

Dive Into Accessibility

在 30 天內打造更具親和力的網站

目錄

第一週

- 前言
- 第一天：Jackie
- 第二天：Michael
- 第三天：Bill
- 第四天：Lillian
- 第五天：Marcus

第二週

- 第六天：選擇 DOCTYPE
- 第七天：指定妳的語言
- 第八天：建立有意義的網頁標題
- 第九天：提供額外的導覽協助
- 第十天：先呈現妳的主要內容

第三週

- 第十一天：跳過導覽列的鏈結
- 第十二天：安全地使用色彩
- 第十三天：使用真實鏈結
- 第十四天：在鏈結上加入標題
- 第十五天：定義快速鍵

第四週

- 第十六天：不要開出新視窗
- 第十七天：定義頭字語
- 第十八天：在月曆中加上真的標題
- 第十九天：使用真的表格標頭
- 第二十天：提供表格摘要

第五週

- 第二十天：忽略卡位圖片
- 第二十二天：使用真正的清單（或以適當的方法偽造）
- 第二十三天：提供能取代圖片的文字
- 第二十四天：提供能取代影像地圖的文字
- 第二十五天：使用真正的水平分割線（或以適當的方法偽造）

第六週

- 第二十六天：使用相對字型尺寸
- 第二十七天：使用真正的標頭
- 第二十八天：加上元件描述
- 第二十九天：加入站台搜尋功能
- 第三十天：撰寫親和力聲明
- 結語

附錄

- 親和力聲明
- 使用聲明
- 譯本

Dive Into Accessibility

在 **30** 天內打造更具親和力的網站

這本書要回答兩個問題：第一個問題是「為什麼該讓我的網站更具親和力？」如果妳根本就沒有網站的話，顯然就不會成為本書的讀者了。第二個問題是「該如何讓我的網站更具親和力？」如果第一個問題的回答說服不了妳，那麼第二個問題大概也將無法引起妳的興趣。

讓我們開始吧。

下載 Dive Into Accessibility

- 下載中文 HTML 版本
- 下載中文 PDF 版本
- 下載原文 HTML 版本
- 下載原文 PDF 版本

閱讀 Dive Into Accessibility

目錄

所有建議皆根據原本出版的順序來編排。建議讀者們第一次閱讀本書時，先按照這個順序閱讀。

依人物編排

按照每種建議的受益者來編排。

依障礙情況編排

按照每種建議所能解決的障礙來編排；所謂的障礙不止包含了身體上的，也包含了技術上的。

依設計原則編排

按照每種建議所依循的 W3C 原則來編排。這指的是所有的頁面都應該要具備可認知 (perceivable)、好操控 (operable)、能導覽 (navigable)、易理解 (understandable) 以及夠強健 (robust) 等原則。

依瀏覽器分類

按照網頁瀏覽器與輔助技術來編排，這些工具將會因這些建議而受益。

依出版工具編排

按照出版工具可採行的建議來編排；每一個建議都會針對適用的出版工具提出修正步驟。

翻譯

這本書已經被翻譯成數種語言。

Dive Into Accessibility

在 30 天內打造更具親和力的網站

前言

這本書定名為《深入親和力：在 30 天內打造更具親和力的網站》，它將會回答兩個問題：第一個問題是「為什麼該讓我的網站更具親和力？」如果你根本就沒有自己的網站的話，顯然就不會成為本書的讀者了。第二個問題是「該如何讓我的網站更具親和力？」如果第一個問題的回答說服不了你，那麼第二個問題大概也將無法引起你的興趣。

為了回答第一個問題，我會先介紹以下五個人的特徵素描：Jackie、Michael、Bill、Lillian及Marcus。這些人有幾個共同點：

1. 她們都有身體上、心理上、以及技術上的障礙，使得網際網路的使用更為困難。
2. 雖然她們都是虛構人物，卻能反映出現實生活中被障礙所困擾的人們；同時她們也以相對應的方式使用網際網路。
3. 她們瀏覽你的網站時，都會遇到困難。

為了回答第二個問題，我會介紹 25 個能夠立即套用在你的網站上的訣竅，讓它更具親和力。雖然這些概念可以應用到所有網站上，但我只會專注在幾種熱門網誌工具 (weblogging tools) 的實作方面。如果你使用的是其他的出版工具或模版系統的話，就得確定如何在那些工具上實作這些訣竅。

每個訣竅都會著重在單一概念上；我會解釋它背後的原因，並指出誰會因為你實作而獲益。這也是我們先談到特徵素描的原因，因為這能把第一個問題的語調從「為什麼我得煩惱這些？」轉變為「誰將因此而獲益？」於是我們可以回答「Marcus 將能從中獲益。」「為什麼 Marcus 能從中獲益？」「好，讓我們從這點來看……」，如此進行。

即使你不是 HTML 專家也甯緊張。就算你的網站只是個人網誌，而且打從你選好模版的第一天後就沒再去碰過它們，也別太擔心。我不是要你徹底地重新設計網站，一邊咒罵你那些巢狀表格，一邊把它們轉換成 XHTML 與 CSS 語法。我們只不過是要聊聊你現有的基礎，然後利用一些小巧卻重要的步驟讓它們變得更好。Jackie、Michael、Bill、Lillian 及 Marcus 都會為你的體貼入微而感動不已。

Dive Into Accessibility

在 30 天內打造更具親和力的網站

第一天：Jackie

Jacquelin（暱稱 Jackie）是個成績優異的 19 歲高中生，和媽媽住在芝加哥伊利諾州的 Park Ridge。她的成績一部分來自於私人家教的協助，但主要還是因為她勤奮學習及廣泛閱讀。她所有的課本都是有聲書，使用特殊的錄音機來撥放；這種錄音機的撥放速度比正常快 3 倍，卻祇會造成一點點失真。她失明已經 8 年了。

Jackie 並非天生失明，所以她還能理解像色彩這樣的視覺概念，而且仍然會跟媽媽聊到「以前」生活裡的色彩。她唯一不談的是害死她爸爸、讓她失明的那場車禍；這在她使用「以前」跟「之後」兩個詞彙時可以看出些許端倪。「就像以前客廳牆壁的那種綠色。」「這像妳以前的那件紫色毛衣，不過更亮一點。」諸如此類。

由於 Jackie 努力地調適生活，以致於她的學業進度比同學們慢了兩年。今年秋天她將進入芝加哥大學，主修比較文學。她對此感到相當興奮，一方面是希望交到新朋友，另一方面則是因為能在線上完成更多課堂作業：閱讀課程規劃，提出論文，與教授、同學們即時傳訊等等。她高中的朋友並不多，多半是跟媽媽還有網際網路在一起。

她每個月花在有聲書、音樂以及一堆小玩意上的錢都超過 300 美金。雖然有越來越多吸引人的讀物能夠下載到電子書閱讀器裡，但是她大部分的有聲書仍舊是錄音帶。她那些音樂 CD 以及小玩意兒都是從線上零售商買來的。

線上購物未必簡單，但是跟自己搭車到賣場然後等店員招呼比起來，實在是好太多了。線上購物也是少數幾件她不需依靠導盲犬，就能完成的事情。她不喜歡現在這隻導盲犬 Arthur；牠不像之前幾隻那麼優秀。Lancelot 跟 Guinevere 都已經退休了，現在跟她們住在一起。她靠項圈來分辨牠們：Arthur 的項圈是光滑的、Lancelot 的帶刺，Guinevere 的則有凹槽。

就像大部分的盲人一樣，Jackie 對布拉耶 (Braille) 點字法了解不多。她有一台點字標籤機可以幫 CD 貼標籤，但她沒有辦法閱讀點字書，因為所有的點字書都是用第二級的點字法寫的，而她從來沒學過。她用最新版的 JAWS 來線上購物及活動，這是一套能在視窗環境下與 Internet Explorer 整合的螢幕朗讀軟體。JAWS 使用了先進的電子語音合成器來朗讀網站內容，也提供一堆深奧的鍵盤便捷鍵來瀏覽網站。Jackie 記得所有按鍵，所以祇要網站設計良好，她閱讀網站的速度就能比閱讀有聲書還要更快。

Dive Into Accessibility

在 30 天內打造更具親和力的網站

第二天：Michael

Michael 現年 27 歲，退休後與女朋友 Christine 住在長島的 Great Neck。他 22 歲的時候創立了一家小公司，專門佈設超高速光纖電纜。兩年後，他在洽當的時機脫手，賣給一家全球性的電訊業集團；同時這家集團也用電話解釋給 Michael 聽，讓他知道為什麼不能讓他那間一個月 4000 美金的公寓有比 56K 撥接更好的服務。看來金錢也有不是萬能的時候。

Michael 眼下的世界跟妳所看到的不同。這可不是甚麼商業的陳腔濫調或是哲學聲明；他看到的世界真的跟妳所看到的不一樣。他是紅綠色盲，沒有辦法分辨紅色跟綠色（某些紅色看起來會非常深，近似於黑色）。他所有的衣服都標上字母來區分顏色，像是用 R 代表紅色、DG 代表深綠等等。他的女朋友設計了一套搭配組合，讓他知道哪些衣服可以搭配在一起。他按照這些字母的指示來穿著，也如此表達；但他不能分辨其間的差異。

當客服人員溫言告訴他說他再也不需要工作的那當兒，Michael 正一如往常地在網路上。現在他退休了，開始把時間都花在蒐集老舊的大型遊戲機台上，然後試著把這些遊戲機台修好，放進櫥櫃裡、加以詳列說明，最後再到拍賣網站上把他們賣掉。事實上大部分的細節都是他女朋友做的，不過其他所有的事都還是 Michael 在處理。他還養了一隻叫 Ralph 的雌鼯；因為 Christine 抱怨說雄鼯的味道太重了，所以他們妥協後，決定按照她的意見。Ralph 大部分的時間都在沙發的角落睡覺；不過早上交通顛峰期的時候，牠就會坐在窗檯上看著窗外的車水馬龍。事實上 Michael 也會。Christine 大概是唯一選擇繼續睡覺的人了。這是專屬於他和鼯的獨處時間。

正如妳所料，Michael 的嗜好仰賴大量的線上互動。他得跟那些客戶們（不論是過去或是未來所遇到的）、供應商們、機台擁有，人以及任何可能在機台零件上有著卓越見解的人打交道。他實在恨透了他的 56K（實際上祇有 28.8K）撥接了，這讓他的瀏覽器幾近爆炸。某些日子裡，他會用一個叫 Links 的純文字瀏覽器，這個瀏覽器就跟其他的圖形瀏覽器一樣，可以畫出複雜的多欄版面，不過就祇有文字，而沒有圖片或字形變化。另一些日子裡，他也會用 Opera 瀏覽器，因為這個瀏覽器允許他在背景載入圖片，而且可以很方便地開啟或關閉圖片選項；他多半會因為頻寬的考量把他們關閉。今天則是使用 Links 的日子。

紅綠燈由上而下分別是紅、黃、綠；可是橫向的紅綠燈位置卻不一定，因此相當危險。Michael 就跟其他也有色盲的 8% 美國人一樣，其實偷看了附近其他人的動作，纔知道離地最近的是綠燈、最遠的是紅燈，而中間的是黃燈。

Dive Into Accessibility

在 30 天內打造更具親和力的網站

第三天：Bill

Bill 住在舊金山的一棟小公寓，唯一的親戚是離他家五哩遠的妹妹。他今年 62 歲，公寓裡總是有條不紊，就像要視察前的軍營一樣。這是當然的，因為 Bill 打從 18 歲就加入軍隊，從士兵開始不斷晉升軍階，並在越戰中得到三面勳章。一直到十年前膝蓋的傷讓他轉任文書工作前，他都維持現役職務。他去年罹患了中風而被迫退休，右手臂完全沒辦法移動，左手則會不由自主地擺動，卻沒人知道為什麼。有 500 個人參加了他的退休宴會，每個人他都認得。

Bill 用一台老舊的筆記型電腦跑 Red Hat Linux 作業系統。他用 Mozilla 瀏覽網站，並使用 Evolution 閱讀信件。其實他完全不懂這些，這整台電腦（還有裡面的軟體）都是他那精通電腦的妹妹送的退休禮物。儘管這台老電腦只有 13 吋螢幕，解析度還是被固定在 1024x768。感謝老天 Bill 看得到，只是偶爾會找不到讀到哪裡。從他中風之後就這樣，而他妹妹認為這越來越嚴重，不過她錯了。

對他來說最大的恩惠是有個延伸的小方向鍵盤，讓他可以更輕鬆的用手碰到它們。這個鍵盤花了 \$29.99 美金，包含 \$5 的郵資。

Bill 退役前從沒上網過，但現在他找到越來越多繼續待在網路上的理由。他每天花四個小時去瀏覽網站以及跟新舊朋友們聊天。由於那些退役同袍的鼓勵，以及妹妹在技術上的協助，他發起了關於退役軍人權利的郵遞論壇。這個論壇已經有 200 個會員了。

令人吃驚地，當 Bill 把左手穩定放在鍵盤底部時，只靠移動手指就可以每分鐘輸入 10 個字（隔年將會是 15 個字）。然而他幾乎沒辦法使用滑鼠，所以只能靠方向鍵、跳格鍵 (tab key)，還有妹妹教他的那一堆令人暈眩的快速鍵組合。他妹妹也教他如何使用某個特殊的程式，讓他可以用方向鍵來移動游標。但那不但慢得煩人，還會經常發生錯誤，所以他通常都不為自己找麻煩。

雖然他自己還不知道接下來會發生的事：明年這份郵遞論壇將會變成一份網誌，接下來這份網誌還會變成政治活動委員會，並且成為主要的政治力。畢竟政治這回事就是看妳認識了多少人，而 Bill 認識每個人。

Dive Into Accessibility

在 30 天內打造更具親和力的網站

第四天：Lillian

據說 Lillian 剛從香港搭船移民來美國。其實她是 30 年前搭飛機來的，不過人們還是如此傳頌著。她今年 54 歲，跟女兒、養子和兩歲大的孫子寡居於堪薩斯市，她們大概就花去了她所有的時間了。雖然她的女兒和養子都熟稔於兩種語言，但是 Lillian 的英語還是支支吾吾，不如廣東話來得流暢。不跟孫子一起玩的時候，她也會試著讀報紙來練英文；她會把報紙攤開在餐桌上，打開天花板上的 100 瓦大燈，用放大鏡慢慢地讀。

Lillian 在一個世界級的電訊業集團裡擔任行政助理。純粹為了簡化的考量，這個世界級的電訊業集團很湊巧地，也就是那個沒辦法提供某個住在長島的二十多歲退休人員高速網路的那一個。不過 Lillian 並不知道這件事。她們的資訊科技部門纔剛大規模地轉移到 Windows XP 和 Internet Explorer 6 的環境裡，而且接下來的九個月還會更為乖戾。他們同時採行了新的網際網路安全政策：不准使用 Java、不准使用 Javascript、不准使用 Flash 也不准使用 ActiveX 控制元件，除非經過了他們自己批准。當時他們顯然還沒有批准任何站台。技術上來說，這表示所有的站台在 Internet Explorer 的安全性頁面裡，都是「限制的網站」，所以所有的腳本都被關閉了；妳得擁有系統管理權限纔有辦法把某個站台加入「信任的網站」裡。

不消說，Lillian 自然是沒有系統管理權限了。

不過她能用 19 吋的大螢幕來跟資訊科技部門的 Matt 的苦口婆心勸阻相抗衡。他常常想起那一次因為電梯壞掉，所以他得幫 Lillian 把大螢幕努力扛到樓上去；所幸他很少不耐煩地抱怨這件事。Lillian 其實還挺喜歡 Matt 的，他可以說是那群人當中最善良的一個了，甚至還幫她把 Internet Explorer 裡的字型調到「最大」，好讓她女兒網誌的字大到能被她閱讀；因此她每天都會去看。但是當她問 Matt 為什麼 CNN.com 的字就沒辦法調得更大時，Matt 啟動了技客演說模式，扔出一大堆技術字眼，最後仍舊祇能說他實在無能為力。

Lillian 好希望可以閱讀更多網站；但既然 Matt 沒辦法修的話，就沒有人有辦法。

Dive Into Accessibility

在 **30** 天內打造更具親和力的網站

第五天：Marcus

Marcus 的父母吸毒，所以他天生視盲。他早產了三個月，在保溫箱裡住了六週，祇有 10% 的機會能夠活下來；不過他現在已經 31 歲了。

他在一家 AT+T 轉譯中心工作，幫忙在聾人及聽人之間轉譯電話。他們是這樣運作的：

1. 假設有個聾人或聽障的人，我們就先叫她 Melissa 好了，用她的 TDD 或電腦撥號到中心來。
2. Melissa 鍵入她想要交談的對象的名字和號碼；就先假設那個人叫 Todd 吧。
3. Marcus 就會打電話給 Todd，然後告訴他有個叫 Melissa 的人從轉譯中心要打電話給他。
4. 不論 Melissa 鍵入了甚麼，Marcus 都會從他的 ALVA 可覆新布拉耶點字顯示器上讀到。任何 Melissa 輸入的字都會被即時地轉換成布拉耶點字。
5. 不論 Todd 說了甚麼，Marcus 都會以每分鐘 110 字的神速打給 Melissa 看。

無論是 Melissa 或 Todd 都不知道（其實也不在乎）Marcus 是盲人。唯一的線索就祇有 ALVA 的針上下移動時所發出的規律噪音，聽起來有點像遠處打字機所傳來的聲音。

Marcus 的眼睛跟大腦之間真的甚麼都沒有了。雖然如此，每個下午當太陽光從窗戶射進他的隔間時，他還是會抱怨著因此頭痛不已，然後把百葉窗拉上。

Marcus 在家裡的時候也會拿 ALVA 搭配全螢幕 DOS 視窗裡的 Lynx 純文字瀏覽器使用。他在家裡讀這些網頁的方法，就跟他工作時的作法一樣：用布拉耶點字，一次讀一列。他討厭螢幕朗讀軟體，就算有一套也不想聽；因為他總是習慣一回到家就把收音機轉到最大聲，直到睡前纔關上。

當他聽著收音機在網路上漫遊時，他也會用電話跟其他人聊天。他每週至少都會跟 100 個人講過電話；這聽起來很不尋常，不過他自己卻完全不曉得。如果妳曾經玩過「六度分離」這個遊戲，試圖尋找妳跟世界上任何另一個人的關聯時，妳很可能會發現 Marcus 就是妳的六度親人之一。他就是那種三姑六婆型的人，即使是最有經驗的市場老手也未必能夠打斷他的話；就是因為有他這種人，所以妳的電話公司纔會不打算提供無限市內通話的選擇。他熱衷於告訴親朋好友們，剛剛看到了多酷的網站。事實上他刻意地使用了「看到」這個字眼，因為他曉得這會讓他們覺得不自在。

Dive Into Accessibility

在 **30** 天內打造更具親和力的網站

第六天：選擇 DOCTYPE

就像句子的開頭會用大寫字母一樣，HTML 也以 DOCTYPE 開頭；這是最基本的文法。

誰因此獲益？

妳從中獲益了。因為妳必須先知道妳所使用的 HTML 版本，纔能夠在日後的課程中，選擇適當的步驟來套用那些訣竅。所以請把它找出來，沒有的話就加上去。

怎麼做

妳可能已經設定好 DOCTYPE 了。請檢查妳的網頁源碼，DOCTYPE（如果有的話）應該在很上面，甚至在 `<html>` 標籤之前。

- 如果妳用了 Movable Type 的預設模版，DOCTYPE 也許會是「XHTML 1.0 Transitional」。這樣很好。
- 如果妳用的是 Radio Userland、Manila 或 Blogger 的預設模版的話，DOCTYPE 也許會是「HTML 4.01 Transitional」。這也很好。
- 其他有效的 DOCTYPE 像是「HTML 4.01 Strict」、「XHTML 1.0 Strict」、「XHTML 1.1」等等，也都很好。

如果妳已經設定好 DOCTYPE 的話，就不用改它了。但是如果在妳源碼中的 `<html>` 前面沒有 DOCTYPE 的話，請加上這些：

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
```

基於技術上的考量，我不想在此深入探討細節（如果有興趣的話，請看「延伸閱讀」部分）；妳可能會發現加入 DOCTYPE 之後，頁面排版有點小變化。如果不幸發生了，妳也可以使用折衷的 DOCTYPE 來代替，就像這樣：

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
```

請注意，網站上所有的頁面都必須包含 DOCTYPE；所以妳得檢查所有的模版。

- Movable Type 的使用者必須檢查「主要索引 (Main Index)」、「主要彙整頁面 (Master Archive Index)」、「分類彙整 (Category Archive)」、「按時彙整 (Date-Based Archive)」、「單篇彙

整 (Individual Entry Archive) 」模版，以及妳手動建立的其他彙整模版。

- Radio Userland 與 Manila 的使用者則必須檢查「 Main template 」跟「 Home page template 」。
- Greymatter 的使用者則必須檢查「 Main Index-Related Templates 」、「 Archive-Related Templates 」，以及「 Entry-Related Templates 」。
- Blogger 的使用者必須在主要模版加上 DOCTYPE。如果妳的「 Archive template 」是分散的頁面（就是說，如果在最上方有 <html> 標籤的話），也需要加上 DOCTYPE。

妳得知道所使用的是 HTML 4（任何版本）、XHTML 1.0（任何版本）還是 XHTML 1.1。明天妳就會知道為什麼了。

延伸閱讀

- A List Apart : Fixing Your Site With The Right DOCTYPE
- MSDN : Quirks mode in IE 6（注意：某些版本的 Netscape 跟 Mozilla 無法瀏覽這個網站；這是 Microsoft 的問題，不是我。）
- Quirks mode in Mozilla

Dive Into Accessibility

在 30 天內打造更具親和力的網站

第七天：指定妳的語言

妳當然知道妳用的是哪種語言，所以請告訴妳的讀者 以及她們所用的軟體。

誰因此獲益？

1. Jackie 從中獲益了。因為她的螢幕朗讀軟體 (JAWS) 必須先知道妳所使用的語言，朗讀時才能準確地發音。如果妳沒有指定，JAWS 就會試著猜測妳所使用的語言，而它可能猜錯；尤其當妳在網頁裡引述源碼，或混雜著其他不屬於語言的內容時。
2. 即使妳寫的就是英文，Google 也從中獲益了，當妳使用的不是英語時更是如此。根據 Google Zeitgeist 的資料顯示，50% 的 Google 使用者用英語之外的語言搜尋，而且許多使用者在 Google preferences 設定只搜尋特定的語言。Google 自動偵測語系的演算法比其他的都好，但是為什麼要讓 Google 的工作更加困難呢？

怎麼做

首先妳得找出正確的雙字元語系碼 (language code)。英文的語系碼是「en」、法文的語系碼是「fr」、德文是「de」、中文則是「zh」。如果妳用的是其他語言，請到這裡找找。語系碼並沒有大小寫的分別。

現在把語系碼放在 <html> 標籤之前，實際的方法會依妳所使用的 HTML 版本而有所不同。請先確定妳的 DOCTYPE，然後依照以下的步驟：

1. 如果妳用的是任一版本的 HTML 4，請修改妳的 <html> 標籤（如果妳用的不是英文，就換上妳自己的語系碼）：

```
<html lang="en">
```

2. 如果妳用的是任一版本的 XHTML 1.0，請修改妳的 <html> 標籤（兩個地方都要用妳的語系碼）：

```
<html xmlns="http://www.w3.org/1999/xhtml" lang="en" xml:lang="en">
```

3. 如果妳用的是任一版本的 XHTML 1.1，請修改妳的 <html> 標籤（當然，用妳自己的語系碼）：

```
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
```

就跟 DOCTYPE 一樣，妳也應該在每個頁面裡指定妳所使用的語言。

另外也請注意：如果同一個頁面裡使用了一種以上的語言，妳也可以在任何元件裡指定使用的語言。比方說，如果妳的網站 (HTML 4) 主要使用德語，但是妳引用了一段英語的文章，那麼可以像這樣標註：

```
<html lang="de">
  ...
  <blockquote lang="en">
    ...
  </blockquote>
```

延伸閱讀

- 語系碼列表
- HTML 標準裡的 lang 屬性。
- Jim Thatcher on lang attribute support in screen readers. 他描述著他們的語系自動判斷演算法相當詭異；他之所以會知道這件事，是因為他設計了 Home Page Reader 的演算法。其他螢幕朗讀軟體自從他寫下了這個評論之後紛紛開始支援 lang 屬性，不過他們的自動判斷演算法仍然很詭異。所以妳更該要講明妳所用的語言了。

Dive Into Accessibility

在 30 天內打造更具親和力的網站

第八天：建立有意義的網頁標題

妳網站上的每個頁面，都該有個獨一無二且具有意義的頁面標題。

- 首頁的標題可以就祇是網站名稱。
- 時序彙整頁面應該包含網站名稱、以及這些頁面的日期或時序。例如在我的網誌上，我就用了像是以「dive into mark/June 19, 2002」為標題的逐日彙整，以及像是以「dive into mark/June 19, 2002」為標題的逐週彙整。
- 分類彙整頁面應該包含網站名稱以及類別名稱。例如我所有關於 CSS 的文章都彙整到一個名為「dive into mark/CSS」的頁面裡。
- 單篇彙整頁面應該包含網站名稱以及項目標題。我沒有讓單篇項目獨立出來，但 Jonathon Delacour 有，而且做得很棒。例如他在 2002 年 6 月 17 日發表的 Accessibility matters，就以「Jonathon Delacour: Accessibility matters」命名。

用哪些標點符號倒是無所謂；不過有一些螢幕朗讀軟體預設會把每個標點符號都唸出來。一般說來，過多的標點符號唸起來就跟看起來一樣笨。

誰因此獲益？

1. Jackie 從中獲益了。因為 JAWS 有個特別的鍵盤快速鍵 (INSERT + F10) 能顯示（並唸出）目前所有開啟視窗的標題。如果開啟的是網頁就會唸出網頁標題。當使用 ALT-TAB 切換視窗時，它也會唸出視窗標題。其他的螢幕朗讀軟體，像是 Home Page Reader 之類的，則會在載入網頁時讀出網頁標題。
2. Marcus 從中獲益了。因為 Lynx 會在第一行顯示網頁標題，所以 Marcus 總是可以先用布拉耶點字讀到標題。
3. Bill 從中獲益了。由於中風的關係，他有時候會搞混或是忽然忘記他讀的東西。視窗標題列的網頁標題就像是個視覺上的錨點，即使是他捲動頁面時仍會留在同一地方；所以他總是可以回頭看一下標題來喚起記憶。
4. Google 從中獲益了。因為 Google 會在搜尋的結果裡顯示網頁標題，而且位於網頁標題的關鍵字會有比較高的排名。這對於你來說是好事，尤其是對於那些單篇彙整頁面來說更是如此（選個好標題總是沒錯的）。

怎麼做

Movable Type 有許多種索引跟彙整的模版。預設的模版已經夠有親和力了；如果妳已經用了它們，就可以不需要做任何改變。

1. 主索引：<title><\$MTBlogName\$></title>
2. 彙整索引：<title><\$MTBlogName\$> 彙整</title>
3. 分類彙整：<title><\$MTBlogName\$>: <\$MTArchiveTitle\$></title>
4. 按時彙整：<title><\$MTBlogName\$>: <\$MTArchiveTitle\$></title>
5. 單篇彙整：<title><\$MTBlogName\$>: <\$MTEntireTitle\$></title>

Greymatter 有一套類似的模版，但是用了不同的語法。由於 Greymatter 沒有特別用來指定網誌名稱的變數，所以妳得一個個加上去：

1. Main Index Template: <title>My Weblog</title>
2. Archive Master Index Template: <title>My Weblog Archives</title>
3. Archive Log Index Template: <title>My Weblog: {{month}} {{year}}</title>
4. Entry Page Template: <title>My Weblog: {{entrysubject}}</title>

Manila 可以讓妳設定每日的標題，所以妳應該用它來取代日期，畢竟這與內文本本身比較相關。

1. Home Page Template: <title>{siteName}</title>
2. Template: <title>{siteName}: {title}</title>

在 Radio 就得用些秘訣了。妳可以用 Radio 的巨集語言為時序彙整頁面加入日期。請小心地複製並貼上這段巨集；它們之間不應該斷行，Radio 會在意這點（感謝 Jake Savin 提供這些步驟）。

1. Home Page Template: <title><%title%></title>
2. Main Template: <title><%title%><%local (d); if radio.weblog.file.getArchiveFileDate (radioResponder.fileBeingRendered, @d) {" : " + string.dateString (d)} else {""}%></title>

很遺憾，我不知道要如何自訂令人滿意的 Blogger 網頁標題。歡迎來信建議。

請記住這些祇是建議而已。妳可以自己決定是否要在按時彙整頁面中加入「彙整」這個字。只要不過量，標點符號的使用並沒有太大的關係。妳也可以改把網站名稱放在結尾；在網頁標題裡放入網站名稱總是不錯的；尤其當人們在許多視窗間切換時，這就能提供一些重要的線索。

延伸閱讀

- Jake Savin: Adding a date to your Radio archive pages

Dive Into Accessibility

在 30 天內打造更具親和力的網站

第九天：提供額外的導覽協助

也許妳對於使用 `<link>` 標籤來連結樣式表這件事並不陌生。但是妳知道可以用類似的語法來指出首頁及前後頁面的關係嗎？舉例來說，在按時彙整頁面裡，妳可以指出前一天以及後一天的（如果有的話）的文章。如果每則文章項目都有獨立的頁面的話，妳也可以指出前後的項目。

```
<link rel="home" title="首頁" href="http://網址/指到/首頁" />
<link rel="prev" title="前一頁的標題" href="http://前一頁/的/網址" />
<link rel="next" title="下一頁的標題" href="http://下一頁/的/網址" />
```

通常這些鏈結在圖形介面瀏覽器（例如 Internet Explorer）裡是看不見的；但是在其他瀏覽器裡則會被顯示出來，並且協助使用者瀏覽妳的網站。妳也許已經提供幾種不同的瀏覽方式了：逐週或逐月的彙整、最近幾篇文章的鏈結、顯示著每日文章的月曆等；也有可能彙整頁面上已經有往前後一日（篇）的鏈結了。這些都很好，請留著它們，然後再加上這些。

誰因此獲益？

1. Marcus 從中獲益了。他的純文字瀏覽器 Lynx 會在頁面頂端顯示這些額外的導覽協助，並顯示我們在 `title` 屬性裡指定的內容。當 Marcus 瀏覽 6 月 18 日的每日彙整頁面時，他看到的是：

```
#Home June 17, 2002 June 19, 2002
```

（# 可以讓 Marcus 瞭解這是詮釋資料，而不是網頁內容。Lynx 對於有意義的頁面標題也是用一樣的處理方式。）

2. Michael 從中獲益了。因為他的純文字瀏覽器 Links（別和 Marcus 的 Lynx 瀏覽器搞混了）也會在頁面頂端顯示這些額外的導覽協助，就像這樣：

```
Link: home
Link: prev
Link: next
```

3. Bill 從中獲益了。因為 Mozilla 會在網站導覽列顯示這些額外的導覽協助。這類的鏈結（「home」、「prev」、「next」）會以按鈕顯示，鏈結的標題則顯示成工具提示（tooltip）。（請注意：網站導覽列是在 Mozilla 0.99 版增加的，曾經在 1.0 版被移除，但在 1.1 版又被加回來了。在有支援的版本裡，妳可以從「檢視」選單、「顯示/隱藏」、「站台導覽列」、「永遠顯示」來顯示）

4. iCab 的使用者從中獲益了。因為 iCab 會在工具列上用下拉式選單來顯示每個網站連結的標題。

怎麼做

在 Movable Type 的按時彙整模版裡，把這幾列附加在 <head> 標籤之後：

```
<link rel="home" href="<$MTBlogURL$" title="Home" />
<MTArchivePrevious>
<link rel="prev" href="<$MTArchiveLink$" title="<$MTArchiveTitle
$>" />
</MTArchivePrevious>
<MTArchiveNext>
<link rel="next" href="<$MTArchiveLink$" title="<$MTArchiveTitle
$>" />
</MTArchiveNext>
```

再來單篇彙整模版裡，把這幾列附加在 <head> 標籤之後：

```
<link rel="home" href="<$MTBlogURL$" title="Home" />
<MTEntryPrevious>
<link rel="prev" href="<$MTEntryLink$" title="<$MTEntryTitle
$>" />
</MTEntryPrevious>
<MTEntryNext>
<link rel="next" href="<$MTEntryLink$" title="<$MTEntryTitle
$>" />
</MTEntryNext>
```

在 Greymatter 的模版裡，把這幾列附加在 <head> 標籤之後：

```
<link rel="home" title="Home" href="{ {pageindexlink} }">
<link rel="prev" title="{ {previousentrysubject} }"
href="{ {entrieswebpath} }/{ {previousentrynumberpadded} }.html">
<link rel="next" title="{ {nextentrysubject} }"
href="{ {entrieswebpath} }/{ {nextentrynumberpadded} }.html">
```

Sjoerd Visscher 提供了一套巨集，可以在 Radio 裡輕鬆做到這些事。

1. 下載導覽鏈結巨集。
2. 解開壓縮檔，並把四個檔案 (navigationLinks.txt, nextDayLink.txt, prevDayLink.txt, permalinkUrl.txt) 複製到 Macros 資料夾裡；按照 Windows 上的標準安裝程序來說，它會被放在 C:\Program Files\Radio UserLand\Macros。
3. 接著在 Main Template 裡，把這一系列附加在 <head> 標籤之後：

```
<%navigationLinks()%>
```

很遺憾地，我不知道要如何在 Manila 或 Blogger 裡實作令人滿意的前後文鏈結，但是至少妳可以在網頁的 <head> 標籤後加上這個鏈結：

```
<link rel="home" title="Home" href="http://url/of/your/home/  
page">
```

延伸閱讀

- Sjoerd Visscher: Navigation links in your Radio Userland weblog.

Dive Into Accessibility

在 30 天內打造更具親和力的網站

第十天：先呈現你的主要內容

純粹祇用 CSS 來處理版面的主要好處之一，是可以輕鬆的重新編排 HTML 源碼中各元件的順序，而不影響到實際的版面。因此當其它資料還在載入時，主要內容早就已經顯示出來了。我發現許多網站仍在使用表格來製作版面，所以我為你撰寫了這個訣竅。

如果你用表格設計版面，對 Marcus 與 Jackie 等失明的使用者來說，放在左側的瀏覽列會比主要內容更早顯示出來。我不知道怎麼形容隨之而來的問題，不過你可以自己看看：

1. 用表格製作的版面。
2. 修改過的版面，這次內文在前面。在圖形介面的瀏覽器上看起來應該一樣，但是在 Lynx 上有著明顯的差異。
3. 原本的版面在 Lynx 下的效果。
4. 修改過的版面在 Lynx 下的效果。會先顯示主要內文，然後才是導覽列。

你不需要從頭規劃整個模版來避免這個問題。我們可以用一些（相對之下）較簡單的技術來讓主要內文先顯示出來，並且讓導覽列維持在左側。我們稱這些技術叫「表格秘訣」。

誰因此獲益？

1. Marcus 從中獲益了。就跟上面的範例所展示的一樣，Lynx 會依照 HTML 源碼的順序顯示內容。這代表 Marcus 每次瀏覽你的網頁時都必須要捲動螢幕來跳過導覽列。捲頁爛透了。
2. Jackie 從中獲益了。因為 JAWS 跟 Lynx 一樣，都會依照 HTML 源碼的順序顯示內容；不過 JAWS 的情況會更糟，因為 Jackie 必須要坐著等 JAWS 唸完整個導覽列後才能聽到真正的內容，而且還沒有保證能直接跳到主要內容的方法。（明天我們就會談到這個問題。）
3. Google 從中獲益了。越接近頂端的內容，Google 會給予越多的加權。這裡的頂端不是指看起來的，而是指 HTML 源碼的頂端。實際上許多瞭解這個技術的人，都參與著搜尋引擎最佳化的研究；對他們來說，網站親和力反倒是其次了。

怎麼做

用 Lynx Viewer 檢查你的網站，確定每天的文章都會比導覽列先顯示。Movable Type 的預設模版就已經是正確的了；如果你用的是預設的模版，大概就不需要再做任何事情了；不過還是用 Lynx Viewer 試試看吧，你會因此對這個議題有更深一層的瞭解。

如果你使用的是 Radio 的預設模版，也許就需要修改表格，讓主要內文放在前面了。這沒有辦法用複製/

貼上的方式做到；妳必須鑽研自己的模版，並且檢視表格的架構。簡單版面與修改過的版面展示了基本的技術。

用表格製作的版面：

```
<table>
<tr>
  <td valign="top" align="left" width="25%">
    ... navigation bar ...
  </td>
  <td valign="top" align="left">
    ... main content ...
  </td>
</tr>
</table>
```

我們做了這樣的修改：

```
<table>
<tr>
  <!-- spacer GIF in upper-left cell -->
  <td></td>
  <!-- main content cell first, with rowspan=2 -->
  <td valign="top" align="left" rowspan="2">
    ... main content ...
  </td>
</tr>
<tr>
  <td valign="top" align="left" width="25%">
    ... navigation bar ...
  </td>
</tr>
</table>
```

延伸閱讀

- [Lynx Viewer.](#)
- [A Promotion Guide: The Table Trick.](#)

Dive Into Accessibility

在 30 天內打造更具親和力的網站

第十一天：跳過導覽列的鏈結

就算妳不想要把模版改成內容在前的樣子，至少也該妥協一下，提供跳過導覽區的鏈結。這並不是多完美的解決方案（先把主要的內容放在前面纔是王道），但至少是許多站台都願意採用的權變之策。

這個「跳過鏈結」就跟其他鏈結一樣，其實就祇是標準的 `<a>` 標籤；不過我們會用 CSS 把他隱藏起來，所以妳在 Internet Explorer 或 Netscape 上完全看不到，也不會影響到妳本來的版面設計。

誰因此獲益？

1. Marcus 從中獲益了。因為當他瀏覽這個頁面時，Lynx 就會顯示出這個鏈結，讓他能夠跳過導覽列，而直接閱讀主要的內容。至於這件事的重要性，請參照第十天：先呈現妳的主要內容裡所舉的例子。
2. Jackie 從中獲益了。因為當他瀏覽這個頁面時，JAWS 就會讀出這個鏈結，讓他能夠跳過導覽列，而直接閱讀主要的內容。

怎麼做

首先請用 Lynx Viewer 來看看妳的主頁面是不是會把導覽列放在主要內容之前。如果妳的主要內容更早出現的話，妳就不需要用到我們今天所要探討的訣竅了；放自己一天假，玩樂去吧！

現在讓我們定義這個跳過鏈結所要用的 CSS 樣式，讓它在視覺性瀏覽器上消失不見。如果妳用了外部樣式表的話，就把這個規則加到最後面（如果妳用了多重外部樣式表的話，請把這個規則加到適用 Netscape 4 的那一個裡）。如果妳本來祇在模版開頭的地方使用 `<style>` 區段來定義樣式的話，請在 `<style>` 標籤之後加入這個規則。

```
.skiplink {display:none}
```

接下來是把這個跳過鏈結直接插入到站台名稱和站台描述之後的地方。找不到嗎？也許妳該搜尋一下相對應的模版變數。

- Movable Type: 尋找 `<$MTBlogTitle$>` 和 `<$MTBlogDescription$>`。
- Greymatter: 沒有特定的模版變數；妳應該直接搜尋網站的名稱和標籤列。
- Radio: 尋找 `<%siteName%>` 和 `<%description%>`。
- Manila: 尋找 `{homePageLink (siteName)}` 和 `{tagLine}`。
- Blogger: 尋找 `<$BlogTitle$>`。

現在找到了嗎？切記要在站台名稱和描述的正下方插入這個跳過鏈結：

```
<a class="skiplink" href="#startcontent">跳過導覽列</a>
```

好，現在你還需要設立跳過連結所要指到的定位錨標籤，理論上也就是指到你的主要內容去。你找不到主要內容嗎？別著急，模版變數再度拯救了今天的課程。

- Movable Type: 尋找 `<MTEntries>`。
- Greymatter: 在你的 Main Index Template 裡尋找 `{{logbody}}`，然後在你的 Entry Page Templates 裡尋找 `{{entrymainbody}}`。
- Radio: 尋找 `<%bodytext%>`。
- Manila: 尋找 `{bodytext}`。
- Blogger: 尋找 `<Blogger>`。

接下來，要使用的定位錨標籤格式則會因你使用的 HTML 不同而有所出入。檢查一下你的 DOCTYPE，然後從底下選一個合適的來用：

1. 如果你用的是任何一種 HTML 4 的話，請在你的主要內容前面加入這個：

```
<a name="startcontent"></a>
```

2. 如果你用的是任何一種 XHTML 1.0 的話，請在你的主要內容前面加入這個：

```
<a name="startcontent" id="startcontent"></a>
```

3. 如果你用的是任何一種 XHTML 1.1 的話，請在你的主要內容前面加入這個：

```
<a id="startcontent"></a>
```

你應該在網站上的每一頁都提供這樣的跳過鏈結，所以在每個模版裡都依樣畫葫蘆，加入這些東西吧。

Dive Into Accessibility

在 30 天內打造更具親和力的網站

第十二天：安全地使用色彩

這個訣竅基本上可以廣泛地套用到網頁設計的各個領域，不過在此我祇會針對網誌裡的一種常見情形加以說明：鏈結文字。

有兩個潛在的問題會跟顏色有關。首先，妳所使用的鏈結文字可能跟底色的對比不夠明顯；任何白色底色上的淡色系文字都很糟，鏈結文字甚至會就這麼消逝於底色之中。同樣地，黑色底色上的深色系文字也很慘。事實上這個訣竅不僅適用於鏈結文字，而是適用於所有的文字；祇不過通常網誌上的一般文字都可以很容易的被閱讀，倒是鏈結文字常常一不小心就變得難以閱讀。這也就是我要在此提及的原因。

第二個潛在的問題是鏈結裝飾。如果妳在 CSS 中把鏈結重新定義成不同的顏色，那麼最好也要確定除了顏色之外，還提供了些別的線索，像是用粗體字、斜體字或加上底線等，讓讀者能夠加以區辨。否則雖然妳的鏈結文字能夠被閱讀，但有色盲的讀者將無法辨別那是否是一個鏈結。我們將在後面的小節加以描述。

誰因此獲益？

1. Michael 從中獲益了。以下是三種不同的鏈結裝飾組合：

1. [This link](#) is underlined, the default behavior.
2. **This second link** is bold and colored instead.
3. **However** this link is distinguished only by color.

正如妳所看到的，第一句裡的鏈結採用了預設的組合，會在視覺性瀏覽器裡顯示成藍色/紫色並加上底線。第二個鏈結套用了兩種文字裝飾，會顯示成紅色粗體字（但是不會加上底線）。第三個鏈結祇有套用一種文字裝飾，所以祇會顯示成紅色。

現在這裡是相同的三個鏈結，祇不過這是 Michael 所看到的樣子。

1. [This link](#) is underlined, the default behavior.
2. **This second link** is bold and colored instead.
3. **However** this link is distinguished only by color.

如妳所見，第一個鏈結仍然可以被清楚地看到；Michael 的色盲並不會影響到藍色的部分。在第

二個句子中，鏈結的紅色褪去了，幾乎變成黑色；不過鏈結本身仍然呈現粗體字，所以 Michael 仍然可以分辨得出來。第三個鏈結就出問題了。因為這個鏈結原本祇能靠紅色來分辨，可是現在紅色褪成黑色，於是單靠視覺將難以將之與其他尋常文字加以區辨。

怎麼做

如果你要檢查你的文字顏色跟背景顏色是否有「顯著對比」，你可以用 VisCheck 來把你的網頁模擬成色盲讀者所看到的樣子。

如果你要檢查鏈結裝飾的問題會不會發生，你可以在 CSS 規則裡尋找 "a" 標籤所會用到的樣式。舉例來說，如果你在 CSS 裡用了下列的規則，那麼你的鏈結將祇能夠用紅色來加以區辨，這樣是不對的：

```
a {
  text-decoration: none;
  color: red;
}
```

你的鏈結可以沿用紅色，但是你得確定你的鏈結同時也呈現成粗體字、斜體字或加上底線。如果你要讓它變成紅色粗體字，就祇要加上一列：

```
a {
  text-decoration: none;
  color: red;
  font-weight: bold; /* 祇需要加上這一行 */
}
```

延伸閱讀

- VisCheck 可以模擬色盲的效果，而讓你看到色盲讀者所會看到的樣子。你不但可以用他來檢查某一張圖片，也可以檢查整個網頁。
- Cal Henderson: Color Vision. 這裡有各種色盲的實際可見色譜範圍。
- Ishihara Test for Color Blindness 有一系列紅綠色盲難以看清甚至完全看不到的圖片。

Dive Into Accessibility

在 30 天內打造更具親和力的網站

第十三天：使用真實鏈結

網頁設計的苦難根源乃是「javascript:」鏈結；這個假鏈結會在妳按下的時候執行一段 Javascript 程式碼。這個問題最常發生於，網誌用這種鏈結來在另一個視窗裡顯示迴響。為什麼這會是個問題呢？因為有 11% 的網際網路使用者因為某些原因而不使用 Javascript；這同時也包括許多殘障的朋友，因為她們所能夠使用的瀏覽器就是不支援。這種假鏈結對她們來說不能用，妳應當改用真實的鏈結。

雖然這個問題很容易描述及修改，但是無論我多麼強調這個訣竅的重要性也都不為過。多數問題，像是不提供跳過導覽列的「跳過鏈結」都祇不過會在某種程度上減低可用性罷了，至少妳的頁面還能夠被讀取。但是現在我們所遇到的這個問題，卻會讓重要的內容幾乎完全無法被取得。如果妳的迴響被藏在某個「javascript:」鏈結之後，就幾乎不存在了。

誰因此獲益？

1. Marcus 從中獲益了。因為 Lynx 並不支援 Javascript。
2. Michael 從中獲益了。因為 Links 並不支援 Javascript。
3. Lillian 從中獲益了。雖然她用的是 Internet Explorer，但是資訊部門祇允許某些獲准的站台使用 Javascript，而把其他站台的 Javascript 給擋掉了；很不幸地，妳的站台並不在這些獲准的站台之中。
4. Google 從中獲益了。因為 Google 會跟著鏈結來尋找更多內容並加以建立索引，但是它卻無法使用「javascript:」鏈結；因為當它索引網頁時並不會執行任何 Javascript 程式碼。

怎麼做

現在 **Movable Type** 和 **Radio** 的預設模版都把這個部分處理好了，所以妳無須再做任何處理。檢視妳的首頁並搜尋「javascript:」；如果完全找不到，那麼也用不到今天的訣竅了。

然而，如果妳的 Movable Type 模版包含有像這樣的鏈結：

```
<a href="javascript:OpenComments(<$MTEntID$>)">Comments (<$MTEntCommentCount$>)</a>
```

就請把它改成這樣：

```
<a href="<$MTCGIPath$>mt-comments.cgi?entry_id=<$MTEntID$>"
onclick="OpenComments(<$MTEntID$>); return false">Comments (<$MTEntCommentCount$>)</a>
```

在支援 Javascript 的瀏覽器裡，原來的鏈結仍會具有相同的作用；因為 `onclick` 屬性會優先於 `href` 屬性。所以在新版裡，仍然會叫用 `OpenComments` 函式，然後彈出新視窗。然而在不支援 Javascript 的瀏覽器（還有 Google）的情況下，就會完全忽略 `onclick` 屬性，而祇從 `href` 屬性來連結，用同一個視窗來顯示評論。

如果你因為其他原因而用了 `javascript:` 假鏈結的話，停。停下來。別繼續下去，別繼續領四千元或之類的。在你的源碼裡套用上述的訣竅，讓不支援 Javascript 的瀏覽器至少還有機會能夠使用真實鏈結。

延伸閱讀

- Jeff Howden: Links & JavaScript Living Together in Harmony.

附註

甚至別開始用那些以 Javascript 做出來的動態選單系統。他們確實讓畫面看起來很酷，不過抽煙也一樣。用真實鏈結吧。

Dive Into Accessibility

在 30 天內打造更具親和力的網站

第十四天：在鏈結上加入標題

既然網頁上充斥著鏈結，妳可能會因此覺得有很多人知道 `title` 屬性；不過我真的很少看到它。很多人可能不知道，所有的鏈結都可以在 `<a>` 標籤裡用 `title` 屬性來指定一個標題。這個標題會額外附加到任何妳指定的鏈結文字上；它可以在視覺性瀏覽器上產生工具提示，而且在非視覺性的瀏覽器上也一樣會有類似的效果。

並非所有的鏈結都該有標題。如果鏈結文字是某一篇文章的標題，就請不要加上標題；因為鏈結文字本身應該就足夠清楚了。但是如果妳閱讀鏈結文字本身時，發現自己在狀況外而不知道它會指往何處的話，就加上標題吧。

誰因此獲益？

1. Jackie 從中獲益了。因為 JAWS 提供了一個選項，可以跟著鏈結文字讀出鏈結標題。（這個選項預設並不會被選取。為了要啟用這個功能，Jackie 按下了 `INSERT+V` 叫出 JAWS 的冗長選項視窗，然後把 "Text links verbosity" 改成 "Alt tag or title"。）
2. Michael 從中獲益了。因為當她把游標移到某個叫 Opera 的鏈結上時，鏈結的標題會顯示在狀態列上，同時也以工具提示的樣子顯示出來。這讓他得以決定是否要把頻寬花在這個鏈結上。
3. Lillian 從中獲益了。因為當她在 Internet Explorer 裡把游標移到鏈結上時，鏈結標題就以工具提示的樣子顯示出來了。
4. Marcus 從中獲益了。因為當他按下了「`l`」之後，Lynx 就會顯示出那一頁的鏈結清單；這份清單同時包含了每個鏈結的標題（如果有的話）。

怎麼做

每一個鏈結文字本身並不一定能提供足夠的資訊，來讓讀者決定是否要按下那個鏈結。這個時候妳應該加入一個 `title` 屬性，像這樣：

1. 在我的導覽列上，有一個連到我的統計頁面的鏈結。這個鏈結的鏈結文字就只是 Statistics 而已，但是 `title` 屬性卻可以提供更多資訊：

```
<a title="referrers and other visitor statistics" href="/stats/">Statistics</a>
```

2. 在我的導覽列上，有一個連到我的書 Dive Into Python 的鏈結，看起來就像這樣：

```
<a title="Free Python book for experienced programmers"
```

`href="http://diveintopython.org/">Dive Into Python`

3. 當我用句子裡的某個詞連結到某一篇文章時，我會試著用 `title` 屬性來提供更多關於這個鏈結的資訊，像是那篇文章的標題或引述之類的。再舉個例子，昨天的訣竅裡就包含這個句子：

為什麼這會是個問題呢？因為``有 11% 的網際網路使用者因為某些原因而不使用 `Javascript` ；這同時也包括許多殘障的朋友，因為她們所能夠使用的瀏覽器就是不支援。

在瀏覽器裡看起來會像這樣：

為什麼這會是個問題呢？因為有 11% 的網際網路使用者因為某些原因而不使用 Javascript ；這同時也包括許多殘障的朋友，因為她們所能夠使用的瀏覽器就是不支援。

切勿濫用 `title` 屬性。凡事皆適可而止。

延伸閱讀

- Jakob Nielsen: Using Link Titles to Help Users Predict Where They Are Going.

Dive Into Accessibility

在 **30** 天內打造更具親和力的網站

第十五天：定義快速鍵

HTML 有一個鮮為人知的功能，就是在鏈結及表單裡還可以使用 `accesskey` 屬性，讓網頁設計師定義常用鏈結或表單區域的快速鍵。在 Windows 上，妳可以按下 ALT + 某個便捷鍵；在 Macintosh 上則是按下 Control + 某個便捷鍵。如果便捷鍵是被定義在某個鏈結上的話，妳的瀏覽器就會跟著連結出去；如果是被定義在表單區域的話，妳的瀏覽器就會把焦點一到那個區域裡。Internet Explorer 從 4.0 版開始支援便捷鍵，Netscape 則是從 6.0 版開始支援。比較舊的瀏覽器會單純地忽略掉這個屬性，所以不會造成甚麼可怕的影響。

雖然那個按鍵對應到甚麼功能並沒有一個標準，但是這裡還是列出了一些常用的快速鍵：

便捷鍵 1

主頁面

便捷鍵 2

跳到主要內容（也就是導覽列跳過鏈結）

便捷鍵 9

回應

誰因此獲益？

1. Jackie 從中獲益了。因為當 JAWS 讀到定義有 `accesskey` 的鏈結時，也會把這個便捷鍵唸出來。舉例來說，`Home pageD` 這個鏈結會被 JAWS 讀成「link: Home page, ALT + 1」。所以 Jackie 可以按下 ALT+1 來把瀏覽器焦點移到那個鏈結，然後再按下 ENTER 就可以從那個鏈結連出去。
2. Bill 從中獲益了。雖然 Bill 受到中風的影響，沒辦法有效率地使用滑鼠，但是他還可以靠鍵盤導覽以及快速鍵的幫忙來在頁面中移動。便捷鍵可以幫他非常有效率地跳到常用的鏈結。Bill 能夠按下 ALT+1，接著 Mozilla 就會馬上連到定義有 `accesskey="1"` 的鏈結去。（註：既然 Mozilla 不會把便捷鍵唸出來，那麼 Bill 要怎麼發現有這個便捷鍵可以用就會變成一個重要的議題了。我們將在後面討論到這個訣竅。）

怎麼做：主頁鏈結

- **Movable Type:** 預設的模版並沒有連回主頁的鏈結，所以妳應該在站台名稱的地方加上鏈結，並且賦予一個 `accesskey`。在妳的模版裡搜尋 `<$MTBlogName$>`，然後改成這樣：

```
<a href="<$MTBlogURL$>" style="color:black; text-decoration:none" accesskey="1"><$MTBlogName$></a>
```

- **Radio:** 在妳的模版中搜尋 {siteName}。這很有可能已經是某個鏈結了，像這樣：

```
<a href="<%radio.macros.weblogUrl (%)%" style="color:black; text-decoration:none"><%siteName%></a>
```

所以妳祇要在這個鏈結裡加上 accesskey 屬性就行了。像這樣：

```
<a href="<%radio.macros.weblogUrl (%)%" style="color:black; text-decoration:none" accesskey="1"><%siteName%></a>
```

- **Blogger:** 在妳的模版裡搜尋 <\$BlogTitle\$>。如果這個標籤已經被 <a> 標籤關上了的話，就在 <a> 標籤裡加上 accesskey="1" 屬性，就跟前面的 Radio 一樣。如果妳的 <\$BlogTitle\$> 標籤還沒有被 <a> 標籤關上的話，就用像這樣的標籤把它關上（別忘記要插入妳自己的主頁網址）：

```
<a href="http://妳的/主頁的/網址" style="color:black; text-decoration:none" accesskey="1"><$BlogTitle$></a>
```

怎麼做：跳過導覽鏈結

妳是否有提供一個跳過導覽列的鏈結呢？如果有的話，就指定 accesskey="2" 給它吧。

```
<a class="skiplink" href="#startcontent" accesskey="2">跳過導覽列</a>
```

怎麼做：回饋鏈結

妳是否有提供連到回饋表單或妳的電子郵件住址的鏈結呢？如果有的話，就指定 accesskey="9" 給它吧。

```
<a href="mailto:me@mydomain.com" accesskey="9">寄信給我</a>
```

注意：Radio 網誌通常會有個連到回饋表單的鏈結（小信封圖示），但是這個鏈結是由巨集所產生的，所以妳沒辦法在上面加上 accesskey。

請確定妳在網站上的每一頁都加上了 accesskey；這表示妳得修改所有相關的模版。

延伸閱讀

- Jukka Korpela: Improving accessibility with `accesskey` in HTML forms and links. Explains why all my suggested `accesskey` codes are numbers, instead of letters.
- Paul Bohman: Access keys, IE6. 這是 Web Accessibility Forum Mailing List 上關於 `accesskey` 的部分討論。

Dive Into Accessibility

在 30 天內打造更具親和力的網站

第十六天：不要開出新視窗

幾乎每個網頁使用者都知道「上一頁」這個按鈕。這幾乎也是瀏覽網頁的必備動作了。連到某個鏈結，回到上一頁；瀏覽某個搜尋結果，然後回到上一頁。就連我老爸也知道可以這樣用，而他本人現在還因為第一次成功地在「Internet」圖示上雙擊成功而興奮不已。

在所有的主流瀏覽器裡，如果你用了 `` 標籤，就會強迫鏈結開出一個新視窗而讓「上一頁」按鈕失效。這個新視窗不會保有前一個視窗的瀏覽歷程，所以「上一頁」按鈕將會失去功效。即使是對有 10 年網頁經驗的我來說，這都會造成嚴重的混淆。在 2002 年的今天人們還會這樣子用，實在是太不可思議了。別這樣做。別強迫鏈結開啟新視窗。

請注意這個訣竅是針對網頁設計師所提出的，而非針對網頁使用者。如果你想要在妳瀏覽網頁的時候開出新視窗，請用後述的方法。在 Windows 上的 Internet Explorer 裡，按下鏈結時同時按住 Shift 鍵，就可以把連結開到新視窗去。在 Netscape 6 和 Mozilla 裡的話，則是同時按住 Control。在 Mac 上的 Internet Explorer 裡則需要按住 Command。（另外像是 Opera 的某些瀏覽器，還支援進階組合鍵，像是 Control + Shift + 點選，這樣會在背景開出連結的新視窗。）這裡的重點在於是否要開出新視窗應該要由使用者來選擇，而不是由網頁設計師。

誰因此獲益？

1. Jackie 從中獲益了。雖然 JAWS 會在連結開出新視窗時唸出「New browser window」，但仍然容易在它唸出鏈結文字和新頁面內容之間被漏掉。Home Page Reader 提供了更好的解決方案：每當有新視窗開啟時，它就會播放某個指定的聲音。另外一個叫 Window Eyes 的螢幕朗讀軟體則完全不會提示有新視窗。

無論如何，「上一頁」按鈕就此爛掉。如果 Jackie 沒有聽到「new browser window」的話，就沒辦法瞥見工作列上開了兩個瀏覽器視窗了。於是她就得去讀整個已開啟視窗清單；可能是透過 JAWS 指定的視窗清單快速鍵 `INSERT+F10` 或標準的 `ALT+TAB`。

2. Lillian 從中獲益了。因為她的 Internet Explorer 視窗總是最大化（這樣她纔看得見），所以新視窗開啟的時候也會預設變成最大化。除此之外 Windows XP 還會把同一個應用程式的多個視窗群組整理在工作列上，所以在視覺上就更難發現有新視窗被開出來了。突然之間，「上一頁」按鈕莫名其妙地失效了，而 Lillian 完全不知道為什麼會這樣。如果你預期她會很高興地閱讀連結後的內容，倒是可以略過這個問題。
3. Bill 從中獲益了。他妹妹把 Mozilla 設定成多頁瀏覽模式，所以 Bill 可以看到分頁，然後很快地記住他開的是那個視窗，並且在那些分頁間切換（在他靈巧華麗的延伸鍵盤上用 `CTRL+PAGEUP` 和 `CTRL+pagedown`）。被強迫開出新視窗的連結會開出另外一個全新的 Mozilla 視窗。這不僅僅會推翻分頁瀏覽的偏好，還會讓所有已開啟的視窗都不見；因為新的視窗上不會有任何前一個視

窗的分頁。

怎麼做

1. 不要用 `` 來強迫連結開出新視窗。
2. 如果你一定得要把某個連結開到新視窗，請務必明確地警告讀者。這祇是個不理想的權變方案，通常是隨著某些「不得與外部內容相關聯」的商業政策而來的。舉例來說，CNN 的「相關站台」頁面就是這樣。
3. 如果你看到「在新視窗開啟鏈結」的核選框，請確定它按照預設值而處於關閉的狀態。

延伸閱讀

- Jakob Nielsen: The Top Ten New Mistakes of Web Design. "#1: Breaking or Slowing Down the Back Button. #2: Opening New Browser Windows."
- W3C Web Accessibility Initiative: Example for Checkpoint 10.1 舉例說明了當某個鏈結一定得開啟新視窗時，你該如何警告使用者。
- W3C Validator mailing list: Re: opening a link in a new window. 身為關心此類事情的人，你應該知道 `<a>` 標籤中的 `target` 屬性是被反對使用的，而且會使得你的頁面無法通過 HTML 4.01 Strict, XHTML 1.0 Strict 或任何未來版本的認證。
- WebAIM mailing list: `mailto:` links open new windows. 這裡提出 `mailto:` 鏈結並非親和力障礙，因為這個行為完全可以由用戶端判斷。網頁介面的電子郵件表單（像是 Radio 所用的）也許是提供親和力的更完整解決方案。網頁介面的表單可以無須整合電子郵件用戶端就讓訪客使用（而且不會有組態不當或情境不允許等問題，像是實驗室的公用電腦），同時也可以在不使用不親和的 Javascript 伎倆的情況下就保護你的電子郵件地址免遭垃圾信發送機器人的攻擊。另一方面，有一些人會因為熟悉感、功能（像是內建拼字檢查）或能夠整理寄出的訊息等原因真的喜歡他們自己的電子郵件用戶端軟體；在這種情況下，我就不會建議你考慮這種方法了。

Dive Into Accessibility

在 30 天內打造更具親和力的網站

第十七天：定義頭字語

我上個月曾經在我自己的網誌上用了 50 個頭字語：ADA, ALT, AOL, API, CGI, CMS, CSS, CTRL, DMV, DNS, DTD, EFF, FAQ, FSF, GFDL, GIA, GPL, HTML, IE, IIRC, IIS, IO, KB, KDE, LONGDESC, MB, MSDN, MSN, MT, Mac, NC, OPML, P2P, PGDN, PGUP, PBS, PDF, PONUR, RSS, RU, SOAP, SSN, TDD, US, VNC, W3C, WCAG, WYSIWYG, Win, XHTML, 和 XML.

如果這 50 個頭字語妳都懂，那實在是可喜可賀；妳顯然是技術編輯的大好人才。否則妳就會感激我實際上用了 `<acronym>` 標籤把他們逐項加以定義了。把妳的游標移到每一個頭字語上，看看他們各代表些甚麼意思。在現今的瀏覽器上這都有效，而對於 Netscape 4 之類的則無傷。

妳應該在任何使用頭字語的時候都加以定義，或在每一篇裡面至少定義過一次。

誰因此獲益？

1. Michael 從中獲益了。因為當 Michael 把游標滑過頭字語的時候，Opera 就會用工具提示的方式顯示頭字語標題。
2. Bill 從中獲益了。因為 Mozilla 甚至會自動在頭字語底下加上點狀底線；當 Bill 把游標滑過頭字語時，Mozilla 還會把游標變成游標 + 問號的形狀，並且以工具提示的方式顯示頭字語標題。（妳可以用樣式表來覆寫掉這個預設的作用，或者用 CSS 來讓其他瀏覽器也有相似的效果）。
3. Google 從中獲益了。因為 Google 不但會用頭字語本身做索引，也會用頭字語標題來做索引，所以人們就算用完整的拼寫當關鍵字也可以找到妳的頁面。
4. 我希望我也能說 Jackie 從中獲益了，可惜並沒有。因為不論 JAWS 或其他任何目前市面上的螢幕朗讀軟體都不會唸出頭文字標題。我希望有朝一日他們將能夠有這個功能，那麼妳就搶先了一步。

怎麼做

當妳第一次使用頭文字的時候，用一個 `<acronym>` 標籤加以標記，像這樣：

```
<acronym title="cascading style sheets">CSS</acronym>
```

Radio 的使用者可以用快速鍵自動地加以標記。從妳的 Radio 主頁裡，在主導覽選單中點選「Shortcuts」然後定義妳常用的頭文字。舉例來說：

Name: CSS

Value: <acronym title="cascading style sheets">CSS</acronym>

(Be sure to change the input type from "WYSIWYG" to "Source" so you can type the HTML directly.)

然後在你的文章裡，只需要鍵入 "CSS"（包含引號），那麼 Radio 就會自動地按照你所定義的內容加上頭文字標籤。

怎麼做：樣式表

你還可以用樣式表來更換所有頭字語看起來的樣子。不單單是 Radio，這對所有的工具都有效。以下是我在所有的瀏覽器（而不單單只是 Mozilla）中產生點狀底線的規則：

```
acronym {
  border-bottom: 1px dotted black;
}
```

這麼做還有額外的好處；我用了下面這個規則，所以當列印我的網頁時，會自動地把頭文字展開成全稱。（其實祇有從 Mozilla 和 Opera 列印時纔有效，不過對於其他瀏覽器來說是無害的）。

```
acronym:after {
  content: " (" attr(title) )";
}
```

延伸閱讀

你可曾用過某些頭文字，卻不知道他們到底是甚麼嗎？現在找出來吧。

- Acronym Finder.
- Acronym Database.
- Acronym Search.

附註

有許多熱衷標記語言的大師會指出說還有一個用於縮寫字的 <abbr> 標籤；很不幸地，Windows 上沒有任何版本的 Internet Explorer 支援這個標籤，所以他們不會為此顯示出工具提示。這時你還是得用 <acronym> 纔行。

Dive Into Accessibility

在 30 天內打造更具親和力的網站

第十八天：在月曆中加上真的標題

「可是，」我聽到妳在哀嚎了，「我的月曆已經有個標題了啊！看那邊，明明在上面就有年份跟月份了呀，我甚至還用了粗體字耶。」

但是，如果妳挖掘過妳的 HTML 源碼，就會發現妳的月曆並沒有真的標題。它可能祇是一個表格的 `<td>` 格子，佔滿著整個第一列，然後用了 CSS 規則來讓它看起來是粗體字。相較之下，如果妳用了真正的 `<caption>` 標籤的話，反而簡單多了。這樣不但在模版裡讀起來更簡單，也可以為妳的頁面多省下幾個位元，在視覺性瀏覽器裡看起來效果也相同，同時還更具親和力。

誰因此獲益？

1. Marcus 從中獲益了。因為 Lynx 顯示標題時，會在前面再加上「CAPTION:」，讓他明白這一行是標題，而不是欄位標頭或表格資料。
2. Jackie 間接地獲益了。因為用了真正的 `<caption>` 標籤能夠奠定使用真正表格標頭的良好基礎；我們將在明天的訣竅中討論到 Jackie 到底能從中得到些甚麼。

怎麼做

這個訣竅祇能夠適用於支援月曆的出版工具（所以 Blogger 就被排除掉了），同時這個出版工具還要能讓妳自訂產生月曆的 HTML 源碼（所以 Manila 也被排除掉了）。

在 Movable Type 中，妳的主索引模版裡可能有個用來當月曆的表格，就像這一個（搜尋「calendarhead」就差不多能找到它了）：

```
<table border="0" cellspacing="4" cellpadding="0">
<tr>
<td colspan="7" align="center"><span class="calendarhead"><
$MTDate format="%B %Y"$></span></td>
</tr>
<tr>
<td align="center"><span class="calendar">Sun</span></td>
...
```

留著這個表格標籤，但是把表格中第一個 `<tr>` 列全部代換成真正的 `<caption>` 標籤，就像這樣：

```
<table border="0" cellspacing="4" cellpadding="0">
```

```
<caption class="calendarhead"><$MDate format="%B %Y"$></caption>
<tr>
<td align="center"><span class="calendar">Sun</span></td>
...
```

剩下來的部分留著別動；我們明天再來加以修改。

<caption> 標籤裡的 class 屬性乃是選用性的；我在這裡把他留著，因為這樣就可以拿來取代掉預設的 Movable Type 樣式，繼續用 CSS 規則來讓年份及月份採用粗體字。像這樣子用 <caption> 標籤之後，妳的頁面看起來仍舊會跟以前一模一樣。

在 Greymatter 中的狀況仍舊相同，但是模版標籤卻不同：

```
<caption>{{monthword}} {{yearyear}}</caption>
```

同樣地，祇要妳這麼做，就可以用樣式表來改變 caption 的視覺樣式。

在 Radio 中，整個過程將會比較複雜，但並非不可能（我得感謝 Tony Bowden 提供了這些指引）。

1. 在 Radio 裡，開啟實際的 Radio 應用程式。如果是在 Windows 的話，從系統列裡在小 Radio 圖示上按滑鼠右鍵，然後選擇「Open Radio」。
2. 在「Tools」選單裡，選擇「Developers」，然後選「Jump...」（Control+J）。跳至「system.verbs.builtins.radio.weblog.drawCalendar」（不含引號）。
3. 現在到「Edit」選單裡，選「Find and Replace」，然後選「Find...」（Control+F）並且尋找「hCalendarTable」。這樣應該會展現出繪製啟始 <table> 標籤和假月曆標題的源碼區塊。
4. 把這個區塊中最後一列（在 <tr> 標籤中寫著 *monthYearString* 的那一列）改成這樣：

```
add ("<caption>" + monthYearString + "</caption>")
```

5. 關閉視窗。妳會被詢問到是否要編譯這個變更，請回答是。
6. 如果妳想要的話，還可以設定標題的樣式。到妳的 Home Page Template（在 Prefs 頁面中）並且新增一個給 caption 用的樣式。以下這個是我自己在用的；在 <style> 區段中原本包含這些東西：

```
body, td, p {
    font-family: verdana, sans-serif;
    font-size: 12px;
}
```

現在則是：

```
body, td, p, caption {
```

```
font-family: verdana, sans-serif;  
font-size: 12px;  
}
```

```
caption {  
  text-align: center;  
  font-weight: bold;  
}
```

延伸閱讀

- [Tony Bowden: Changing the Calendar in Radio.](#)

Dive Into Accessibility

在 30 天內打造更具親和力的網站

第十九天：使用真的表格標頭

如果妳的站台上有一個月曆，就應該是以 HTML 表格畫出來的。試圖做出純 CSS 月曆的人並不多，而且那樣子其實反而不對；因為月曆本來就是資料表格，所以就應該用表格來做出來。

資料表格中最重要的一件事就是要做出適當的標頭。在月曆這個例子裡，就是指最上面的星期幾。妳應該明確地在表格中囊括這是星期幾的標頭；如果妳不想要讓他們被看見，就該用 CSS 來處理掉（我在我的網誌上就是這樣子用的）。但是標頭無論如何都該存在，因為螢幕朗讀軟體會靠著這些東西來讓盲胞讀者得以瀏覽表格而不會迷失。

月曆這玩意兒（其實任何資料表格都一樣，真的，但是我們今天祇討論月曆的部分），如果妳一次看整個的話用起來就很簡單；但是如果妳一次祇看得到一天的話用起來就會很困難。想像一下妳桌上如果有一份日曆，但是上面祇寫出當天的日期，卻不標示那是星期幾。就這樣一頁一頁下去：1, 2, 3, 4, 5, 6, 7... 今天是 4 號，而我還知道這是星期四。現在繼續翻到 11, 12, 13. 快一點：13 號是星期幾？這一頁不會告訴妳，妳得自己去追查出來。

這就跟盲胞讀者在瀏覽沒有合適標頭的月曆時所發生的狀況相似。妳祇會得到一大堆數字，卻沒有能夠加以追蹤的情境線索。在月曆中加上適當的標頭能夠讓螢幕朗讀軟體把表格標頭（星期幾）跟表格資料（日期）聯繫在一起，並且一起唸出來。「Thursday 4, Thursday 11, Friday 12, Saturday 13」，嗯，原來是星期六啊。

請注意我說的是適當的標頭。單單祇把星期幾放在第一列的 `<td>` 標籤裡是不夠的，妳還得改用 `<th>` 標籤纔行。多數的網誌模版在這裡都弄錯了，不過要修好其實很簡單，而且妳的月曆並不會因此而看起來有所不同。

誰因此獲益？

1. Jackie 從中獲益了。因為當她遇到妳的月曆時，JAWS 會先念出標題，然後再唸出標頭，接著 Jackie 可以按著 `Control + ALT` 不放，就能夠用方向鍵在表格裡移動。當她移動的時候，JAWS 還會同時唸出標頭（星期幾）和格子裡的資料（日期）。

所有的主流螢幕朗讀軟體都支援這類的表格導覽功能。Home Page Reader 就允許使用者切換到「表格導覽」模式 (`ALT+T`)，然後使用者無須再按住任何按鍵，就能夠在月曆裡瀏覽。Home Page Reader 在這裡確實略勝 JAWS 一籌。我們馬上就會看到，妳還能為每一個表格標頭定義簡短（或更長）的標題（有點像在為鏈結增加標題一樣），這時 Home Page Reader 還會改念這個標題。這表示妳可以在視覺上把妳的星期顯示成「Sun」、「Mon」、「Tue」，而讓 Home Page Reader 把他們唸成「Sunday」、「Monday」、「Tuesday」。真酷！

怎麼做

如果你從來沒有這樣做過，請先確定你的月曆有真的標題。 <caption> 標籤一定要緊接在 <table> 標籤之後，然後 <th> 標籤列纔再緊接在這之後。

在 Movable Type 中，從你的主索引模版裡尋找月曆的部分（同樣地，你可以搜尋「calendarhead」就差不多找到了）。緊接在 <caption> 之後，你可以看到星期幾被定義著：

```
<tr>
<td align="center"><span class="calendar">日</span></td>
<td align="center"><span class="calendar">一</span></td>
<td align="center"><span class="calendar">二</span></td>
<td align="center"><span class="calendar">三</span></td>
<td align="center"><span class="calendar">四</span></td>
<td align="center"><span class="calendar">五</span></td>
<td align="center"><span class="calendar">六</span></td>
</tr>
```

請把它改成這樣：

```
<tr>
<th abbr="星期日" align="center"><span class="calendar">日</span></th>
<th abbr="星期一" align="center"><span class="calendar">一</span></th>
<th abbr="星期二" align="center"><span class="calendar">二</span></th>
<th abbr="星期三" align="center"><span class="calendar">三</span></th>
<th abbr="星期四" align="center"><span class="calendar">四</span></th>
<th abbr="星期五" align="center"><span class="calendar">五</span></th>
<th abbr="星期六" align="center"><span class="calendar">六</span></th>
</tr>
```

這裡有兩個要點：所有的 <td> 標籤都換成了 <th>，而且他們也都被加進了「abbr」屬性來指定星期幾的全稱（「abbr」屬性會有兩個功用。對於非常長的表格標頭來說，它可以被當成縮寫來用，一如其名；但對於非常短的表格標頭來說，它卻能提供較長的版本。這也就是我們在此所做的事。）

在 Radio 中，程序就跟昨天你在給表格真的標題時很像。

1. 在 Radio 裡，開啟實際的 Radio 應用程式。如果是在 Windows 的話，從系統列裡在小 Radio 圖示上按滑鼠右鍵，然後選擇「Open Radio」。
2. 在「Tools」選單裡，選擇「Developers」，然後選「Jump...」(Control+J)。跳至「system.verbs.builtins.radio.weblog.drawCalendar」（不含引號）。
3. 現在到「Edit」選單裡，選「Find and Replace」，然後選「Find...」(Control+F)並且尋找「addDayName」。這樣應該會展現出好幾個高亮度的 addDayName 函式。在三角形上雙擊以顯示出函式碼，他們看起來應該像這樣：

```
on addDayName (name)
  add ("<td width=\"19\" height=\"10\" align=\"center\"
style=\"font-size:9px\">" + name + "</td>")
```

4. 請把他們改成這樣：

```
on addDayName (name, fullname)
  add ("<th abbr=\"\" + fullname + \"\" width=\"19\" height=
\"10\" align=\"center\" style=\"font-size:9px\">" + name +
"</th>")
```

5. 現在下面的「for i = 1 to 7」那一系列上雙擊，妳應當會看到這樣：

```
for i = 1 to 7
  addDayName (radio.string.getLocalizedNameString
("dayOfWeekShort." + i))
```

6. 請改成這樣：

```
for i = 1 to 7
  addDayName (radio.string.getLocalizedNameString
("dayOfWeekShort." + i), radio.string.getLocalizedNameString
("dayOfWeek." + i))
```

關於表格排版的極重要注意事項

用來處理視覺排版的表格則有完全不同的一套規則。別在排版表格裡使用 <th> 標籤。此外別忘了適當地調整表格，讓你的主要內容先呈現出來；要讓排版表格更具親和力能做的事情不多，我們將在明天拿一點出來討論。

延伸閱讀

如果你要把資料表格作得比月曆還複雜（像是有多層標頭和次標頭的表格）時，請自己想辦法。這裡列

出一些開頭的好方向：

- [Jim Byrne: Table Manners.](#)
- [WebAIM: Create tables that transform gracefully.](#)
- [Kynn Bartlett: Understanding Accessible Table Markup.](#)

Dive Into Accessibility

在 30 天內打造更具親和力的網站

第二十天：提供表格摘要

最後一件製作表格的重要步驟乃是提供摘要。表格摘要永遠不會在視覺性瀏覽器中顯示出來，而是設計給螢幕朗讀軟體跟語音瀏覽器用的。它的功能正如其名：就是一份摘要、一份比標題還要長的描述。通常摘要會跟在標題之後被唸出來。

每個表格都該有個摘要。如果妳用了月曆，這個摘要就很有可能只是「帶有每一天文章鏈結的月曆」。如果妳拿表格來排版，就該在摘要的地方留空，指示說這個表格僅是拿來用於排版之用，而不是要用來呈現表格化的資料。（這就很像妳在祇用來的圖片裡的時候，把 ALT 屬性留空一樣。我們將在星期一討論這個「卡位圖片」。）

誰因此獲益？

1. Jackie 從中獲益了。因為當 JAWS 遇到妳的月曆時，Jackie 可以聽見「Summary: Monthly calendar with links to each day's posts.」。然後她還會聽見標題，接著她聽見了表格標頭，於是她得以導覽於月曆中。
2. iCab 的使用者從中獲益了。因為 iCab 可以用 Mac OS 內建的文字轉語音能力把網頁唸出來，而它也會把所有有被定義的表格摘要唸出來。

怎麼做：月曆

在 Movable Type 中，從妳的主索引模版裡尋找月曆的部分（同樣地，妳可以搜尋「calendarhead」就差不多能找到了）。妳會看到像這樣的 <table>：

```
<table border="0" cellspacing="4" cellpadding="0">
```

請改成這樣：

```
<table border="0" cellspacing="4" cellpadding="0" summary="帶有每  
一天文章鏈結的月曆">
```

在 Radio 中，程序就跟我們前幾天所做的很像。

1. 在 Radio 裡，開啟實際的 Radio 應用程式。如果是在 Windows 的話，從系統列裡在小 Radio 圖示上按滑鼠右鍵，然後選擇「Open Radio」。
2. 在「Tools」選單裡，選擇「Developers」，然後選「Jump...」（Control+J）。跳至「system.verbs.builtins.radio.weblog.drawCalendar」（不含引號）。

3. 現在到「Edit」選單裡，選「Find and Replace」，然後選「Find...」(Control+F) 並且尋找「draw the month and year」。這樣應該會展現出某列寫著「bundle // draw the month and year」的東西，並且設成高亮度。在三角形上雙擊以顯示出函式碼，它們看起來應該像這樣：

```
add ("<table cellpadding=\"0\" border=\"0\" class=
    \"hCalendarTable\">"); indentLevel++
```

4. 請改成這樣：

```
add ("<table summary=\"Monthly calendar with links to each
day's posts\" cellpadding=\"0\" border=\"0\" class=
    \"hCalendarTable\">"); indentLevel++
```

怎麼做：排版表格

如果你把表格用於排版，請在每個表格裡加入 `summary=""`。這可以用搜尋和取代輕易完成。請搜尋這個：

```
<table
```

然後取代成這樣：

```
<table summary=""
```

Dive Into Accessibility

在 30 天內打造更具親和力的網站

第二十天：忽略卡位圖片

許多設計師喜歡用透明的卡位圖片來控制網頁在視覺性瀏覽器上的排版效果。妳當然也可以隨意使用，但是同時也該在每一個卡位圖片上明確地指定空的 alt 屬性，這樣子那些非視覺性瀏覽器纔知道要忽略掉這些東西。

誰因此獲益？

1. Marcus 從中獲益了。因為在預設的情況下，Lynx 會在任何圖片未包含有 alt 屬性時，顯示出這些圖片的檔名。許多有名的網誌模版甚至在站台名稱之前就會先用了許多的卡位圖片了。妳在視覺性瀏覽器上當然不會注意到他們，但是 Marcus 看到的卻可能像這樣：

```
[shim.gif] [shim.gif]
[shim.gif]
[shim.gif]
Welcome to My Web Site
[ciblueHeader2.gif]

[ciblueCurve2.gif]
```

2. Jackie 從中獲益了。因為在預設的情況下，JAWS 會在任何圖片未包含有 alt 屬性時，念出這些圖片的檔名。如果妳覺得 Marcus 那樣子就算被干擾到了，試想一下 Jackie 聽到的是些多麼繁雜的東西：

```
graphic shim dot gif graphic shim dot gif graphic shim dot gif graphic shim dot gif
welcome to my web site graphic c i blue header two dot gif graphic c i blue curve
two dot gif
```

如果妳在現實世界裡也用這種方法介紹妳自己的話，就再也沒有人會理妳了。

怎麼做

Radio 的使用者今天可以放假一天；因為上星期一的課程裡，Radio 已經會在所有的卡位圖片裡自動產生空的 alt 屬性了。（真是多謝 Jake 了）。如果妳檢視主頁源碼時，沒有看到任何卡位圖片裡有 alt="" 的話，請更新 Radio.root 並重建妳的站台。

其他出版工具的使用者應該在妳們的模版裡，尋找後面跟著像是「spacer.gif」、「shim.

gif」、「1.gif」之類的圖檔檔名、或其他在模版裡出現了很多次且每次 width 和 height 屬性都不同的圖檔檔名的 標籤。

舉例來說，每一個卡位圖片看起來都可能像這樣：

```

```

請改成這樣：

```

```

如果你重複使用了相同的卡位圖檔好幾次，可能用全域搜尋和取代會是最簡單的。

不可以做的事

1. 不可以定義成 alt=" "。alt 屬性應該是空字串，而不是一個空白字元。
2. 不可以你的 <body> 標籤裡指定 alt 屬性，就算你在那裡定義了背景圖片也一樣。對於所有的非視覺性瀏覽器來說，這個背景圖片本來就都會被忽略掉。這種標籤看起來就像：

```
<body background="http://網址/指到/image.gif">
```

3. 不可以你的 <td> 標籤裡指定 alt 屬性，就算你在那裡定義了背景圖片也一樣。對於所有的非視覺性瀏覽器來說，這個背景圖片本來就都會被忽略掉。這種標籤看起來就像：

```
<td background="http://網址/指到/image.gif">
```

延伸閱讀

- [WebAIM: How to Create Accessible Graphics.](#)

Dive Into Accessibility

在 30 天內打造更具親和力的網站

第二十二天：使用真正的清單（或以適當的方法偽造）

假設妳有這三個鏈結的網誌目錄：Slashdot、The Register 和 dive into mark；可是妳不想要用醜陋的黑色項目符號來條列，而想要在每個鏈結前加上華麗無比的圖片。這時候妳能怎麼辦？最常見的解決方法是在每個鏈結前用上 `` 標籤。這不但可行，而且還可以在每個圖片加上適當的 `alt` 來增加親和力。

然而妳還可以更進一步，使用真正的清單標記（`` 和 `` 標籤），再用 CSS 把醜陋的黑色項目符號換成圖片。無論妳在不在乎這種學術熱誠，這個技術不但是「正確的方法」，還能夠帶來額外的親和力。

我將在一分鐘內展示這兩種技術。

誰因此獲益？

1. Marcus 從中獲益了。因為就跟我們昨天所看到的一樣，Lynx 會在任何圖片未包含有 `alt` 屬性時，顯示出這些圖片的檔名。
2. Michael 從中獲益了。因為 Links 從不顯示圖片；但是在預設的情況下，當任何圖片未包含有 `alt` 屬性時，它也不會顯示任何資訊。昨天當我們想要忽略卡位圖片時，倒可以接受這個結果，但今天我們還希望有些圖片，能夠用來指出那是一份清單，所以我們得用上 `alt` 文字。

同樣地，當 Michael 把瀏覽器的圖形選項關閉後，Opera 會在原本出現「缺少圖形」的區塊顯示 `alt` 文字。如果妳用了進階技術的話，Opera 還可以有更傑出的表現：它會恢復顯示那些醜陋的黑色項目符號，而不是顯示「缺少圖形」的區塊。

3. Jackie 從中受益了。因為就跟昨天我們所看到的一樣，JAWS 會在任何圖片未包含有 `alt` 屬性時，念出這些圖片的檔名；最後鏈結就會失落於不規則檔名汪洋之中。Jackie 可能聽到像這樣的東西：

```
fancy dot dot gif link slashdot, fancy dot dot gif link the register, fancy dot dot gif
link dive into mark
```

在 `alt` 文字裡加上星號能有莫大的幫助。JAWS 會知道這個圖片是用來當作清單前面的符號用的，於是就不唸出來。然而 Home Page Reader 還是會明確地把星號唸出來，所以使用者就會聽到像這樣的東西：

```
asterisk link slashdot, asterisk link the register, asterisk link dive into mark
```

使用真正的清單標記纔是最好的方法。因為所有的視覺呈現都會在 CSS 描述裡，完全不會打斷你的頁面，同時 JAWS 和 Home Page Reader 也都會唸出你的清單當作清單來念。現在聽起來就會像這樣：

link slashdot, link the register, link dive into mark

怎麼做

如果你有個網誌目錄，看起來像這樣：

```
 <a href="http://www.slashdot.org/">Slashdot</a> <br>
 <a href="http://www.theregister.co.uk/">The Register</a> <br>
 <a href="http://diveintomark.org/">dive into mark</a> <br>
```

你應當在清單項目圖片裡提供 alt 屬性。在 alt 文字裡用上星號，模擬成用真的清單標記時所會看到的樣子。（為了要避免在視覺性瀏覽器上跑出工具提示，你也應該提供空的 title 屬性）。

```
 <a href="http://www.slashdot.org/">Slashdot</a> <br>
 <a href="http://www.theregister.co.uk/">The Register</a> <br>
 <a href="http://diveintomark.org/">dive into mark</a> <br>
```

怎麼做：進階版

進階（也比較好）的技術是使用 CSS 來定義你的清單項目圖片。

```
<style type="text/css">
ul.blogroll {
  list-style: url(/images/fancydot.gif) disc;
}
</style>
```

然後在你的模版裡，就可以用真的清單標記來列出清單：

```
<ul class="blogroll">
<li><a href="http://www.slashdot.org/">Slashdot</a></li>
```

```
<li><a href="http://www.theregister.co.uk/">The Register</a></li>
<li><a href="http://diveintomark.org/">dive into mark</a></li>
</ul>
```

結果：

- Modern browsers will display the image as the list bullet.
- Browsers with images turned off will display the boring black bullet.
- Netscape 4 will always display the boring black bullet.
- Text-only browsers always ignore CSS, so they will display the list however they normally display lists (usually rendering the list bullet as an asterisk).

附註：不加項目符號的清單

要建立前面不帶圖片的鏈結清單，還有一個常用的方法，就是把鏈結堆積起來，可能再設定一個向右對齊，就像這樣：

```
<div align="right">
<a href="http://www.slashdot.org/">Slashdot</a><br>
<a href="http://www.theregister.co.uk/">The Register</a><br>
<a href="http://diveintomark.org/">dive into mark</a><br>
</div>
```

這也可以用 CSS 和真的清單標記來辦到：

```
<style type="text/css">
ul.blogroll {
  list-style: none;
  text-align: right;
}
</style>
```

或者，如果你想讓鏈結靠左對齊的話，可以改成這樣：

```
<style type="text/css">
ul.blogroll {
  list-style: none;
  margin-left: 0;
  padding-left: 0;
}
</style>
```

無論如何，清單標記都不會跟前一個例子裡有所不同：

```
<ul class="blogroll">
<li><a href="http://www.slashdot.org/">Slashdot</a></li>
<li><a href="http://www.theregister.co.uk/">The Register</a></li>
<li><a href="http://diveintomark.org/">dive into mark</a></li>
</ul>
```

「list-style: none」這一系列會讓視覺性瀏覽器裡不要出現黑色項目符號。對所有的瀏覽器，甚至是 Netscape 4 來說，這都會有效。在此感謝 Tobias Schmidt 提醒我可以用這個訣竅。

延伸閱讀

- Tobias Schmidt: Styling lists with CSS.
- W3Schools: CSS List Properties.
- Eric Meyer: Lists and Indentation.
- Eric B. Bednarz: Manipulating Margin and Padding of Lists With CSS.

Dive Into Accessibility

在 30 天內打造更具親和力的網站

第二十三天：提供能取代圖片的文字

這是整個系列中最重要的一天，所以把照子放亮點吧：

妳站台上每個頁面裡的每一張圖片，都該提供足以取代的文字部分，也就是所謂的「alt 文字」；這是在 `` 標籤裡用 `alt` 屬性所加以指定的。

螢幕朗讀軟體可以讀得到這段文字，純文字瀏覽器會把這段文字顯示出來，Google 會為這段文字建立索引，視覺性瀏覽器則會用工具提示或者在狀態列顯示出來（雖然身為設計師的妳，還是可以覆寫掉這個作用）。我們已經看到要如何指定卡位圖片的空 alt 文字，還有許多用項目圖片建立親和性高的清單的方法。還有漏掉了些甚麼的嗎？

- 靜態鏈結圖示
- 「架站系統」圖示
- Mail-to 圖示
- XML 圖示
- 浮在妳的文章項目內的小圖形
- 任何其他妳加進樣本裡的圖片

這些都需要適當的 alt 文字。

誰因此獲益？

1. Jackie 從中獲益了。因為 JAWS 會唸出 alt 文字。如果沒有有效的 alt 的話，Jackie 就會聽到檔名，而這聽起來真的很可怕。
2. Marcus 從中獲益了。因為 Lynx 這個純文字瀏覽器並不會顯示任何文字，而祇會顯示 alt 文字。如果沒有 alt 文字的話，Lynx 就會顯示檔名，這看起來就跟 JAWS 聽起來一樣糟。
3. Michael 從中獲益了。因為 Links 這個純文字瀏覽器並不會顯示任何文字，而祇會顯示 alt 文字。如果沒有 alt 文字的話，Links 就不會顯示任何跟這個圖片有關的資訊（除非這個圖片也是個鏈結，在這種情況下 Links 就祇會顯示出「[IMG]」）。當使用 Opera 瀏覽而把圖形選項關閉時，Michael 會在圖片的位置看到 alt 文字。
4. Lillian 從中獲益了。因為 Internet Explorer 會用工具提示的樣子顯示 alt 文字（當然身為設計師的妳，還是可以讓這件事不要發生）。
5. Google 從中獲益了。因為 Googlebot 會對 alt 文字建立索引；這不僅對一般的搜尋有效，更會用於圖片搜尋功能（不然妳以為那是怎麼辦到的？）

怎麼做

預設的 Movable Type 模版不會包含任何的 `` 標籤；如果你用了「Powered by Movable Type」圖形的話，你就該確定 `` 標籤裡同時也包含了 `alt="Powered by Movable Type"` 屬性。

預設的 Greymatter 模版祇包含了一個圖片，這是由 `{{gmicon}}` 模版變數所產生的。實際上會產生一個 `` 標籤，同時還會包含有合適的 `alt` 文字：「Powered by Greymatter」。

Radio 會為下列的圖示自動產生合適的 `alt` 文字：

- XML 咖啡杯： `alt="Subscribe to <site name> in Radio UserLand."`
- XML 圖示： `alt="Click to see the XML version of this web page."`
- Mailto 圖示： `alt="Click here to send an email to the editor of this weblog."`

然而 Radio 的使用者還得需要手動為自訂圖片指定 `alt` 文字。請到 `Prefs`，然後按下 `Customized Images`（在 `Templates` 裡），並新增這些 `alt` 屬性：

- Day-level permalink: `alt="Permanent link: <%longDate%>".`
- Item-level permalink: `alt="Permanent link".`
- Source link icon: `alt="source".`
- Enclosure link icon: `alt="enclosure".`

你同時也該增加一個 `title=""`，讓視覺性瀏覽器不要顯示工具提示。

當然，不論你用了甚麼出版工具，如果你在模版裡增加了自己的圖片，或者文章項目裡會浮著小圖形的話，他們都該有個合適的 `alt` 文字。因為我從範例裡學得最好，所以以下就是一些範例。更多泛用的原則和範例會被列在「延伸閱讀」小節裡。

alt 文字的不良示範

- 對於任何 HTML 標記來說，`alt` 文字都祇能用純文字，不能有任何標籤。
- `alt="filename.jpg"`. 這不能提供我們任何資訊。圖片的功能是甚麼？我們並不真的在乎它叫甚麼名稱。
- `alt="alt text"`. 這是由某些 HTML 編輯器所插入、用來提醒的文字，顯然是被某些沒概念的設計師遺留下來了。
- `alt="Click here!"`. 沒有甚麼太有用的功能（除非圖形上面寫的也就是「Click here!」）。
- `alt="Turn images on!"`. 這就跟去問盲人時鐘上的時間，然後對她說「張開你的眼睛！」圖片可能因為某個原因而被關掉（像是 Michael 的頻寬不夠）、無法使用（像是 Marcus 的純文字瀏覽器）、或者連讓你決定要不要關掉的餘地都沒有（像是 Jackie 的螢幕朗讀軟體就祇會把 `alt` 文字大聲唸出來）。
- 更多 `alt` 文字的不良示範。

alt 文字的良好示範

- Jonathon Delacour 在他的站台標幟上有一個中文字的圖形。 alt="Site logo: xin, the Chinese character for heart".
- Leslie Harpold 在頁面標幟用了包含站台名稱「 The Historical Present 」和標語「 Hypermodernism has a posse 」的圖形。 alt="the historical present: hypermodernism has a posse".
- Simon Willison 用了一個「 W3C XHTML 1.0 」貼紙。 alt="Valid XHTML 1.0!"
- Jeffrey Zeldman 用了像文字的圖形來當導覽列，拉下來的時候，每一個圖形都會用 Javascript 在狀態列放上一小列標語。當然視盲的讀者會錯過這些東西，所以 Zeldman 也在每一個圖片上用了 alt 文字放進相同的標語。真邪惡。
- Dean Allen 在他的頁面標幟上用了一個包含站台名稱及標語的圖形。他的 alt 文字也一樣長，但卻包含了略微不同的標語（當然會造成一些混淆），不過 Dean 確實酷到可以僥倖成功的地步。 alt="Textism is an ephemeris focused on the composition, design, and reading of text. In addition, there will be pie". 請注意：妳很可能沒有那麼酷。請還是保持簡單為妙。

延伸閱讀

- A. J. Flavell: ALT texts in IMG.
- Jukka Korpela: Simple guidelines on using ALT texts in IMG elements.
- Ian Hickson: Mini-FAQ about the alternate text of images.
- Watchfire.com: Provide alternative text for all images.
- All My FAQs Wiki: ALT attribute.
- WebAIM: How to Create Accessible Graphics.
- Martin Schrode: On accessible advertising.
- Section 508 Federal Accessibility Guidelines: What is meant by a text equivalent?

Dive Into Accessibility

在 30 天內打造更具親和力的網站

第二十四天：提供能取代影像地圖的文字

我對於居然有這麼多知名站台都用了用戶端影像地圖而大感震驚。我自己根本不會用這種東西，而且我不認為會有任何預設的網誌模版使用這個訣竅，但是真的有不少人提出來了。如果妳不知道影像地圖是甚麼，Leslie Harpold 在她的主頁面底下就用了一個來當作導覽鏈結。它們全是圖片，但是在「archives」這個字上面按一下，就會把妳帶到某一頁；按下「by category」就會連到另一頁，以此類推。

影像地圖聽起來像是親和力的夢魘，不過他們並不是。就跟每一個圖片都需要足以取代的文字一樣，每一個影像地圖裡每一個可以按下的區域也都需要一個足以取代的文字。妳可以提供 alt 文字給圖片本身（在 標籤裡），並且也提供給影像地圖（在跟 <map> 標籤相關聯的 <area> 標籤裡，這些標籤是用來定義哪些區域可以按、以及會連結到哪裡去的）。

誰因此獲益？

1. Marcus 從中獲益了。因為 Lynx 會把圖片的 alt 文字顯示成鏈結。所以當 Marcus 按下 ENTER 後，Lynx 就會另外用一個單獨的頁面顯示影像地圖裡的鏈結清單。每一個鏈結都會標上影像地圖裡 area 的 alt 文字。如果沒有 alt 文字的話，Lynx 就會顯示每一個 area 的鏈結網址，不過這顯然會變得很難以理解。

如果 Leslie 在她的影像地圖上沒有 alt 文字的話，Marcus 在她的主頁底下所看到的鏈結就會像這樣：

```
[USEMAP:hpfooter.gif]
```

從這裡連結出去，Marcus 會看到影像地圖上所有鏈結的清單。因為沒有 alt 文字，所以 Lynx 就祇能顯示每一個網址，他們看起來就會像這樣：

```
[USEMAP:hpfooter.gif]
```

```
MAP: http://leslie.harpold.com/#Map
```

1. http://leslie.harpold.com/archives.html
2. http://leslie.harpold.com/category/
3. http://leslie.harpold.com/links.html
4. http://leslie.harpold.com/leslie.html
5. http://www.moveabletype.org

然而實際上，Leslie 確實在影像地圖跟每一個區域上用了合適的 alt 文字。所以 Marcus 真正會在她的主頁底下看到的鏈結其實像這樣：

```
Site navigation links
```

從這裡連結出去會讓 Marcus 看到像是這樣的頁面：

```
Site navigation links
```

```
MAP: http://leslie.harpold.com/#Map
```

1. previously...
2. by category
3. about the site
4. about leslie
5. Powered by Movable Type

2. Michael 從中獲益了。因為 Links 會把 alt 文字顯示成鏈結的樣子。當 Michael 按下了 ENTER 後，Links 就會彈出一個所有定義在地圖上的鏈結的選單。每個鏈結都會標記著 area 上的 alt 文字。如果沒有這個 alt 文字的話，Links 就會顯示出每個 area 的鏈結，而這看起來將會相當難以理解。
3. Jackie 從中獲益了。因為 JAWS 會把影像地圖中每個 area 的 alt 文字按照你在 HTML 源碼裡定義出來的順序顯示出來。Jackie 可以按下 ENTER 來連結。如果沒有 alt 文字的話 JAWS 就會把每個 area 裡的鏈結位址唸出來。這真的完全的讓人無法理解。（妳有沒有試過在電話裡念超級長的網址？）
4. Lillian 從中獲益了。因為 Internet Explorer 會在妳的游標滑過每個影像地圖裡的 area 時，用工具提示顯示出 alt 文字。
5. Google 從中獲益了。因為 Googlebot 會對主要影像以及影像地圖裡每一個 area 的 alt 文字建立索引。同時 alt 文字也是用來判斷妳的頁面是否含有某個關鍵字的重要因子；同時每個鏈結是否跟某個關鍵字有關，也是用 area 裡的 alt 文字來判斷的。

怎麼做

如果妳用了像是這樣的影像地圖：

```
  
  
<map name="Map">  
<area shape="rect" coords="203,114,258,129" href="/archives.html">  
<area shape="rect" coords="277,113,348,129" href="/category/">  
<area shape="rect" coords="364,113,401,128" href="links.html">
```

```
<area shape="rect" coords="418,114,488,130" href="leslie.html">
<area shape="rect" coords="-4,190,131,210" href="http://www.
moveabletype.org">
</map>
```

請在主影像跟每一個影像地圖裡被連結的 area 加入 alt 文字，像這樣：

```


<map name="Map">
<area alt="previously..." shape="rect" coords="203,114,258,129"
href="/archives.html">
<area alt="by category" shape="rect" coords="277,113,348,129"
href="/category/">
<area alt="about the site" shape="rect" coords="364,113,401,128"
href="links.html">
<area alt="about leslie" shape="rect" coords="418,114,488,130"
href="leslie.html">
<area alt="Powered by Movable Type" shape="rect" coords="-
4,190,131,210" href="http://www.moveabletype.org">
</map>
```

所有為圖片寫出好的 alt 文字的規則也一樣適用於影像地圖。妳也應該在主要的 標籤和每個 <area> 標籤裡加入 title=""，讓視覺性瀏覽器不要出現工具提示。

不可以做的事

不可以建立伺服器端的影像地圖、把影像被點擊的座標資料傳回伺服器做後續處理。這對於 Jackie 這樣子的 JAWS 使用者、或對於使用純文字瀏覽器的 Michael 和 Marcus、或對於 Bill 這種祇能用鍵盤的使用者、甚至是對於像 Google 這樣的搜索引擎來說，都完全沒有親和力可言。如果妳一定得用伺服器端的影像地圖，那妳也該用真的文字鏈結來新增一個純文字導覽列，讓使用者能夠連結到每一個妳放在影像地圖裡的鏈結。

延伸閱讀

- Leslie Harpold: The Historical Present. Leslie 很好心地讓我用她的網誌來當作今日範例的基礎。

Dive Into Accessibility

在 30 天內打造更具親和力的網站

第二十五天：使用真正的水平分割線（或以適當的方法偽造）

假設妳想要從文章中間一分为二，但是典型的水平分隔線（也就是 `<hr>` 標籤）很無趣，所以妳改用了另外一個圖片。這當然行得通，而且還可以輕易地加上 `alt` 文字來增加親和力。

然而妳還可以更進一步，使用真實的水平分隔線，然後再用 CSS 手法，使其在主流瀏覽器裡看起來跟圖片一樣。比較舊的瀏覽器或者是純文字瀏覽器則會忽略掉 CSS 的部分，然後用原來的樣子畫出水平分割線（純文字瀏覽器通常會用一系列跟螢幕等寬的底線或橫線來表示）。

我馬上就會展示這兩種技術。

誰因此獲益？

1. Jackie 從中獲益了。正如我們之前所看到的，JAWS 會在圖片沒有 `alt` 屬性時，把檔名唸出來。
2. Marcus 從中獲益了。因為 Lynx 會在圖片沒有 `alt` 屬性時，顯示出檔名。如果妳用了真的水平分割線，那麼 Lynx 就會畫出一條跟妳的螢幕等寬的底線。
3. Michael 從中獲益了。因為 Links 在圖片沒有 `alt` 屬性時，甚麼也不會顯示出來；如此一來 Michael 將無法得知這裡有個分割線。我們至少需要 `alt` 文字，最好是直接使用真的 `<hr>` 標籤，這麼一來 Links 就會畫出一條跟妳的螢幕等寬的橫線。

怎麼做

如果妳把一張圖片當作是水平分割線來用的話，增加親和力的最簡單方法就是在妳的 `` 標籤裡增加 `alt` 屬性。同時妳也應該再加一個空的 `title` 屬性，這樣子在視覺性瀏覽器裡纔不會出現工具提示。所以如果本來像這樣：

```

```

請改成這樣：

```

```

請不要發瘋地在 `alt` 文字裡填上八十個橫線。兩三個就夠了。

怎麼做：進階版

進階（而且也比較好）的技術就是用 `<hr>` 標籤。雖然在最理想的情況下，瀏覽器很神奇地能夠支援直接定義樣式的 `<hr>` 標籤，我們在此仍舊沿用笨笨的 `<div>` 標籤來顯示影像。把下列的 CSS 放在妳模版頂端的 `<style>` 區段裡（如果妳用了像是 `style-sites.css` 的外部樣式表的話，就在那個檔案裡面隨便找個地方放；如果妳用了多重樣式表的話，請放在 Netscape 4 適用的那一個裡）。

```
div.hr {display: none}
/*/*/a{}
div.hr {
  display: block;
  height: 25px;
  background-image: url(/images/fancyrule.gif);
  background-repeat: no-repeat;
  background-position: center center;
  margin: 1em 0 1em 0;
}
hr {display:none}
/* */
```

（在 `height` 裡的是妳的圖片高度；在 `background-image` 裡的是妳的圖片位址。）

然後當妳想要在模版中插入這種華麗的分隔線時，就這樣做：

```
<div class="hr"></div><hr />
```

結果：

- 所有的主流瀏覽器現在都不會顯示普通的水平分隔線，而是顯示出妳的圖片。
- Netscape 4 祇會顯示出普通的水平分隔線。
- 純文字瀏覽器會忽略掉 CSS，所以他們也祇會顯示出普通的水平分隔線（通常是畫成一堆底線或斜線）。

延伸閱讀

- Hiding CSS From Netscape 4 而不使用額外的樣式表。這也就是我們在前面的進階範例裡所用到的技術。
- CSS1 and the Decorative HR. 如果妳格外勇猛而想要在 Netscape 4 的 CSS 裡使用華麗的水平分隔線的話，這裡有人幸運地辦到了。

Dive Into Accessibility

在 30 天內打造更具親和力的網站

第二十六天：使用相對字型尺寸

網站的重點就是文字，幾乎沒什麼例外。新聞、輿論、想法、理念、故事、創新的文學作品、電子商務：全都是文字。視覺設計跟圖像當然很重要，但是如果人們連妳的文字都讀不到，那有什麼用呢？

2000 年秋天時，Jeffrey Zeldman 發表了一篇著名的文章，指出使用相對字型尺寸是不可能的，這是因為瀏覽器都有一大堆的錯誤，包括 Netscape 4 以及之後的眾多瀏覽器也是如此。雖然 Netscape 4 仍然沒有修正錯誤，而且也繼續在被使用，但是至少我們都從中學到一些實作相對字型尺寸的方法（Zeldman 也是。他最近重新設計的 Web Standards Project 網站也使用下面描述的技術）。

我們在能支援的瀏覽器裡使用相對字型尺寸，而在不支援的 Netscape 4 裡使用絕對字型尺寸。即使不使用多個樣式表也能完成這項任務。我馬上就會提供一個可以直接複製/貼上的解決方案，讓妳套用在 Movable Type 預設模版跟 Radio 的所有預設主題上。接著也會對技術本身作些囉唆的解釋，好讓妳能夠在其他模版上採行。

誰因此獲益？

1. Lillian 從中獲益了。因為 Lillian 上了年紀，眼睛沒辦法看清楚網頁。就像其他 80% 的網際網路人口一樣，她使用的是 Windows 上的 Internet Explorer。除非網站設計師明確地指定相對字型尺寸，否則這將沒辦法支援相對字型。Lillian 雖然有修改過瀏覽器的預設字型尺寸（在「檢視」選單下的「字型」選項裡），但它在使用絕對字型尺寸的網站上表現並不好。這些網站包括所有現有的網誌模版。例如對 Lillian 來說，Movable Type 的預設模版看起來就像這樣：

TEST POST

Lorem ipsum dolor sit amet, consectetur adipis
minim veniam, quis nostrud exerci tation ullam
vulputate velit esse molestie consequat, vel illum
delenit augue duis dolore te feugait nulla facilis

Ut wisi enim ad minim veniam, quis nostrud ex
hendrerit in vulputate velit esse molestie conse
luptatum zril delenit augue duis dolore te feuga
ut laoreet dolore magna aliquam erat volutpat.

Duis autem vel eum irure dolor in hendrerit in
dignissim qui blandit praesent luptatum zzril de
nonummy nibh euismod tincidunt ut laoreet de
lobortis nisl ut aliquip ex ea commodo consequat

Posted by f8dy at June 26, 2002 09:48 AM

假使這個模版使用的是相對字型尺寸，對大部分不需要修改字型尺寸的人來說，看起來是完全相同的。但是在 Lillian 眼中會是這樣：

TEST POST

Lorem ipsum dolor sit amet, consa
dolore magna aliquam erat volutpa
lobortis nisl ut aliquip ex ea comm
molestie consequat, vel illum dolo
praesent luptatum zzril delenit aug

Ut wisi enim ad minim veniam, qui
consequat. Duis autem vel eum iri
feugiat nulla facilisis at vero eros et
duis dolore te feugait nulla facilisi.
euismod tincidunt ut laoreet dolor

我再強調一次：如果人們連妳的文字都讀不到，那有什麼用呢？

怎麼做：**Radio**

找出妳 Home Page Template 裡的 <style> 部分，最頂端的 CSS 規則看起來應該像這樣：

```
body, td, th, p {
    font-family: verdana, sans-serif;
    font-size: 12px;
}
```

保持原狀，但是在後面接上這些東西：

```
/**/a{}
body,
body td,
body th,
body p {
    font-size: x-small;
    voice-family: "\"}\"";
    voice-family: inherit;
    font-size: small;
}
html>body,
html>body td,
html>body th
html>body p {
    font-size: small;
}
/* */
```

請確定妳有在開頭跟結尾的地方都加上那些註解，它們是一切的關鍵。後面將會加以說明。

怎麼做：**Movable Type**

Movable Type 的預設模版比 Radio 要複雜，不過我們做的事情還是一樣，只不過這次份量又多了點。請在樣式表模版 (styles-site.css) 結尾處加上這些東西：

```
/**/a{}
body,
body a,
body .calendar,
body .calendarhead,
body .title,
body .sidetitle,
body .syndicate,
body .powered,
```

```
body .comments-post,  
body .posted {  
    font-size: xx-small;  
    voice-family: "\"}\"\"";  
    voice-family: inherit;  
    font-size: x-small;  
}  
html>body,  
html>body a,  
html>body .calendar,  
html>body .calendarhead,  
html>body .title,  
html>body .sidetitle,  
html>body .syndicate,  
html>body .powered,  
html>body .comments-post,  
html>body .posted {  
    font-size: x-small;  
}
```

```
body .date {  
    font-size: x-small;  
    voice-family: "\"}\"\"";  
    voice-family: inherit;  
    font-size: small;  
}
```

```
html>body .date {  
    font-size: small;  
}
```

```
body #banner {  
    font-size: 200%;  
}
```

```
body .description {  
    font-size: 60%;  
}
```

```
body .blogbody {  
    font-size: 110%;  
}
```

```
body .blogbody,  
body .calendar,
```

```
body .calendarhead,  
body .side,  
body .title,  
body .sidetitle,  
body .syndicate,  
body .powered,  
body .comments-body {  
    line-height: 128%;  
}  
/* */
```

也請確定妳有在開頭跟結尾的地方都加上那些註解了。

怎麼做：深入探討

基本概念是使用字型尺寸關鍵字。它們（因為舊版瀏覽器的錯誤而）很少被用到，但其實它們有三個有趣的特性：

1. 它們不會加成 (compound)。如果妳在「main」段落裡使用了 90% 的字型尺寸，又在其中的「post」段落使用 90% 的字型尺寸，有些瀏覽器會用 81% (90% x 90%) 來顯示「post」，但有些會用 90% 來顯示。隨著一層層的巢狀套用（在使用表格製作版面時很常見），文字馬上就會隨著百分比的加成而縮小到看不見。然而如果妳的「main」段落使用了 small 的字型尺寸，又在其中的「post」段落使用 small 的字型尺寸時，所有的瀏覽器都會用 small 的字型尺寸來顯示「post」。
2. 它們改變字型尺寸的特性可以在 Windows 的 Internet Explorer 下正確顯示。這點蠻神奇的，因為 small 聽起來就像是個絕對字型尺寸（尤其是加上前面關於巢狀的說明後），但是我發誓它真的可以運作。
3. 它們永遠不會太小。Lillian 在 Internet Explorer 裡使用的「字型」設定可以自由修改尺寸。許多視力正常的人比較希望看到比正常字型來得小一點的字。使用百分比設定的字型尺寸當與預設的最小字型尺寸合成之後，就會變得模糊不清。但使用字型尺寸關鍵字總是可以至少維持在 9px，這在任何字體下都可以閱讀（假設妳的視力不錯）。

所以我們使用字型尺寸關鍵字來設定我們的基礎大小。如果需要控制得更加精細的話，我們還可以使用百分比，但只能用在末端的文字上（像是可以用在「post」，但不能用在「main」），以避免百分比的加成及過小的問題。

以下是使用字型尺寸關鍵字的基本概念：

```
p {  
    font-size: 12px;  
}  
  
/* */a{}
```

```

body p {
  font-size: x-small;
  voice-family: "\"}\"";
  voice-family: inherit;
  font-size: small;
}
html>body p {
  font-size: small;
}
/* */

```

接下來的都很重要，所以請注意。

1. 首先，我們定義 <p> 的絕對尺寸 (12px)。包括 Netscape 4 在內，所有的瀏覽器都會接受這個樣式設定。
2. 接著我們加入「`/*/*`」這個古怪的註解。根據 Netscape 4 的錯誤，所有在這個註解之後的東西都會被忽略掉。如此一來，只有 Netscape 4 以外的瀏覽器才會看到之後的樣式。
3. 緊接在那個古怪的註解之後，我們又加入一個空白的樣式「`a {}`」。Mac 上充滿錯誤的 Opera 5 會忽略這個樣式（只有這一個），而接受其他所有的。
4. 現在我們已經切出一個範圍了，在這個範圍內所定義的規則只會套用在 Netscape 4 以外的其他瀏覽器上。接下來我們就可以開始定義相對字體尺寸（這也就是 Netscape 4 所無法處理的）了。首先我們要用「`body p`」選擇子來重新定義 p 標籤的作用。因為 CSS 可行的關係，這將會覆寫掉先前的 p 選擇子（用技術名詞來描述的話，就是「`body p`」選擇子比「`p`」選擇子還要更具特定性）。
5. 我們把所有 <p> 標籤的字體尺寸重新定義成 `x-small`。這個字體尺寸關鍵字在預設的情況下，會被 Windows 上的 Internet Explorer 5 轉譯成 12px。然而祇要使用者（從檢視選單）更改了「字型」，這些文字就可能依照使用者的設定而變得更大或更小。這也正是我們的目的（請注意：我們現在已經對 IE5/Win 定義了兩次字體尺寸；不過這並沒有關係，因為更特定的選擇子總是會勝出，而讓前一個選擇子被忽略掉）。
6. 很不幸地 IE5/Win 在字體尺寸關鍵字上有一刻度的錯誤：世界上所有其他瀏覽器（包括 IE5/Mac、Netscape 6、Mozilla、IE6/Win）都會把 `x-small` 轉譯成 10px，而非 12px。不過很幸運地，IE5/Win 另外還有一個分析錯誤之處，可以被我們拿來加以利用：當他看到那個看起來很奇怪的 `voice-family` 時，會誤以為整個「`body p`」選擇子已經結束了，然後就忽略掉「`}`」之後的那幾列。
7. 現在我們又切出了一個更小的範圍了，在這個範圍內所定義的規則只會套用在 IE5/Win 以外（至於 Netscape 4 早就被排除在外了）的瀏覽器上。如此一來我們就可以再把字體尺寸重新定義成 `small` 了。這麼一來，在非 IE5/Win（也就是會一路解讀到這一層的那些）瀏覽器上就會被正確地解讀成 12px 了（在預設的情況下）。當然，如果使用者把「字體尺寸」設成較大，這裡的文字也會變得比較大。這仍舊是我們所想要的功能。
8. 但是等一下！Opera 5 也有跟 IE5/Win 一樣的分析錯誤，所以有可能也會被 `voice-family` 黑克手法弄混，可是他又會正確地轉譯字體尺寸關鍵字，結果就是我們的文字在 Opera 5 底下將會變得太小。很幸運地，Opera 5 還支援第三種選擇子：「`html>body p`」（而且這個選擇子比「`body p`」還要「更具特定性」，所以會覆寫掉前一個選擇子的內容）。IE5/Win 並不支援這

種選擇子，所以會加以忽略，而這也正是我們所需要的（因為我們已經處理過那個一刻度的錯誤了，所以現在當然不會想要再把這個偏差放回去）。IE6/Win 同樣地也不支援這個選擇子，不過這並沒有甚麼關係，因為在「body p」選擇子裡，我們把「font-size: small」放在黑克用的「voice-family: inherit」之後。所有其他的瀏覽器都支援「html>body」選擇子；所以對他們來說，我們足足定義了四次字體尺寸。這當然沒有問題，因為祇有最具特定性的選擇子纔會生效，其他的都祇會被忽略掉。

9. 最後我們還得設定一個空註解：`/* */`。這樣纔會讓 Netscape 4 繼續分析之後的部分。任何在這個空註解之後定義的規則，也都會套用到所有的瀏覽器（包括 Netscape 4）上。

現在讓我們反過來看：

1. 無論使用者設定了甚麼，Netscape 4 都會把 `<p>` 文字顯示成 12px。
2. Windows 上的 Internet Explorer 5 會把 `<p>` 文字顯示成 `x-small`。依照預設值的話，也就是 12px；不過如果妳在檢視選單裡把「字型」設成較大的話，這裡的文字看起來也會比較大。
3. Windows 上的 Internet Explorer 6 會因為「body p」選擇子裡的「font-size: small」的關係，而把 `<p>` 文字顯示成 `small`。依照預設值的話，也就是 12px；不過如果妳在檢視選單裡把「字型」設成較大的話，這裡的文字看起來也會比較大。
4. Mac 上的 Internet Explorer 5，和所有平台上的 Opera、Netscape 6、Mozilla 以及（希望是啦）所有未來的瀏覽器都會因為「html>body p」選擇子裡的「font-size: small」規則，而把 `<p>` 文字顯示成 `small`。依照預設值的話，也就是 12px；不過如果妳用了「字體縮放」功能時，這裡的文字看起來也會比較大。

延伸閱讀

- Mark Pilgrim: Relative Font Sizing HOWTO. 跟這裡一樣提供了所有必要的解釋；但是那一頁本身就是這個技術的範例了，所以妳可以實際看出來到底會有甚麼作用。
- Todd Fahrner: Size Matters: Making Font Size Keywords Work.
- Caio Chassot: Hiding CSS from Netscape 4 但不使用多重樣式表。
- Tantek Çelik: Box Model Hack. How to hide CSS from Internet Explorer 5 for Windows.
- The Web Standards Project 同時也用了字體尺寸關鍵字來黑克 IE5/Win；祇不過他們使用了 Javascript 的解決方案（而不是用內嵌註解黑克法）來處理 Netscape 4 的問題。
- Owen Briggs: Text Sizing. 這裡有許多相對字體尺寸技術在各種不同瀏覽器、平台以及預設字體設定下的畫面擷取圖片。

Dive Into Accessibility

在 **30** 天內打造更具親和力的網站

第二十七天：使用真正的標頭

想像一下妳的網站輪廓：最頂層標出了網站的名稱；而在妳的網站上，列出了這幾天的文章，所以第二層標明了日期敘述：「Tuesday, July 16, 2002」之類的。因為每一天妳都寫了許多文章，所以每篇文章都該有自己的標題。因此第三層就會標明文章的標題。

現在照著剛剛的輪廓，用真的 `<h1>`、`<h2>` 和 `<h3>` 等標籤加以標記；螢幕朗讀軟體必須靠這些標籤來解讀妳的頁面結構。妳的網頁當然有結構，但是若沒有適當的標頭標籤，螢幕朗讀軟體就沒辦法找到他們了。我馬上就會展示給妳看，要如何用 CSS 讓妳的標頭看起來跟妳現在用 `` 怪物做出來的一樣。

誰因此獲益？

1. Jackie 從中獲益了。因為當 Jackie 按下某個頁面後，JAWS 就會唸出這個頁面裡鏈結跟標頭的數量。此時 Jackie 可以按下 `INSERT+F6` 來聽到頁面上所有的標頭；或是按下 `CTRL+INSERT+ENTER` 來跳到下一個標頭的地方。
2. Michael 從中獲益了。因為在 Opera 瀏覽器裡，他可以按下 `s` 跳到下一個標頭、或者用 `w` 跳到前一個。
3. Google 從中獲益了。因為 Google 欣賞結構良好的頁面，並給予標頭裡的關鍵字較高的評比（又是另一個撰寫好文章標題的原因）。

怎麼做：Movable Type

1. 定義站台標頭的樣式。在妳的樣式表模版 (`styles-sites.css`) 裡，加入以下這幾列：

```
h1, h2, h3 {
    margin: 0;
    padding: 0;
}

h1 {
    font-size: 20px;
    font-weight: normal;
}

/* */a{}
h1 {font-size: 100%}
/* */
```

2. 用 `<h1>` 標籤定義妳的站台標頭。在妳的四個主要模版（主索引、分類彙整、按時彙整、單篇彙整）裡，搜尋這些東西：

```
<div id="banner">
<$MTBlogName$><br />
```

然後把他們代換成這樣：

```
<div id="banner">
<h1><$MTBlogName$></h1>
```

3. 用 `<h2>` 標籤定義日期標頭。我們已經定義了一個用於此的樣式級別了，所以無須再修改樣式表，而祇需要修改標籤就好。在妳的四個主要模版中，搜尋這些東西：

```
<div class="date">
<$MTEnterDate format="%B %d, %Y"$>
</div>
```

然後把他們代換成這樣：

```
<h2 class="date">
<$MTEnterDate format="%B %d, %Y"$>
</h2>
```

4. 用 `<h3>` 標籤定義妳的文章標題。同樣地祇需要修改標籤就好，而不用修改樣式表。在妳的四個主要模版中，搜尋這些東西：

```
<span class="title"><$MTEnterTitle$></span>
```

然後把他們代換成這樣：

```
<h3 class="title"><$MTEnterTitle$></h3>
```

怎麼做：Radio

1. 定義妳的標頭樣式。預設的 Radio 佈景主題並沒有使用任何的真實標頭標籤，所以我們得先在樣式表裡自己定義出來（這其實得量身訂做纔行；不過這裡的範例應該可以讓妳的頁面在視覺性瀏覽器裡看起來跟以前一樣纔對）。

事實上在我們開始之前，請先在妳的 Home Page Template 裡搜尋「`h2 {`」。如果妳找到像這樣的規則的話，請把它移掉。這個規則實際上沒有在任何地方被用到，卻會妨礙到我們：

```
h2 {
font-family: Verdana, Arial, Helvetica, sans-serif; font-
size: 24px; font-weight: bold
}
```

2. 好，現在在妳的 Home Page Template 的 <style> 區段中任何位置，加入這些樣式：

```
h1, h2 {
margin: 0;
padding: 0;
}

h1 {font-size: 24px}
h2 {font-size: 13px}

/**/a{}
h1 {
font-size: large;
voice-family: "\"}\"";
voice-family: inherit;
font-size: x-large;
}
html>body h1 {
font-size: x-large;
}
h2 {
font-size: x-small;
voice-family: "\"}\"";
voice-family: inherit;
font-size: small;
}
html>body h2 {
font-size: small;
}
/* */
```

請注意我們在支援相對字型尺寸的瀏覽器裡使用相對字型，而在 Netscape 4 裡則使用絕對字型。這技術看起來應該很眼熟，因為我們昨天才剛做過一樣的事情。

3. 定義站台名稱的標頭。在妳的 Home Page Template 中搜尋「<%siteName%>」，並尋找像這樣的某一行：

```
<font size="+2"><b><a href="<%radio.macros.weblogUrl (%)%"
```

```
style="color:Black; text-decoration:none"><%siteName%></a></b></font>
```

然後把他們代換成這樣：

```
<h1><a href="<%radio.macros.weblogUrl (%)%" style="color:Black; text-decoration:none"><%siteName%></a></h1>
```

4. 定義日期標頭。在你的 Day Template 中搜尋「 <%longDate%> 」，並尋找像這樣的某一行：

```
<b><%longDate%></b>
```

然後把他們代換成這樣：

```
<h2><%longDate%></h2>
```

延伸閱讀

- Shirley Kaiser: Don't Fake Your Markup: CSS 的親和力議題。

Dive Into Accessibility

在 **30** 天內打造更具親和力的網站

第二十八天：加上元件描述

妳是否曾為網頁表單的難用所苦？舉例來說，標準的 GUI 應用程式裡，妳可以在核選框標籤上任何一個地方核選或取消核選；但是在網頁應用程式裡，妳祇能在核選框的框框裡纔能這麼做。這樣雖然煩人但還不至於無可挽回。但對於視盲的讀者來說，這個狀況可能更糟。即使是像迴響發表表單這樣子簡單的東西，一旦妳無法看到了，就可能非常難以使用（我們曾經注意過表格裡的相似問題：網誌月曆祇有在能夠一次被整個看到時纔會好用；如果一次祇能看到一天的話，就會變得相當困難）。

有一些 HTML 標籤可以讓表單比較容易使用。在此我所要談的是 `<label>` 標籤；至於其他的那些，妳可以在「延伸閱讀」裡找到。

`<label>` 標籤可以把任何表單輸入元件連結到表單稱號：文字盒、多列文字區域、核選框、圓形按鈕等等。這會讓螢幕朗讀軟體聰明地唸出輸入元件的稱號，讓使用者知道那到底是甚麼。除此之外，如果妳把核選框 (`<input type="checkbox">`) 加上了 `<label>` 標籤，寫著其後所出現的文字的話，妳的網頁表單也會跟 GUI 應用程式一樣：在文字稱號讓任何地方點選，都可以對核選框起作用。

誰因此獲益？

1. Jackie 從中獲益了。因為當 Jackie 在表單間跳躍時，JAWS 會把每一個元件的名稱（從他們的 name）唸出來；但這樣可能並不明智。不過如果這些表單元件也有稱號的話，JAWS 就會改成去念這些稱號文字。「Text: name.」(TAB)「Text: email address.」(TAB)「Text: URL.」(TAB)「Text area: comments.」之類的。
2. Lillian 從中獲益了。因為一旦元件被貼上了稱號，Lillian 就能按下核選框後的文字，還能生效。這麼一來，就能夠大幅提高用把滑鼠游標對到核選框的容錯範圍；而以她有限的視力來說，這個範圍顯然是越大越好。
3. Bill 應該也要從中獲益了，可惜還沒有。他主要是透過鍵盤，也就是 TAB 鍵來導覽。當他跳到表單裡的核選框時，Mozilla 應該要把焦點移到整個稱號外的矩形外框，可惜並沒有這樣，焦點祇移到核選框本身的矩形外框而已（然而 Internet Explorer 的結果是對的，甚至 Netscape 4 也能弄對。Mozilla 這回遜掉了）。

怎麼做：Movable Type

在 Movable Type 裡，編輯妳的迴響清單模版。在接近底下的地方，妳會看到有個表單包含了像是這樣的元件：

```
名字:<br />
<input name="author" /><br /><br />
```

```
Email 地址:<br />
<input name="email" /><br /><br />

網址:<br />
<input name="url" /><br /><br />

迴響:<br />
<textarea name="text" rows="10" cols="50"></textarea><br /><br />

<input type="checkbox" name="bakecookie" />記住我的資訊?<br /><br />
>
```

每一個裸露的稱號都應該要被裝進 `<label>` 標籤裡。同時，因為 `<label>` 標籤會用 ID（而不是 name）指到表單元件去，所以每個 `<input>` 標籤也都還要再加上 ID 屬性。全部加起來，最後看起來應該會像這樣：

```
<label for="author">名字:</label><br />
<input id="author" name="author" /><br /><br />

<label for="email">Email 地址:</label><br />
<input id="email" name="email" /><br /><br />

<label for="url">網址:</label><br />
<input id="url" name="url" /><br /><br />

<label for="text">迴響:</label><br />
<textarea id="text" name="text" rows="10" cols="50"></
textarea><br /><br />

<input type="checkbox" id="bakecookie" name="bakecookie" /
><label for="bakecookie">記住我的資訊?</label><br /><br />
```

請確定妳也在迴響預覽模版、迴響出錯模版和單篇彙整模版裡都做了相同的修改。

怎麼做: Greymatter

在「Edit Karma & Comments-Related Templates」裡，妳應該會看到一個叫「`{{entrycommentsform}}` Posting form」的模版，裡面包含著這樣的東西：

```
Name
<BR>
<INPUT TYPE=TEXT NAME="newcommentauthor" SIZE=40>
<P>
E-Mail (optional)
```

```
<BR>
<INPUT TYPE=TEXT NAME="newcommentemail" SIZE=40>
<P>
Homepage (optional)
<BR>
<INPUT TYPE=TEXT NAME="newcommenthomepage" SIZE=40>
<P>
Comments
<BR>
<TEXTAREA NAME="newcommentbody" COLS=35 ROWS=10 WRAP=VIRTUAL></
TEXTAREA>
```

請改成這樣：

```
<label for="newcommentauthor">Name</label>
<BR>
<INPUT TYPE=TEXT id="newcommentauthor" NAME="newcommentauthor"
SIZE=40>
<P>
<label for="newcommentemail">E-Mail (optional)</label>
<BR>
<INPUT TYPE=TEXT id="newcommentemail" NAME="newcommentemail"
SIZE=40>
<P>
<label for="newcommenthomepage">Homepage (optional)</label>
<BR>
<INPUT TYPE=TEXT id="newcommenthomepage"
NAME="newcommenthomepage" SIZE=40>
<P>
<label for="newcommentbody">Comments</label>
<BR>
<TEXTAREA id="newcommentbody" NAME="newcommentbody" COLS=35
ROWS=10 WRAP=VIRTUAL></TEXTAREA>
```

延伸閱讀

- WebAIM: How to Create Accessible Forms. 這裡介紹了更多複雜的表單，以及其他像 `<legend>` 和 `<fieldset>` 這些跟親和力有關的標籤。這些教學文件不但會告訴你他們是甚麼，同時也會讓你知知道該怎麼用。
- W3C: Forms in HTML Documents: The LABEL element.

Dive Into Accessibility

在 30 天內打造更具親和力的網站

第二十九天：加入站台搜尋功能

每個網站都需要站台搜尋功能。至少就當代來說是如此。

正確的站台搜尋準則：

1. 每一頁都放搜尋對話盒，最好還要是「在褶層之上」（不需要捲動就看得得到）。
2. 預設會搜尋所有的東西。如果你可以讓使用者選擇祇搜尋文章、祇搜尋迴響或兩者皆搜尋的話，請預設「兩者都搜尋」。
3. 別讓進階選項（像是布林邏輯、大小寫敏感或正規表示式）弄亂了你的搜尋對話盒。請預設使用類似 Google 的作用（比對所有的字，不比對部分的字，大小寫不敏感，不使用正規表示式），同時在另一個「進階搜尋」的頁面裡，提供所有的選項。
4. 讓你的搜尋對話盒有合適的標籤以及一組便捷鍵。在此我建議使用 `accesskey="4"`（請注意：在第十五天裡，我示範了如何指定你的搜尋表單的 `accesskey`，但是我弄錯了。為了要達到最大的跨瀏覽器相容性，你得把 `accesskey` 定義在 `<label>`，而不是 `<input>`。正確的語法請見下面的範例）。

誰因此獲益？

Jackie、Michael、Bill、Lillian、Marcus 以及世界上其他許多人都從這個妥善採行的站台搜尋功能中獲益了。尤其是對於網誌或新聞導向的站台來說，內容主要是按照時間所整理；如果使用者每次尋找特定的文章時，都得要將頁面不斷地往下捲動的話，實在是讓人苦不堪言。很少有人知道 Google 的「`site:domainname.com`」語法（用來把搜尋結果線訂在某個網域裡），而且 Google 每個月祇會重新建立一次索引，所以你該提供自己的站台搜尋功能。

怎麼做

Greymatter 內建有搜尋功能，但是你得稍微修改模版，纔能提供搜尋對話盒的 `accesskey`。到「Edit Templates」，「Miscellaneous Templates」，「Search Form Template」，然後找到像這樣的表單：

```
<FORM ACTION="{ {cgiwebpath} }/gm-comments.cgi" METHOD=POST>
<INPUT TYPE=TEXT NAME="gmsearch" SIZE=20>
<INPUT TYPE=SUBMIT VALUE="Search">
</FORM>
```

然後把他改成這樣：

```
<FORM ACTION="{ {cgiwebpath} }/gm-comments.cgi" METHOD=POST>
<label for="gmsearch" accesskey="4">尋找：</label>
<INPUT TYPE=TEXT id="gmsearch" NAME="gmsearch" SIZE=20>
<INPUT TYPE=SUBMIT VALUE="搜尋">
</FORM>
```

Movable Type 的使用者可以試試看 MT-Search 外掛模組（譯註：現在這個外掛模組已經整合到 Movable Type 套件裡了）。我自己正在用由將來臨的 MT 架起來的站台。我在自己的主要網誌（超過九百篇文章）上試了一下，稍微慢了點，不過在比較小的站台上應該還不賴纔是。雖然它不會繼續被發展下去，但就算是在最新版的 Movable Type 上，也還動得不錯。（請注意：如果你用的是 MySQL 版本的 Movable Type 2.2，你還得稍微調整一下 mt-search.cgi 裡的設定）。

如果你的網誌有自己的網域名稱，你還可以使用第三方的搜尋引擎來為你的內容建立索引，並按要求提供搜尋結果。比較為人所熟知的代用品包括像是 Blogger.com 所建議的 Atomz Express Search 和我在我的網誌上所採用的 Google Free Web Search。這兩者都提供了某些程度的自由度，而且也相當快；祇不過他們建立你的站台索引後，並不會刻刻更新。這兩個搜尋引擎都允許你自訂搜尋對話盒的外貌。我用 Google 做出來的搜尋表單就像這樣（請留意 <label> 和 accesskey 的使用）：

```
<form id="searchform" method="get" action="http://www.google.com/
custom">
<p id="searchlabel"><label for="q" accesskey="4">Search this
site:</label></p>
<p id="searchinput"><input type="text" id="q" name="q" size="18"
maxlength="255" value=" " /></p>
<p id="searchsubmit"><input type="submit" value="Search" />
<input type="hidden" name="cof" value="LW:116;L:http://
diveintomark.org/images/eyes.jpg;LH:68;AH:left;GL:0;S:http://
diveintomark.org/;AWFID:0d8ffcebe359c844;" />
<input type="hidden" name="domains" value="diveintomark.org" />
<input type="hidden" name="sitesearch" value="diveintomark.org" /
>
</p>
</form>
```

請注意：你不能夠就直接把這一段剪下來，然後貼進你的網誌裡。如果你想要用 Google Free Web Search 的話，你得先登入 Google，並且取得後來會被放到隱藏的 cof 欄位裡的金鑰。

延伸閱讀

- Jay Allen: MT-Search, Movable Type 的搜尋外掛模組。
- Phil Ringnalda: mt-search.cgi and MySQL. 這裡有如何讓 MT-Search 在 Movable Type 2.2 和 MySQL 裡運作順利的重要訊息。

- Atomz Express Search.
- Google Free Web Search.
- Blogger.com: How do I add a search engine to my blog? 她們推薦使用 Atomz ，但是其他鏈結都沒有列出。
- Jukka Korpela: Improving accessibility with `accesskey` in HTML forms and links. 接近尾聲的地方，解釋了為什麼站台搜尋功能的 `accesskey` 要用「4」。
- Jakob Nielsen: Is Navigation Useful? Users often rely on search as their main hunting strategy.
- Jakob Nielsen: Search Usability. 這個地方雖然已經五歲了，但仍然不可思議地一直在更新。
- PHP.net: URL HOWTO. PHP.net 上有我所看過最神奇的站台搜尋功能。除了標準的搜尋對話盒外，她們還有自訂的 404 錯誤文件（通常別人祇會丟給妳一個「頁面找不到」的錯誤訊息而已），並且根據網址來聰明地尋找站台。所以妳可以直接在瀏覽器的位址列輸入像 `php.net/phpinfo` 的位址，然後 `php.net` 會把妳轉向到 `phpinfo` 函式的參考頁面。

Dive Into Accessibility

在 30 天內打造更具親和力的網站

第三十天：撰寫親和力聲明

既然妳已經採行了這一系列的訣竅，現在就該撰寫一份親和力聲明，詳細列出站台上所有的親和力特色。

誰因此獲益？

1. Bill 從中獲益了。因為他十分仰賴鍵盤來導覽，而且如果他能夠知道相關訊息的話，就很可能會用到妳的 `accesskey` 快速鍵。很不幸地，Mozilla 不會用任何方法把這些設定唸出來或者顯示出來給使用者看，所以妳得自己列出來。
2. 許多其他使用者都從中獲益了，因為他們能夠由此得知，如何把她們自己的軟體用得更出神入化。舉例來說，Lillian 並不知道她可以設定瀏覽器的預設文字尺寸，直到有人跟她說可以這麼做；現在她終於可以更容易地閱讀妳的站台了（當然，妳得先使用相對字體大小）。Netscape 和 Mozilla 的使用者也許並不知道她們能夠開啟網站導覽列，從妳以鏈結為主的導覽協助中獲益（「檢視」選單，「顯示 / 隱藏」，「網站導覽列」，「需要時才顯示」。很不幸地，這個功能在最後一刻從 Mozilla 1.0 最終版裡被拿掉了，不過 1.1 版時又被加回來了）。
3. 其他網頁設計師從中獲益了，因為她們將能夠看到且學到妳所花時間採行的這些親和力訣竅。

怎麼做

在 Radio 或 Manila 中，妳可以把親和力聲明當作是一則故事（Radio 或 Manila 工具列上的「Stories」）來發表，所以它將有自己的網址。要不然就是直接建立起新的文章，並且在儲存後把它的網址記下來；稍後妳將會用上。

妳第一件該做的事情是列出妳所定義過的 `accesskey` 快速鍵，因為多數的瀏覽器既不會把這些快速鍵唸出來，也不會讓他們被顯示出來。同時也要記得列出親和力聲明本身的 `accesskey`（通常會是 `accesskey="0"`，我們馬上就會看到了）。

在這之後，妳還該列出所有妳採行的親和力特色。如果妳祇有採行某幾個訣竅的話，就祇提出妳真的用的那幾個就好。這會是很有啟發性的經驗，就跟妳在更新履歷時一樣；當妳第一次得到工作時，妳將瞭解到妳到底曾經做過了些甚麼事。妳已經做很多事了。

如果妳高興的話，還可以概略地描述一下為什麼要採行這些特色（像是「對於螢幕朗讀軟體的使用者來說會更好用」、「這是設計給純文字瀏覽器的使用者的」、「這是為了無法分辨顏色的使用者所設計的」之類）。請記住，親和力聲明可能會被更多殘障的使用者所閱讀，所以它不但該具備詳盡的情報性（給那些直接獲益的人），還該具備教育性（給那些拿妳的站台來當範例的網頁設計師）。或者除了妳

自己解釋之外，妳還可以連結到這本書裡的任一天的內容，當然也可以就祇是連結到本書的首頁。

最後，讓妳的站台裡每一頁都連結到妳的親和力聲明；妳可以放在全站的導覽列，也可以放在每一頁的註腳。指定這個鏈結的標題以及 `accesskey="0"`，就像這樣：

```
<a href="accessibility_statement.html" title="本站的親和力特色"
accesskey="0">親和力聲明</a>
```

延伸閱讀

- diveintomark.org 的親和力聲明。
- diveintoaccessibility.org 的親和力聲明。妳可以把頁面的結構、字句及延伸閱讀的鏈結這些當成模版，自由取用。

Dive Into Accessibility

在 30 天內打造更具親和力的網站

結語

以上就是「深入親和力：在 30 天內打造更具親和力的網站」。謹代表 Jackie、Michael、Bill、Lillian 和 Marcus 感謝您對此議題的關注。

延伸閱讀：我推薦的閱讀書目

1. Joe Clark: Building Accessible Websites. 我是這本書的技術編輯，這真的是本博而不濫的好書。
2. Jim Thatcher and others: Constructing Accessible Web Sites. 比 Joe 的書精簡一點，但是它所囊括的主題深度更為過之。這本書還提供了許多的擷取畫面，展示出各種螢幕朗讀軟體及瀏覽器在各種不同的標籤及標記上所顯示的效果；同時還有一個章節描述著跟親和力需求有關的法律聲明。
3. Steve Krug: Don't Make Me Think: A Common Sense Approach to Web Usability. 可用性及親和力在許多方面都會相重疊；Steve 在可用（且親和）的導覽上提出了許多很棒的忠告。
4. Owen Briggs, Steve Champeon, Eric Costello, Matthew Patterson: Cascading Style Sheets: Separating Content from Presentation. 就如同我們在這一系列的課程理所看到的一樣，CSS 乃是親和力中不可分割的一部份，因為他能讓妳在 HTML 標記（這也是輔助科技所在乎的部分）裡「做正確的事」，而仍舊讓妳在視覺性瀏覽器裡做出妳想要的效果。
5. Eric Meyer: Eric Meyer on CSS: Mastering the Language of Web Design. 在這本書的指南網站上也有一些免費的材料。

Dive Into Accessibility

在 **30** 天內打造更具親和力的網站

親和力聲明

以下是深入親和力的正式親和力聲明。如果你有任何疑問或建議的話，歡迎來信 feedback@diveintoaccessibility.org。

便捷鍵

大部分的瀏覽器都支援用網站上定義的按鍵來跳到特定的鏈結。在 Windows 上的話，你可以按下 ALT + 某個便捷鍵；在 Macintosh 上的話，你可以按下 Control + 某個便捷鍵。

這個站台裡所有的頁面都定義了下列的便捷鍵：

- 便捷鍵 1 - 主頁面
- 便捷鍵 3 - 目錄
- 便捷鍵 4 - 搜尋
- 便捷鍵 8 - 使用聲明
- 便捷鍵 9 - 回饋
- 便捷鍵 0 - 親和力聲明

遵循標準

1. 本站所有的頁面都通過了 Bobby AAA 認證，遵循所有的 Bobby 指導原則。這一直都是主觀判斷下的結果；許多親和力要素可以被測得，但是也有很多沒辦法。我已經回顧了所有的指導原則，相信所有的頁面都確實遵照了。
2. 本站所有的頁面都通過了 WCAG AAA 認證，遵循 W3C Web Content Accessibility Guidelines 的所有第一優先、第二優先和第三優先指導原則。同樣地，這也是主觀判斷下的結果；許多指導原則都刻意模糊而無法被自動測試。我也回顧了所有的指導原則，相信所有的頁面都確實遵照了。
3. 這個站台上所有的頁面都通過了 Section 508 認證，遵循所有的美國聯邦政府 Section 508 指導原則。再一次地，這還是主觀判斷下的結果。我已經回顧了所有的指導原則，相信所有的頁面都確實遵照了。
4. 本站所有的頁面都通過了 XHTML 1.0 Strict 認證。這並非主觀判斷；有一個程式能夠 100% 精確地判斷某個頁面是否為有效的 XHTML。舉例來說，檢查主頁面是否為有效的 XHTML。
5. 本站所有的頁面都使用了結構化語意標記。H2 標籤都用於主標題，H3 標籤都用於副標題。舉例來說，在這一頁裡，JAWS 的使用者可以按下 ALT+INSERT+3 來跳到親和力聲明的下一個小節。

導覽協助

1. 所有的頁面都有 `rel=previous`, `next`, `up`, 和 `home` 鏈結，來協助純文字瀏覽器導覽。Netscape 6 和 Mozilla 的使用者也可以從檢視選單、顯示 / 隱藏，站台導覽列、只在需要的時候顯示（或永遠顯示）來啟用這個進階功能。
2. 所有的訣竅都可以用多種不同的方法交叉參考。妳可以依人物、依障礙、依設計原則、依協助科技或依出版工具來瀏覽這些訣竅。
3. 主頁面以及所有的彙整頁面都有一個搜尋對話盒（便捷鍵 4）。進階搜尋選項也可以在進階搜尋頁面裡使用。

鏈結

1. 許多的鏈結都以 `title` 屬性來更詳細地描述鏈結，否則就是因為鏈結的文字已經完全能夠描述連結的目標（像是文章的標題）了。
2. 鏈結即使離開了情境依舊有意義。

圖片

1. 本站所有的內容圖片都包括了描述性的 `ALT` 屬性。單純的裝適用圖片則包含了空的 `ALT` 屬性。
2. 複雜圖片則會包含了 `LONGDESC` 屬性或者是內嵌了描述，為非視覺的讀者描述每一張圖片的含意。

視覺設計

1. 本站使用了樣式表串聯來處理視覺排版。
2. 本站僅使用相對字體大小，與使用者在視覺性瀏覽器指定的「字體大小」相容。
3. 如果妳的瀏覽器或瀏覽裝置完全不支援樣式表的話，每一頁的內容仍舊可閱讀。

親和力參考文獻

1. W3 親和力指導原則解釋了每一個指導原則背後的理由。
2. W3 親和力技術解釋了如何採行每一個指導原則。
3. W3 親和力核對表可以作為忙碌開發者的親和力指引。
4. 美國聯邦政府 Section 508 親和力指導原則。

親和力軟體

1. JAWS 是 Windows 上的螢幕朗讀軟體，這裡提供了可以下載的限時展示版。
2. Home Page Reader 是 Windows 上的螢幕朗讀軟體，這裡提供了可以下載的展示版。
3. Lynx 是一個免費的純文字瀏覽器，搭配可覆新的布拉耶點字顯示器後就可以給盲人使用。
4. Links 是一個給頻寬不足的視覺性使用者用的純文字瀏覽器。
5. Opera 是一個配備許多親和力功能的視覺性瀏覽器，包括有文字縮放、自訂樣式表、圖片標定等

多樣功能。有提供免費下載的版本。與 Windows、Macintosh、Linux 和許多其他的作業系統相容。

親和力服務

1. Bobby 是一個分析網頁是否遵照親和力指導原則的免費服務。也提供全功能的商業版本。
2. HTML Validator 是一個用來檢查網頁是否吻合已公開的 HTML 標準的免費服務。
3. Web Page Backward Compatibility Viewer 這個工具可以用來檢視當妳的網頁，看看若沒有各種流行的瀏覽器功能時，它會變成怎樣。
4. Lynx Viewer 是一個讓妳檢視網頁在 Lynx 裡看起來會像甚麼的免費服務。

相關資源

1. WebAIM 是一個投注於改善線上學習材料親和力的非營利組織。
2. Designing More Usable Web Sites 列出了大量額外的資源。

我建議的親和力書籍

1. Joe Clark: Building Accessible Websites. 我是這本書的技術編輯，這真的是本博而不濫的好書。
2. Jim Thatcher and others: Constructing Accessible Web Sites. 比 Joe 的書精簡一點，但是它所囊括的主題深度更為過之。這本書還提供了許多的擷取畫面，展示出各種螢幕朗讀軟體及瀏覽器在各種不同的標籤及標記上所顯示的效果；同時還有一個章節描述著跟親和力需求有關的法律聲明。

Dive Into Accessibility

在 **30** 天內打造更具親和力的網站

使用聲明

這本書是自由的。這表示妳在任何情況下，無論是否涉及商業，都可以任意取用本書裡的資訊，而無須以任何方式付酬或簽約。

不但如此，妳還被允許能在遵照 GNU 自由文件許可證 1.1 版或任何未來由自由軟體基金會所發佈的版本下，任意地複製、散佈以及/或修改此書的內容，而無須包含任何固定的章節、封面文字和封底文字。這份許可證的副本如後。

GNU Free Documentation License

Version 1.1, March 2000

Copyright (C) 2000 Free Software Foundation, Inc.
59 Temple Place, Suite 330, Boston, MA
02111-1307 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other written document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondly, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you".

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (For example, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, whose contents can be viewed and edited directly and straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup has been designed to thwart or discourage subsequent modification by readers is not Transparent. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML designed for human modification. Opaque formats include PostScript, PDF, proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies of the Document numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a publicly-accessible computer-network location containing a complete Transparent copy of the Document, free of added material, which the general network-using public has access to download anonymously at no charge using public-standard network protocols. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- **A.** Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
- **B.** List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has less than five).
- **C.** State on the Title page the name of the publisher of the Modified Version, as the publisher.
- **D.** Preserve all the copyright notices of the Document.
- **E.** Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- **F.** Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
- **G.** Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
- **H.** Include an unaltered copy of this License.
- **I.** Preserve the section entitled "History", and its title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
- **J.** Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

- **K.** In any section entitled "Acknowledgements" or "Dedications", preserve the section's title, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
- **L.** Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
- **M.** Delete any section entitled "Endorsements". Such a section may not be included in the Modified Version.
- **N.** Do not retitle any existing section as "Endorsements" or to conflict in title with any Invariant Section.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties--for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in

parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections entitled "History" in the various original documents, forming one section entitled "History"; likewise combine any sections entitled "Acknowledgements", and any sections entitled "Dedications". You must delete all sections entitled "Endorsements."

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, does not as a whole count as a Modified Version of the Document, provided no compilation copyright is claimed for the compilation. Such a compilation is called an "aggregate", and this License does not apply to the other self-contained works thus compiled with the Document, on account of their being thus compiled, if they are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one quarter of the entire aggregate, the Document's Cover Texts may be placed on covers that surround only the Document within the aggregate. Otherwise they must appear on covers around the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License provided that you also include the original English version of this License. In case of a disagreement between the translation and the original English version of this License, the original English version will prevail.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See <http://www.gnu.org/copyleft/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

```
Copyright (c) YEAR YOUR NAME. Permission is granted to copy,
distribute and/or modify this document under the terms of the
GNU Free Documentation License, Version 1.1 or any later version
published by the Free Software Foundation; with the Invariant
Sections being LIST THEIR TITLES, with the Front-Cover Texts
being LIST, and with the Back-Cover Texts being LIST. A copy of
the license is included in the section entitled "GNU Free
Documentation License".
```

If you have no Invariant Sections, write "with no Invariant Sections" instead of saying which ones are invariant. If you have no Front-Cover Texts, write "no Front-Cover Texts" instead of "Front-Cover Texts being LIST"; likewise for Back-Cover Texts.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.

Dive Into Accessibility

在 **30** 天內打造更具親和力的網站

譯本

如同在使用聲明所提到的一樣，本書的翻譯或按原有內容出版的權利都在相同的自由許可裡。我們非常歡迎妳在妳自己的站台上發表妳的翻譯版本。此外我也計畫最後在主 Dive Into Accessibility 站台上放置各個譯本。

如果妳把 Dive Into Accessibility 翻譯成別的語言，並且想要列在這裡的話，請聯絡我 translate@diveintoaccessibility.org。請注意妳無須詢問過我的意見就可以開始翻譯了；妳已經獲得我的首肯了。

Karl Dubost 提到了在本書裡的人物特徵都是美式的。妳可以自由地修改這些人物，改變她們的住處、房租甚至是名字。唯一要注意的就是請完整保持她們的障礙，因為那纔是這整本書所要提到的。

中文

Jedi 和 Liang-Bin Hsueh 已經發表了繁體中文版（ UTF-8 ）和簡體中文版（ UTF-8 ）的翻譯。

愛沙尼亞語

Priit Laes 正在著手進行愛沙尼亞語版本的翻譯。

芬蘭語

Visa Kopu 和 Marjut Mutanen 正在著手進行芬蘭語版本的翻譯。

法語

Karl Dubost 已經發表了他的法語版翻譯。

德語

Haiko Hebig 和許多其他的譯者正在著手進行德語版的翻譯。

義大利語

Dario Di Ludovico 正在著手進行義大利語版本的翻譯。

挪威語

Jesro Christoffer Cena 正在著手進行挪威語版本的翻譯。

羅馬尼亞語

Petru Paler 正在著手進行羅馬尼亞語版本的翻譯。

西班牙語

Iván de Cincomonos 和其他譯者正在著手進行西班牙語版本的翻譯。

瑞典語

Marko Injac 正在著手翻譯瑞典語版本的翻譯。