

Joomla Template Tutorial

Last Updated Friday, 04 August 2006

在本教程中，我們將完成創建 Joomla 範本這一步。Joomla 是一個開放原始程式碼 Content Management System (CMS)，該系統可以自由（真正的）使用並且由一個很強大的線上社區支持。具體地說，我們將創建一個範本，使用 cascading style sheets (css) 而不是表格來生產佈局。這是一個很合理的目標，因為它意味著範本代碼會更容易符合 World Wide Web Consortium (W3C) 標準。它裝載的更快，更容易維持並能在搜尋引擎中執行地更好。本教程稍後將討論這些問題的細節，它包含以下章節。

-什麼是 Joomla 範本？

闡述 Joomla 範本執行的功能有哪些，不包含內容的範本與添加了內容到 CMS 的範本之間的區別。

-本地設計過程

設計過程與設計一個靜態 (x) HTML 網頁之間的不同之處。

-W3C 與無表格設計

涉及到 Joomla 中無表格設計和 W3C 標準，可用性與無障礙環境三者之間的關係。

-範本元件

Joomla 範本由什麼檔組成，它們各具有什麼功能。

-使用 CSS 創建佈局

如何使用 CSS 代替表格來創建一個源定 3 欄的佈局

-默認的 Joomla CSS

介紹基本的 CSS 樣式，應該與 Joomla 一起使用，還有被 Joomla core 使用的類型預設名單。

-模組

如何放置，包含圓角新技術的類型模組。

-菜單

一個簡單的策略用於生產精益的 CSS 功能表，模仿出那些用 javascript 寫的菜單效果。

-隱藏欄目

如何控制什麼時候顯示欄，當沒有內容時隱藏欄。

-結論

-附錄 A：提示與技巧

可變頁寬度，圓角，文本成型模以及更多

{mospagebreak title=What is a Joomla Template?}

什麼是 Joomla 範本？

Joomla 範本就是在 Joomla CMS 裡面的一系列檔，用來控制內容的顯示。Joomla 範本既不是一個網站，也不是一個完整的網站設計。該木板對於查看你的 Joomla！網站是一個基礎。要生產出一個“完整”網站的效果，範本會和存儲在 Joomla 資料庫中的內容密切合作。以下你可看到一個這方面的例子：

這個截圖顯示了與樣本內容的使用。圖 B 顯示了範本，它可能看著與一個原始 Joomla 安裝一起，幾乎沒有內容。範本被格式化了，這樣當你的內容被插入禁區時，將繼承定義在範本中的樣式表，比如連接類型，功能表，巡覽列，文本大小和顏色等。注意圖片和內容是如何聯繫的（人們的相片），該內容不是範本的一部分，而在標題那的圖片是範本的成分。

為一個 CMS 使用一個範本，就像 Joomla 一樣，有很多優點和缺點：

-內容和外觀完全分離，特別是當 CSS 用於佈局時（因為反對在 index.php 檔中存在表格）。這是網站主要標準的一個，符合現代網站標準。

-一個新的範本,因此對網站而言一個全新的外觀可以立刻被應用上.這甚至可以和顏色圖片一樣有不同的內容地點/定位

-如果一個網站內部想有不同的版面配置,這很難實現.雖然不同的範本可以應用到不同的頁面,這種內置的功能並不可靠.大多數設計者選擇使用各種 PHP 代碼來顯示/隱藏欄取決於是否在那個位置公佈了內容.(在提示與技巧中討論到了).

{mospagebreak title=Localhost Design Process}

本地設計過程

你在強大的 Joomla 網站上看到的頁面並不是靜態的.這就說明它是動態地從存儲在資料庫中的內容產生的.你所看到的頁面是通過在範本中各種 PHP 命令創建的.這種設計階段提出了一些困難.

現在使用一個“所見即所得”(WYSIWYG)的 HTML 編輯器已經很普遍了,比如 Dreamweaver.這就意味著設計者不需要再對 HTML 進行編碼.而在 Joomla 範本設計過程中這是不可能的.所見即所得編輯器不能顯示一個動態網頁.這說明設計者必須“用手”編寫代碼,然後從一個服務端頁面的 PHP 查看輸出頁面.只有 web 伺服器擁有足夠快速的連接,但是大多數設計者使用他們自己電腦上的“local server”.這是一款服務於設計者自己電腦上網頁的軟體.

沒有辦法“立刻”創建一個頁面,它取決於設計者自己的背景.那些更圖形化的頁面在圖形軟體中比如 photoshop 製作頁面的“形象”,然後將圖形分開來使用(名為片和塊).高技術的設計師往往直接跳到 CSS 進行編碼.然而,上面我們提到,Joomla 範本設計者是受限制的,他不能在相同的編輯器中立刻看到她/他編碼的效果.改進的設計過程如下:

- 在 HTML 編輯器中做出修改,保存修改
- 在背景下由本機伺服器來“運行”Joomla
- 在網站瀏覽器中查看編輯
- 回到第一步

本機伺服器選項

幾個本地主機伺服器可用:

- JSAS - Joomla Stand Alone Server. WAMP

支援個人電腦上 windows 系統下 Joomla 的伺服器.一個自身包含了 Apache-MySQL-PHP 伺服器.

jsas.joomlasolutions.com

- XAMPP

在Linux,Windows, Mac OS X和Solaris上安裝Apache Distribution很簡單.該包包含了Apache web server, MySQL, SQLite,PHP等.

www.apachefriends.org/en/xampp.html

JSAS確實有大量的廣告在上面,應該會吸引你.

Easy CSS Styling

是設計過程更有效的令人滿意的技術是為你正在設計的網頁服務,然後拷貝粘貼原始程式碼到一個編輯器中.例如,一旦你的 CSS 佈局成立,你可以使用其中以本機伺服器來支援一個頁面, View_Source,然後將它拷貝粘貼到你的編輯器中,現在你就可以很簡單地使用 CSS 來為頁面定義類型,且不需要重複上述步驟.

Some HTML Editor Options for Joomla Designers

對於那些不能支付商業編輯器比如 Dreamweaver 的人,這有一些免費的編輯器可用:

Nvu 是一個堅實的選擇,並確認已經建成..它還是一個 Mambo/Joomla 的延伸,會非常有用. Nvu 是 100%的開放原始程式碼.這就意味著任何人都歡迎來免費下載 Nvu,如果要做特殊的修改的話還包含原始程式碼.你可以從 Manboforge

得到延伸版,是在這裡建立完整的項目.

另外一個選在實際上也是一種驗證, "CSE HTML Validator"是集 HTML,XHTML,CSS,連結,拼寫和無障礙環境檢查於一體的驗證.你可以在這裡取得免費的 html 驗證器版本.

請記住,這些都不是"所見即所得"的 html 編輯器.

{mospagebreak title=W3C and Tableless Design}

W3C 與無表格設計

可用性,可訪問性和搜尋引擎優化都是用來形容今天萬維網中高品質網頁的片語,現實中,這三者之間存在大量的重疊,一個網頁演示了其中一個的特徵也就顯示了三者的特徵.最簡單的方法來達到這三個目標就是使用 W3C 網頁標準中的 framework.

例如,一個(x)html 語義結構的網站(xhtml 解釋檔,而不是它看來如何)很容易讓某些視力不好的人在一個螢幕閱讀器中閱讀.對於一個搜尋引擎 spider 閱讀也很簡單.Google 在如何閱讀你的網站方面是很盲目的.

一個符合 W3C 網站標準的網站有一個更好的基礎使其變得可訪問,可用以及搜尋引擎優化.把這些想像成你家房子的建築法規.一個擁有這些特性的網站會更強大,更安全.你可以用 W3C HTML 驗證服務來檢查你的頁面.免費的.最簡單的,一個符合 W3C 驗證的網站運用 (x) html 語義結構並通過 CSS 將內容從外觀中分離。

為了幫助你理解網站標準的來源,回顧歷史是個好方法.實際上很多網頁是針對舊版本的瀏覽器設計的.為什麼?自從 www 開始後,瀏覽器不斷地演變.新版本已經出現而舊版本已經消失(還記得 Netscape?).另一個複雜的因素是,不同的瀏覽器製造商(比如微軟)試圖讓他們的瀏覽器以稍微不同的方式去解釋 html/xhtml.這導致了網站設計師不得不設計他們的網站以支持舊版本的瀏覽器而不是新版本.往往網頁設計成與這些“遺留的”瀏覽器相匹配是必須的。

網站標準為所有的網頁瀏覽器投入了一套共用的“規則”來顯示一個網頁。而推動這些標準的主要組織就是 W3C，其負責人，Tim Berners-Lee 在 1989 年發明了 world wide web。

如果你問 5 個設計者：什麼是 web 標準？將會有 5 個不同的答案。但是大多數認為它們是基於以下內容的：

-有效的代碼，無論是 html 或者 xhtml（或其它）

之前我們使用了一個創建結構代碼的例子。描述製作網頁代碼時的標準已經發展成就有連貫性。在 validator.w3.org 中檢查你的代碼很簡單。當你驗證你的代碼時確保你使用了正確的 doctype。在 Compass Design 上的這篇文章描述了一個有效的 joomla doctype。

-代碼語義正確

我們前面提到,要有語義,即在網頁中的(x)html 只描述內容,而不是外觀.特別是指有結構組織 h1/h2 等的標記和只用表格來存放表格資料,而不佈置一個網頁.

-Cascading Style Sheets (CSS)

與有語義的代碼密切相關的,是使用CSS來控制外觀和網頁的佈局.CSS是添加樣式(例如字體,顏色,間距)到web檔的一個簡單機制. (Source: www.w3.org/Style/CSS/). 它們存在與(x)html類似之處,這就可以讓你將內容(語義代碼)從外觀(CSS)中完全分離. 有關這方面最好的例子是CSS Zen Garden,一個有相同語義的xhtml網站用不同的CSS樣式以不同獨特的方式成形.結果是很多網頁看起來不一樣但是卻有相同的核心內容.

設計 Joomla 強大的網站,要滿足驗證標準目前來說是相當大的挑戰.在當前發佈的一系列新聞中,1.0.X,使用了大量的表格來輸出它的頁面.它既不使用 CSS 來設計外觀,也不產生語義化正確的代碼.這個問題與事實混合在一起,也就是極少數的協力廠商開發者既使用 CSS,也用表格來生成代碼.然而,無表格設計與 valid 並不一樣.它很有可能在網站上使用表格來驗證,這會使其更難.在 Joomla 論壇上有一個很有用的執行緒,進入瞭解更多詳情:

輕鬆的手法從 Joomla!輸出標準中清除一些表格.

幸運的是, Joomla Core Development 團隊意識到了 Joomla 這個問題.當 1.5 版本中從核心內容中移除表格沒有變化,在 1.6 版本中已經定義了路線來記錄位址.

無論如何.當創建一個範本時,核心內容仍然會採用,這樣它變得方便(比如可升級的字體大小),實用(比如清晰的巡覽列),搜尋引擎優化(比如資源整齊).

```
{mospagebreak title=The Template Components}
```

The Template Components 範本元件

爲了瞭解範本的內容.我們開始查看一個空白的 Joomla 範本.在這個檔中,包含了各種組成一個 Joomla 範本的檔和資料夾.這些檔必須放在 Joomla 安裝程式的/templates 目錄下.因此,如果你安裝了兩個範本,我們的目錄就會看起來像:

```
/templates/JS_Smoothportal
```

```
/templates/JS_Synergy
```

請注意範本的目錄名必須和範本的名稱一樣.既然這樣, JS_Smoothportal 和 JS_Synergy,很顯然它們是區分大小寫的,並且不包含空格.傳統上最初的设计者或者名字是用作首碼,在範本的目錄中,有很多重要的文件:

```
/JS_Smoothportal/templateDetails.xml
```

```
/JS_Smoothportal/index.php
```

這兩個檔案名和位置必須和 Joomla 腳本調用時一致.

- templateDetails.xml

(記錄大寫的"D")當裝載一個使用該範本的網頁時,一個 XML 格式的中繼資料檔會告訴 Joomla!其他檔需要什麼.它還詳細介紹了作者,版權以及範本的組成檔(包括所有使用的圖片).最後使用該檔是在管理員端安裝範本時.

- index.php

這個文件是最重要的.它涉及網站並告訴 Joomla CMS 在哪放置不同的元件和模組.它是 PHP 和(x)HTML 的結合.

幾乎在所有的模組中,額外的檔都會使用到.像如下顯示的那樣對它們進行命名和定位是常規:

```
/JS_Smoothportal/template_thumbnail.png
```

```
/JS_Smoothportal/css/template_css.css
```

```
/JS_Smoothportal/images/
```

```
logo.png
```

- template_thumbnail.png

一個網站瀏覽器,模組的截圖(往往減小到大約寬 140 圖元,高 90 圖元).安裝完範本後,功能"Preview image"可在

joomla! 執行中看到.

Template Manager.範本管理者

- css/template_css.css

範本的 CSS 樣式.資料夾的位置是可選的,但是你必须指定放在哪.請注意,檔案名稱只有在引用到 `index.php` 文件中時是重要的.你可以隨意稱呼它.通常顯示的名字會被用到,但是我們稍後將看到存在其它 CSS 檔會有優勢.

- images/logo.png

任意與範本匹配的圖片.又是組織原因,大部分設計師把它放到圖片資料夾中.這裡我們將 `logo.png` 圖片檔作為例子.

要添加範本(再次,存在于豐富的教程中),你要到自己網站的管理端,通過上傳壓縮檔安裝範本.注意事實上你也可以一個個地添加檔(而不是壓縮檔的形式).你必须把它們放到你網站上的 `.com/templates` 中.

templateDetails.xml

templateDetails.xml 必須包含範本組成部分的所有檔.以及作者和版權資訊.有的顯示在 **Template Manager** 的管理員後端.

如下是一個 xml 檔例子 :

```
<mosinstall type="template" version="1.0.x">
```

```
<name>YourTemplate</name>
```

```
<creationDate>March 06</creationDate>
```

```
<author>Barrie North</author>
```

```
<copyright>GNU/GPL</copyright>
```

```
<authorEmail>compassdesigns@gmail.com</authorEmail>
```

```
<authorUrl>www.compassdesigns.net</authorUrl>
```

```
<version>1.0</version>
```

```
<description>
```

```
An example template that shows a basic xml details file </description>
```

```
<files>
```

```
<filename>index.php</filename>
```

```
<filename>js/ie.js</filename>
```

```
<filename>template_thumbnail.png</filename>
```

</files>

<images>

<filename>images/header.png</filename>

<filename>images/background.png</filename>

<filename>template_thumbnail.png</filename>

</images>

<css>

<filename>css/base.css</filename>

<filename>css/norightcol.css</filename>

<filename>css/template_css.css</filename>

</css>

</mosinstall>

讓我們來解釋一下這些代碼：

- mosinstall

XML 檔的內容用於安裝說明。選項類型 type= “template” 告訴安裝者我們正在安裝一個範本。

- name:

定義你範本的名稱。你在這裡輸入的名字將被用來在範本目錄中創建目錄。因此，它不應該包含檔案系統不能識別的字元，比如空格。如果手動安裝，你要創建一個目錄與範本名相同。

- createDate:

範本創建的日期。這是一個自由形態的區域，可以是任意像這樣的格式：May 2005, 08-June-1978, 01/01/2004 等。

- author:

該範本作者的名字-大多時候是你的名字。

- copyright:

任何版權資訊都進入這一部分。對於 Developers & Designers 來說是一本許可證，能在 Joomla 論壇找到。

- authorEmail:

Email 位址，能找到該範本的作者。

- authorURL:

作者網站的 URL

- version:

該範本的版本

- files:

“files” 部分包含了所有通用的檔，像範本的 PHP 原始檔案，範本預覽的縮略圖。每一個列出在這個部分的檔都

由<filename> </filename>括起來。還有一些額外的檔被包含進來，這裡我們使用一個範本需要的 JavaScript 檔例子。

- images:

所有範本使用的圖片檔都列在圖片區。同樣的，所有列出的檔由<filename> </filename>括起來。檔的路徑資訊與範本的根目錄息息相關，比如，如果你的範本在目錄中名為'YourTemplate'，並且所有的圖片在'YourTemplate'裡面的目錄'images'中，那麼正確的路徑是<filename>images/my_image.jpg</filename>。

- css:

樣式表列在 CSS 區域。同樣的，檔案名被包含在<filename> </filename>中，且它的路徑與範本的根路徑相關。使用大量由主 template_css.css 導入的樣式表會很有說明的。我們將在教程後面的內容中詳細介紹。

The index.php

在 index.php 中到底有什麼？它是(X)HTML 和 PHP 的結合，決定有關佈局和頁面外觀的一切。

首先我們會研究取得驗證範本的關鍵部分，在 index.php 文件頂端的 DOCTYPE。這段代碼在任意網頁的頂端運行。在我們頁面的頂端，我們把這段代碼放到範本中：

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
```

```
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

```
<html xmlns="http://www.w3.org/1999/xhtml"
```

```
lang="<?php echo _LANGUAGE; ?>"
```

```
xml:lang="<?php echo _LANGUAGE; ?>">
```

一個頁面 DOCTYPE 是基本元件的一部分，基本元件是由瀏覽器顯示網頁的那一部分，特別是，瀏覽器如何解釋 CSS。為了讓你領悟，從 alistapart.com 上觀察顯示：

[information on W3C's site about doctypes is] "由 geeks 為 geeks 寫的。當我說到 geeks 時，我並不是說普通的網站專業人員像你和我。我是指那些在她收到郵件的第一天使我們這些人看起來像 Grandma 的人。

無論如何，有一些doctype你可以使用。基本上，doctype告訴瀏覽器如何解析頁面。這裡"strict" and "transitional"兩個詞開始四周浮動(float:left and float:right通常)。本質上講，自從有了WWW，不同的瀏覽器在不同程度上支持CSS。這就說明，比如，Internet Explorer不理解"min-width"命令是用來設置最小頁寬。為了達到同樣的效果，你必須使用CSS中的"hacks"。

Strict是指html(或者xhtml)將解析成和標準規定的完全一樣。一個transitional doctype是指頁面將允許一些與標準不同的地方。

對於複雜的東西，有一種名為"quirks"樣式，如果doctype是錯誤的，過時的，或者不存在的，那麼瀏覽器會進入quirks模式。這是一個與後端相容的嘗試，所以比如Internet Explorer，會呈現出假裝成IE4的頁面。

不幸的是，有時候人們會意外地結束quirks模式。通常以兩種方式發生：

-他們直接從WC3頁面使用doctype聲明，連結因此中斷：

```
DTD/xhtml1-strict.dtd
```

除了它是WC3伺服器上一個相關的連結外。你需要如上所示的完整路徑。

微軟建立了IE6，這樣你就可以擁有有效的頁面，但是在quirks模式下。將一個"xml prolog"放在doctype前面後，這就會發生。

```
<?xml version="1.0" encoding="iso-8859-1"?>
```

關於IE6quirks模式部分是很重要的。在本教程中我們將只為IE6+設計，因此我們會確保它在標準模式下運行。這樣可以減少我們之後必須處理的hacks。Xml prolog不管怎樣都是沒必要的，我們會注意之後joomla的發行情況然後把xml prolog停止掉。

要製作一個符合標準的頁面，一個你能在頂端看到"valid xhtml"的頁面，並不意味著很難編碼，或者很難理解標籤。它僅僅說明你使用的代碼與你說的doctype匹配。僅此而已！把你的網站設計成符合標準可以在一定程度上減少成說明你所做的，然後做你所說的。

一些有用的連結：

- <http://www.quirksmode.org/css/quirksmode.html>
- <http://www.alistapart.com/stories/doctype>
- <http://www.w3.org/QA/2002/04/Web-Quality>
- <http://forum.joomla.org/index.php/topic,7537.0.html>
- <http://forum.joomla.org/index.php/topic,6048.0.html>

在 index.php 中又是什麼呢？

首先讓我們看看標題的結構，我們希望越小越好，卻仍然足夠產生一個網站。我們將使用的標題資訊是：

```
<?php defined( '_VALID_MOS' ) or die( 'Direct Access to this location is not allowed.' ); ?>
```

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
```

```
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

```
<html xmlns="http://www.w3.org/1999/xhtml" lang="<?php echo _LANGUAGE; ?>"
```

```
xml:lang="<?php echo _LANGUAGE; ?>"
```

```
<head>
```

```
<meta http-equiv="Content-Type" content="text/html; <?php echo _ISO; ?>" />
```

```
<?php
```

```
if ($my->id) {
```

```
initEditor();
```

```
}
```

```
?>
```

```
<?php mosShowHead(); ?>
```

```
<script type="text/javascript"> </script>
```

```
<!--http://www.bluerobot.com/web/css/fouc.asp-->
```



```
<link href="templates/<?php echo $cur_template; ?>/css/template_css.css" rel="stylesheet"
type="text/css" media="screen" />
```

```
</head>
```

這些都是什麼意思呢？

```
<?php defined( '_VALID_MOS' ) or die( 'Direct Access to this location is not allowed.' ); ?>
```

確保檔不會被直接訪問。

```
<?php defined( '_VALID_MOS' ) or die( 'Direct Access to this location is not allowed.' ); ?>
```

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
```

```
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

```
<html xmlns="http://www.w3.org/1999/xhtml" lang="<?php echo _LANGUAGE; ?>"
```

```
xml:lang="<?php echo _LANGUAGE; ?>"
```

```
<head>
```

關於這個上面我們已經講到。"<?php echo _LANGUAGE; ?>"是從網站的整體配置上引入語言。

```
<meta http-equiv="Content-Type" content="text/html; <?php echo _ISO; ?>" />
```

我們所使用的字元集，_ISO 是一個特殊的用來定義字元集編碼的常量。

```
<?php
```

```
if ($my->id) {
```

```
initEditor();
```

```
}
```

```
?>
```

這是一個腳本變數，如果一個使用者登錄到你的網站時，它非零。如果一個用戶登錄，WYSIWYG 編輯器被預載進來。你可以，如果你願意，經常預載該編輯器，但是總體上一個匿名的訪客不需要添加內容。這樣可以為你的網站的普通流覽節省一些腳本 overhead。

```
<?php mosShowHead(); ?>
```

標題資訊再次在整體配置中設置。它包含以下標記（在預設安裝的情況下）：

```
<title>A Complete Guide to Creating a Joomla Template </title>
```

```
<meta name="description" content="Installing Joomla, doctype and the blank joomla template" />
```

```
<meta name="keywords" content="installing joomla, joomla doctype, blank joomla tempate" />
```

```
<meta name="Generator" content="Joomla! - Copyright (C) 2005 Open Source Matters. All rights reserved." />
```

```
<meta name="robots" content="index, follow" />
```

```
<link rel="shortcut icon" xhref="images/favicon.ico" />
```

```
<script type="text/javascript"> </script>
```

要阻止一個 bug，即作為一個無樣式內容的 flash。Blue Robot 的 Details courtesy。注意這個可以是任意指令檔，所以如果我們添加一個的話，我們可以移除這一行。

```
<link href="templates/<?php echo $cur_template; ?>/css/template_css.css"
```

```
rel="stylesheet" type="text/css" media="screen" />
```

這一行為範本連結到 CSS 檔。PHP 代碼<?php echo \$cur_template; ?>將返回當前範本的名字。這就生成"portable"這行。當你創建一個新的範本，你可以拷貝它（沿著整個標題代碼）且不用擔心編輯任何事情。

正如你稍後會看到，在 `temmplate_css.css` 檔中，我們將使用 `@import` 作為阻止網站與 Netscape 4 破裂的一種方式。使用舊版本瀏覽器的使用者不能取得 CSS 樣式表，因此將看到我們完全無樣式的內容。如果你想配合這些瀏覽器，我們會有許多 CSS hacks，於是我們這樣做。一個空白的 `joolma` 範本主體

這將非常簡單，準備好了嗎？

```
<body>
```

```
<!-- 1 --><?php echo $mosConfig_sitename;?>
```

```
<!-- 2 --><?php mospathway();?>
```

```
<!-- 3 --><?php mosLoadModules('top');?>
```

```
<!-- 4 --><?php mosLoadModules('left');?>
```

```
<!-- 5 --><?php mosMainBody();?>
```

```
<!-- 6 --><?php mosLoadModules('right');?>
```

```
<!-- 7 --><?php include_once( $mosConfig_absolute_path './includes/footer.php' );?>
```

```
</body>
```

```
</html>
```

我們有一個合理的邏輯順序：

- 網站的名稱
- 路徑
- 頂端模組
- 左模組
- 主要內容
- 右模組
- 預設頁腳模組

目的在於嘗試並盡可能接近語義標記。從網站角度來看，它意味著網頁可以由任何人，瀏覽器，spider 或者螢幕閱讀器閱讀。語義佈局是可訪問性的基石。

現在值得注意的是，在這裡我們真正擁有的只是語義佈局的可能性。如果一個向前進了，並且把隨機的模組放到任意位置，就會引起一片混亂。對於 CMS 網站重要的的顧慮是，一個範本只是和內容的 `population` 一樣好。正是這個問題經常使設計者在試圖驗證他們網站時犯錯。

{mospagebreak title=Using CSS to create a layout}

使用 CSS 創建佈局

我們將使用 CSS 來給 Joomla 範本製作一個三欄佈局。並且會把它製作成一個流體佈局。網站頁面配置有兩個主要的類型，固定的和流體的，這兩種類型都提到如何控制頁面的寬度。

頁寬成爲一個問題是因爲人們上網所使用的瀏覽器解析度。儘管百分比在下降，但是有 20%的線民使用 800 X 600 的解析度。主要的 76%使用 1024 X 768 或者更高的解析度(source:www.upsdell.com)。製作一個流體佈局意味著你寶貴的網頁將不會是一個 1024 解析度狹窄的欄目，且在更小的顯示器中都可視。

一個典型的設計可能會使用表來對頁面配置。它們在你必須設計欄寬爲百分比的情況下很有說明，但是它們也有一些缺點：

-與 CSS 佈局相比，它們有很多額外的代碼。這將導致更長的載入時間（線民不喜歡）以及搜尋引擎中更差的表現。該代碼大致上可以雙重大小，不僅僅是標記還有一些名爲"spacer gifs"的東西。即使是大公司，有時候也會落入表陷阱中，被一份關於新 disney.co.uk 網站的 contorversey 看到。

-它們很難維持。要改變一些你必須弄清楚的，所有的 td/tr 正在做的東西。使用 CSS，只要幾行代碼來檢查。

-這些內容不能作爲有序的資源。

很多線民在瀏覽器上看不到網頁。那些使用一個文本瀏覽器或者螢幕閱讀器流覽的人將從左上角到右下角進行閱讀。這就意味著他們在看到中間欄即重要資訊之前首先流覽標題和左欄的所有東西（對於一個 3 欄佈局）。而一個 CSS 佈局允許"source-ordered"內容，也就是內容可以在代碼或者資源中進行排列。可能你最重要的網站訪問者是 Google，它爲所有目的地和意圖使用了一個螢幕閱讀器。

讓我們看看使用 CSS 設置的佈局。使用 CSS 你可以以很多方式放置元素（東西）。快速地介紹一個很好的資源 Brainjar's CSS Positioning.

如果對於 CSS 你是新手，你至少閱讀了"beginners guide to CSS".這裡有一些建議：

Kevin Hale's - An Overview of Current CSS Layout Techniques

htmldog's CSS Beginner's Guide

Mulder's Stylesheets Tutorial

yourhtmlsource.com

我們將使用浮點來佈置內容。由於其最基本性，範本可能看起來像這樣：

```
<?php defined( '_VALID_MOS' ) or die( 'Direct Access to this location is not allowed.' ); ?>
```

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
```

```
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

```
<html xmlns="http://www.w3.org/1999/xhtml" lang="<?php echo _LANGUAGE; ?>"
```

```
xml:lang="<?php echo _LANGUAGE; ?>"
```

```
<head>

<meta http-equiv="Content-Type" content="text/html; <?php echo _ISO; ?>" />

<?php

if ($my->id) {
initEditor();
}
?>

<?php mosShowHead(); ?>

<script type="text/javascript"> </script>

<!--http://www.bluerobot.com/web/css/fouc.asp-->

<link xhref="templates/<?php echo $cur_template; ?>/css/template_css.css" rel="stylesheet"
type="text/css" media="screen" />

<style type="text/css">
<!--

#wrap {width:80%;}

#header {}

#sidebar {float:left;width:20%;}

#content {float:left;width:60%;}

#sidebar-2 {float:left;width:20%;}

#footer {clear:both;}

-->
</style>

</head>

<body>

<div id="wrap">
```

```
<div id="header">

<?php echo $mosConfig_sitename; ?>
<?php mospathway() ?>

</div>

<div id="sidebar">

<?php mosLoadModules('left');?>

</div>

<div id="content">
<?php mosLoadModules('top');?>
<?php mosMainBody(); ?>

</div>

<div id="sidebar-2">

<?php mosLoadModules('right');?>

</div>

<div id="footer">

<?php include_once( $mosConfig_absolute_path .'/includes/footer.php');?>

</div>

</div> <!--end of wrap-->

</body>
</html>
```

CSS 樣式在檔的開頭被定義來顯示正在進行的事件，但是一般的它們是在 `template_css.css` 文件中。一切都載在一個名為 `#wrap` 的盒子或者元素中。該盒子在任何時間都擁有 80% 的可視寬度。

CSS Shorthand

通過使用“shortland”減少 CSS 的數量是可行的。這方面的一個例子就是應用於某個元素的填充和 `margin` 類型。

```
margin-top:5px;
```

```
margin-bottom:5px;
```

```
margin-left:10px;
```

`margin-right:10px;`

可以由如下代替：

`margin: 5px 10px;`

在最開始的類型定義中有“shorthand”類型。一旦你已經揣摩出該類型，填充 shorthand 版本然後刪除 long 版本。語法是：

這是一個例子，而非這個.....

`font-size:1em;`

`font-family:Arial,Helvetica,sans-serif;`

`font-style:italic;`

`font-weight:bold;`

`line-height:1.3em;`

有這個：

`font:bold 1em/1.3em Arial,Helvetica,sans-serif italic;`

在 [An Introduction to CSS](#) 閱讀更多關於該語法的 shorthand 屬性。

左邊，中間和右邊的欄均給出自己的元素。每個欄是浮動離開然後給出了一個百分比寬度加起來為 100%。`clear`：在頁腳的兩種類型告訴瀏覽器“stop floating”並使頁腳橫跨這三欄。

要提高佈局並給內容添加一些空間，我們需要添加一些列間距，俗稱“gutter”。不幸的是，有個問題。你知道 Internet Explorer 並沒有很正確地翻譯 CSS。有個問題就是不同地計算寬度。我們通過使用一些填充或者一些有寬度的邊界來解決這個問題。要獲得 gutter，我們添加另一個<div>到列中。顯示如下：

```
<div id="sidebar">
```

```
<div class="inside">
```

```
<?php mosLoadModules('left');?>
```

```
</div></div>
```

```
<div id="content">
```

```
<div class="inside">
```

```
<?php mosLoadModules('top');?>
```

```
<?php mosMainBody(); ?>
```

```
</div></div>
```

```
<div id="sidebar-2">
```

```
<div class="inside">
```

```
<?php mosLoadModules('right');?>
```

```
</div></div>
```

給 CSS 添加：

```
.inside {padding:10px;}
```

這個簡單的佈局是一個用於學習如何將 CSS 和 Joomla 一起使用的很好的例子。它給出了 CSS 超越表格基於佈局的兩個優點,較少的代碼,很容易維持。然而,它並不是有序資源。因此我們必須使用一種更先進的佈局名為”nested float”。有了他的允許,我們將使用一個由 Dan Cederholm 開發的佈局,在本教程中有詳細的描述。

Source Ordered Three Column CSS Layout

爲了幫助解釋我們如何做這個例子,讓我們首先看一下結果。

[TO DO: PICTURE OF NESTED FLOAT HERE]

該頁被分爲兩個主要 floats,第一,#main-body 左邊浮動,#sidebar-2 是右邊浮動。這和我們以前做的一樣,main-body 浮動將首先出現在原代碼中。現在,在 main-body 內部,我們有另外兩個浮動:#content 是右邊浮動,#sidebar 是左邊浮動。只要我們正確設置寬度,#content 浮動就可以首先出現在原始程式碼中。

```
<div id="wrap">
```

```
<div id="header">
```

```
<?php echo $mosConfig_sitename; ?>
```

```
<?php mospathway() ?>
```

```
</div>
```

```
<div id="main-body">
```

```
<div id="content">
```

```
<div class="inside">
```

```
<?php mosLoadModules('top');?>
```

```
<?php mosMainBody(); ?>
```

```
</div></div>
```

```
<div id="sidebar">
```

```
<div class="inside">
```

```
<?php mosLoadModules('left');?>
```

```
</div></div>
```

```
</div> <!--end of main-body-->
```

```
<div id="sidebar-2">
```

```
<div class="inside">
```

```
<?php mosLoadModules('right');?>
```

```
</div></div>
```

```
<div id="footer">
```

```
<?php include_once($mosConfig_absolute_path ./includes/footer.php);?>
```

```
</div>
```

```
</div> <!--end of wrap-->
```

現在我們原始程式碼中的浮動順序是：

```
#content
```

```
#sidebar
```

```
#sidebar-2
```

要知道寬度，我們現在需要做一點數學題。我們都希望每個邊寬為 25%，#sidebar-2 很簡單，它將是 25% 的邊寬。

然而#sidebar 需要一個基於其在<div>內部一個 75% 的寬度來定義。它的寬度要為 33%。

所以， $33\% \text{ of } 75\% = 25\%$

#content 的寬度將成為剩餘部分。我們會把它設置成 66%。最後的 1% 我們會在#content 和#sidebar 之間分割。

CSS 會是：

```
#wrap {width:80%;}
```

```
#header {}
```

```
#footer {
```

```
clear:both;
```

```
}
```

```
#main-body {
```

```
float:left;
```

```
width:75%;
```

```
}
```

```
#sidebar-2 {
```

```
float:right;
```

```
width:25%;
```

```
}
```

```
#content {
```

```
float:right;
```

```
width:66.5%;
```

```
}
```

```
#sidebar {
```



```
float:left;
width:33.5%;
}
.inside {
padding:10px;
}
```

一些 CSS 設計師會建議通過讓方柱略微小一點來建立一個很小的 gutter。這有助於佈局防止在 Internet Explorer 中被破壞。如果你希望很這樣做，只要修改#sidebar-2 的寬度為 24%。

範本的代碼如下顯示。它在一個滾動框中，所以你可以拷貝並將其粘貼到 index.php 中。注意我們已經將 CSS 佈局從頭部刪除。我們要把它放到一個單獨的檔中。

```
<?php defined( '_VALID_MOS' ) or die( 'Direct Access to this location is not allowed.' ); ?>
```

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
```

```
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

```
<html xmlns="http://www.w3.org/1999/xhtml" lang="<?php echo _LANGUAGE; ?>"
```

```
xml:lang="<?php echo _LANGUAGE; ?>"
```

```
<head>
```

```
<meta http-equiv="Content-Type" content="text/html; <?php echo _ISO; ?>" />
```

```
<?php
```

```
if ($my->id) {
```

```
initEditor();
```

```
}
```

```
?>
```

```
<?php mosShowHead(); ?>
```

```
<script type="text/javascript"> </script>
```

```
<!--http://www.bluerobot.com/web/css/fouc.asp-->
```

```
<link href="templates/<?php echo $cur_template; ?>/css/template_css.css" rel="stylesheet"
```

```
type="text/css" media="screen" />
```

```
</head>
```

```
<body>
```

```
<div id="wrap">
```

```
<div id="header">
```

```
<?php echo $mosConfig_sitename; ?>
```

```
<?php mospathway() ?>
```

```
</div>
```

```
<div id="main-body">
```

```
<div id="content">
```

```
<div class="inside">
```

```
<?php mosLoadModules('top');?>
```

```
<?php mosMainBody(); ?>
```

```
</div></div>
```

```
<div id="sidebar">
```

```
<div class="inside">
```

```
<?php mosLoadModules('left');?>
```

```
</div></div>
```

```
</div> <!--end of main-body-->
```

```
<div id="sidebar-2">
```

```
<div class="inside">
```

```
<?php mosLoadModules('right');?>
```

```
</div></div>
```

```
<div id="footer">
```

```
<?php include_once($mosConfig_absolute_path './includes/footer.php');?>
```

```
</div>
```

```
</div> <!--end of wrap-->
```

```
</body>
</html>
```

```
{mostitle=The Default Joomla CSS}
The Default Joomla CSS
```

目前所有我們的 CSS 都只是關於佈局的。這將製作一個既漂亮又簡單的頁面，所以讓我們添加一些格式吧。我們還會把所有 index.php 標題中的 CSS 移出到 CSS 文件中。

儘管這樣會減慢你網站的速度一點點，但是讓 CSS 含有子檔將會更靈活，然後通過 `template_css.css` 全部載入，一個例子看起來像這樣：

```
/*Compass Design template CSS file*/

@import url("layout.css"); /*layout css file*/

/*
@import url("color.css"); color css file*/

@import url("customize.css"); /*Use this file to customize your website*/
```

之前提到過，我們使用 `@import` 是因為 Netscape 4 不理解這個命令。同時它也不理解 CSS，因此它將視我們未設計格式的內容為一個文本流覽器。

所有與佈局相關的 CSS 會在 `layout.css` 文件中。一旦設置好了，你只能離開它，並要知道你對其它樣式表所作的任何改變不會有任何 drastic。Color.css 檔可能包含了與顏色相關的任意資訊（它被指出與這裡顯示的一樣）。你可以快速簡單地做一些修改或者設置“color packs”。最後，我們所有的排版和 Joomla 樣式將放在我們的 `customize.css` 文件中。

我們的 `layout.css` 檔現在是：

```
/*Compass Design layout.css CSS file*/
body {
text-align:center; /*center hack*/
}
#wrap {
width:80%; /*center hack*/
margin:0 auto; /*center hack*/
text-align:left;
}
#header {
text-align:left;
}
#footer {
clear:both;
```

```

}
#main-body {
float:left;
width:75%;
}
#sidebar-2 {
float:right;
width:25%;
overflow:hidden;
margin-left:-3px;
}
#content {
float:right;
width:66.5%;
overflow:hidden;
}
#sidebar {
float:left;
width:33.5%;
overflow:hidden;
}
.inside {
padding:10px;
}

```

我們通過使用一個小型的 **hack** 來使頁面集中。由於 **Internet Explorer** 我們不得不這樣做。有標準相容瀏覽器，我們只要頁面空白為：0.10%；要使頁面集中，但是 **IE** 並不識別它。因此我們讓整個頁面的“text”集中然後調整它回到列的左邊。

我們已經添加了兩個規則。一個是 **overflow**：隱藏到每一列。這就使得頁面更頻繁地“break”如果我們減少它的寬度。第二是我們添加了一個導航邊到 **sidebar-2**。這僅僅是給 **Internet Explorer** 用 **CSS** 記錄一些它的問題。我們可以通過添加一個 **hack**（**Tan hack**）應用該規則到 **IE**：

```
* html #sidebar-2 {margin-left:-3px;}
```

然而，**hacks** 通常很麻煩。最好將其應用到所有的瀏覽器（本書作者的觀點），畢竟，它只有 3 圖元。

在 **customize.css** 檔的開頭，我們將設置一些 **overall** 樣式並包含一個名為“**global reset**”的東西。

```

/*Compass Design Customize.css file */
* {
margin:0;

padding:0;

}

```

```

h1,h2,h3,h4,h5,h6,p,blockquote,form,label,ul,ol,dl,fieldset,address {
margin: 0.5em 0;
}
li,dd {
margin-left:1em;
}
fieldset {
padding:.5em;
}
body {
font-size:76.1%;
font-family:Verdana, Arial, Helvetica, sans-serif;
line-height:1.3em;
}
#header {
background:#0099FF;
}
#footer {
background:#0099FF;
}
#main-body {
background: #CC0000;
}
#sidebar-2 {
background:#009933;
}
#content {
background: #999999;
}
#sidebar {
background: #009933;
}

```

所有的東西都只給出了 0 邊，填充，然後所有 **block** 級別的元素給出了一個底部邊。這有助於實現瀏覽器的一致性。你可以閱讀更多在 [clagnut](#) 中與 [global reset](#) 和 [左對齊](#) 相關的資訊。

字體大小設置成 76.1%。原因是要嘗試並獲得整個瀏覽器更多一致的字體大小。所有字體大小設置為 **em**。包含線高：1.3em 有助於閱讀。這就意味著頁面將更容易訪問，瀏覽器也會據自己的愛好來縮放字體。這在以下討論更多：一個在 [The Noodle Incident](#) (Owen Briggs) 上的排版方面的例子

最後，我們添加了一些背景顏色，這樣就可以看到列在哪了。

有一個新的預設 Joomla!1.0.8 的安裝，範本現在看起來像這樣：

注意邊列並不到達它們的注腳。這是因為它們只延伸到內容，即左邊紅色右邊白色區域，它們並不存在。如果我們有一個範本，該範本的三個列都是白色背景，這沒有問題。我們將使用這個方法並讓 **box** 圍繞元件周圍。如果相同

高度的列需要著色或者需要 box，那麼你必要使用一個垂直鋪開的背景圖片。這種技術叫做"Faux Columns"，是由 Douglas Bowman and Eric Meyer 描述的。

[TO DO: DESCRIPTION OF FAUX COLUMNS HERE]

不幸的是，該技術在 Internet Explorer 上引起了一些問題。在有些環境中，列背景會消失。這就是有名的"Peekaboo bug"，在 Position Is Everything 中有詳細的描述。這是通過使用 HOLLY HACK 修補的（在 IE 中分配一個 1% 的高度）。這裡，稍微做了點修改，因此只有 IE6 旨在使用一個!Important 聲明。這就意味著沒有實際的 hack，使用的是無效的 i.e.CSS.

```
#wrap{  
border:1px solid #999;  
background: url(..images/threecol-r.gif) repeat-y 75% 0;  
height:100% !Important;height:1%;  
}
```

```
#wrap-inner {  
background: url(..images/threecol-l.gif) repeat-y 25.125% 0;  
height:100% !Important;height:1%;  
}
```

注意在 Internet Explorer 很小的螢幕寬度上，佈局開始破壞。很有可能通過 hacking 一個最低的高度來修補，但是我們將把這個問題留給設計者作為練習。

Joomla Specific CSS

在編寫期間。當前穩定的 Joomla 是 1.0.X 系列。該發行仍然使用重要的表格來輸出內容到主體中。緊跟著這些表格的是 CSS 輸出，可用于設計師設計樣式頁面。基於各種社區成員的一些調查，當前清單顯示如下。注意它並不包含一般的網頁樣式如 H1,H2,p,u,l,a,form 等。

```
#active_menu  
#blockrandom  
#contact_email_copy  
#contact_text  
#emailForm  
#mod_login_password  
#mod_login_remember  
#mod_login_username  
#poll  
#search_ordering  
#search_searchword  
#searchphraseall  
#searchphraseany  
#searchphraseexact  
#voteid1,#voteid2....  
.adminform  
.article_seperator
```

.back_button
.blog
.blog_more
.blogsection
.button
.buttonheading
.category
.clr
.componentheading
.contact_email
.content_rating
.content_vote
.contentdescription
.contentheading
.contentpagetitle
.contentpane
.contentpaneopen
.contenttoc
.createdate
.fase4rdf
.footer
.frontpageheader
.inputbox
.latestnews
.mainlevel
.message
.modifydate
.module
.moduletable
.mostread
.newsfeed
.newsfeeddate
.newsfeedheading
.pagenav
.pagenav_next
.pagenav_prev
.pagenavbar
.pagenavcounter
.pathway
.polls
.pollsborder
.pollstableborder
.readon
.search

.searchintro
.sectionentry1
.sectionentry2
.sectionheader
.sitetitle
.small
.smalldark
.sublevel
.syndicate
.syndicate_text
.text_area
.toclink
.weblinks
.wrapper

關於這個清單重要的筆記。

很多你可能看到的設計實際上已經給出 CSS 樣式，該樣式在他們的定義中更規範。基本上，一個更規範的規則將優先於一個不規範的規則。

例如：

```
a {color:blue;}  
a.link {color:red;}  
.contentheading {color:blue;}  
div.contentheading {color:red;}
```

一個連結的顏色和 .Contentheading 的顏色是紅色，是由於規則更規範（.Contentheading 是包含在一個<div>中的）

在 Joomla 範本的案例中，你將看到所使用的更規範的規則。這經常在當類在表中時發生。更多例子：

```
.moduletable  
table.moduletable
```

- .moduletable 是<div>的名字，它包含了一個元件。table.moduletable 將只把樣式應用到一個表中，該表有 class="moduletable"在上面

- .moduletable 將使用該樣式，不管類處於什麼元素上。

```
a.contentpagetitle:link  
.contentpagetitle a:link
```

- a.contentpagetitle:link 將把樣式應用到任意一個標記，a.contentpagetitle 類在上面即一個連結。

- .contentpagetitle a:link 將把樣式應用到在.contentpagetitle 內部的任意元素即一個連結。

規範性並不容易理解，它往往更容易使用最常見的樣式開始，如果結果不是你想要的，它就會變得更規範。

一些關於規範的連結：

<http://www.htmldog.com/guides/cssadvanced/specificity/>

<http://www.meyerweb.com/eric/css/link-specificity.html>

http://www.stuffandnonsense.co.uk/archives/css_specificity_wars.html

目前我們的範本使用了很多表，實際上有 20 張！之前提到，這會減慢頁面運行速度並且使它們很難更新。要減少表格的數量，當我們調用元件時，需要使用准 index.php 中的 \$style 下標。

```
{mospagebreak title=Modules}
```

Modules

當一個元件中 index.php 中被調用時，它有若干個關於如何顯示的選擇。語法是：

```
mosLoadModules('$position_name[, $style] )
```

\$style 是可選的，可以為 0，1，-1，-2，或者-3。

0=（預設顯示）元件可以顯示在一個列中。以下顯示一個輸出例子：

```
<table cellpadding="0" cellspacing="0" class="moduletable">
```

```
<tr>
```

```
<th valign="top">Module Title</th>
```

```
</tr>
```

```
<tr>
```

```
<td>Module output</td>
```

```
</tr>
```

```
</table>
```

1=元件可以水準地顯示。每個元件輸出在意個包裹表的小組中。以下顯示了輸出例子：

```
<!-- Module wrapper -->
```

```
<table cellspacing="1" cellpadding="0" border="0" width="100%">
```

```
<tr>
```

```
<td align="top">
```

```
<table cellpadding="0" cellspacing="0" class="moduletable">
```

```
<tr>
```

```
<th valign="top">Module Title</th>
```

```
</tr>
```

```
<tr>
```

```
<td>
```

```
Module output</td>
```

```
</tr>
```

```
</table>
```

```
</td>
```

```
<td align="top">
```

```
<!-- ...the next module... -->
```

```
</td>
```

```
</tr>
```

```
</table>
```

-1=元件可以顯示作為原始輸出且不帶標題。以下顯示一個輸出例子：

```
Module Output
```

-2=元件可以以 CSS 格式顯示，由<div>括起來。以下顯示一個輸出例子：

```
<div class="moduletable">
```

```
<h3>Module Title</h3>
```

```
Module output
```

```
</div>
```

-3=元件可以一個允許的格式顯示，例如，**stretchable rounded corners**。如果該\$style 被使用，<div>的名字從 moduletable 改變成 module。以下顯示了一個輸出例子：

```
<div class="module">
```

```
<div>
```

```
<div>
```

```
<div>
```

```
<h3>Module Title</h3>
```

```
Module output
```

```
</div>
```

```
</div>
```

```
</div>
```

```
</div>
```

你可以看到 CSS 選項 (-1, -2 和-3) 在代碼中更簡潔，使網頁更容易設置樣式。本教程作者並不建議使用 0 或者 1 的 suffixes 除非明顯需要。

要開發你的範本，我們將把一個 \$style 為-2 的範本應用到所有的範本上。

```
<body>
```

```
<div id="wrap">
```

```
<div id="header">
```

```
<?php echo $mosConfig_sitename; ?>
```

```
<?php mospathway() ?>
```

```
</div>
```

```
<div id="main-body">
```

```
<div id="content">
```

```
<div class="inside">
```

```
<?php mosLoadModules('top',-2);?>
```

```
<?php mosMainBody(); ?>
```

```
</div></div>
```

```
<div id="sidebar">
```

```
<div class="inside">
```

```
<?php mosLoadModules('left',-2);?>
```

```
</div></div>
```

```
</div> <!--end of main-body-->
```

```
<div id="sidebar-2">
```

```
<div class="inside">
```

```
<?php mosLoadModules('right',-2);?>

</div></div>

<div id="footer">

<?php include_once( $mosConfig_absolute_path .'/includes/footer.php' ); ?>

</div>

</div> <!--end of wrap-->
</body>
```

注意我們不能把所有的範本樣式應用到任意以下例子中，因為它們都不是範本。

```
<?php echo $mosConfig_sitename; ?>
<?php mospathway() ?>
<?php mosMainBody(); ?>
<?php include_once( $mosConfig_absolute_path .'/includes/footer.php' ); ?>
```

設置範本到一個 CSS 報告已經減少表格的數量到 14 張了。現在讓我們添加一些簡單的樣式到範本來獲取結果顯示如下：

首先我們將把網站標題置於一個<H1>標記內部。它在語義上更準確且在 SEO 中有幫助。

```
<h1><?php echo $mosConfig_sitename; ?></h1>
```

我們還將添加一些 CSS 來給範本設置樣式，給標題設置邊框和背景。

我們的 customize.css 現在看起來像這樣：

```
/*Compass Design Customize.css file */

* {

margin:0;

padding:0;

}

h1,h2,h3,h4,h5,h6,p,blockquote,form,label,ul,ol,dl,fieldset,address{

margin:0.5em 0;

}
```

```
ul{

margin-left:2em;

}

fieldset{

padding:.5em;

}

body{

font-size:76.1%;

font-family:Verdana,Arial,Helvetica,sans-serif;

line-height:1.3em;

margin:1em 0;

}

#wrap{

border:1px solid #999;

background: url(../images/threecol-r.gif) repeat-y 75% 0;

height:100% !Important;
height:1%;

}

#wrap-inner {

background: url(../images/threecol-l.gif) repeat-y 25.125% 0;

height:100% !Important;
height:1%;

}
```

```
#header{

border-bottom: 1px solid #999;

padding:10px;

}

#footer{

border-top: 1px solid #999;

padding:5px;

}

a{

text-decoration:none;

}

a:hover{

text-decoration:underline;

}

h1,.componentheading{

font-size:1.7em;

line-height:1.7em;

}

h2,.contentheading{

font-size:1.5em;

line-height:1.5em;

}
```

```
h3{

font-size:1.3em;

line-height:1.3em;

}

h4{

font-size:1.2em;

line-height:1.2em;

}

h5{

font-size:1.1em;

line-height:1.1em;

}

h6{

font-size:1em;

line-height:1em;

font-weight:bold;

}

#footer,.small,.createdate,.modifydate,.mosimage_caption{

font:0.8em Arial,Helvetica,sans-serif;

color:#999;

}

.moduletable{
```

```

margin-bottom:1em;

padding:0 10px;
/*padding for inside text*/
border:1px #CCC solid;

}

.moduletable h3{

background:#666;

color:#fff;

padding:0.25em 0;

text-align:center;

font-size:1.1em;

margin:0 -10px 0.5em -10px;
/*negative padding to pull h3 back out from .moduletable padding*/
}
{mospagebreak title=Menus}
Menus

```

設置控制如何輸出一個功能表在出版它的範本中被控制了。當你首先製作一個範本，你將看到一個提示，告訴你有一個相同名字的範本已經創建。

在那個功能表的範本中，有一些針對功能表如何顯示的選項：

-垂直的

功能表顯示為一個垂直的表

-水準的

功能表顯示為一個水準的表

-統一列表

功能表顯示為一個統一的CSS 列表

表格式生成代碼如：

```
<div class="moduletable">
```

```
<h3>Main Menu</h3>
```

```
<table width="100%" border="0" cellpadding="0" cellspacing="0">
```

```
<tr align="left">
```



```
<td>
<a href="index.php?option=com_frontpage&Itemid=1" class="mainlevel" id="active_menu">Home</a>

</td>

</tr>

<tr align="left">

<td>
<a href="index.php?option=com_content&task=view&id=5&Itemid=6"
class="mainlevel" >Joomla! License</a>
</td></tr> <tr align="left"><td><a href="index.php?option=com_content&task=section&id=1&Itemid=2"
class="mainlevel" >News</a>

</td>

</tr>
<tr align="left">
<td>
<a href="index.php?option=com_content&task=blogsection&id=0&Itemid=9"
class="mainlevel" >Blog</a>

</td>

</tr>

<tr align="left">

<td>
<a href="index.php?option=com_weblinks&Itemid=23" class="mainlevel" >Links</a>
</td></tr>
<tr align="left"><td>
<a href="index.php?option=com_contact&Itemid=3" class="mainlevel" >Contact Us</a>

</td>

</tr>
<tr align="left">

<td>
<a href="index.php?option=com_search&Itemid=5" class="mainlevel" >Search</a>

</td>
```

</tr>

<tr align="left">

<td>

News Feeds

</td>

</tr>

<tr align="left">

<td>

FAQs

</td>

</tr>

<tr align="left"><td>

Wrapper

</td>

</tr>

</table>

</div>

統一清單將生成(相同的功能表):

<div class="moduletable">

<h3>Main Menu</h3>

<ul id="mainlevel">

Home

Joomla! License

```

<li><a href="index.php?option=com_content&task=section&id=1&Itemid=2"
class="mainlevel" >News</a></li>
<li><a href="index.php?option=com_content&task=blogsection&id=0&Itemid=9"
class="mainlevel" >Blog</a></li>

<li><a href="index.php?option=com_weblinks&Itemid=23" class="mainlevel" >Links</a></li>

<li><a href="index.php?option=com_contact&Itemid=3" class="mainlevel" >Contact Us</a></li>
<li><a href="index.php?option=com_search&Itemid=5" class="mainlevel" >Search</a></li>

<li><a href="index.php?option=com_newsfeeds&Itemid=7" class="mainlevel" >News Feeds</a></li>

<li><a href="index.php?option=com_content&task=category&
sectionid=3&id=7&Itemid=25" class="mainlevel"
>FAQs</a></li>

<li><a href="index.php?option=com_wrapper&Itemid=8" class="mainlevel" > Wrapper</a></li>

</ul>

</div>

```

而且，使用 CSS 清單而不是表格減少了代碼，卻又使標記更簡單了。在設置所有的功能表到統一清單後，我們只剩下 12 張表格了（剩下的在沒有 **hacking** 的情況下不能被刪除）。清單比表格要好 是因為基於文本的瀏覽器，螢幕閱讀器，不支援 CSS 瀏覽器，關閉 CSS 的瀏覽器以及搜索 bot 都將更容易地訪問你的內容。

使用 CSS 做功能表的一個其它的優點是在各種 CSS 設計師網站有很多實例代碼。讓我們來看一個例子並理解它是如何被使用的。

一個在 maxdesign.com 的網頁有一個超過 30 個功能表的選擇，這些功能表都是使用相同的基礎代碼。被稱為 **Listamatic**。這些代碼有輕微的區別，我們必須做一些修改來讓這些功能表適用於 Joomla 。

這些列表使用以下代碼：

```

<div id="navcontainer">

<ul id="navlist">

<li id="active"><a href="#" id="current">Item one</a></li>

<li><a href="#">Item two</a></li>
<li><a href="#">Item three</a></li>

<li><a href="#">Item four</a></li>

```

```
<li><a xhref="#">Item five</a></li>
```

```
</ul>
```

```
</div>
```

這就意味著有一個名為 `navcontainer` 的 `<div>`。 `` 有一個 `navlist` 的 `id`。要複製這個效果到 Joomla 中，我們需要添加一個元件位置，讓我們使用 `user1` 在左模組之前。

```
<div id="sidebar">
```

```
<div class="inside">
```

```
<div id="navcontainer">
```

```
<?php mosLoadModules('user1',-2);?>
```

```
</div>
```

```
<?php mosLoadModules('left',-2);?>
```

```
</div></div>
```

注意我們將 `mostLoadModules` 置於 `<div>` 內部來匹配 `ListaMatic`。這可以是任何名字，除了本教程的目的，它有助於簡單地使用所有這些功能表。

接下來，如果有任何與 `navlist` 相關的，對於 Joomla 將輸出作為 `mainlevel`：

```
<ul id="mainlevel">
```

最後，我們需要添加一個 CSS 模組 `suffix` 到管理後端的 `user1` 模組，這樣模組就能接收唯一的 CSS 樣式。顯然這意味著我們正把自己想要的功能表放在這個位置。

一個 `Module Class Suffix` 可以應用於任意模組。當輸出時，模組 `<div>` 的名字將附加上 `suffix`。

所以在這種情況下：

-如果使用一個-2 樣式，它將是 `.moduletable-leftnav`。

-如果使用一個-3 樣式，它將是 `.module-leftnav`。

這種對模組類 `suffix` 的使用很有幫助。我們將在 `Tips and Tricks` 部分看到它允許不同顏色的箱子，僅僅在模組類 `suffix` 上做了一個簡單的修改。

對於我們的網站，我們將使用 `List 10 by Mark Newhouse`。我們的 CSS 會是：

```
.moduletable-leftnav{
```

```
margin-bottom:1em;
```

```
padding:0; /*the padding is removed so the menu fills the whole module box*/
```

```
border:1px #CCC solid;

}

.moduletable-leftnav h3{

background:#666;

width:100%;
color:#fff;

padding:0.25em 0;

text-align:center;

font-size:1.1em;

margin:0;
/*now we have no padding in the module, we don't need the negative margins*/
border-bottom: 1px solid #CCC;

}

#navcontainer{

padding:0;

color: #333;

}

#navcontainer ul{

list-style: none;

margin: 0;

padding: 0;

}

#navcontainer li{
```

```
border-bottom: 1px solid #ccc;
```

```
margin: 0;
```

```
}
```

```
#navcontainer li a{
```

```
display: block;
```

```
padding: 3px 5px 3px 0.5em;
```

```
border-left: 10px solid #333;
```

```
border-right: 10px solid #9D9D9D;
```

```
background-color:#666;
```

```
color: #fff;
```

```
text-decoration: none;
```

```
}
```

```
html>body #navcontainer li a {
```

```
width: auto;
```

```
}
```

```
#navcontainer li a:hover,a#active_menu:link,a#active_menu:visited{
```

```
border-left: 10px solid #1c64d1;
```

```
border-right: 10px solid #5ba3e0;
```

```
background-color: #2586d7;
```

```
color: #fff;
```

```
}
```

Designer's Tip

當嘗試獲取一個特殊的功能表工作時，這是一個很有用的提示。

創建一個預設的 Joomla 安裝，然後查看主功能表的代碼

拷貝並粘貼該代碼到一個 HTML 編輯器中（比如 Dreamweaver）

用“#”重置所有的連結，然後你可以添加 CSS 規則知道你想要的效果已經看到了。功能表創建樣式的代碼顯示如下：

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />

<title>Untitled Document</title>

<style type="text/css">

<!--
.thisisyourCSS {

/* put a rule here*/

}
-->

</style>

</head>

<body>
<div class="moduletable">

<h3>Main Menu</h3>

<ul id="mainlevel">

<li><a href="#" class="mainlevel" id="active_menu">Home</a></li>

<li><a href="#" class="mainlevel" >Joomla! License</a></li>

<li><a href="#" class="mainlevel" >News</a></li>

<li><a href="#" class="mainlevel" >Blog</a></li>
```

```
<li><a href="#" class="mainlevel" >Links</a></li>

<li><a href="#" class="mainlevel" >Contact Us</a></li>

<li><a href="#" class="mainlevel" >Search</a></li>

<li><a href="#" class="mainlevel" >News Feeds</a></li>

<li><a href="#" class="mainlevel" >FAQs</a></li>

<li><a href="#" class="mainlevel" >Wrapper</a></li>

</ul>

</div>

</body>

</html>
```

注意 CSS 是被嵌入而非連結來使編輯更容易。

```
{mospagebreak title=Hiding Columns}
```

Hiding Columns

到目前為止，我們使佈局經常含有三列。不管這裡面是否有內容。我們希望能自動地“轉動”一個列，或在沒有內容時“折疊”一個列。

最簡單的方式是包含一個很小塊的 PHP 到頁面的<head>中。

```
<?php if ( mosCountModules( 'right' ) <= 0) { ?>
```

```
<style type="text/css" media="screen">
```

```
#main-body {width:100%;}
```

```
#content{width:75%;}
```

```
#sidebar{width:25%;}
```

```
#sidebar-2{display:none;}
```

```
</style>
```

```
<?php } ?>
```


`mosCountModules` 將返回那個位置的模組數。如果該數等於 0 或者小於 0，i.e.，沒有任何東西，樣式規則將被調整。這個 `php` 必須出現在連結到 `template_css.css` 文件那一行的後面。這是因為如果有兩個相同的 CSS 樣式規則，後面的一個將覆蓋所有其它的。

這可以通過一個相似的方法來完成，也就是讓 `if` 聲明載入一個子 CSS 檔。

Hiding Module Code

當創建可折疊欄時，好的做法就是設置模組不被輸出，如果沒有內容的話。如果這一步沒有完成，頁面將有一個空的 `<div>` 在裡面，會導致橫跨瀏覽器問題。

要隱藏空的 `<div>`，使用以下 `if` 聲明：

```
<?php if (mosCountModules('left')) { ?><?php mosLoadModules( 'left', -2 );?>
```

```
<?php } ?>
```

Anything can be placed inside the `if` statement, so we can put our `<div>` there:

```
<?php if (mosCountModules('left')) { ?>
```

```
<div id="sidebar">
```

```
<div class="inside">
```

```
<?php mosLoadModules( 'left', -2 );?>
```

```
</div></div>
```

```
<?php } ?>
```

使用以上代碼，如果沒有東西發佈在左邊，`#sidebar` 將不會輸出。

條件聲明同樣可以使用。我們的 `sidebar` 列還包含 `user1` 在裡面和左邊，因此我們可以包含一個 `OR` 聲明：

```
<?php if (mosCountModules('left') || mosCountModules('user1')) { ?>
```

```
<div id="sidebar">
```

```
<div class="inside">
```

```
<div id="navcontainer">
```

```
<?php mosLoadModules('user1',-2);?>
```

```
</div>
```

```
<?php mosLoadModules('left',-2);?>
```

```
</div></div>
```

```
<?php } ?>
```

所以如果任何資訊發佈在“左邊”或者“user1”，那麼這塊代碼將被輸出。

使用該技術到左右列。我們的 index.php 檔現在看起來如下：

```
<?php defined( '_VALID_MOS' ) or die( 'Direct Access to this location is not allowed.' ); ?>
```

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
```

```
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

```
<html xmlns="http://www.w3.org/1999/xhtml" lang="<?php echo _LANGUAGE; ?>"
```

```
xml:lang="<?php echo _LANGUAGE; ?>">
```

```
<head>
```

```
<meta http-equiv="Content-Type" content="text/html; <?php echo _ISO; ?>" />
```

```
<?php
```

```
if ($my->id) {
```

```
initEditor();
```

```
}
```

```
?>
```

```
<?php mosShowHead(); ?>
```

```
<script type="text/javascript"> </script>
```

```
<!--http://www.bluerobot.com/web/css/fouc.asp-->
```

```
<link href="templates/<?php echo $cur_template; ?>/css/template_css.css" rel="stylesheet"
```

```
type="text/css" media="screen" />
```

```
<?php if ( mosCountModules( 'right' ) <= 0 ) { ?>
```

```
<style type="text/css" media="screen">
```

```
#main-body {width:100%;}
```

```
#content{width:75%;}
```

```
#sidebar{width:25%;
```

```
#wrap{background:none;}
```

```
</style>
```

```
<?php } ?>
```

```
</head>

<body>

<div id="wrap">

<div id="header">

<h1><?php echo $mosConfig_sitename; ?></h1>
<?php mospathway() ?>

</div>

<div id="main-body">
<div id="content">

<div class="inside">

<?php mosLoadModules('top',-2);?>

<?php mosMainBody(); ?>

</div></div>

<?php if (mosCountModules('left') || mosCountModules('user1')) { ?>

<div id="sidebar">

<div class="inside">

<div id="navcontainer">

<?php mosLoadModules('user1',-2);?>

</div>
<?php mosLoadModules('left',-2);?>

</div></div>

<?php } ?>

</div> <!--end of main-body-->
<?php if (mosCountModules('right')) { ?>
```

```

<div id="sidebar-2">

<div class="inside">

<?php mosLoadModules('right',-2);?>

</div></div>

<?php } ?>

<div id="footer">
<?php mosLoadModules(footer,-2);?>
</div>

</div> <!--end of wrap-->

</body>

</html>

```

在這一點上，我還建議對注腳做一些修改。包含的 footer.php 使修改文本更難。我寧願將其放置作為一個模組。甚至有一個模組位置被稱為注腳！！注意在與該嚮導有關的範本檔中，我沒有在可下載的範本中修改該注腳代碼，但是我顯示在這裡給你選項供你選擇。

Designer's Tip

在 Joomla 中，有一些與模組相關的名字：baner,left,right,user1,footer 等。要明白一件很重要的事情，就是名字與任意指定的位置無關。一個模組的位置完全由範本設計師控制，就像我們所看到的一樣。將它們置於一個與其名字連結的位置是常見的，但不是要求。

```
{mospagebreak title=Conclusion}
```

Conclusion

本教程涉及了創建一個 joomla 範本的步驟。範本使用階層式樣式表 (CSS) 來生成一個使用很少表格的佈局。這是一個可取的目標，因為它意味著範本代碼更容易得到 World Wide Web Consortium(W3C)標準的驗證。它還往往載入更快，更容易維持且在搜尋引擎中執行的更好。同時範本對於在 SEO 中優化性能是有序的資源。

```
本文獻必須全文轉載{mospagebreak title=Appendix A:Tips and Tricks}
```

Appendix A: Tips and Tricks

Variable Page Widths, Rounded Corners, Text Resizers and More

Variable Page Widths

在這裡討論的設計是一個流動頁面。它將隨著包含左右 10% 頁面寬度空白的瀏覽器視窗進行縮放。即使我們已經設置列寬為百分比，我們仍然可以通過設置 #width 為一個具體數字來固定頁寬。以這種方式設計意味著更靈活，還能通過只編輯一行代碼來改變頁寬。

當工作于流動頁面設計時，一方面要注意的是圖片。任意一個我們需要延伸的方向，往往需要是一個可以延伸的可

靠的顏色部分。通常圖片的一小塊被分割然後重複。

如同你在以上例子中所看到的，在中間有一個區域可以分級的。CSS 可能看起來像這樣：

```
background:url(..images/FluidImageSlicingMIDDLE.png) 0 0 repeat-x;
```

將這個和使用一張照片的標題相比較：

在這裡沒有什麼區域可以被重複，且這種類型的標題不能運行於一個流動的設計。該原理應用於任意類型的規模，垂直的或者水準的。圖像一定有某個部分可以重複。

讓範本使用基於列寬的百分比還有另外一個優點就是對一個寬度縮放器的使用。通過使用分離的樣式表，很有可能根據使用者的愛好來修改頁寬。你可以在 [Compass Design](#) 中看到該動作。

Rounded Corners

大概最常見的設計，看看目前的模組，就是給箱子使用圓角。當前的標準是使用“-3”模組樣式來生成 3<div>。這三個和主要的.module<div>用來放置箱子周圍的 4 張圖片。

你可以在www.joomla.org 看到該動作。這是 4 張圖片：

- http://www.joomla.org/templates/jw_joomla/images/jos_box_grey_bl.png
- http://www.joomla.org/templates/jw_joomla/images/jos_box_grey_br.png
- http://www.joomla.org/templates/jw_joomla/images/jos_box_grey_tl.png
- http://www.joomla.org/templates/jw_joomla/images/jos_box_grey_tr.png

這是代碼：

```
div.module-grey h3, div.module-table-grey h3 {
```

```
font-family: Helvetica, Arial, sans-serif;
```

```
font-size: 1em;
```

```
font-weight: bold;
```

```
color: #333;
```

```
margin: -2px -8px 0 -8px;
```

```
border-bottom: 1px solid #cdcdd;
```

```
padding-left: 10px;
```

```
padding-bottom: 2px;
```

```
}
```

```
div.module-grey, div.module-table-grey {
```

```
background: url(../images/jos_box_grey_tl.png) 0 0 no-repeat;
```

```
margin: 0;
```

```
padding: 0;
```

```
margin-bottom: 20px;
```

```
}
```

```
div.module-grey div, div.moduletable-grey div {
```

```
background: url(../images/jos_box_grey_tr.png) 100% 0 no-repeat;
```

```
}
```

```
div.module-grey div div, div.moduletable-grey div div {
```

```
background: url(../images/jos_box_grey_bl.png) 0 100% no-repeat;
```

```
}
```

```
div.module-grey div div div, div.moduletable-grey div div div {
```

```
background: url(../images/jos_box_grey_br.png) 100% 100% no-repeat;
```

```
padding: 8px;
```

```
width: auto !important;
```

```
width: 100%;
```

```
}
```

```
div.module-grey ul, div.moduletable-grey ul {
```

```
margin: 10px 0;
```

```
padding-left: 20px;
```

```
}
```

存在很多技術用於生成圓角。兩個都包含這些摘要的網站是cssdiscuss.incutio.com and Smiley Cat.

Fixed Width Boxes with Two Images

首先在 Joomla! 中開發，很有可能用於一個盒子中的圖片數量減半。該技術將給方塊垂直比例而不是水準比例。

使用的兩張圖片是：

以及

用來產生效果的 CSS 是：

```
.moduletable-box {  
  
margin:0 0 10px 0;  
  
padding:0 0 10px 0;  
  
background:url(..images/bottom.png) bottom left no-repeat;  
}  
  
.moduletable-box h3 {  
  
padding:8px 10px 6px 15px;  
  
margin-bottom:8px;  
  
text-align:left;  
  
font:bold 1.1em Arial,Helvetica,sans-serif;  
  
color:#fff;  
  
background:url(..images/top.png) top left no-repeat;  
}
```

這個技術特別有幫助是因為以下幾點：

- 底部箱子圖片可以由不同的顏色重複使用。
- H3 頭部的背景將由文本縮放來分級（joomla.org 技術不是）
- 在代碼中很簡潔
- 一半的圖片以雙倍的速度載入。

它要求模組 \$style suffix 為 -2 以準確運行。

Fluid Width and Height Boxes with Two Images

一個更高級的技術由 Compass Design 開發出來。它要求模組 \$style suffix 為 -3 以正確運行。此外，每一個 <div> 用於放置一個箱子的角，但是只有兩張圖片被使用。箱子的一個模組類 suffix 已經在這裡使用了。

以及

```
.module-box {  
  
background: url(..images/boxright.png) top right no-repeat;
```

```
padding:0;
```

```
margin:0 0 10px 0;
```

```
}
```

```
.module-box h3 {
```

```
margin:0;
```

```
padding:0 0 4px 0;
```

```
border-bottom:#ccc 1px solid;
```

```
color: #666;
```

```
font: bold 1em Tahoma, Arial, Helvetica, sans-serif;
```

```
text-align:center;
```

```
}
```

```
.module-box div {
```

```
background: url(../images/boxleft.png) top left no-repeat;
```

```
margin:0;
```

```
padding:6px 0 0 0;
```

```
}
```

```
.module-box div div{
```

```
background: url(../images/boxleft.png) bottom left no-repeat;
```

```
padding:0 0 0 5px;
```

```
}
```

```
.module-box div div div{
```

```
background: url(../images/boxright.png) bottom right no-repeat;
```



```
padding:0 5px 5px 0;
```

```
height:auto !important;
```

```
height:1%;
```

```
}
```

該技術將以垂直和水準分比例。

幾乎任何來自之前給出的那兩個網站的技術都能起作用。特殊的好處就是 456bereastreet 的靈活的箱子，都有自定的角和邊。這只適用一張圖片，所以要改變顏色很簡單。該技術的實現留下來作為讀者的練習。

Text Resizers

由於現在很多撒小已經變得更容易，所以把文本 resizers 納入設計中。作為一個一般規則要達到最好的跨流覽器一致性，你需要給文本 resizer 按鈕做一些事情以讓其很好地運行：

1. 在主體標記中，定義一個字體大小為 76%

```
body {
```

```
font-size:76%;
```

```
}
```

2. 定義所有的大小在"em"中,這是一個相關的單元，例如：

```
p {
```

```
font-size:1em;
```

```
}
```

3. 使所有的文字方塊都為動態的。該方面的一個例子不能運行是在www.joomla.org。如果你使其變大，橫向功能表容器不變大，如果你使用FF來go past你可以按下按鈕實現的大小（ctrl+在FF中），佈局破壞。你不通過將固定高度置於容器上來取得它，你可以在www.joomlashack.com中看到其運行。

4. 下載 joomla.org 的 font style switcher 文件

(http://forum.joomla.org/Themes/joomla/md_stylechanger.js)

5. 把那個檔放在你正在使用的範本資料夾中。

6. 加上 A+,A-,並重置在你範本圖片資料夾中的圖片。

7. 粘貼以下代碼片段到你範本的 index.php 文件中。

```
<script type="text/javascript" language="javascript" xsrc="<?php echo $mosConfig_live_site;?>/templates/<?php echo $mainframe->getTemplate(); ?>/__1__">
```

```
</script>
```

```
<a href="index.php" title="Increase size" onclick="changeFontSize(1);return false;">
```

```

```

```
</a>
```

```
<a href="index.php" title="Decrease size" onclick="changeFontSize(-1);return false;">
```

```

```

```
</a>
```

```
<a xhref="index.php" title="Revert styles to default" onclick="revertStyles(); return false;">

</a>
```

8. 做以下練習：

- Replace ___1___ with the location in your template folder where you saved the .js file
- Replace ___2___ with the name of your A+ image
- Replace ___3___ with the name of your A- image
- Replace ___4___ with the name of your Reset image

很感謝 r0tt3n 和他在 Joomla 論壇的 FAQ 對於該技術的部分。

Drop Down Menus

由於種種原因，其中的 SEO 和無障礙環境，作者不偏向 flash 和 JavaScript。更是強調 W3C 驗證代碼和簡潔頁面，沒有一個是由這兩種方法幫助的。又如何實現重複這些效果的功能表呢，比如一個下拉式功能表。

然而，有很多技術你可以與 CSS 一起使用來取得視覺上更吸引人的功能表。它們都使用無序列表（符號清單）來創建功能表。讓我們看一下，一個下拉式功能表。該功能表就是被創造為“suckerfish”的功能表，它是純 CSS 的，很精簡，自由的 hack 並且用於 IE 只有 12 行 JavaScript 語言。

你可以看一個 demo：

www.html5dog.com/articles/suckerfish/dropdowns/example/

你可以找出這些東西如何在兩個網站上運行的指南：

www.html5dog.com/articles/suckerfish/dropdowns/

www.alistapart.com/articles/dropdowns/

現在，你可能已經注意到你需要輸出你的功能表作為一個很好的精簡的列表。它就是這樣發生的，有一個模組做這個工作，而我們將需要它。它被稱為 `extend_menu`，你可以在這裡找出：

de.siteof.de/extended-menu.html

如果你給它一個功能表和模組類 suffix，將會最簡單。我給兩個都是用“mainnav”(你將在以下的 CSS 中看到)。你將需要其它兩個設置：

-菜單類型：Tree List

-延伸菜單：Yes

```
.moduletablemainnav{
```

```
position:relative;
```

```
z-index:100;
```

```
font:0.9em Verdana, Arial, Helvetica, sans-serif;
```

```
margin:0;
```

```
padding:0;
```

```
}
```

```
#mainlevelmainnav,#mainlevelmainnav ul{
```

```
float:left;
```

```
list-style:none;
```

```
line-height:1em;
```

```
background:transparent;
```

```
font-weight:700;
```

```
margin:0;
```

```
padding:0;
```

```
}
```

```
#mainlevelmainnav a{
```

```
display:block;
```

```
color:#f90;
```

```
text-decoration:none;
```

```
margin-right:15px;
```

```
padding:0.3em;
```

```
}
```

```
#mainlevelmainnav li{
```

```
float:left;
```

padding:0;

}

#mainlevelmainnav li ul{

position:absolute;

left:-999em;

height:auto;

width:11em;

font-weight:400;

background:#36f;

border:#00C 1px solid;

margin:0;

}

#mainlevelmainnav li li{

width:11em;

}

#mainlevelmainnav li ul a{

width:11em;

color:#fff;

font-size:0.9em;

line-height:1em;

font-weight:400;

}

```
#mainlevelmainnav li:hover ul,#mainlevelmainnav li:hover ul ul,
#mainlevelmainnav li.sfhover ul ul,#mainlevelmainnav li.sfhover ul ul ul{
left:-999em;

}
```

```
#mainlevelmainnav li:hover ul,#mainlevelmainnav li li:hover ul,
#mainlevelmainnav li li li:hover ul,#mainlevelmainnav li.sfhover ul,
#mainlevelmainnav li li.sfhover ul,#mainlevelmainnav li li li.sfhover ul{

left:auto;

z-index:6000;

}
```

```
#mainlevelmainnav li li:hover,#mainlevelmainnav li li.sfhover{

background:#039 url(../images/soccerball.gif) 98% 50% no-repeat;

}
```

現在，只要確保你把 z-indexes 設置正確了，且記住有一個 z-index,元素需要一些位置類型，如果不明顯就要相關的。

最後，但不是至少，你需要給 IE 添加 JavaScript 到範本 index.php 文件的頭部（或者以個 js 文件），並不說明：準確地 hover 。

```
<script type="text/javascript"><!--><![CDATA[//><!--
sfHover = function() {
var sfEls = document.getElementById("mainlevelmainnav").getElementsByTagName("LI");
for (var i=0; i<sfEls.length; i++) {
sfEls[i].onmouseover=function() {
this.className+=" sfhover";
}
sfEls[i].onmouseout=function() {
this.className=this.className.replace(new RegExp(" sfhover\b"), "");
}
}
}
if (window.attachEvent) window.attachEvent("onload", sfHover);
//--><![></script>
```

該技術在 Joomla 論壇中將進一步討論。

作者，Barrie North在www.compassdesigns.net中創建了 Joomla Templates and Tutorials。本作品在Creative Commons Attribution NonCommercial ShareAlike 2.5 License得到許可。

