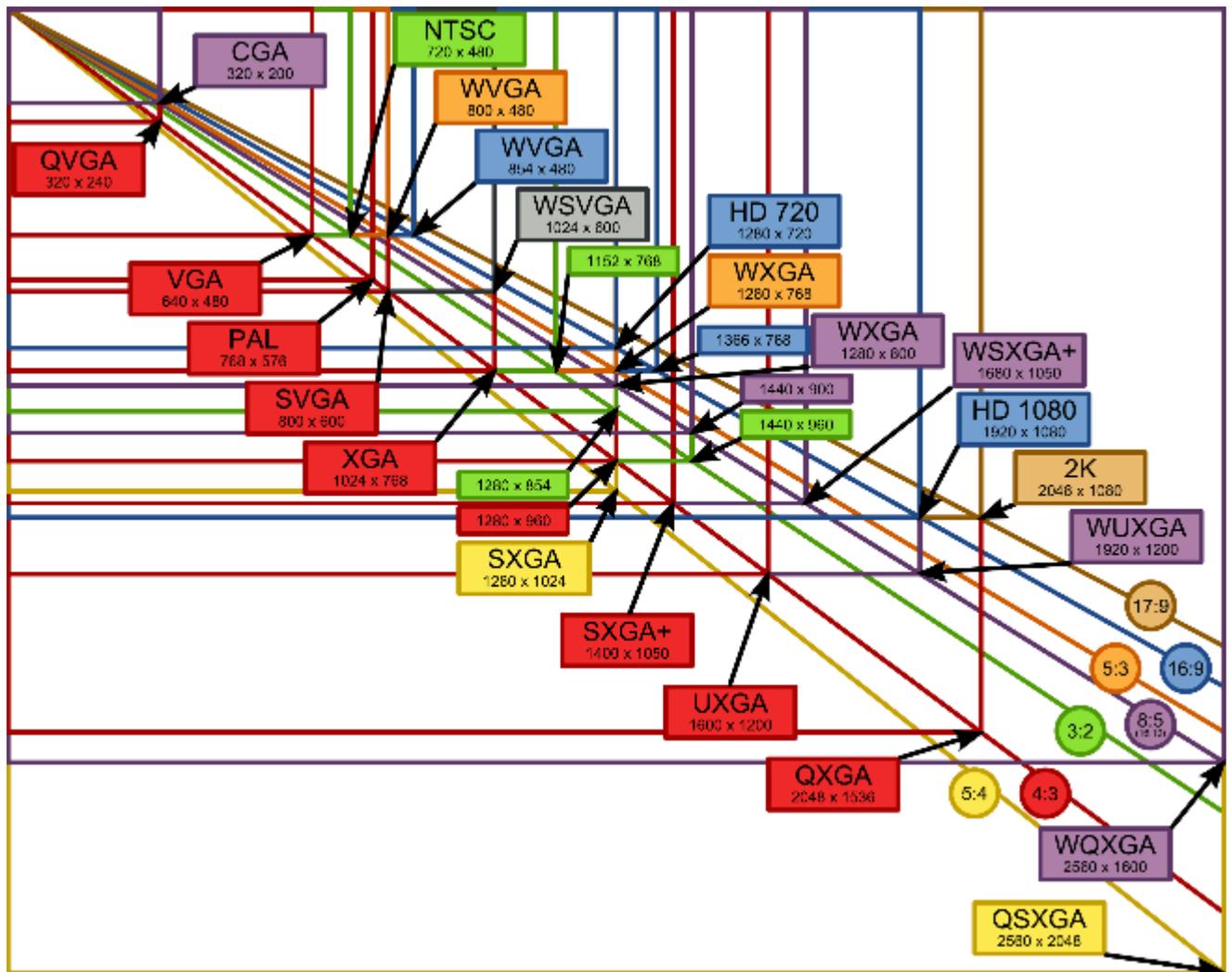
在網頁設計的世界當中，相當於印製用的紙張的，是網頁使用者的螢幕；螢幕也有一些尺寸規格，但是光是這部份就比平面印刷用的紙張還要複雜許多。螢幕的尺寸是以「對角線」的長度來計算，小至 1.5 吋、大至 200 吋皆有之；但是對角線的長度僅能大致描述螢幕帶來的使用感受，而無法確實描述其「規格」——不同螢幕的長、寬比例並不相同，常見的螢幕長、寬比例包括 5:4、

4:3、3:2、8:5 (16:10)、5:3、16:9、17:9，即使是對角線等長的兩個螢幕，所能呈現內容的面積也有可能不同。

更進一步來說，面積較大的螢幕，也不見得能夠用來描繪較多的畫面內容。這關係到設計內容呈現時所使用的單位。

在平面設計的世界中，內容的呈現係按照度量衡單位來處理：設計師畫出一個直徑 10 公分的圓形，則長、寬窄於 10 公分的紙張無論如何也無法完整呈現這個圓形。在螢幕上，內容的設計與呈現則是以「像素」為單位；「像素」是螢幕上用來顯示一個色點的最小單位，這個色點實際佔據的面積會因為螢幕的差異而有所不同，也不見得會是正多邊型；通常我們會用「解析度」（此處的「解析度」與「dpi」是兩回事，切勿混淆！）來描述螢幕內的像素數量，例如「1024×768」表示這個螢幕寬有 1024 個像素、高有 768 個像素。像素越多的螢幕，就能容納更多的畫面內容；換句話說，一個 12.1 吋、解析度為 1600×1200 的螢幕，比一個 20 吋、解析度為 1024×768 的螢幕，能呈現更多內容。在螢幕解析度的規格當中，有些特定的解析度另外會有一個名稱，但是這些名稱與解析度之間並非一對一的關係，因此較少被使用；常見的解析度規格包括了：320×200 (CGA)、320×240 (QVGA)、640×480 (VGA)、720×480 (NTSC)、800×480 (WVGA)、854×480 (WVGA)、768×576 (PAL)、800×600 (SVGA)、1024×600 (WSVGA)、1024×768 (XGA)、1152×768、1280×720 (HD 720)、1280×768 (WXGA)、1280×800 (WXGA)、1366×768、1280×854、1280×960、1280×1024 (SXGA)、1440×900、1440×960、1400×1050 (SXGA+)、1680×1050 (WSXGA+)、1600×1200 (UXGA)、1920×1080 (HD 1080)、2048×1080 (2K)、1920×1200 (WUXGA)、2048×1536 (QXGA)、2560×1600 (WQXGA)、2560×2048 (QSXGA) 等。這麼多種不同的解析

度，並不是讓網頁設計變得複雜的主因；真正的原因在於：網頁設計師不可能預期網頁使用者的螢幕解析度到底是哪一種。（實際上則是：通通都有可能！）



接著回到平面設計。設計一本菊十六開大小的書籍時，其實書本內頁的大小並不會是菊十六開，而會是略小於菊十六開；因為印刷的過程中會裁切掉部分紙張邊緣以求平整對齊，而且靠近書脊裝訂處的部份無法完全展開，所以在平面設計的慣例上，會標示出這樣的裁切參考線，設計內容版面時也會留天、留地、留邊，有需要的時候纔做「出血」處理。設計網頁時也是類似的情況：使用者可能會把網頁瀏覽器（也就是用來操作網頁的軟體）放大到佔滿整個螢幕的空間，也有可能因種種需求，而縮成較小的視窗；不論視窗或大或小，也不能保證視窗的尺寸就是使用者看到網頁的尺寸，因為不同的瀏覽器會有不同的

工具列及選單配置方式，使用者也可能任意調整或增刪這些工具列，因此幾乎沒有辦法預測網頁在所有使用者的電腦上，會以何種面貌呈現。

在這麼充滿不確定的設計挑戰之下，網頁設計師仍然必須要盡可能善用使用者網頁瀏覽器中的可視區域。「可視區域」指的是網頁瀏覽器當中，實際用來呈現網頁內容的區域；當網頁尺寸超過瀏覽器可視區域的尺寸時，往往會出現垂直及水平的捲動控制元件，讓使用者得以逐步瀏覽網頁的不同部分。當網頁剛載入時，最先呈現在瀏覽器可視區域之中的部分，往往就是網頁內容最左上方的區塊（當然，如果是設計由右至左的語言的網頁時，例如阿拉伯文的網頁，就會是最右上方的區塊），這個部分相當於報紙頭版的上半段，因此也有「折線以上」的說法——報紙通常是以頭版朝外對折的樣子陳列，這是報紙最搶眼的版面，也會是廣告最昂貴的版面。

即使是經營網路服務聞名的 **Google**，到現在仍然會犯下這樣的錯誤：**Google** 設計出的網頁寬度超出多數使用者的可視區域寬度，結果使用者看不到「下載檔案」的按鈕，因而離開網頁，而錯過使用相關服務的機會。

從以上這些比較當中，可以發現在平面設計時，設計師可以充分掌握讀者體驗——設計師決定要用多大的紙張、決定作品內頁的呈現方式、決定哪些內容一定會讓使用者馬上映入眼簾；但是在網頁設計當中，這些變因卻由使用者操控——不同使用者會有不同的螢幕、不同的瀏覽器視窗配置，看到的網頁呈現結果也就因此而有所不同；這種呈現結果的不同，不但出現在不同使用者之間，就連同一個使用者，每次使用網頁時也會有不同的結果與感受。這是網頁設計師必須要面對的考驗，也是從平面設計轉向網頁設計時，首先要突破的觀念。

小結

本節的重點是比較平面設計與網頁設計當中，「畫布」——施展設計的區域——的對應關係，茲表列如下：

平面	網頁
紙張	螢幕
內頁	視窗
折線	可視區域
設計師控制	使用者控制

網頁的動態特性

前一個小節最後提到了 Google 設計網頁時的失誤，這個失誤源自於 Google 對網頁設計的不熟悉。Google 的執行長多年前在一次訪問當中提到，其網頁搜尋首頁之所以如此簡潔，最早是因為 Google 不懂如何設計繁複的網頁；多年後再度回顧這番話，就會明白這並不祇是謙虛之詞，而是對 Google 內部能力的真實描述。此刻在 Google 工作的工程師多半使用很大的電腦螢幕，例如 30 吋、解析度 2560×1600 以上的螢幕就是其員工配發的標準選項之一，許多 Google 的工程師按照平面設計的概念在製作網頁，偏好固定的版面設計，並且以為所有的使用者都會看到他們所看到的網頁呈現，因而造成了這樣的慘劇。

我們可以從 Google 的慘劇之中學到許多教訓：

當網頁尺寸超出可視區域時，「往右」（或「往左」，如果文字方向是由右至左的網頁）的內容比「往下」的內容更容易錯過。這是因為多數的使用者輸入控制設備（例如滑鼠滾輪）都是以上下捲動為主要設計，而且對橫書文字來說，左右距離過長的內容並不便於閱讀。如果網頁尺寸無論如何無法控制在一個可視區域之中（基本上幾乎所有的網頁都沒辦法），那麼就應該盡可能把內

容挪至目前可視區域的底下。其實，當網頁設計師略過網頁版面設計不理時，這也是網頁的預設行爲。

網頁設計與平面設計最大的不同之處之一，在於網頁的版面配置可以動態變化，不但隨著不同的可視區域來調整，甚至可以在使用者變動可視區域（例如調整瀏覽器視窗大小、開關工具列）的時候，跟著即時變動。

相對於「固定」版面設計，這種會隨著可視區域而變動的作法，則常被稱作「流動」版面設計或「彈性」版面設計，也有人稱之爲「相對」版面設計。固定版面設計與流動／彈性版面設計也可以混合搭配使用，亦即讓部分的排版元件的尺寸固定，其餘則依情況來調整，或者是分別指定彈性組件的最大及最小尺寸，這在網頁設計當中也都是可行的，並被稱作「混合」版面設計。以下並列舉這些版面設計技法的簡要說明：

- **固定排版**：跟平面設計最相似的作法，但是也最容易造成網頁使用上的困擾
- **流動排版**：通常使用百分比作爲長度單位，讓版面隨著當時的可視區域尺寸自動縮放
- **彈性排版**：通常使用 **em** 作爲長度單位，讓版面依據當時的內容文字尺寸跟著自動調整
- **相對排版**：流動排版與彈性排版的合稱
- **混合排版**：部分內容使用固定排版，部分內容則使用相對（流動／彈性）排版的作法，可以兼具兩者的優點

這些版面設計的技術細節在此先略過不談，此處的重點是要請讀者明白網頁排版與平面排版在技法上有如此差異。

小結

本節提到了一些網頁設計的技法：（關於長度單位的部份，請參考〈長度單位與解析度〉小節的說明）

排版技法 使用長度單位 說明

固定	px	跟平面設計最相似，但容易造成使用困擾
相對流動	%	隨著可視區域尺寸縮放
彈性	em	隨著內容文字尺寸調整
混合	px、%、em	混合使用以上的技法

色彩

在平面設計當中，色彩是按照 **CMYK**（青色、洋紅色、黃色、黑色）色域來表示的「減法混色」，亦即在白色的介質（例如紙張）上，使用不同比例的四色油墨，再透過光的反射來呈色。由於網頁的主要呈現媒介並不是紙張，而是螢幕，並不適合採用 **CMYK** 色域來表達顏色，而是採用 **RGB**（紅色、綠色、藍色）色域來表示的「加法混色」，也就是從什麼都沒有的黑暗開始，疊加不同比例的三色光，來產生顏色。

最近幾年開始有網頁瀏覽器特別留意處理以 **CMYK** 表色的圖片，但是一般而論，設計網頁時最好直接根據 **RGB** 色域來定色，纔能避免色偏的問題。

設計網頁時，如果要指定顏色，有兩種方法：

- 使用色彩名稱

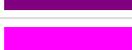
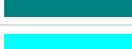
- 使用十六進制色碼

茲分別說明如下：

色彩名稱

全球資訊網協會在制定 HTML 4.01 規格的時候，定義了 16 種色彩名稱，並沿用至 CSS 1 與 CSS 2 規格。這 16 種色彩分別是：

色彩	名稱	色碼	中文色彩名稱
----	----	----	--------

	black	#000000	黑色
	silver	#c0c0c0	銀色
	gray	#808080	灰色
	white	#ffffff	白色
	maroon	#800000	栗色
	red	#ff0000	紅色
	purple	#800080	紫色
	fuchsia	#ff00ff	品紅色
	green	#008000	綠色
	lime	#00ff00	萊姆色
	olive	#808000	橄欖色
	yellow	#ffff00	黃色
	navy	#000080	藏青色
	blue	#0000ff	藍色
	teal	#008080	亮綠色
	aqua	#00ffff	水色

這 16 種色彩又稱為「HTML 色彩」，同時也是 Windows 的 VGA 色盤當中所支援的 16 種標準色彩。

後來全球資訊網協會在 CSS 2.1 規格當中，又多加入了 1 種色彩名稱：

色彩	名稱	色碼	中文色彩名稱
----	----	----	--------

	orange	#ffa500	橙色
---	--------	---------	----

多出來的這個橙色，跟前述的 16 種 HTML 色彩，則合稱「CSS 色彩」。

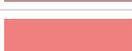
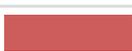
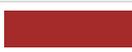
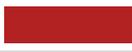
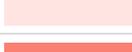
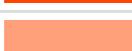
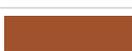
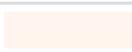
在 CSS 2 以及 CSS 2.1 當中，除了上述的 17 種 CSS 色彩外，另外還制定了 28 種「系統色彩」；所謂系統色彩，指的是根據使用者各自電腦當中所選擇的佈景主題而決定的顏色，因此這 28 種系統色彩並沒有固定的樣子。這 28 種系統色彩包括了：

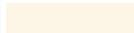
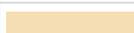
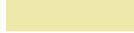
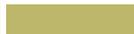
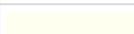
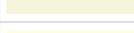
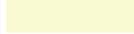
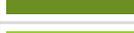
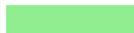
色彩名稱	說明
ActiveBorder	使用中視窗的邊框顏色
ActiveCaption	使用中視窗的視窗標題列底色
AppWorkspace	應用程式工作區域底色
Background	桌面底色
ButtonFace	按鈕底色
ButtonHighlight	被選取的按鈕的底色
ButtonShadow	按鈕的陰影顏色
ButtonText	按鈕上的文字顏色
CaptionText	使用中視窗的視窗標題列文字顏色
GrayText	非作用中的文字顏色
Highlight	反白選取文字時的底色
HighlightText	反白選取文字時的文字顏色
InactiveBorder	非作用中視窗的邊框顏色
InactiveCaption	非作用中視窗的視窗標題列底色
InactiveCaptionText	非作用中視窗的視窗標題列文字顏色
InfoBackground	資訊對話方塊的底色
InfoText	資訊對話方塊的文字顏色
Menu	選單列底色
MenuItemText	選單列文字顏色
Scrollbar	捲動控制元件的顏色
ThreeDDarkShadow	立體按鈕的深色陰影顏色
ThreeDFace	立體按鈕的底色
ThreeDHighlight	被選取的立體按鈕的底色
ThreeDLightShadow	立體按鈕的淺色陰影顏色
ThreeDShadow	立體按鈕的陰影中間色
Window	視窗元件底色
WindowFrame	視窗元件邊框顏色
WindowText	視窗元件文字顏色

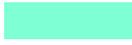
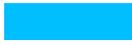
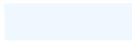
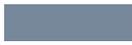
使用這 28 種系統色彩，可以用來在網頁上製作與使用者其他應用程式配色一致的操作介面。當然，使用者可能會自訂出很奇怪的色彩搭配，連帶地把用了系統色彩名稱的網頁搞得一團亂，這也是必須事先考慮到的可能後果。

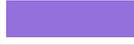
除了 HTML 色彩、CSS 色彩、系統色彩外，另外有一組非標準的 HTML 色

彩，共有 141 種顏色：

色彩	名稱	色碼	中文色彩名稱
	black	#000000	黑色
	dimgray	#696969	昏灰色
	gray	#808080	灰色
	darkgray	#a9a9a9	暗灰色
	silver	#c0c0c0	銀色
	lightgrey	#d3d3d3	亮灰色
	gainsboro	#dcdcdc	庚斯博羅灰色
	whitesmoke	#f5f5f5	白煙色
	white	ffffff	白色
	snow	fffafa	雪色
	rosybrown	#bc8f8f	玫瑰褐色
	lightcoral	#f08080	亮珊瑚色
	indianred	#cd5858	印第安紅色
	brown	#a52a2a	褐色
	firebrick	#b22222	耐火磚紅色
	maroon	#800000	栗色
	darkred	#8b0000	暗紅色
	red	ff0000	紅色
	mistyrose	ffe4e1	霧玫瑰色
	salmon	fa8072	鮭紅色
	tomato	ff6347	番茄紅色
	darksalmon	e9967a	暗鮭紅色
	coral	ff7f50	珊瑚紅色
	orangered	ff4500	橙紅色
	lightsalmon	ffa07a	亮鮭紅色
	sienna	a0522d	黃土赭色
	seashell	fff5ee	海貝色
	saddlebrown	8b4513	鞍褐色
	chocolate	d2691e	巧克力色
	peachpuff	ffdab9	粉撲桃色
	sandybrown	f4a460	沙褐色
	linen	faf0e6	亞麻色
	peru	cd853f	秘魯色
	bisque	ffe4c4	陶坯黃色
	darkorange	ff8c00	暗橙色
	antiquewhite	faebd7	古董白色

	tan	#d2b48c	日曬色
	burlywood	#deb887	硬木色
	blanchedalmond	#ffebcd	杏仁白色
	navajowhite	#ffdead	那瓦霍白色
	papayawhip	#ffefd5	番木瓜色
	khaki	#996b1f	卡其色
	moccasin	#ffe4b5	鹿皮鞋色
	oldlace	#fdf5e6	舊蕾絲色
	wheat	#f5deb3	小麥色
	orange	#ffa500	橙色
	floralwhite	#fffaf0	花卉白色
	goldenrod	#daa520	金菊色
	darkgoldenrod	#b8860b	暗金菊色
	cornsilk	#fff8dc	玉米絲色
	gold	#ffd700	金色
	lemonchiffon	#fffacd	檸檬綢色
	palegoldenrod	#eee8aa	灰金菊色
	darkkhaki	#bdb76b	暗卡其色
	ivory	#ffff0	象牙色
	beige	#f5f5dc	米黃色
	lightyellow	#ffffe0	亮黃色
	lightgoldenrodyellow	#fafad2	亮金菊黃色
	olive	#808000	橄欖色
	yellow	#ffff00	黃色
	olivedrab	#6b8e23	橄欖軍服綠色
	yellowgreen	#9acd32	黃綠色
	darkolivegreen	#556b2f	暗橄欖綠色
	greenyellow	#adff2f	綠黃色
	lawngreen	#7cfc00	草坪綠色
	chartreuse	#7fff00	查特酒綠色
	honeydew	#f0fff0	蜜瓜綠色
	darkseagreen	#8fbc8f	暗海綠色
	lightgreen	#90ee90	亮綠色
	palegreen	#98fb98	灰綠色
	forestgreen	#228b22	森林綠色
	limegreen	#32cd32	萊姆綠色
	darkgreen	#006400	暗綠色
	green	#008000	綠色
	lime	#00ff00	萊姆色

	springgreen	#00ff80	春綠色
	mediumspringgreen	#00fa9a	中春綠色
	mediumaquamarine	#66cdaa	中碧藍色
	aquamarine	#7fffd4	碧藍色
	turquoise	#30d5c8	綠松石色
	lightseagreen	#20b2aa	亮海綠色
	mediumturquoise	#48d1cc	中綠松石色
	lightcyan	#e0ffff	亮青色
	paleturquoise	#afeeee	灰綠松石色
	darkslategray	#2f4f4f	暗岩灰色
	teal	#008080	靉綠色
	darkcyan	#008b8b	暗青色
	cyan	#00ffff	青色
	aqua	#00ffff	水色
	darkturquoise	#00ced1	暗綠松石色
	cadetblue	#5f9ea0	軍服藍色
	powderblue	#e0e0e6	嬰兒粉藍色
	lightblue	#add8e6	亮藍色
	deepskyblue	#00bfff	深天藍色
	skyblue	#87ceeb	天藍色
	lightskyblue	#87cefa	亮天藍色
	steelblue	#4682b4	鋼青色
	aliceblue	#f0f8ff	愛麗絲藍色
	slategray	#708090	岩灰色
	lightslategray	#778899	亮岩灰色
	dodgerblue	#1e90ff	道奇藍色
	azure	#f0ffff	湛藍色
	lightsteelblue	#b0c4de	亮鋼青色
	cornflowerblue	#6495ed	矢車菊藍色
	royalblue	#4169e1	品藍色
	ghostwhite	#f8f8ff	幽靈白色
	lavender	#e6e6fa	薰衣草紫色
	midnightblue	#191970	午夜藍色
	navy	#000080	藏青色
	navyblue	#9afddf	藏青藍色
	darkblue	#00008b	暗藍色
	mediumblue	#0000cd	中藍色
	blue	#0000ff	藍色

	darkslateblue	#483d8b	暗岩藍色
	slateblue	#6a5acd	岩藍色
	mediumslateblue	#7b68ee	中岩藍色
	mediumpurple	#9370db	中紫紅色
	blueviolet	#8a2be2	藍紫色
	violet	#8b00ff	紫羅蘭色
	indigo	#4b0080	靛色
	darkorchid	#9932cc	暗蘭紫色
	darkviolet	#9400d3	暗紫色
	mediumorchid	#ba55d3	中蘭紫色
	thistle	#d8bfd8	薊紫色
	plum	#dda0dd	李紫色
	purple	#800080	紫色
	darkmagenta	#8b008b	暗洋紅色
	magenta	#ff00ff	洋紅色
	fuchsia	#f400a1	品紅色
	orchid	#da70d6	蘭紫色
	mediumvioletred	#c71585	中青紫紅色
	deeppink	#ff1493	深粉紅色
	hotpink	#ff69b4	暖粉紅色
	lavenderblush	#fff0f5	薰衣草紫紅色
	palevioletred	#db7093	灰紫紅色
	crimson	#dc143c	緋紅色
	pink	#ffc0cb	粉紅色
	lightpink	#ffb6c1	亮粉紅色

這 141 種顏色當中，除了與前述 CSS 顏色相同的 17 種之外，其餘的 124 種並不一定會按照上表來表示，不同的瀏覽器或上網設備，有可能會做出不同的解讀，因此盡量還是不要使用為佳。

十六進制色碼

除了以色彩的名稱來表達顏色外，也可以根據色彩在 RGB 色域當中的「座標」來表達，這種表達方式稱作「色碼」。在網頁的世界中，對每一個原色光的解像力是 8 位元，這句話的意思是說，三原色（紅、綠、藍）光按照其亮度，分別分成 2^8 等份，若分別以二進制來表達不同的程度，則從完全沒有

(00000000) 到最亮 (11111111)，共有 256 種可能；以上是一種原色光的情況，因為總共有三原色光，所以可以表達 $2^8 \times 2^8 \times 2^8 = 2^{24} = 16,777,216$ 種色彩，這也就是電腦當中的 24 位元色彩「True Color」的由來。

由於以二進制來表達 8 位元數字要用到 8 位數字（所以纔叫「8 位元」），撰寫上過於麻煩，因此實際在網頁設計當中，色碼是採用十六進制來表達，如此 2 位數字（從 00 到 ff）就足夠表達一個原色光的情況；三原色光總共 24 位元，用十六進制來表達則是 6 位數字。

以十六進制數字來表達色碼時，會在最前面加上一個「#」號，接著是表達紅色光的 2 位數字，然後是表達綠色光的 2 位數字，最後是表達藍色光的 2 位數字，從 # 號到接下來的 6 位數字間都沒有空格或其他符號隔開，一路寫到底，因此 #ffffff 就是白色，而 #000000 則是黑色。前面幾個色彩名稱的表格當中，都有附註十六進制的色碼，讀者可以一併參考使用。

設計出這種色碼表示法的是資訊背景的人，這類人們有個很重要的人格特質：非常懶惰，因此當下列三個條件都成立的時候，色碼就可以用「縮寫」的方式來表達：

- 表達紅色光的兩位十六進制數字是一樣的
- 表達綠色光的兩位十六進制數字是一樣的
- 表達藍色光的兩位十六進制數字是一樣的

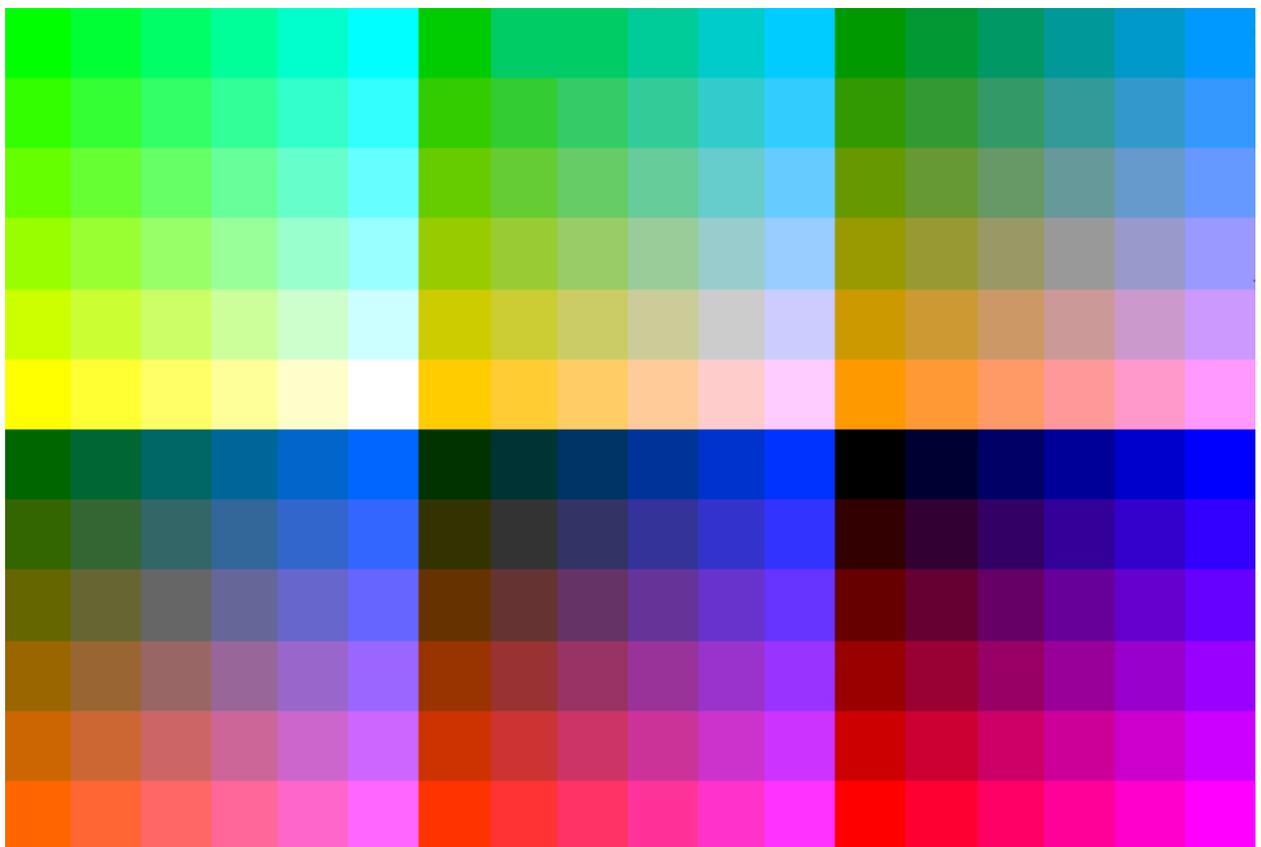
在這三個條件都成立的時候，原本的 6 位十六進制數字，就可以縮寫成 3 位，例如把 #22bb99 縮寫成 #2b9，把 #000000 縮寫成 #000，依此類推。

安全色

設計網頁時使用的色彩，也會遇到跟處理網頁版面時一樣的問題：不同的瀏覽器、不同的作業系統、不同的螢幕及顯示設備，可能會把同樣色碼的顏色，呈

現成不同的樣子。做平面設計時，設計師的螢幕會按照印刷設備的規格來校對色彩，確保電腦上看到的色彩，跟實際印製出來的成品一致；然而絕大多數的網頁使用者，並不會校對螢幕色彩，所以設計師根本不可能確認設計網頁時所看到的色彩，與使用者看到的會一致。

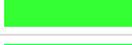
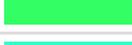
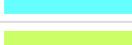
絕大多數的色彩都不可信賴，色偏的問題必然會發生，因此在網頁設計時，16,777,216 種顏色當中有 4,096 種會比較常用——沒錯，就是可以縮寫的那 4,096 種（#000000、#000011、#000022、……、#ffffdd、#ffffee、#ffffff，總共有 $16 \times 16 \times 16 = 4,096$ 個色碼），這些色彩統稱為「WebSmart 色板」；WebSmart 色板當中，僅由 00、33、66、99、cc、ff 數值組成的色彩（#000000、#000033、#000066、……、#ffff99、#ffffcc、#ffffff，總共有 $6 \times 6 \times 6 = 216$ 個色碼），在不同瀏覽器上的表現幾乎是一致的，因此被稱為「WebSafe 色板」或「BrowserSafe 色板」。



很不幸地，由於不同作業系統預設的 gamma 曲線不同（例如在 Windows 上，預設的 2.5 gamma 螢幕會把 0%、20%、40%、60%、80%、100% 等不同濃

度顯示成 0%、2%、10%、28%、57%、100%)，所以 216 種 WebSafe 色彩當中，祇有 22 種是真正可以完全信賴的：

色彩 色碼

	#000000
	#000033
	#0000ff
	#00ff00
	#00ff66
	#00ffcc
	#00ffff
	#33ff33
	#33ff66
	#33ffcc
	#33ffff
	#66ff00
	#66ff33
	#66ffff
	#ccff66
	#ff0000
	#ff0033
	#ff00ff
	#ffff00
	#ffff33
	#ffff66
	#ffffff

這 22 種色彩因此有了「WebReallySafe 色板」的稱號。

除了十六進制色碼外，其實還有其他的色碼寫法，例如以十進制來撰寫成：`rgb(rrr,ggg,bbb)`，或根據百分比來撰寫成 `rgb(rrr%,ggg%,bbb%)` 等；`#369` 跟 `#336699` 跟 `rgb(51,102,153)` 跟 `rgb(20%,40%,60%)` 四種寫法所代表的色彩是一樣的，不過實務上還是以十六進制的色碼寫法最為常用。

小結

本節介紹了設計網頁時使用色彩的相關事項。根據本節介紹的知識，使用色彩時的優先順序應該如下：

1. 使用 22 種 WebReallySafe 色板中的色彩
2. 使用 216 種 WebSafe 色板中的色彩
3. 使用 4,096 種 WebSmart 色板中的色彩
4. 使用十六進制的色碼來指定色彩
5. 使用 17 種 CSS 色彩名稱來指定色彩
6. 使用 28 種系統色彩名稱以及 141 種非標準的 HTML 色彩名稱來指定色彩

長度單位與解析度

網頁設計當中，計算長度的方式也跟平面設計有很大的不同。做平面設計時，長度的計算單位就是現實生活中會使用的公分、吋，處理文字時會比較複雜一點，因為字型尺寸的計算方式是從鉛字印刷、照相打字時代一路走來，所以在此需要做點換算。

鉛字（活字）排版使用的字型大小是按照「號」來區分的，從最大的「初號」開始，依次為「一號」、「二號」、「三號」、「四號」、「五號」、「六號」，以及很少使用的「七號」與「八號」，後來並有「新初號」至「新七號」的尺寸，因此總共計有 17 種固定大小。初號字相當於照相打字中的 72 級字大小。

「級」是照相打字的字型大小計算單位，每一級相當於 0.025 公分；實際施用的「級」數祇有固定的 24 個尺寸：從最大的 100 級開始，依次為 90 級、80 級、70 級、62 級、56 級、50 級、44 級、38 級、32 級、28 級、24 級、20

級、18 級、16 級、15 級、14 級、13 級、12 級、11 級、10 級、9 級、8 級，以及最小的 7 級。

進入電腦排版的時代後，這些字型的計算單位也跟著轉移。早期電腦平面設計的天下幾乎都是麥金塔，麥金塔固定採用 72 dpi 的解析度，意思就是每 1 吋有 72 PT；1 吋為 2.54 公分，約略近於 2.5 公分，也就是 100 級的照相打字尺寸（其實 1 吋比較接近 102 級，但是照相打字的級數是固定的，所以取最接近的 100 級），因此在平面出版時，通常以 100 級等於 72 PT 來換算。麥金塔以外的其他電腦，則不一定採用 72 dpi 的解析度配置；舉例來說，有些電腦可能會以 300 dpi 作為印刷輸出的設定，此時 2.5 公分則約略等於 296 點，所以 100 級就要換算成 296 點。

鉛字 照相打字 印刷點數 (300 dpi) 螢幕點數 (72 dpi) 實際長度

	100 級	296 點	72 PT	2.5 公分
初號	62 級	183 點	44.6 PT	1.55 公分

平面設計時所採用的長度計算單位，可以用上表來加以換算。至於為什麼不是把 100 級換算成比較接近的 70.9 PT，則是因為電腦上的字型在 72 PT 大小時會比 70.9 PT 要來得漂亮，這是考慮到實務面之後的取捨結果。

設計網頁時，可能採用的長度單位就更複雜了。網頁當中可以使用的長度單位分成兩大類：

- 絕對單位
- 相對單位

「絕對單位」指的是沿用真實世界當中的實際長度單位，像是吋 (in)、公分 (cm)、公釐 (mm)，以及沿用自印刷界所慣用的長度單位，像是 PT (pt) 與 pica (pc)。在網頁設計中的 PT 是根據 72 dpi 解析度所假想出來的

「點」，因此不論使用者所指定的解析度為何，1 PT 就一定等於 1/72 吋；pica 則也是根據印刷出版的規格來制定，因此 1 pica 就一定等於 12 pt，也就是 1/6 吋。

然而稍早在介紹網頁排版與平面排版的異同時就有提到，設計師無從得知使用者的螢幕設置方式，因此設計網頁時指定的絕對長度單位，並不能想成跟真實世界中的長度單位一模一樣。一個在網頁上指定為 5in (5 吋) 長的線段，呈現在某個使用者的螢幕上時，可能長達 7.3 吋，而在另一個使用者的螢幕上卻祇有 2.2 吋。要推測這些長度實際上呈現出來的情況，就必須先把絕對單位換算成相對單位。

「相對單位」指的是會根據使用者的操作配置而有所不同的長度單位，包括：

- 像素 (px)
- 目前字型大小 (em)
- 目前字型的 x 高 (ex)

「像素」是螢幕上顯示一個色點的最小單元；Windows 作業系統採用 96 dpi 的設定，也就是以「1 吋等於 96 像素」作為理想值，即每個像素約 4.10 公釐，但是實際上的像素長度並不能這樣推論，而是要實際考慮使用者的螢幕尺寸與螢幕解析度。舉例來說，Dell 2408WFP 螢幕的顯示面板長 20.4 吋、高 12.7 吋，最高支援到 1920×1200 的螢幕解析度，因此在這樣的解析度下，此螢幕橫向 20.4 吋當中有 1920 像素，相當於 $1920/20.4 \doteq 94.12$ dpi，每個像素寬約 4.18 公釐；縱向 12.7 吋當中有 1200 像素，相當於 $1200/12.7 \doteq 94.49$ dpi，每個像素高約 4.17 公釐。沒錯，螢幕上的像素有可能是長方形的。另外有一些輸出設備，例如 TTY 電傳打字機，是以「字元」為長度單位，則根本無法接受以「像素」作為單位的長度描述。

要把絕對單位換算成相對單位時，要根據使用者電腦作業系統配置的 dpi 設定來處理前述的假想點 (pt)。如果使用者的電腦設定成 96 dpi (也就是 Windows 的預設值)，則 1in 就可換算成 96px；設定成 120 dpi (Windows 設定中的「大字型」) 則 1in 可換算成 120px；設定成 72 dpi (Mac 的預設值) 則 1in 可換算成 72px。至於其他的絕對單位，則可以依照「1in = 72pt = 6pc」來加以換算，換算成 px 之後還可以再根據前述實際丈量螢幕顯示面板尺寸的方法，推算出實際表現出來的長度。

即使使用相同的螢幕以及螢幕解析度設定，由於 Windows 與 Mac 預設 dpi 的不同，會導致使用絕對單位時，反而會造成大小差異，即同樣指定為 72pt 大小的文字，在 Mac 上看起來會比較小 (等於 72px)，而在 Windows 上看起來會比較大 (等於 96px)，所以設計網頁時，應該要盡量避免採用絕對長度單位，這樣設計出來的網頁纔能在不同作業系統間，獲得較為一致的呈現結果。

em 與 ex 則是比 px 更為彈性的相對長度單位，很有趣地，這兩個單位的名稱由來也是源自印刷界。em 當中的 m 指的是 Mutton，這是在活字印刷術當中用來形容大寫字母 M 的寬度——大寫 M 是所有字母當中最寬的一個，所以後來就用這個名稱來指定字型的尺寸。如果網頁中，目前的排版區塊使用的是 48pt 的字型尺寸，則此時 1em 就會等於 48pt；許多網頁瀏覽器都提供使用者隨時動態調整文字尺寸縮放的功能，當網頁設計師用 em 來指定內容長度時，就也能跟著文字尺寸的縮放，而即時自動調整。

ex 指的則是目前字型當中的「x 高」——小寫 x 字母的高度，也就是從基線 (所有大寫字母以及大多數小寫字母所坐落的假想位置) 到中線 (也就是小寫 x 以及其他多數小寫字母的上緣的假想位置) 的距離。同樣地，使用 ex 作為長度單位時，就可以按照使用者對網頁文字的縮放操作，來即時自動調整。

除了前述的絕對長度單位 (in、cm、mm、pt、pc) 與相對長度單位 (px、em、ex) 外，網頁設計時也可以使用比例單位 (%) 來描述長度，此時就會根據繼承到的長度乘上指定的百分比，得出該有的長度。不過這裡必須要注意的是，使用了 % 比例單位時，就算上層元件用的是 em 或 ex 這種相對長度單位，也會根據該情境中的字型尺寸「展開」成 px 或其他絕對單位的長度，而非沿用 em 或 ex 單位，所以接下來若再改變情境中的字型尺寸，這些使用 % 比例單位的長度也不會再跟著變動了。

不論使用哪一種長度單位，都可以有正、負值，例如指定「往左移動 -5px」的話，效果就會跟「往右移動 +5px」是一樣的；當數值為正值時，也可以省略 + 符號不寫。當長度為 0 的時候，也不用加上任何單位。

小結

本節描述了設計網頁時，長度單位的使用方法：

- 絕對長度單位的換算方式為 $1\text{in} = 2.54\text{cm} = 25.4\text{mm} = 72\text{pt} = 6\text{pc}$
- 根據 dpi 將 in 換算為相對長度單位 px，公式為 $\text{in} \times \text{dpi} = \text{px}$
- 根據目前字型尺寸而變動的相對長度單位 em 與 ex
- 用 % 來描述長度比例
- 正值、負值、0

字元編碼

製作非純英文的網頁，就會遭遇字元編碼的問題，而這是個為時已久、繁複、充滿政治文化角力的問題。不過在此還是可以給讀者一個簡要版本的說明：

早年的中文網頁多半使用 **Big5** 字元編碼，這是個由我國民間資訊相關公司所共同制定編碼方式，也稱作「大五碼」，但是 **Big5** 所收錄的字數不多，諸如「堃」、「喆」等字均無法使用，如果需要與日文、韓文、越南文以及其他語言的內容同時使用，則更是不可能。

再加上 **Big5** 先天設計上的問題，有不少字（例如「許」、「功」、「蓋」）含有控制字元或逸出字元，使得在電腦程式或網頁當中，很容易因為這些字而發生問題。這兩方面的困難，都使得 **Big5** 字元編碼越來越不適合在網頁時代繼續使用下去。

近年來由於作業系統與其他應用程式的更新，對於萬國碼（**Unicode**）的支援已經大大改善，使得網頁的發展也走向萬國碼的懷抱，其中又以 **UTF-8** 編碼為主流。**UTF-8** 收錄的漢字字數為 **Big5** 的幾倍以上，並且可以混合處理及顯示中、日、韓、越、歐、美、亞、非的各種語言，實在是沒有不採用的道理。

然而要注意在萬國碼的規格當中，有個稱作 **BOM**（**Byte Order Mark**，位元組順序記號）的特殊字元，這個特殊字元對於 **UTF-8** 編碼並沒有作用，但是很多應用程式在處理 **UTF-8** 文件時卻會把它加到檔案開頭處，使得網頁檔案無法正確運作。挑選設計網頁用的編輯器時，務必要選擇可以正確處理 **UTF-8** 字元編碼、並且可以關閉 **BOM** 的軟體。

小結

關於網頁使用的字元編碼，祇有兩個要點：

- 使用 **UTF-8**
- 不要有 **BOM**

圖檔格式

資訊世界中的圖檔可分成兩大類：

- 點陣圖檔
- 向量圖檔

點陣圖檔的技術原理，是把組成圖片的「像素」逐一記錄；向量圖檔是以向量方式，來描述每一個線段與色塊等物件。如果以應用程式來說明的話，由 **Adobe Photoshop** 處理過的圖檔通常就是點陣圖檔，由 **Adobe Illustrator** 處理過的則是向量圖檔。向量圖檔的優勢在於可以無限放大、自由旋轉仍不失真，點陣圖檔的優勢則是在小尺寸的情況下，更能讓設計師掌握所有的小細節，而有較好的呈現效果。

點陣圖檔

設計網頁時，常見的點陣圖檔格式包括了：

- PNG (*.png)
- JPEG (*.jpg、*.jpeg、*.jpe)
- GIF (*.gif)

PNG 是由全球資訊網協會 (**W3C**) 所開發、制定的點陣圖型格式，支援 **24** 位元全彩，並可再多加 **8** 位元的半透明 α 通道 (總共就是 **32** 位元)，另外也支援 **16** 位元灰階以及色盤式配色，採用無專利疑慮的非破壞性壓縮演算法，因此可以在絕佳的影像品質與小巧的檔案尺寸間，取得理想的平衡。

各瀏覽器均支援 **PNG** 圖型格式，但是微軟的 **Internet Explorer** 在第 **6** 版及更早的版本中，並不支援 **PNG** 格式中的半透明 α 通道，這一點必須留意。

JPEG 是由聯合圖像專家小組 (**Joint Photographic Experts Group**) 制定的點陣圖型格式，支援 **24** 位元全彩，以及 **8** 位元灰階，但是並不支援透明或半透

明效果。JPEG 圖型採用破壞性壓縮法，因此同樣尺寸的圖片，採用 JPEG 格式會小得多；但因為採用的壓縮演算法是破壞性的，因此 JPEG 圖檔一定會失真，並容易產生雜訊。

GIF 是由 CompuServe 公司開發的點陣圖型格式，僅支援 8 位元色彩（總共祇有 256 色），但是可自訂色盤，因此在技術上，仍然可以從 24 位元全彩色域當中，挑選 256 種色彩出來使用。GIF 圖檔採用非破壞性的 LZW 壓縮演算法，並且僅使用 256 色色盤，所以在檔案尺寸上可獲得優勢，同時保持圖片內容不失真；然而 LZW 壓縮演算法有專利授權費用的問題，因此近年來有被 PNG 格式取代的趨勢。

GIF 可以在一個圖檔當中置入多幀畫面，因此可以用來製作動畫；儘管 GIF 不具半透明 α 通道的能力，但仍然可以在色盤當中，指定其中一種色彩為「透明色」，實現簡易的「去背」效果。各瀏覽器對 GIF 圖檔格式的支援也比 PNG 格式要來得完善，使得 GIF 即便在面臨專利問題與色彩數僅有 8 位元的限制下，仍然能繼續存活在網路世界中。

為網頁製作圖片時，有些影像處理軟體或繪圖軟體可以指定圖片的解析度資訊，預設的圖片解析度就是與作業系統的 dpi 設定一致，所以在螢幕上以幾個像素來呈現的圖片，實際上也就是按照相同數量的像素來儲存；為平面設計準備圖片時，為了顧及印刷效果，通常會把圖片的解析度設為 300 dpi 以上，這時處理圖片的軟體就會多存入額外的像素，圖片檔案也會因此變大。網頁上的圖片主要就祇是用於螢幕顯示之用，因此設定太高的解析度除了浪費流量頻寬、讓使用者花更多時間下載外，並沒有顯著的好處；用於螢幕顯示的圖片，設定為 72 dpi（Mac 的預設解析度）至 120 dpi（Windows 的「大字型」解析度）其實就足夠了。不過在本節所介紹的三種點陣圖檔格式當中，僅有 PNG

與 JPEG 格式可以如此指定 dpi；GIF 格式無法儲存 dpi 資訊，所以看起來多大、實際上就是多大。

向量圖檔

設計網頁時，也可以使用向量圖檔，常見的向量圖檔格式包括：

- SVG (*.svg)
- Flash (*.swf)

SVG 是由全球資訊網協會制定的網頁標準，本質上是一個 XML 檔案，因此任何應用程式理論上都可以輕易地取得 SVG 向量圖檔當中的文字內容。SVG 可以有動畫效果，也能夠與使用者的操作互動，但是無法嵌入視訊及音訊媒體，而且編撰軟體較少，瀏覽器支援的情況也是到最近一、兩年間纔有改善。

Flash 則是由 Macromedia 公司制定的私有格式，後來被 Adobe 併購——這裡有個有趣的歷史沿革：多年前 Macromedia 是 Adobe 的主要競爭對手，因此 Macromedia 推動 Flash 格式的時候，Adobe 則是致力於 SVG 格式；當 Adobe 併購 Macromedia 後，反而開始推動起 Flash，而不理會 SVG 了。

Flash 是個私有的專屬格式，要取得 Flash 內的文字內容較不容易，但並非不可能；Flash 在多媒體的整合上，尤其是視訊及音訊方面的能力，則是優於 SVG，而且編撰 Flash 的工具也較成熟、普遍。

現今多數的瀏覽器都已經內建 SVG 的支援能力，但是若要考慮到舊版的瀏覽器，情況則不理想；Flash 到目前為止都不是各瀏覽器能夠預設支援的，但是從很久以前，幾乎各主要瀏覽器就都有對應的外掛模組可以安裝、使用。不過，即使到了現在，iPhone（以及未來將要上市的 iPad）仍然不支援 Flash，也沒有對應的外掛模組可用，算是支援上的遺珠之憾。

除了 SVG 與 Flash 外，目前尚在研議中的 HTML5 也有加入向量繪圖支援的打算，不過幾年內應該都不會有定案，所以有興趣的讀者可以嘗試，但大概還不能過於信賴。

小結

本節介紹、比較了 3 種點陣圖檔格式：

功能	PNG	JPEG	GIF
色彩	24 位元	24 位元	8 位元
灰階	16 位元	8 位元	8 位元
檔案大小	小	極小	小
壓縮	無失真	破壞性	無失真
動畫	無	無	支援
透明	8 位元半透明 α 通道	不支援	單色
瀏覽器支援	好	很好	很好

以上述的點陣格式製備圖檔時，除了 GIF 格式外，PNG 格式與 JPEG 格式都要留意使用的解析度應介於 72~120 dpi 之間。

本節另外還介紹、比較了 2 種向量圖檔格式：

功能	SVG	Flash
檔案型態	純文字開放格式 XML	私有格式二進制
動畫	支援	支援
操作互動	支援	支援
視訊及音訊	不支援	支援
瀏覽器支援	新版瀏覽器：好 舊版瀏覽器：差	安裝外掛後佳 iPhone 仍不支援
入門難度	較高	較低
網頁標準	是	否

瀏覽器

網頁設計之路上，一定會遇到的難題還有一個：瀏覽器相容性。雖然全球資訊網協會（W3C）制定了許多詳盡的「網頁標準」，但是在現實中，沒有任何一個瀏覽器完整實踐了所有的網頁標準；另一方面，幾乎每一個廠牌的瀏覽器也都在標準以外「多做了些事」，其中有些是尚未定案的新標準提案，例如 HTML5 與 CSS3 的部份規格，另外有些則是瀏覽器廠商自己發明的規格。

因此，即便是完全按照網頁標準來撰寫網頁，也一定會遭遇不同瀏覽器的支援不良的情況，而需要進一步針對各瀏覽器來修補——這些修補要很謹慎，以免瀏覽器更新至新版、對網頁標準的支援改善後，反而被「修補」給補壞了。

對於甫進入網頁設計領域的人，最好盡可能把寫好的網頁實際拿到各個不同的瀏覽器上，看看呈現的結果是否與想像中的樣子接近。本課程在未來講述技術實務細節時，也會盡可能提出不同瀏覽器的支援情況以及修補方法。

本講義以創用 CC 姓名標示—非商業性—相同方式分享 2.5 台灣版授權條款釋出，您可在此前提下任意使用，並請註明作者「Jedi」及本講義網址「<http://Jedi.org/p4/slide/cite2010/lesson01.html>」。如有其他使用需求，請務必先行聯繫。