

# Drupal 精华教程集 2

cck 的工作是集合需要的 fields 之后, 集中到作为一个 node 处理

而 contemplate, template, css 就会负责显示 node 部分

至于 contemplate 和 template 之间,

contemplate 和 template 都对某一个指定的 content type 起作用

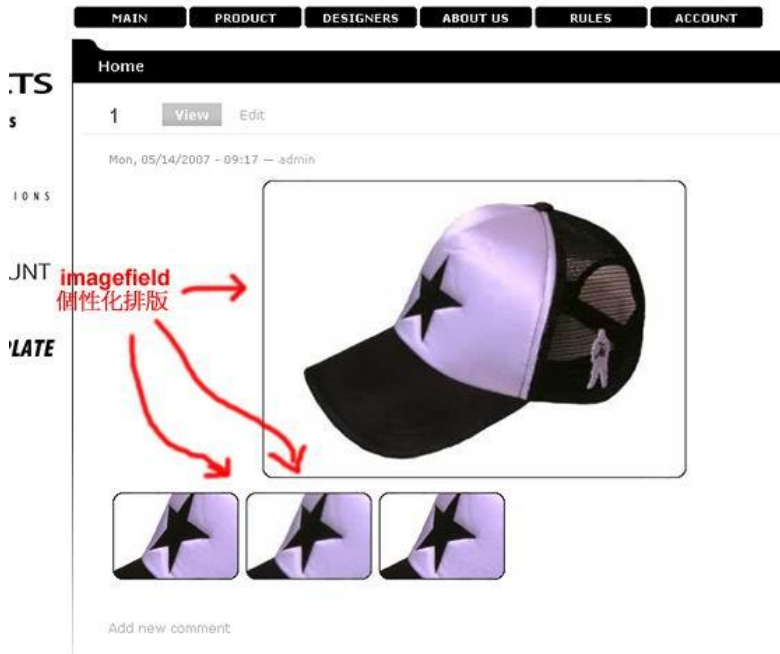
他们的分别, 在于被修改的对象

contemplate 只可修改 cck 新增的 fields

例如 imagefield, textarea, user\_ref, node\_ref.....

改动什麼都可以, 加减 tags, class, id, fields 之间的排位

分成多少行, 两排, 三排等等(图一)



但不可以改动原有的 node title, author, submit time,

body, comment 的出现位置之类的部分

这些都留给了 template 负责

node.tpl.php 中应该就 submitted time, content, node title 等排版



设定相关的 class, id, tags.

又, 要注意,

以上的这些改变都对用 views 做的 table, list view 的情况完全没有影响

full node, teaser view 等便有影响

contemplate 的设定介面分成三个群组

teaser, body, RSS (图中红框)

teaser 是撮要, 一般的首页都使用撮要的显示

虽然撮要的内容可能和正文一样

题外话:

Drupal 内建的撮要机制令撮要独立的放到一个 field 中处理

所以当储存 node 的时候没有设定撮要,

则及后再开撮要都没有显示撮要

contemplate 可以对 teaser, body, RSS 作特定的显示方式

而修改的方法则完全一样

左面的窗口是输出的 php, html 代码 (图中蓝框)

右面为一些可用的 php 变数 (图中绿框)

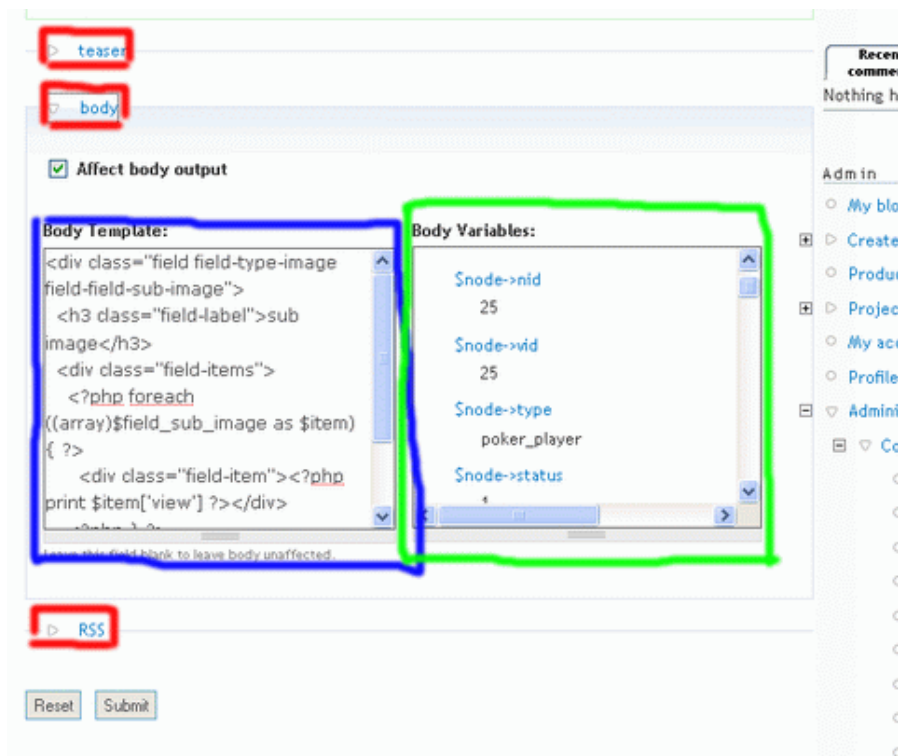
先按 Affect body output 激活左面窗口

然后随意修改 html, php

再在右面的窗口按你想加入的变数, 如 \$node->title, \$node->type 等

相关的 code 就会加到左面的窗口了

按 submit 就大功告成了



\*\*\*\*\*END\*\*\*\*\*

## CCK 图像字段:Imagefield

**简介:** Image field 为 CCK 提供一个图像字段。**特性:**views 整合 多图片支持 图片大小限制  
 imagecache 整合 //Imagefield 不提供缩略图或派生图生成, Views 可以使用 imagecache 生成  
 派生的 imagefield 图片, 但 imagecache 配置起来可能困难重重, 而且文档也不齐全。

imagecache 配合 imagefield 使用方法

第一步

在 imagecache 设置中, 新建两个 presets

第一个 (正文中的图)

namespace: yourcontentnode\_mainimage (文件名, 随便选, 最后图片会在这个目录下生成)

action: scale

width: 400 (根据你的喜好)

height: 400 (根据你的喜好)

scale to fit: inside dimensions

weight: 1

第二个 (摘要中的图)

namespace: yourcontentnode\_thumbnail

action: scale

width: 100

height: 100

scale to fit: inside dimensions

weight: 0



项目	
<a href="#">列表</a>	<a href="#">添加术语</a>
名称	操作
个人贷款系统	<a href="#">编辑</a>
地方财政系统	<a href="#">编辑</a>
<b>重要客户系统</b> <small>~ http://mysite/taxonomy/term/3</small>	<a href="#">编辑</a>

用 **Taxonomy Redirect** 模块定制分类的对应 Link:好处是我可以尽情地定制分类的显示方式(^\_^)。

<input type="checkbox"/>	项目	taxonomy_redirect	taxonomy	project/term/.
--------------------------	----	-------------------	----------	----------------

用 **Panels** 模块定制页面

项目分类	project/term	<a href="#">编辑</a>   <a href="#">删除</a>
项目管理	project	<a href="#">编辑</a>   <a href="#">删除</a>

先说一说 项目管理 **Project** 这个页面: 这个页面没有什么特别, 其中包含了一个 **View** 用来显示 **Taxonomy Redirect** 定制后的项目分类 Link。

项目分类

- 类别: **重要客户系统** ~ http://mysite/project/term/3  
成员: 白云涛  
重要客户详细信息说明
- 白云涛
- 类别: **地方财政系统**  
成员: 白云涛  
地方财政系统详细信息说明
- 白云涛
- 类别: **个人贷款系统**  
个人贷款系统详细信息

再说说 项目分类 **Project/term** 这个页面: 这个页面包含了四个 **View** 分别对应 项目信息、项目合同、培训信息、设备信息。在这 4 个 **View** 中需要设置一下 **Arguments** 为对应的分类 ID

Arguments

Argument Type: 默认

Taxonomy: Term ID: **Display All Values**

The argument will filter by a taxonomy term ID. F

在 Panels 中设置对应 View 中的 View 参数



好了，现在看一看结果：



再看看相关的合同文书 Node 中的表现



现在，基本的关联已经搞定了，再来看看相关区块的实现

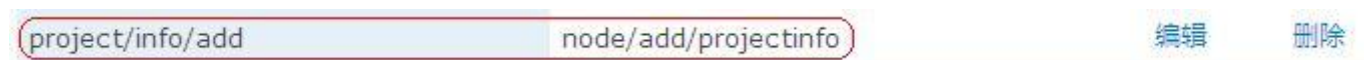


对区块的基本要求是：应只在与项目管理和项目信息有关的页面中显示。

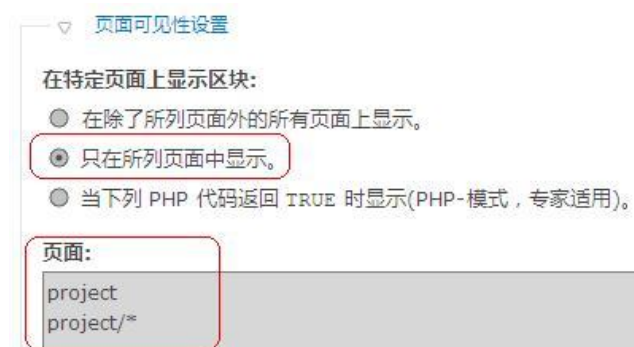
使用 Pathauto 模块定制 项目信息 node 的 Link



在“URL 别名”中设定



好了，现在所有“项目管理”与“项目信息”相关的页面都有一个共同点了。都采用 [http://mysite/project/\\*](http://mysite/project/*) 这种格式的 Link。现在我们建立一个“项目管理”的菜单。在“区块”管理中找到对应的区块，将它显示在侧边栏中。进入“项目管理”区块的配置：



也可根据用户角色来配置



OK，完成，现在“项目管理”区块已经可以根据我们的要求在特定的页面显示了。



### 后续工作

最基本的工作已经完成，接下来的工作就是细化与美工了。用 CCK 详细定制“项目信息”、“合同文书”、“培训信息”等内容类型的各种字段。

用 Views Theme wizard 定制和美化各种相关 Views 的表现形式。

用 主题模版 定制和美化 Panels 生成的各种页面。

……这里就不一一说了。^\_^

END



## Views 2 教程：创建一个显示相关网页的区块

当我们阅读一个节点时，希望能把相关文章的标题也显示在一个区块中，利用 Views 2 for Drupal 6 很容易就可以做到这一点。下面这个教程演示了实现这个功能的步骤，同时也可以帮助我们熟悉 Views 2 的操作界面。Views 2 进行了很大的改版，使用界面与 Views 1 也有很大的不同。

### 安装并启用模块

Views 模块的主页在这里：<http://drupal.org/project/views>

下载 Views 2 for Drupal 6，安装并启用模块，对于 Views 2 至少要启用 Views 和 Views UI 两个模块，Views 只提供了 API，而 Views UI 则提供了操作界面，如下图所示：

Enabled	Name	Version	Description
<input checked="" type="checkbox"/>	Views	6.x-2.2	Create customized lists and queries. Required by: Panels (disabled), Views (disabled), Mini panels (disabled)
<input type="checkbox"/>	Views exporter	6.x-2.2	Allows exporting multiple views. Depends on: Views (disabled)
<input checked="" type="checkbox"/>	Views UI	6.x-2.2	Administrative interface to views. Depends on: Views (disabled)

### 添加一个新的 View

下面我们添加一个新的 View，在导航菜单中选择 Administer > Site building > Views，点击 add 选项卡，View name 填入"relative\_articles"，View description 填入"相关文章"，在 View type 中选择"node"，如下图所示：

Home > Administer > Site building > Views

Views [List](#) [Add](#) [Import](#) [Tools](#)

**View name: \***  
  
This is the unique name of the view. It must contain only alphanumeric characters and underscores. It is used to generate the themeing template names for this view. If overriding a module provided view, the name must be prefixed with the module name.

**View description:**  
  
This description will appear on the Views administrative UI to tell you what this view does.

**View tag:**  
  
Enter an optional tag for this view; it is used only to help sort views.

**View type:**

- Node  
Nodes are a Drupal site's primary content.
- Comment  
Comments are responses to node content.
- File  
Files maintained by Drupal and various modules.
- Node revision

### 编辑新的 View

点击下面的 "Next" 按钮，进入 Views 的编辑界面，这个 UI 与前任的 Views 1 截然不同，需要使用一段时间才能熟悉：

View *relative\_articles*, displaying items of type Node. [Export](#) [Clone](#) [New view](#)

**Defaults** ▸ Defaults *Default settings for this view.*

**View settings**  
Tag: None

**Basic settings**  
Name: Defaults  
Title: None  
Style: Unformatted ✖  
Row style: Fields ✖  
Use AJAX: No  
Use pager: No  
Items to display: 10  
More link: No  
Distinct: No  
Access: Unrestricted  
Exposed form in block: No  
Header: None  
Footer: None  
Empty text: None  
Theme: Information

**Relationships** + ↕  
None defined

**Arguments** + ↕  
None defined

**Fields** + ↕  
None defined

**Sort criteria** + ↕  
None defined

**Filters** + ↕  
None defined

Click on an item to edit that item's details.

### 详细设置

接下来，我们对 Views 进行一些详细的设置，Title 设置为“相关文章”，Style 设置为“HTML List”，Row Style 保持原来的“Fields”，Use AJAX 设置为“Yes”，Item to display 设置为“5”，More Link 设置为“Yes”，如下图所示：

View *relative\_articles*, displaying items of type Node.

**Defaults** ▸ Defaults *Default settings for this view.*

**View settings**  
Tag: None

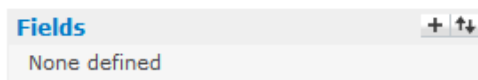
**Basic settings**  
Name: Defaults  
**Title: 相关文章**  
Style: HTML List ✖  
Row style: Fields ✖  
Use AJAX: Yes  
Use pager: No  
Items to display: 5  
More link: Yes  
Distinct: No  
Access: Unrestricted  
Exposed form in block: No  
Header: None  
Footer: None  
Empty text: None  
Theme: Information

**Defaults: The title of this view**  
相关文章  
This title will be displayed with the view, when

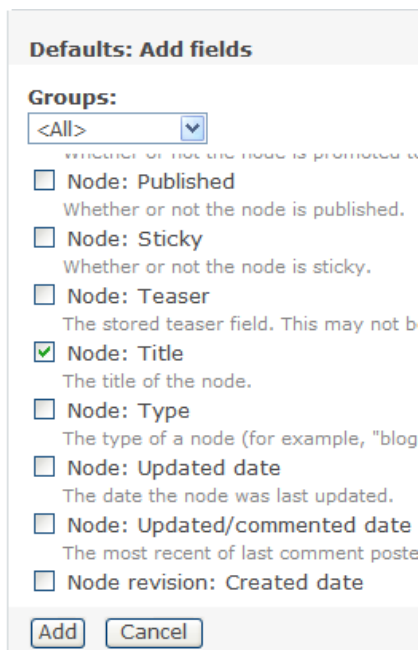
接着，我们点击左下角的“Save”按钮，但是，这时候 Drupal 会给出一个错误提示：

Display "Defaults" uses fields but there are none defined for it or all are excluded.

由于我们把 Row Style 设置成了“Fields”，但是我们还没有给出可用的 Field，所以才有这个警告出现。接下来，我们定义一个 Field，点击 Fields 旁边的 + 号：



下面会出现 Fields 详细设置表单，因为我们只想在区块中显示文章的标题，因此，我们只定义一个 Node Title 就可以了：

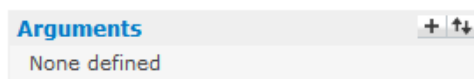


点击 add 按钮，进入 Title 的详细配置界面，清除标签中的“Title”字样，因为我们只想显示标题本身，而不是显示成“Title: 这是一个标题”。选中“Link this field to its node”，然后点击“Update”按钮，如下图：



### 设置 Arguments

要想显示相关文章，关键在于 Arguments 的设置，我们希望找出正在阅读的节点的 Taxonomy items，然后选出具有同样 taxonomy items 的文章，点击 Arguments 旁边的“+”号：



下面会出现 Arguments 的设置界面，我们选中“Taxonomy: Term ID”做为参数，然后点击“add”按钮：

**Defaults: Add arguments**

**Groups:**

<All> ▾

Taxonomy: Term  
Taxonomy term name.

Taxonomy: Term ID  
The taxonomy term ID

Taxonomy: Term ID (with depth)  
The depth filter is more complex, so prov

Taxonomy: Term ID depth modifier  
Allows the "depth" for Taxonomy: Term I

Taxonomy: Term synonym  
Term synonyms may be used to find terr

Taxonomy: Vocabulary ID  
The taxonomy vocabulary ID

User: Name  
The user or author name.

User: Roles

接下来在出现的 Arguments 详细设置界面中，在 Action to take if argument is not present 中，选中“Provide default argument”，然后在 Default argument type 中选中“PHP code”，并填入如下代码：

```
$node = node_load(arg(1));
if ($node && $node->taxonomy){
    foreach($node->taxonomy as $term) {
        $terms[]=$term->tid;
    }
    return implode('+',$terms);
} else {
    return;
}
```

如下图所示：

**Defaults: Configure Argument *Taxonomy: Term ID***

**Title:**

The title to use when this argument is present. It will override the title of the view and titles from substitution here to replace with argument titles. Use "%1" for the first argument, "%2" for the se

**Action to take if argument is not present:**

- Display all values
- Hide view / Page not found (404)
- Display empty text
- Summary, sorted ascending
- Summary, sorted descending
- Provide default argument

**Wildcard:**

If this value is received ignored; i.e, "all values"

**Wildcard title:**

The title to use for the w

[Provide default argument options](#)

**Default argument type:**

- Fixed entry
- Node ID from URL
- PHP Code
- User ID from URL
- User ID from logged in user

**PHP argument code:**

```
$node = node_load(arg(1));
if ($node && $node->taxonomy){
foreach($node->taxonomy as $term){$terms[]=$term->tid;}
return implode('+',$terms);
}else{return;}

```

在下面的选项中选中“Allow multiple terms per argument.”和“Reduce duplicates”，这样允许参数为“1+2+3”这样的形式，同时消除重复的节点，完成后点击“Update”按钮，如下图所示：

**Allow multiple terms per argument.**  
 If selected, users can enter multiple arguments in the form

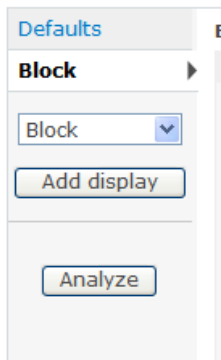
**Allow multiple arguments to work together.**  
 If selected, multiple instances of this argument can work together. This setting is not compatible with the "Reduce duplicates" setting.

**Do not display items with no value in summary**

**Reduce duplicates**  
 This filter can cause items that have more than one of the same value to occur, this checkbox can reduce those duplicates; however, use this with caution.

**Set the breadcrumb for the term parents**  
 If selected, the breadcrumb trail will include all parent term IDs received.

现在，我们创建一个区块，以便以后在 **Administor > Site building > Blocks** 中把它放置在适当的位置，如下图，在 **Views** 编辑界面的左上角，下拉菜单中选中 **block**，然后点击“Add display”按钮，如下图所示：



现在大功告成了，把这个区块放在 **Left Sidebar** 或 **Right Sidebar** 上，当我们浏览文章时，相关文章就会显示在这个区块中。不过还有点小问题，我们当前正在浏览的节点也会显示在区块中，怎么才能把它去掉呢？

Administer > Site building > Views，在下面的列表中找到我们的 `relative_articles`，点击“Edit”，进入编辑界面，再添加一个 Arguments，类型为“Node: Nid”，选择“Provide default argument”，Default argument type: 设置为“Node ID from URL”，最后选中下面的“Exclude the argument”，如下图所示：

**Defaults: Configure Argument Node: Nid**

**Title:**  
  
The title to use when this argument is present. It will override the title substitution here to replace with argument titles. Use "%1" for the first.

**Action to take if argument is not present:**

- Display all values
- Hide view / Page not found (404)
- Display empty text
- Summary, sorted ascending
- Summary, sorted descending
- Provide default argument

**Provide default argument options**

**Default argument type:**

- Fixed entry
- Node ID from URL
- PHP Code
- User ID from URL
- User ID from logged in user

**Validator options**

**Validator:**  
<Basic validation>

**Action to take if argument does not validate:**  
Hide view / Page not found (404)

Allow multiple terms per argument.  
If selected, users can enter multiple arguments in the form of 1+2+3

Exclude the argument  
If selected, the numbers entered in the argument will be excluded rather than summed.

怎么样，这回可以了吧？

----- E N D -----

## IEWS 實作公告+分類

首頁

## 所辦公告

[公告一覽](#)
[公告](#)
[徵才](#)
[徵稿](#)
[新聞](#)
[網站](#)

所有最新哲學所辦公室發佈公告消息

類別	標題	期限	發佈時間
公告	97學年度研究生顏鳳論文大綱口試時間表		4週6日前
徵才	國立清華大學哲學研究所 徵聘專任教師啟事		6週12小時前
甄試	哲學研究所98學年度 碩士班甄試 正式榜單		10週12小時前
甄試	哲學研究所98學年度甄試初試錄取名單暨複試流程表		12週3日前

根據您的貼圖個人估計是在同一個 content type 下給定一個特定的分類 (Taxonomy)，再使用 Views 來進行整理與顯示。作法如下：

1.先新增一個 Content type-Name:公告 Type:announcement。其他關於 content type 的設定，暫不贅述。

Home > Administer > Content management

公告 [Edit](#) [Manage fields](#) [Display fields](#)

Identification

**Name:** \*

公告

The human-readable name of this content type. This text will be displayed with a capital letter and contain only letters, numbers, and spaces. This name must be unique.

**Type:** \*

announcement

The machine-readable name of this content type. This text will be used to identify the content type. It must contain only lowercase letters, numbers, and underscores. Underscores must be unique.

2.新增一個分類 Add vocabulary

Home > Administer > Content management

Taxonomy [List](#) [Add vocabulary](#)

[View help](#)

Name	Type
Forums	Forum topic

Home > Administer > Content management > Taxonomy

**Taxonomy**   List   [Add vocabulary](#)

[View help](#)

▼ Identification

**Vocabulary name:** \*

公告類別

The name for this vocabulary, e.g., "Tags".

**Description:**

Description of the vocabulary; can be used by modules

**Help text:**

Instructions to present to the user when selecting term

▼ Content types

**Content types:**

- Blog entry
- Book page
- Event
- Forum topic
- Page
- Poll
- Story
- 公告

### 3. 新增公告類別下的 terms:

**Taxonomy**   [List](#)   [Add vocabulary](#)

Created new vocabulary 公告類別.

[View help](#)

Name	Type	Operations	
+	Forums	Forum topic	<a href="#">edit vocabulary</a> <a href="#">list terms</a> <a href="#">add terms</a>
+	公告類別	公告	<a href="#">edit vocabulary</a> <a href="#">list terms</a> <a href="#">add terms</a>

[Save](#)

Home > Administer > Content management > Taxonomy

**Edit term**

▼ Identification

**Term name:** \*

公告

The name of this term.

**Description:**

A description of the term. To be displayed on taxonom feeds.

— ▷ [Advanced options](#)

[Save](#)   [Delete](#)



Home > Administer > Content management > Taxonomy

Terms in 公告類別 [List](#) [Add term](#)

Updated term 甄試.

[View help](#)

Name	Operations
<a href="#">+</a> 公告	<a href="#">edit</a>
<a href="#">+</a> 徵才	<a href="#">edit</a>
<a href="#">+</a> 甄試	<a href="#">edit</a>

4.好了，現在讓我們新增一個"公告"的內容：您可以發現多了一個欄位如下圖：

Home > Create content

Create 公告

Title: \*

公告類別:

公告  
- None selected -  
公告  
徵才  
甄試

請填入一些資料，接下來進行 Views 的部份

細節請容我不多說了，貼幾張圖，請您參考參考，請特別注意"HR"這個 page. 需要多加一個 **Filter**，就是選擇分類中屬於徵才的分類。

#### 1.新增一個 View

Home > Administer > Site building > Views

Views [List](#) [Add](#) [Import](#)

View name: \*

announcement

This is the unique name of the view. It must be unique across all template names for this view. If overriding a

View description:

announcement

This description will appear on the Views admin page.

View tag:

announcement

Enter an optional tag for this view; it is used to group views.

View type:

Node  
Nodes are a Drupal site's primary content.

Comment

## 2.Views 中的設定如下三個圖

Home > Administer > Site building > Views

**Edit view *announcement*** List Add **Edit** Import Tools

View *announcement*, displaying items Export Clone View "list\_all" View "HR" of type **Node**.

**Defaults**

list\_all  
HR

Page

**Defaults** *Default settings for this view.*

**View settings**  
Tag: announcement

**Basic settings**  
Name: Defaults  
Title: None  
Style: Table   
Use AJAX: Yes  
Use pager: Yes  
Items per page: 20  
More link: No  
Distinct: No  
Access: Unrestricted  
Link display: list\_all  
Exposed form in block: No  
Header: None  
Footer: None  
Empty text: None  
Theme: Information

**Relationships** + ↑ ↓  
None defined

**Arguments** + ↑ ↓  
None defined

**Fields** + ↑ ↓  
Taxonomy: Term 類別  
Node: Title 標題  
Node: Post date 發佈時間

**Sort criteria** + ↑ ↓  
Node: Post date desc

**Filters** + ↑ ↓  
Node: Published Yes  
Node: Type = 公告

Home > Administer > Site building > Views

**Edit view *announcement*** List Add **Edit** Import Tools

View *announcement*, displaying items Export Clone View "list\_all" View "HR" of type **Node**.

Defaults

**list\_all**

HR

Page

**list\_all** *Display the view as a page, with a URL and menu links.*

**Basic settings**  
Name: list\_all  
Title: None  
Style: Table   
Use AJAX: Yes  
Use pager: Yes  
Items per page: 20  
Distinct: No  
Access: Unrestricted  
Exposed form in block: No  
Header: None  
Footer: None  
Empty text: None  
Theme: Information

**Page settings**  
Path: announcement/lis...  
Menu: Tab: 公告一

**Relationships**  
None defined

**Arguments**  
None defined

**Fields**  
Taxonomy: Term 類別  
Node: Title 標題  
Node: Post date 發佈時間

**Sort criteria**  
Node: Post date desc

**Filters**  
Node: Published Yes  
Node: Type = 公告

Home > Administer > Site building > Views

## Edit view *announcement*

List Add **Edit** Import Tools

View *announcement*, displaying items of type **Node**. Export Clone View "list\_all" View "HR"

Defaults

list\_all

**HR**

Page v

Add display

Analyze

**HR**

*Display the view as a page, with a URL and menu links.*

**Basic settings**

Name: HR

Title: None

Style: Table \*

Use AJAX: Yes

Use pager: Yes

Items per page: 20

Distinct: No

Access: Unrestricted

Exposed form in block: No

Header: None

Footer: None

Empty text: None

Theme: Information

**Page settings**

Path: announcement/hr

Menu: Tab: 徵才

Remove display

**Relationships**

None defined

**Arguments**

None defined

**Fields**

Taxonomy: Term 類別

Node: Title 標題

Node: Post date 發佈時間

**Sort criteria**

Node: Post date desc

**Filters** + ↕

Node: Published Yes

Node: Type = 公告

Taxonomy: Term = 徵才 \*

*Click on an item to edit that item's details.*

3.效果圖如下:

Home

公告一覽 徵才

類別	標題	發佈時間
甄試	哲學研究所98學年度碩士班甄試 正式榜單	36 min 16 sec ago
徵才	國立清華大學哲學研究所 徵聘專任教師啟事	37 min 13 sec ago
公告	97學年度研究生願願論文大綱口試時間表	38 min 18 sec ago

Home > 公告一覽

公告一覽 徵才

類別	標題	發佈時間
徵才	國立清華大學哲學研究所 徵聘專任教師啟事	40 min 13 sec ago

呼~~~，不知道您是否改了。

真的很謝謝你們的幫助

然後有照步驟做過一次~=\_="（汗）

列表 Add 編輯 匯入 Tools

View announcement, displaying items of type 內容節點. 輸出 Clone View "list\_all" View "HR"

Defaults

- list\_all
- HR

頁面 [v]  
Add display  
Analyze

Defaults Default settings for this view.

View settings  
Tag: announcement

基本設置  
名稱: Defaults  
標題: None  
Style: 表格  
Use AJAX: 是  
Use pager: Mini  
Items per page: 20  
More link: 否  
Distinct: 否  
Access: Unrestricted  
Link display: list\_all  
Exposed form in block: 否  
頁首: None  
頁尾: None  
Empty text: None  
Theme: 檔資訊

Relationships + ↑ ↓  
None defined

Sort criteria + ↑ ↓  
None defined

Arguments + ↑ ↓  
None defined

Fields + ↑ ↓  
分類: Term Term  
內容節點: 標題 標題  
內容節點: Post date Post date

過濾器 + ↑ ↓  
內容節點: 已發表 是  
內容節點: 類型 = 公告

Click on an item to edit that item's details.

儲存 取消 刪除

Live preview

列表 Add 編輯 匯入 Tools

View announcement, displaying items of type 內容節點. 輸出 Clone View "list\_all" View "HR"

Defaults

- list\_all
- HR

頁面 [v]  
Add display  
Analyze

list\_all Display the view as a page, with a URL and menu links. Remove display

基本設置  
名稱: list\_all  
標題: None  
Style: 表格  
Use AJAX: 是  
Use pager: Mini  
Items per page: 20  
Distinct: 否  
Access: Unrestricted  
Exposed form in block: 否  
頁首: None  
頁尾: None  
Empty text: None  
Theme: 檔資訊

Relationships + ↑ ↓  
None defined

Sort criteria + ↑ ↓  
None defined

Arguments + ↑ ↓  
None defined

Fields + ↑ ↓  
分類: Term Term  
內容節點: 標題 標題  
內容節點: Post date Post date

過濾器 + ↑ ↓  
內容節點: 已發表 是  
內容節點: 類型 = 公告

Page settings  
路徑:  
announcement/lis...  
選單: Tab: 公告一覽..

Click on an item to edit that item's details.

儲存 取消 刪除

Live preview

列表 Add 編輯 匯入 Tools

View announcement, displaying items of type 內容節點. 輸出 Clone View "list\_all" View "HR"

Defaults  
list\_all  
HR

頁面  
Add display  
Analyze

HR Display the view as a page, with a URL and menu links. Remove display

**基本設置**

名稱: HR  
標題: None  
Style: 表格  
Use AJAX: 是  
Use pager: Mini  
Items per page: 20  
Distinct: 否  
Access: Unrestricted  
Exposed form in block: 否  
頁首: None  
頁尾: None  
Empty text: None  
Theme: 檔案資訊

**Relationships**  
None defined

**Arguments**  
None defined

**Fields**  
分類: Term Term  
內容節點: 標題 標題  
內容節點: Post date Post date

**Sort criteria**  
None defined

**過濾器**  
內容節點: 已發表 是  
內容節點: 類型 = 公告

Page settings  
路徑: announcement/hr  
選單: Tab: 徵才

Click on an item to edit that item's details.

儲存 取消 刪除

Live preview

Display: Arguments:  
Defaults 預覽  
Separate arguments with a / as though they were a URL path.

您是指 tab 的問題吧，還是請您在看看之前的文章喔。

另外，hr 的 page 中的"過濾器"少個條件。

- 1.滑鼠先點擊過濾器，告訴 Views 這個頁面的過濾器要與 Defaults 不同。
- 2.然後，新增一個過濾條件。

六, 2009-01-31 13:23 — thomasfan



Re: 實作公告+分類

可以使用 Views 來產生您所需要的下拉式選單

首先，到 Views 中編輯要產生下拉式選單的 View 或 page 中。

- 2.到過濾器的方塊中，選取您將允許使用者使用的選項

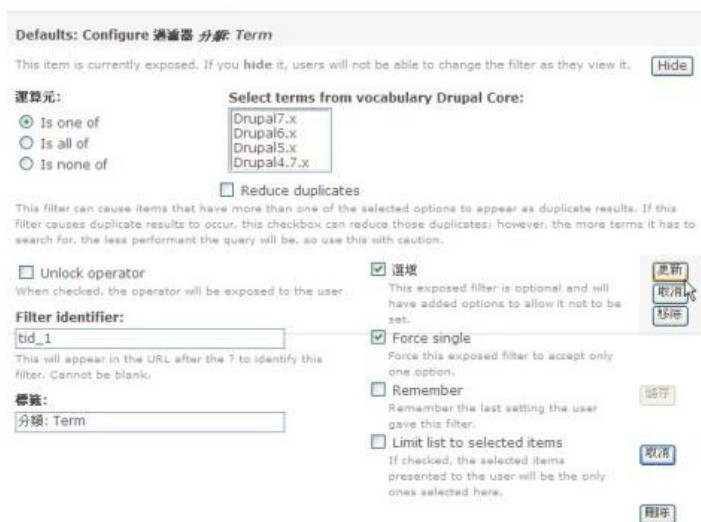
**過濾器** + ↑ ↓

內容節點: 類型 = Modules  
introduction  
內容節點: 已發表 是  
分類: Term exposed  
分類: Term exposed

3. 如下圖中，有個"開放使用"的按鈕，表明允許使用者能透過下拉式選單或文字框的方式來進行內容篩選的功能。



按下按鈕後，會出現如下的畫面



4. 如果您想要讓這個選單能夠以區塊(block)的方式呈現在每個區域(region)中，您只需要在 Basic settings 方塊下的 Exposed form in block 選項中選擇"是"，就可以了。最後，請您到管理 > 網站設置 > 區塊 下，將這個區塊放置到您所想要置放的區域中即可。



\*\*\*\*\*END\*\*\*\*\*

## Views 2 與 Tab

以下將以一個簡單的例子介紹：如何使用 Views 模組，在單一頁面中，以 tab 的方式顯示多頁。  
效果如下：



由於本次針對性比較強，所以請原諒我在此不一一介紹 Views 的操作，您可以參考方先生的教學影片。  
首先，我們先增一個 View，並且產生將要顯示的頁面，如下圖的：myProject, allProject



此例中，我們希望 myProject 與 allProject 能用 tab 放在同一頁。所以，在路徑上我們必須先設定一個主路徑（Project），然後再根據個別頁面給你子路徑(my)。

列表 Add 编辑 导入 Tools

View Project\_List, displaying items of type 节点. 导出 Clone View "myProject" View "allProject"

Defaults

myProject Display the view as a page, with a URL and menu links. Remove display

myProject

allProject

页面

Add display Analyze

**Basic settings**

名称: myProject  
标题: 我的项目  
Style: 表格  
Use AJAX: 是  
Use pager: 是  
Items per page: 10  
Distinct: 否  
Access: Multiple roles  
Exposed form in block: 否  
标题: None  
风格: None  
Empty text: Filtered HTML  
Theme: Information

**Relationships**

None defined

**Arguments**

None defined

**Sort criteria**

None defined

**过滤器**

用户: Current 是  
节点: 已发表 是  
节点: 类型 = 项目

**Fields**

节点: 标题 项目名称  
节点: 正文 项目说明  
内容: 客户名称 -  
(field\_workfor) Title (link)  
个人资料: 姓名 创建者  
节点: Post date 创建日期

**Page settings**

路径: Project/my  
菜单: Tab: 我的项目

**myProject: The menu path or URL of this view**

http://ims.dingwow.com.cn/Project/my

This view will be displayed by visiting this path on your site. You may use "%" in your URL to represent values that will be used for arguments: For example, "node/%/feed".

更新 取消

**Page settings**

路径: Project/my  
菜单: Tab: 我的项目

**myProject: Default tab options**

When providing a menu item as a tab, Drupal needs to know what the parent menu item of that tab will be. Sometimes the parent will need to have one created. The path of a parent menu item should be the path to the parent menu item with the last part left off. i.e, if the path to the parent menu item would be *foo/bar*.

**Parent menu item:**

**标题:**

我的项目

If creating a parent menu item,

Already exists

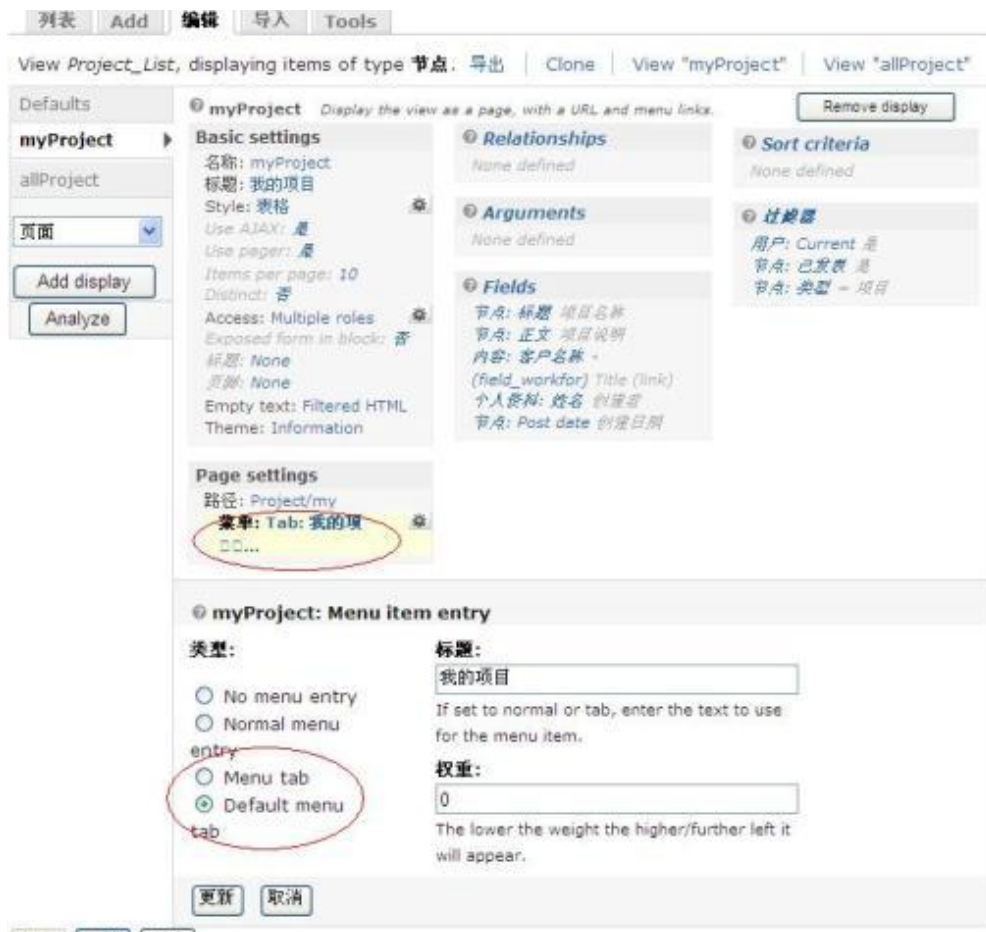
Normal menu item

Menu tab

更新 取消



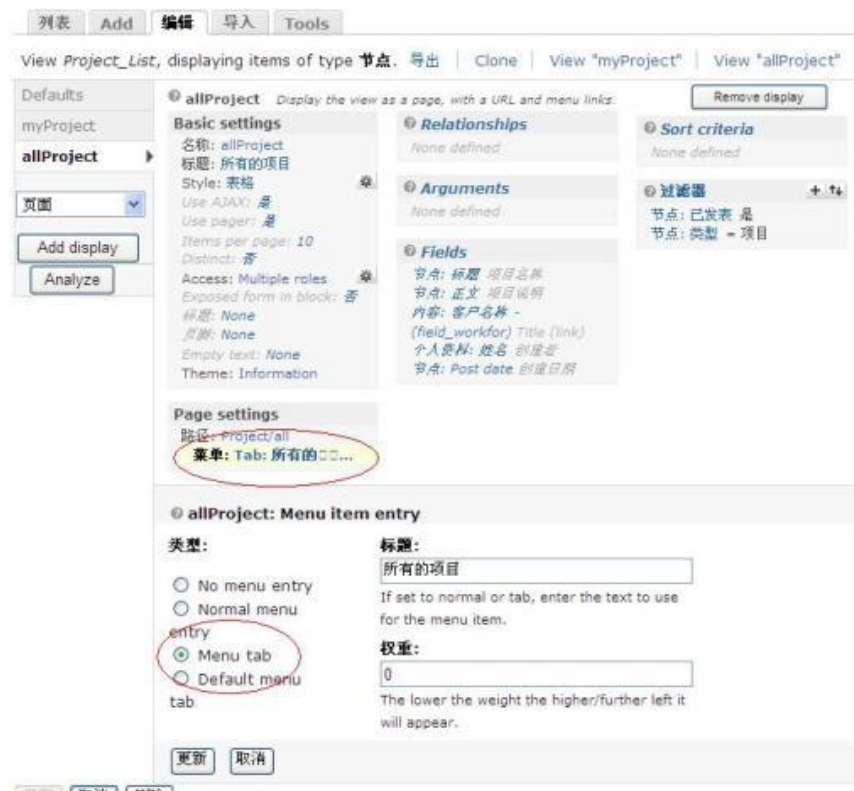
接下來，就要注意菜單選項的部份了。一個頁面中我們必須選擇一個頁面為初始頁面，因此在 Page settings > Menu item entry 中，要選擇 "Default menu tab"



接下來開始設定 allProject 這一頁面，一樣的，我們將目光放在 "Page settings" 這個方塊中。路徑的部份如上頁一樣，主路徑 ("Project")，次路徑 ("all")。



然後，就是設定這一頁的菜單選項了，請選擇"Menu tab"。



即便是您有第三，四...頁，作法都是一樣的。

如有錯誤請指點，如果問題歡迎交流討論。

相關模組：

Views 2

\*\*\*\*\*END\*\*\*\*\*

## Drupal 项目实战：公司员工订餐系统（一）

### 简单的需求

我们要做的是公司内部的一个员工订餐网站，它的主要功能是：

1. 员工可以浏览各种菜的信息：为了简单，我们先假设可以浏览的是“套餐”
2. 员工在选择好菜后，就可以下订单：订单中可以选择订菜的数量
3. 饭店可以查看员工所下订单：并根据订单进行送餐，订单的有效期是当日上午 10:00（假设我们订的是午餐）
4. 菜单的主要属性：标题、说明、照片、价格

OK. 非常简单易懂，基本上三个需求。这个系统中，主要的角色是：

- 公司员工
- 饭店
- 订餐管理员

我们要做的就是实现这样一个系统。首先快速做出原型，然后逐步迭代，即采用敏捷开发[1]方法。可以发挥同学的想像力，逐步完善。

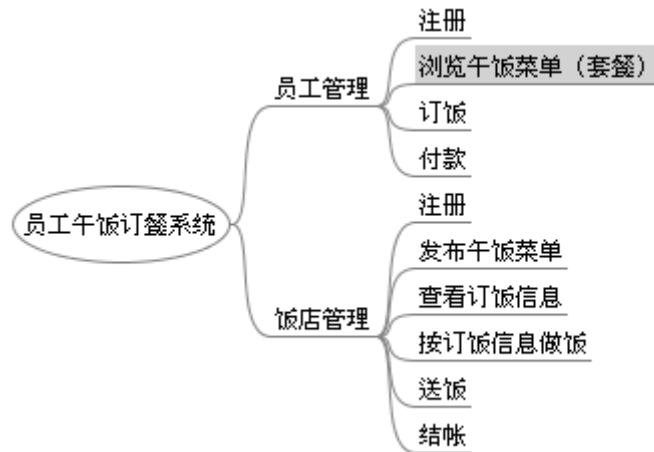


图 1 员工午饭订餐系统结构图（用户视角）

## 客户

在了解了需求后，我们可以对系统的功能进行分析，以便进行技术架构的分析，进而估计出，要实现这样一个系统，大概要多少成本。这是一个很现实的问题。在实际的项目中，客户一般只关心花多少钱来完成这样一个项目，至于用哪种技术，用不用 Drupal，其实他们并不关心。因此，某些开发者有时过多的向客户介绍 Drupal 的优势，其实是完全没有必要的。只需要让客户了解我们采用了“先进的”、“高效的”、“便于扩展”的技术架构就可以了。但是，对于开发者来说，技术框架是非常重要的，因为这样我们才能准确的估计出是否能顺利完成项目，以及其开发成本。

## UML

我个人是比较主张使用 UML 的。因为 UML 的用例图，状态图和时序图等，可以很好很方便的对系统进行建模，而且便于和世界各地的用户交流。这一点很重要，因为如果我们的客户和我们使用的不是同一种语言时，这种图型就成为了很好的交流媒介。

## 分析

我们力求简化开发过程，不必要的就直接省略。对于这个项目，我认为接下来应该分析如何用 Drupal 来实现。

首先，要找出系统中的“元数据”。因为 Drupal 是 CMS，且核心就是 Node。那么分析系统时，首先就是找出系统中的“Node”，即“元数据”。从面向对象的角度看，就是“类”。我认为这是面向数据和面向对象分析的综合运用。

菜单应该是系统中的“元数据”，因为所有的行为都是围绕这个进行的。菜的主要属性已经在需求中说明了，那么我们可以马上联想到，“菜单”即是 Node 的子类，是 Node 的继承，扩展 Node 的属性（因为 Node 只有标题和内容两个内容属性），就可以完成“菜单”的管理。我们把“菜单”类称为 Food。

除了 Food，还有一个类是“订单（Order）”。订单用于记录员工的订餐情况。它也是一个“元数据（meta-data）”。那么它是否也可以继承 Node 类呢？我们知道，使用 Drupal 的一个原则，就是尽可能使用 Drupal 本身的模块功能来完成业务需求。简单的讲，能少使用模块就少使用模块。现在有两个第三方模块可以实现电子商务的功能：**e-Commerce** 和 **Ubercart**。



前者历史比较悠久，在 Drupal 5 时代非常流行，不过进入 Drupal 6 时代就暂停开发了。听说最近又重新开始开发了，估计作者前一段时间持观望态度。Ubercart 是后起之秀，它也是以“商品”为主的电子商务网站的解决方案。而且 Ubercart 也于最近推出了 Drupal 6 的 beta 版本。我没有对这两个插件作过横向对比，刚好我们借这个实战项目，也可以进行一下实验。不过，在使用这两个模块之前，我想先使用“纯 Drupal”来实现我们的功能（尝试），如果能实现，那么效率应该还是比较高的，而且更简单。

### 预计要使用的模块

有人说，Drupal 开发就是“拼模块”，此话有一定道理。而且大多数人在用 Drupal 开发时，马上头脑中就无数个模块翻腾。其实这不是一个很好的现象，很容易在还没真正弄清楚需求前就陷入了技术误区。当然，我们还是需要在开始前设想一下需要使用哪些模块，因为这样可以提前下载下来，做下尝试。不过我更倾向于：边做边找的模式。也许这不是最佳实践，不过对于比较小的项目还是很有效率的。

- CCK、Views、Panels：这三个模块基本没什么悬念，任何一个网站几乎都会用到。Panels 可能会损伤性能，不过它可以让我们避免陷入枯燥的 CSS 定位调试中。
- Image Field：用于添加 Image 的 CCK Field
- Date：Date 模块主要是一些日期相关的功能，也用于添加 Date 类型的 CCK Field
- Vote API：这个模块是用于评论时投票的，与 FiveStar 等模块连用，可以实现“小星星评分”，可爱又实用。
- Webform：在“下订单”时，我们有可能用到，先列于此。
- 自定义模块：如果自己开发模块能够高效的解决问题，那么自定义模块也必不可少。

### 主题模板

Drupal 中的主题模板(Theme)是表示层的术语，除了样式、色彩等，我们还要对内容的显示布局。默认的节点显示页面只适合简单的内容类型，如果我们使用了 CCK 扩展节点字段的话，那么就需要自定义模板文件了。本项目对网站风格的要求是简洁，同时为了开发方便考虑，先使用 Garland（Drupal 默认模板）主题，然后再进行定制。

### 工具

我喜欢使用 EditPlus 作为开发 PHP 的工具，然后配以 Firefox+Firebug 和 IE+Developer Tools（或 IE8）进行调试。在工具方面，仁者见仁，智者见智，没有好坏之分，只有熟练与否。Choose whatever you like.

同时，我发现微软的 Visual Web Developer Express 是一个非常不错的编辑器，尤其是网页开发，可用于页面设计，有兴趣的朋友们也可以试一试。

### 小结

本文叙述了项目的需求，并进行了分析和设计。采用 Drupal 的开发与普通的软件开发还是有区别的，因为我们是“站在 Drupal 的肩膀”上，因此对软件开发过程可以做一定的“剪裁”。至于我的方法是不是“Best Practice”，那还需要大家一起共同探讨了。

详细的开发流程从第二篇开始。

### Drupal 项目实战：公司员工订餐系统（二）

#### 创建菜单节点

首先创建菜单（Food）的节点类型，增加几个字段后，效果见下图。由于属基本操作，就不再赘述了。

Food			
Label	Name	Type	Weight
title	title	title	-5
body	body_filter	body_filter	0
有效日期	field_effective_date	Date	1
预览图	field_food_pic	Image	2
单价	field_price	Decimal	3

在完成菜单类型的创建后，发现默认的节点显示页面，实在是比较简陋。这需要我们自己动手来完善一下。在开始之前，必须介绍一个很重要的模块—Devel (<http://drupal.org/project/devel>)。此模块为开发者必备，主要功能有：

查看节点的变量

清空缓存（模块的）

生成测试数据（节点等）

我常用的是这几个，对于 D6，它还可以调试主题模板，比较有效。不过，这个模块有时会用其它模块的 Javascript 脚本有冲突，目前还不太明确与哪些模块有冲突。如果朋友们在调试 Drupal 时发现莫名其妙的 JS 问题，可以考虑先关闭 Devel 模块，然后再试。

开发主题，当然还需要能方便查看和动态更改 CSS 的浏览器支持。Firefox+Firebug 组合当仁不让成为现时的首选。不过目前，IE8 和 Chrome 也正努力赶上。IE8 中按 F12 可以调出和 IE Developer Tools 相似的 HTML,CSS 代码调试框，基本上 Firebug 功能类似。Chrome 目前可以定位 HTML 页面元素，以及查看其 CSS 样式，不过还不能即时修改。选择哪个就看大家自己的喜好了。一般情况下，我们都要调试 Firefox 和 IE 下的兼容性。尤其是 Div+CSS 的布局，如果没有很高的技巧，确实需要调试很长时间。

#### 创建菜单显示模板

在开始创建之前，当然要进行设计。我们的项目目前没“网页设计”，那我们就直接挑一个差不多的作参考吧。如果你的项目组中有美工人员，这部分应该交给他/她去做了。对于 Drupal 页面的设计，我的意见是：尽量符合 Drupal 默认的页面布局，这样交给 Drupal 开发者时，只需要做少量修改即可。否则就需要花很多时间在主题和模板开发上。毕竟 CSS 技巧高的又懂 Drupal 的同志，还是不大容易找到的。

我看中了一个国外网站的“菜单”信息的显示页面，分享一下：

<http://www.foodnetwork.com/recipes/paula-deen/christmas-ham-recipe/index.html>。截图如下。



## Christmas Ham

Recipe courtesy Paula Deen, 2007

Show: [Paula's Party](#) • Episode: [Holiday Party](#)

[RECIPE](#)
[COMMENTS & REVIEWS \(31\)](#)


31 Reviews

### Cook Time

40 min

### Level

Easy

### Yield

8 servings

### Ingredients

- 1 butt [end](#) smoked ham, about 7 pounds
- 1/2 cup Dijon mustard
- 1/2 cup orange marmalade
- 1 teaspoon ground black pepper
- 3 cans [crescent rolls](#)
- Egg wash

### Recipe Tools:

- [Print Recipe](#)
- [Get Card](#)
- [Save Recipe](#)

### All About...

- [Ham](#)
- [Christmas](#)
- [American](#)
- [Oranges](#)
- [More Topics](#)

### Directions

Preheat oven to 350 degrees F.

Place the ham in a roasting pan and lightly score it with a sharp

对于 Drupal 项目，我认为首先应该开发针对不同的节点类型的页面的显示模板，即：`node-content_type.tpl.php`。它是显示的模板，它包含节点的各个元素的显示方式的设定（如标题、内容、以及用 CCK 生成的字段等），但不对整个页面的布局进行设定。此处需要提一个需要注意的问题：通过 Devel 的"Dev load"查看的变量，并不是在 `node.tpl.php` 中直接可用的变量。它显示的是通过 `node_load` 函数得到的 `$node` 节点，与 `node.tpl.php` 中的 `$node` 节点还是有区别的。同时，大家在使用节点变量时，也可以参考一下 `$node->content` 变量中的内容，它是经过 Drupal 处理后的 HTML 内容。

关于 Drupal 主题模板开发的顺序，下面是我的看法：

开发节点类型的模板：`node-content_type.tpl.php`

开发评论的模板：`comment.tpl.php`

开发普通页面的模板：`page.tpl.php`

开发首页模板：`page-front.tpl.php`

整理 CSS 类别和区块

制作全站的 CSS，写入 `style.css`，或创建单独的样式表文件，在 `style.css` 中 `import`。

为生产站点作 CSS 优化：如使用 Drupal 的 CSS Aggregation 将 CSS 文件整合为一个文件或重新设置每页加载的 CSS 文件。

我喜欢的开发顺序是：先把所有的 HTML 写出，然后创建好每个 CSS 类和 ID 等，最后再进行“上色”。流程如下所示：

原始页面

## 台东坡套餐

[View](#)[Dev load](#)

Mon, 12/15/2008 - 21:41 — blade

这可是非常有趣的一道菜

**有效日期:**

12/15/2008 21:41

**预览图:**



**单价:**

12元

[Add new comment](#)

第一次布局修改:  
分左中右三栏  
左侧为图像, 中间为  
基本信息, 右侧为评  
论。

## 台东坡套餐

[View](#)[Dev load](#)

套餐名称: 台东坡套餐

提供时间: 2008

单价: 12元

[Add new comment](#)

第二次布局修改  
加入样式

采用 table 作布局

## 台东坡套餐

[View](#)[Dev load](#)[Dev render](#)[Edit](#)

套餐名称: [台东坡套餐](#)

种类:

提供时间: [2008-12-15 09:41:00](#)

单价: [12元](#)

### 简介:

东坡肉，杭州一道传统名菜，相传为北宋诗人苏东坡所创制。苏东坡《食猪肉》诗云：“……慢着火，少着水，火候足时他自美。”东坡肉色、香、味俱佳，深受人们喜爱。既然如此，就赶快来看看这样的美味是怎么做的吧，下面给大家介绍的东坡肉做法，不仅有传统做法、微波炉做法，还有素东坡肉的做法，最后再给大家奉上关于东坡肉的来历故事。

做成像 [cnbeta.com](#) 那样的评论

Drupal 默认的评论在最下方，我想把它放在页面的右侧，这样用户打开这个页面，无需浏览到底部，就可以看到最新的评论。[cnbeta.com](#) 的形式也不错，它将“支持”者多的评论，放置在右侧。下面我们看看在 D 中怎样实现。

汉王科技明年花1亿推广电子书阅读器

上一篇新闻: 美打造怪异蜘蛛船可连续航行8000公里 下一篇新闻: 巨星列农借数字技术“复活”为O...

汉王科技明年花1亿推广电子书阅读器

alect发布于 2008-12-28 11:57:36 | 3752 次阅读 字体: 大小 打印 预览

12月28日消息,“我看好电纸书这个产品,一心扎下去了”汉王科技董事长刘迎建在汉王电纸书新品广州发布会表示。这位出生中科院的IT老兵无比看好这款产品的未来。据了解,电子书概念目前正在欧美掀起热潮,被认为是类似“iPod”的下一个热点数码产品,吸引了无数风险投资商的介入。

面对如此大的数码金矿,汉王科技在国内市场抢推了电纸书产品汉王电纸书N516,这一5寸...

热门评论

匿名人士 发表于 2008-12-28 12:00:27

楼主装B中。

回复 支持(107) 反对(9) 举报

匿名人士 发表于 2008-12-28 12:06:07

这里指的电子书是指以“电子纸”为屏幕的电子阅读器产品,跟以LCD为屏幕的Mp4、手机等介质不同。

回复 支持(56) 反对(1) 举报

按正常思路,评论就应该在 `page.tpl.php` 中设置。但是,在 `page.tpl.php` 中只能找到 `$content` 变量。因此,将评论更换显示位置就显得有点“棘手”了。我目前的解决方法是:使用 Views 创建某个节点的评论列表,然后创建自定义模块,调用 `views_build_view` 函数,将列表以区块的形式,放置在右侧区域。那么,这部分应该以区块形式存在,放置在 `right` 区域,也无须对 `node.tpl.php` 进行修改了。

(未完待续)

\*\*\*\*\* E N D \*\*\*\*\*

## [Drupal 模块-CCK 和 Views 的使用](#)

网站要求介绍

我们现在假设有一位客户 D 先生,他需要创建一个网站,这个网站是一个招聘类的网站,网站的管理员可以通过后台发布招聘信息,每一条招聘信息包括应聘职位、工作部门、工作类型、联系发布者、专业类型、待遇如何、对应聘者有何要求等等,这些招聘信息能被所有网站访客查看,同时,所有这些招聘信息还以工作部门进行了归类。



另外，在管理能发布招聘信息的同时，网站的注册会员还能应聘，所有的空缺职位被列举在一个单独的页面中，这个页面将所有的职位以工作部分进行分组展示，当用户点击相关的组名称时，才显示这个组下的职位列表。当注册用户查看这些职位的时候，还可以查看发布者资料、联系发布者，同时还可以上传自己相关的信息文档（如简历等）。

实现方法

Drupal 核心为我们提供了一个很好的起点，它提供了我们需要的用户管理功能以及权限分配功能，还提供了几种基本的内容类型，如 Story、Page 等等，还允许我们创建其它的内容类型，如 Application、Job 等，但是 Drupal 的框架性架构，使得每一种新的内容类型都只有简单的标题、主体等几个固定的属性，现在我们需要对其进行足够的扩展，使其能被应用于更加广泛的领域，当然，包括了本章节所要讨论的招聘网站的应用。

自定义输入形式

这个网站最核心的需求就是将数据以各种类型插入到数据库中，而且还要提供相关的输入手段，比如用一个下拉菜单来选择部门、使用一个简单的文本字段来输入单选文本或者电话号码之类的内容。

列表

当我们把招聘信息都发布在网站上，应聘者也都提交了自己的应聘书之后，我们还需要把这些信息都列举到网站上，这个时候我们可以创建表格，这些表格可以把所有招聘信息按一定的组织规则列举在网页上，或者把应聘者的应聘书也按一定的规则列举出来，同时我还可以控制这些表格的可访问性，比如，招聘信息可以给任何人看，但是应聘书却只能由管理员查看等等。

文件上传

虽然 Drupal 系统核心模块已经提供了文件上传功能，但是它有如下不足之处：

- 它允许用户上传任何类型的文件，但是我们有时候却只希望被允许的几种文件格式可被上传。
- 所有文件都会被上传到系统默认的文件存储路径，也就是“Root”目录，这使得当文件数量越来越多之后，显得十分的混乱。
- 它允许一个节点可以上传无限数量的文件，但有时候我们只希望某个节点只上传限定数量的文件。
- 通常它上传的文件都是作为一个节点的附属部件来对待的，但是有时候我们需要把这个文件也做为一个节点。
- 它全使用的设定都是整个网站的全局设定，但是我们需要的是对它进行更为细致的设定。
- 它不允许你为某个文件设定特殊的名称等。

幸运的是，上面这些不足之处我们都可以通过基于 CCK 的模块 FileField 模块来解决。

聚焦-CCK 聚焦-CCK

CCK 是一种极其灵活的框架，它使得我们可以在 Drupal 上很轻松的建立任何我们需要的内容类型，虽然 Drupal 核心已经提供创建新的内容类型的功能（比如我们可以不使用使用新增模块就可能创建招聘信息、应聘信息等内容类型），但是这样创建的类型太过于单调，或许并不是我们真正所需求的，比如我们无法为应聘信息这个内容类型单独指定一个字段，用来上传单个文件！但是 CCK 却可以很容易的帮我们实现，所以，几乎使用一个使用 Drupal 搭建的网站，在安装好了 Drupal 系统之后，第一步就是安装 CCK 这个模块。

当我们全新安装好一个 Drupal 系统之后，我们可以看到有两种默认的内容类型 Story 和 Page，但其它这两者几乎是一样的，每一种类型无非就是一个包含标题和主体的集合，任何附加内容也都是由一个标题和一个主体组成，但是我们需要添加更多的额外信息，这个时候，我们就会用到 CCK 了。

当我们安装好了 CCK 模块之后，几乎无限数量无限类型的字段都可以被添加要任何一个内容类型上，比如我们可以创建一个新的内容类型，取名为“招聘信息”，之后，我们再为这个内容类型添加一个标题字段、主体字段，还可以添加一个字段用来存储工作部分信息、还可以再添加一个字段用来存储工作地点信息或者工资待遇信息，除了这些，我们还可以再为其添加一个字段，用来上传相关的文档等等。

字段（Fields）

从功能上来讲，CCK 为 Drupal 站点创建了两个最终用户功能，第一个就是字段，字段允许用户将数据保存到你的网站数据库中，字段可以存储的是这数据的类型，比如整数、小数、文本等。当为一个内容类型添加一个字段时，要做的第一件事情就是为这个字段指定一个类型以告诉系统什么样的内容将被保存在这个字段中，比如工作职位名称的字段类型为单行文本，电话号码的字段类型为整数等等。下表列举了所有 CCK 模块核心所支持的数据类型：

CCK 模块核心内置的数据类型	
字段类型	字段的一般使用
整数 Integer	最常被用来保存一个数字，比如产品数量，ID，或者任何可以使用整数来表示的内容，比如某个相册中的照片数或者某个事件的参与人数等等。
浮点数 Decimal	一般都保存含有小数点的数值，比如产品价格等。
浮点数 Float	最常被用来保存需要有较高精确度的数值，比如汇率等。
文本 Text	可以保存文本数据，比如标题、简介等，当然，还可以用来保存更长一点的内容，比如传记、说明等等。
节点引用 Node Reference	它可以在一个节点中生成一个对另一个节点的引用，将两个或者多个节点链接到一起。
用户引用 User Reference	与节点引用功能相似，只不过它是将用户与用户，用户与节点进行链接，比如可以在一则招聘信息下方列举出所有应聘者的网址等。

### CCK 或 Taxonomy?

CCK 与 Taxonomy 都能以将内容进行列表展示，在实在应该中，我们应该选择哪一种呢？以下面对 CCK 和 Taxonomy 进行一些对比，让大家在项目中可以按需而定：

- Taxonomy 是用来产品分类的，所以，如果你想把很多内容节点进行分类，你应该使用 Taxonomy。当然，你还可以使用 CCK 创建一个叫作 Category 的字段来完成与 Taxonomy 一样的功能，但是想想，Taxonomy 已经完全能够胜任这个任务了，而且很多其它的模块都已经支持 Taxonomy。
- Taxonomy 可以很容易的实现内容的分类与整理，同样，它还可以很容易地生成一个目录树。
- Taxonomy 只为很一个 Term 提供名称 (Title) 和简介 (Description)，所以，如果你需要更为精确的 Term，那么就应该使用 CCK。
- 如果你想将来即使移除了某个用来分类的字段，但是使用了此字段的节点却仍然存在，那么就应该是使用 Taxonomy。一个被归类在“Technology”分类下面的文章在 Technology 这个分类被删除之后，他仍然是存在的，而且还可能 被归类到其它分类下面。

### Widgets

一旦数据类型确定了，那么就是时候考虑以何种形式在表单中展示这些数据了，在 CCK 的术语中，一个表单元素被称之为 Widgets。一个 Widget 可以是一个下拉菜单、一个筛选框群或者一个单行文本等。

**注意：** 可用的 Widget 将根据你所选择的不同数据类型有所不同。

下表列举了 CCK 模块核心支持的 Widget 类型：

CCK 模块核心支持的 Widget 类型	
Widget 类型	作用以及一般使用
文本字段 Text Field	此类型允许单行文本的输入，比如姓名、电话号码等，同时支持纯文本或者格式化文本的输入。
文本区域 Text Area	此类型允许多行文本的输入，比如简介、主体、博客文章内容等等，同时支持纯文本或者格式化文件的输入。
单个选框 Single on/off CheckBox	当某事只允许出现“是”或者“否”时，我们可以使用这个 Widget 类型，比如：是否成人？是否为中国人？或者询问用户是否愿意接收网站的邮件等。
多个选框 Checkboxes/Radio buttons	当某事允许出现多种情况时，可以使用，比如：用户的兴趣列表、用户现有的 IT 装备列表等等，或者如用户喜欢的颜色，这些都是可以进行复选的。

CCK 模块核心支持的 Widget 类型	
Widget 类型	作用以及一般使用
选择列表 Select list	与多个选框类似，但是它将所有的选项都存放在一个下拉列表里面，一般只能选择一项。
自动填充文本字段 Autocomplete text field	自动填充的文本字段，当用户输入“St”是，会自动提示“Student”，“Stewart”等等，当一个菜单项目很多时，可以使用这种类型的字段。

#### 格式化 (Formatters)

当完成一个 Field 的添加之后，Formatter 可以让我们完全的控制此 Field 的输出，比如一个名为单价的 Field，我们可以定义其输出格式为 ¥9999.99，这样，任何输入都将被转换成这样的格式，比如现在我输入一个产品的单价为 23.456，那么在输出时，将被输出为¥23.46。

其它的模块也可以添加核心 CCK 所没有 Formatter，比如再之后的章节中将讨论到的 ImageCache 模块就可以提供一个生成并展示缩略图的 Formatter。

**注意：**可用的 Formatter 是根据 Field 的类型提供的，不同的 Field 类型将有不同的 Formatter，如果你想对一个数值进行格式化，就一定要将此 Field 的类型设置为整数、小数或者浮点数，而不能设置成为 Text，Field 的类型在创建之后将不能再进行更改或转换，你需要删除再重新创建一个新的 Field。

#### 动手操作：CCK

我们现在要开始正式制作一个前面所讨论的招聘网站，首先，让我们思考一下这个网站所需要的内容类型，有两个内容类型是必须的：

##### 招聘信息

这个内容包括招聘信息的标题、简介等等。

##### 应聘信息

由应聘者提供信息的内容类型，对招聘信息相对应。

我们还需要将应聘信息与相关的招聘信息联系到一起，这将使用到 CCK 中的节点引用字段，而应聘信息又应该与应聘者联系到一起，这将会使用到 CCK 中的用户引用字段。

现在我们先要激活 CCK 和 FileField 这两个模块，如下图所示：

进入站点构建->模块页面，我们共激活了 CCK 中的 Content、FileField、Node Reference、Number、Option Widgets、Text、User Reference 这几个，其中，Content 模块属于 CCK 的核心 模块，所以是需要要激活的，同时 FileField 并不是 CCK 内置的模块，而是单独的一个 CCK 扩展。

#### 动手创建“招聘信息”内容类型

激活 CCK 之后，我们就可以开始使用 CCK 来创建前面所说的“招聘信息”的内容类型了。

我们将创建的“招聘信息”这个内容类型，其包括如下字段：

- 招聘信息名称
- 招聘信息简单介绍
- 职位所属部门（一个下拉菜单选择）
- 与用户相联系（User Reference）
- 工资待遇（一个整数字段）

首先，进入管理->内容管理->内容类型->点击“添加内容类型”，之后我们会进入一个页面，要求我们填写一些普通的信息，比如类型名称等，如下图所示：

接上图所示填写完成之后，再点击页面下端的“保存内容类型”，这时我们会返回到内容类型列表页面，现在我们要为刚刚创建的名为招聘信息的内容类型添加一些新的 Field 了，点击“招聘信息”这个内容类型后面的“Manage Fields”链接，我们开始为其添加非 Drupal 内置的 Fields。

工作岗位名称和简介已经有了，所以我们现在先添加“工作岗位所属部门”，如下图所示：

在 News Field 的第一个 Label 中填写“工作岗位所属部门”，在 Field name 这里填写 department，这个将是用来写入数据库时使用的，字段数据类型为 Text（即文本），Widget 我们选择 Select List（单选菜单），点击保存。

保存之后，进入的一个新页面将要求我们对刚才添加的 Field 作更细致的设置，在这里我设定帮助文本（Help Text）为“选择此工作职位所属的部门”，然后，可选择的部门列表如下：

- 网站管理员
- 论坛管理员
- 主机销售员
- 论坛版主
- 服务器相关
- .....

当然，我们还可以填写更多，但是在这里我们就以这几个来作为示例，现地我们再一次保存之后，Drupal 会提示你将的字段内容已经添加进入你的数据库。

现在我们可以添加工资待遇 Field 了，在 Field 管理页面中添加新的 Field，如下图所示：

保存设置，进入设置页面，在帮助文本区块输入：“此工作职位一年的工资待遇”，接着，我们还应该设置这个字段的前缀（Prefix），也就是进行简单的格式化，我们输入：“¥”，表示这里所输入的工资数值按人民币计算。点击保存。

再接下来我们添加一个联系字段，将被发布的招聘信息与发布人联系到一起，如下图所示：

我们需要只有管理员才能发表招聘信息，所以，这里我们也应该只允许管理员才能进行管理，所以，在创建了新的 Field 之后，我们应该设置其权限为“只有管理员可访问”。同时，我们在帮助文本栏里设置“请选择应聘者第一联系人”，即如果某人要应聘，他应该第一联系谁。“User roles that can be referenced”这里是设置“哪一等级的用户能被联系到招聘信息上，我们选择管理员，“User status that can be referenced”设置哪一类型用户能被联系到招聘信息上，我们选择 Active，即已经激活的用户帐户。点击保存。

动手操作：自定义 Field 输出

++++++END++++++

## Drupal 6 中添加 "相关链接" 的方法

浏览别人网站时，总是会在文章底下或者侧边栏上看到很酷的“相关链接”或者“related blog”。而且随着网站里面文章越来越多，怎么让读者找到相关的内容很重要。在网上搜索了很久，加上自己的摸索，终于搞定了在 Drupal 中的实现。下面分别用 Drupal 6 的 CCK 模块，Drupal 的 Relevant content 模块，Drupal 的 similar entries 模块和 Drupal 的 similar by terms 模块实现了该功能。

方法 1：用 Drupal CCK 模块

当然，装完要先到[管理]>[站点构建]>[模块]里启用，注意一定要启用[Node Reference]这个字模块。

CCK 模块下载

<http://drupal.org/project/cck>

CCK 中文介绍（繁体）

[http://drupal taiwan.org/module/content\\_construction\\_kit\\_cck](http://drupal taiwan.org/module/content_construction_kit_cck)

CCK Handbook（英文）

<http://drupal.org/node/101723>

下面来说对 CCK 的设置：

1. [管理]>[内容管理]>[内容类型]> [编辑] Story > [增加栏位]

这里平民是对 Story 类型更改为例，当然你也可以照葫芦用到其他内容类型中。

进入[增加栏位]的页面后，如果之前有新增过栏位，会出现[增加现有的栏位]的群组（Group），如果没有的话，表示您真的没有用过 CCK 模块来增加栏位，但这不重要，重要的是下面的[建立新的栏位]群组，先填入[名称]，最好用英文平民

把它命名为 "relatenode", 选好名字后, 再来是下面的[栏位类型], 这个一定要选 [Node Reference]下的[Select List], 再检查一下, 有没选对, 确认没问题后, 按[建立栏位]。

按下去后, 会跳出另一个设定画面, 先把[工具设定]群组的[标签]改成中文的, 这个是会显示出来给使用者看的, 例如平民就改为[站内相关连结], 不改的话, 默认会用栏位的名字, 而上面那[工具]是选栏位的类型, [Select List]是下拉式选单的栏位[Autocomplete Text Field]则是会依打入的字自动寻接相似的文件。在[可以被参照的内容类型:]里选择范围, 平民选择了全部。

2.[管理]>[内容管理]>[内容类型]>[编辑] Story > [Manage Fields]

在这裡就会看到刚刚新增的栏位, 名称会变成以 field\_ 开头, 这裡是要设定这个栏位在表单裡的位置, 以 Story 来说, 已经有[Title]和[Body]两个栏位了, 如果要排在这两个栏位之后, 一般默认是可以, 保险起见, [顺序]设为 3。

3. [管理]>[内容管理]>[内容类型]>[编辑] Story > [Display Fields]

新增的栏位可以限定其要被显示出来的时机和位置 (是在阅读的时候, 跟编辑的画面无关), 总共有三项设定:

\* [标签]设定新增栏位的标签摆放位置, 平民选 "Above", 所以 "相关连结" 的标签下面才会出现所选的相关博文。

\* [Teaser]设定在 [Teaser]模式下的显示方式, 平民设为 "Hidden"。

\* [Full]设定全文模式下的显示方式, 平民是设为 "Title (link)"。

4. 去新增一篇 Story, 下面会多出一个下拉式的栏位, 叫做 "站内相关链接" (或者你自己起的别的名字), 在里面就可以为这篇 story 选择相关的节点了。

注意一个 CCK 栏位可以给多个内容类型使用, 方法就是利用那个 [增加现有的栏位]的方式, 来增加一个现有栏位给指定的内容类型。

方法 2: 用 Drupal 的 Relevant Content 模块

Relevant Content 模块下载

[http://drupal.org/project/relevant\\_content](http://drupal.org/project/relevant_content)

装完, 启用。这个模组可以将相关链接放到 block 里, 也可以将相关链接直接放到 CCK 栏目中显示 (效果和上面的相同, 但是自动生成而不是手动选取)。选择相关性基于共有的 tag 数量来判断。在论坛里反响不错, 但平民试验后发现在自己的网站上不能在文章正文后显示。遂卸载。你也可以试验下, 如果成功了确实很方便。

方法 3: 用 Drupal 的 Similar Entries 模块

Similar Entries 模块下载

<http://drupal.org/project/similar>

该模块会产生一个相关链接的 block。它基于标题的相似来判断相关性, 而不是基于 tag。小巧易用。注意它暂时不支持 CCK 的栏位显示。

方法 4: 用 Drupal 的 Similar by Terms 模块

Similar Entries 模块下载

<http://drupal.org/project/similarterms>

和 Similar Entries 相似, 但它基于 tag 来判断相关性。也不支持 CCK 的栏位显示。但平民参照了在 Drupal 6 中将区块放入页面任何地方将 block 插入文章内容后进行显示。就是你现在看到的效果。

注意在[站点设置]>[Similar terms]中可以设置是基于所有的分类进行相关性判断, 还是只基于 tags 分类进行判断。平民选择了 tags (先到[内容管理]>[分类]中去建立 tags 词汇, 注意将它设置成可以自由加标签的那种类型)。平民这样做就可以做到这个词汇和其他词汇之间的独立了, 也就是 tags 只有来做这个相关链接。

Bookmark/Search this post with:

## 用 CCK 模组做一个 "站内相关连结" 的栏位

随着网站裡贴的文越来越多, 文章与文章之间开始有互相参考的需要, 阿舍一开始很勤劳的用手动方式来 Copy & Past 站内相关文件的网址和标题, 可是越贴越觉得不对劲, 这麽常用的功能, 怎麽会没有人写成模组呢? 所以..... 阿舍又去网路逛大街, 爬文找资料了。

原来 CCK 模组就有这个功能, 不过也不是装了 CCK 模组就可以使用, 还是要设定一下, 是要花一点时间的。(若要达到更好的效果, View 这个模组是一定要装的哩.....)

首先当然要去下载 CCK 这个模组及中文 PO 档 ( 请参考文尾的连结列表 ), 接着便是启用 CCK 模组, 要注意, 有个 CCK 模组的子模组叫 "Node Reference" 的要启用, 因为今天要说的功能就是靠它完成的。

接下来就说明设定方式, 要有点耐心才行

### 1. 「管理」>「内容管理」>「内容类型」>「编辑」 Story > 「增加栏位」

阿舍是以 Story 这个「内容类型」来修改, 如果大德要修改的是别种「内容类型」, 就「编辑」要修改的「内容类型」, 例如 Page 或部落格文章。

进入「增加栏位」的页面后, 如果之前有新增过栏位, 会出现「增加现有的栏位」的群组 ( Group ), 如果没有的话, 没关系, 表示这位大德真的没有用过 CCK 模组来增加栏位, 但这不重要, 重要的是下面的「建立新的栏位」群组, 先填入「名称: 」, 千万别打中文, 这个是给程式用的, 最好只用英数字, 阿舍把它命名为 "ref", 选好名字后, 再来是下面的「栏位类型: 」, 这个是第一个重点了, 请点选「Node Reference」下的「Select List」, 再检查一下, 有没选对, 确认没问题后, 按「建立栏位」。

按下去后, 会跳出另一个设定画面, 不是结束哩, 不要冲动..., 先把「工具设定」群组的「标籤: 」改成中文的, 这个是会显示出来给使用者看的, 所以要看的懂, 例如阿舍就改为「站内相关连结」, 不改的话, 就会用栏位的名字当作标籤用, 而上面那「工具」是选栏位的类型, 「Select List」是下拉式选单的栏位「Autocomplete Text Field」则是会依打入的字自动寻找相似的文件, 阿舍试不出来, 所以还是乖乖的用「Select List」.....

接下来的部分, 各位大德可以很简单的只填「可以被参照的内容类型: 」这个栏位就好, 或是复杂一点的把「Advanced - Nodes that can be referenced (View)」群组点开, 来指定这个新增栏位可以参照的文件别。

由于阿舍并没有用到「检视」, 只在「可以被参照的内容类型: 」的栏位裡选了 "Stroy", 就按「储存建立栏位」, 所以这个新增的栏位就会显示出所有用 Story 所建立的文件 ( node ), 这样比较单纯, 等阿舍的分类作好了再来调整。

补最底下的「Advanced - Nodes that can be referenced (View)」群组的栏位是用来帮助将文件分类的, 当各位大德在网站内的文件数多到一定程度时, 要找到一份文件来参照并不容易了, 这时就要配合 View 模组来达成, 至于怎麽用 View 模组来设定, 阿舍并没有打算在这裡详谈, 因为 View 模组博大精深, 不是一、二十个字可以说完的 ( 其实是, 阿舍还在研究中, 等学有所成就会写出来和各位大德分享.....=\_=!), 不过只要装过 View 模组, 在「Advanced - Nodes that can be referenced (View)」群组裡的「检视: 」栏位就会有一些预设的「检视 (View)」可以选了, 至于什麽是检视? 简单地说, 就是依指定条件所整理出来的文件 ( node ) 列表。

### 2. 「管理」>「内容管理」>「内容类型」>「编辑」 Story > 「Manage Fields」

在这裡就会看到各位大德刚刚新增的栏位, 名称会变成以 field\_ 开头, 阿舍新增的栏位名称会变成 field\_ref, 如果栏位名称就只有 "field\_" 怎麽办? 没关系还是可以用, 只是丑了一点, 想要改名称怎麽办? 那就只能移除再重建了.....-!!。回到重点, 这裡主要目的是要设定这个栏位在表单裡的位置, 以 Story 来说, 已经有「Title:」和「Body:」两个栏位了, 如果要排在这两个栏位之后, 「顺序」设为 3 会比较保险的作法, 如果是 0, 那就要试试看了, 阿舍不确定预设的这两个栏位都是 0 或是 1 和 2。(为什麽没有讲 Group 的设定? 因为一开始就没讲, 所以会另外再写一篇来介绍哩...)

赞助连结

### 3. 「管理」>「内容管理」>「内容类型」>「编辑」 Story > 「Display Fields」

新增的栏位可以限定其要被显示出来的时机和位置 ( 是在阅读的时候, 跟编辑的画面无关 ), 总共有三项设定:

\* 「标籤」设定新增栏位的标籤摆放位置, 阿舍选 "Above", 所以 "相关连结" 的标籤下面才会出现所选的参考文件。

\* 「Teaser」设定在「Teaser」模式下的显示方式, 阿舍是设为 "", 就是隐藏的意思, 因为阿舍觉得都没看完内文, 显示相关连结的意义应该不大, 所以设定不显示。

\* 「Full」设定全文模式下的显示方式, 阿舍是设为 "Title (link)", 就是会显示文件的标题, 然后按下去就会开启文件的那一种。

好啦, 这样算搞定了, 去新增一篇 Stroy 看看, 是不是多出了一个下拉式的栏位, 叫做 "站内相关连结" ( 如果都照阿舍的方式做的话 )。

注.

1. 阿舍用的 CCK 模组的中文 PO 档, 有些部分并没有翻, 所以就直接用英文的词来描述, 如果不清楚的地方, 就请提出.....

2. 一个 CCK 栏位可以给多个内容类型使用，方法就是利用那个「增加现有的栏位」的方式，来增加一个现有栏位给指定的内容类型。

CCK 模组

<http://drupal.org/project/cck>

CCK 模组的中文介绍

[http://drupaltaiwan.org/module/content\\_construction\\_kit\\_cck](http://drupaltaiwan.org/module/content_construction_kit_cck)

CCK 模组正体中文 PO 档下载

<http://drupaltaiwan.org/forum/20070126/826>

CCK Handbook (英文)

<http://drupal.org/node/101723>

后记.(08/03/18)

阿舍今天在 Drupal Taiwan 裡有看到一篇讨论文在讨论利用「Best Related Links Project」模组来达到 "相关文章" 的作法，有兴趣的大德们可以参考一下囉！

对中文内容比较准确的相关主题文章模组

<http://drupaltaiwan.org/forum/20080317/1959#comment>

-----END-----

## Drupal 综合实战：CCK+VIEWS=CMS

我简单的介绍一下 Drupal，并为一个虚拟的客户建立一个简单的网站。我将首先概述客户的需求，安装和配置一些模块，创建用户角色和分配权限，最后我们将新增一些内容。我们不会修改主题，因为它有点超出了本文的范围。

### 虚拟客户

让我们先虚拟一个客户：SmartMarks 是一个很小的市场营销咨询公司，4 名员工。每一位员工想拥有一个自己的博客。该网站除了博客还需要几个页面：

- Home(首页) About(关于) Contact(联络) Links(链接) Blogs(博客)

Shannon，企业老板，要完全控制网站。其余的员工（Bill, Jean, 和 Terry）应只允许写博客，但是 Bill 允许发布 Links(链接)。

联络方式是按用户的姓名，电话，电子邮件和短信息。提交的联络信息只发送给 Shannon。

听起来很简单，是吧？我们将利用 Drupal，一些核心功能已经内置。我们可以使用很多核心功能，当然我们需要添加一些其他模块来建立这个网站。

### 安装某些东西

首先，开始安装 Drupal，我将在本地计算机进行开发，当然您可以安装在任何您想要的地方。本教程将使用 Drupal 6.x。

安装 Drupal，先下载 (<http://drupal.org>) 解压并创建数据库，并访问 <http://localhost/>（或您自己的开发网址）。安装过程非常简单。

您需要创建配置文件，可以复制/webroot/sites/default/default.settings.php 到 /webroot/sites/default/settings.php，并在您的服务器上设置为可写。同时保留一个 sites/default/default.settings.php 文件的备份，Drupal 在安装过程中将要用到。

当您的配置文件已经创建，可以继续安装 Drupal。

下一步将创建您的网站的第一个用户，这个用户将是管理员或超级用户，此用户拥有本网站的全部权限。

接下来进行一些服务器设置，如果你的服务器开启了 `mod_rewrite`，你可以继续设置是否启用简洁链接，例如 `/?q=node/3` 将重写为 `/node/3`。

安装完成为，您就可以访问网站，并以超级用户登录了。

## 安装这些模块

当你下载完上面的模块并解压到 `/webroot/sites/all/modules` 后你可以安装它们。

访问 `http://localhost/admin/build/modules`，制作本网站需要启用下面的模块

- Administration - Administration Menu
- Core - Blog
- Core - Contact
- Core - Path
- Content - Content
- Content - Text
- Other - Meta tags
- Other - Pathauto
- Other - Token
- Views - Views
- Views - Views UI

## 内容类型

在我们创建用户和角色前我们先创建内容类型，我们需要创建 **Links**（链接）内容类型，每个链接需要一个标题、URL 和一段简短的说明。

### 什么是节点？

几乎所有的内容存储在 **Drupal** 中都可以称为一个节点。所有的节点拥有标题及描述（可选）。通过创建内容类型，你可以添加自定义字段扩展节点。

对于我们而言，需要为 **Links**（链接）创建一个 URL 字段。

## 创建一个 **Links**（链接）内容类型

我们先创建一个为“**Links**（链接）”的内容类型，然后为本内容类型创建一个“URL”字段。

访问 `http://localhost/admin/content/types/add`

在名称字段中输入一个名字，在类型字段中输入一个独一无二的类型名称。系统将在内部使用这个名字，你可以修改它，通常它被视为一个变量名，使用小写和下划线。接着输入内容类型的简短描述。

接下来，我们对内容类型进行一些修改。在“Submission form settings”组中的“Body field label”中输入“Short



Description”。

The screenshot shows the 'Submission form settings' for a content type. It includes a 'Title field label' set to 'Title', a 'Body field label' set to 'Short Description', and a 'Minimum number of words' spinner set to 0. There is also a large empty text area for 'Explanation or submission guidelines'.

下一步，我们编辑“Workflow settings”，选定 Links 保存时默认为“published”。

The screenshot shows the 'Workflow settings' for a content type. Under 'Default options', the 'Published' checkbox is checked, while 'Promoted to front page', 'Sticky at top of lists', and 'Create new revision' are unchecked. The 'Allow editing of meta tags' checkbox is also checked.

最后，禁止评论。

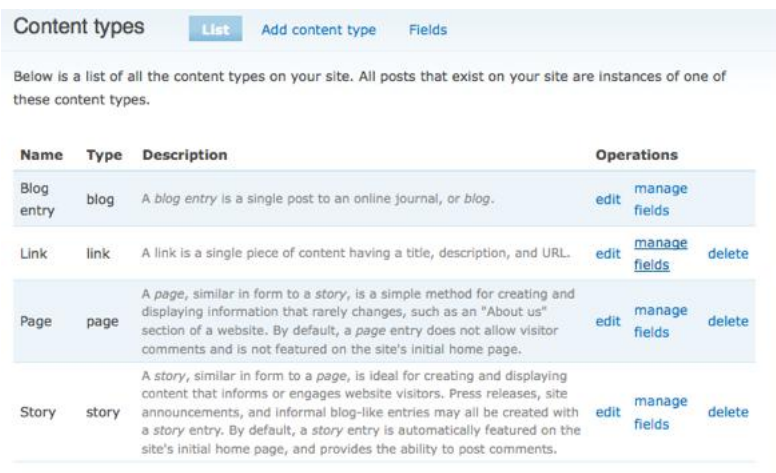
The screenshot shows the 'Comment settings' for a content type. Under 'Default comment setting', the 'Disabled' radio button is selected. Under 'Default display mode', both 'Flat list - collapsed' and 'Flat list - expanded' radio buttons are unselected.

保存内容类型，当你访问“创建内容”页面时你可以看到刚才创建的“Link”内容类型。

The screenshot shows the 'Create content' page on a Drupal site. It lists several content types: 'Blog entry', 'Link', 'Page', and 'Story'. The 'Link' content type is highlighted, with a description: 'A link is a single piece of content having a title, description, and URL.'

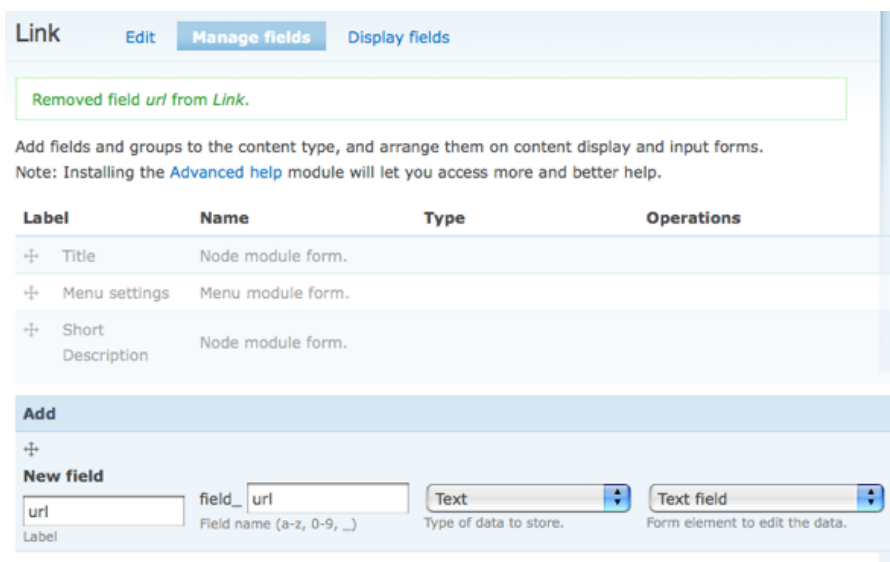
## 通过 CCK 添加字段

现在，我们已经创建了一个基本的 Link 内容类型，接下来我们添加为本内容类型添加 URL 字段用来输入连接。访问 <http://localhost/content/types>，点击 Link 内容类型的“Manage fields”。



Name	Type	Description	Operations
Blog entry	blog	A <i>blog entry</i> is a single post to an online journal, or <i>blog</i> .	edit manage fields
Link	link	A link is a single piece of content having a title, description, and URL.	edit manage fields delete
Page	page	A <i>page</i> , similar in form to a <i>story</i> , is a simple method for creating and displaying information that rarely changes, such as an "About us" section of a website. By default, a <i>page</i> entry does not allow visitor comments and is not featured on the site's initial home page.	edit manage fields delete
Story	story	A <i>story</i> , similar in form to a <i>page</i> , is ideal for creating and displaying content that informs or engages website visitors. Press releases, site announcements, and informal blog-like entries may all be created with a <i>story</i> entry. By default, a <i>story</i> entry is automatically featured on the site's initial home page, and provides the ability to post comments.	edit manage fields delete

添加一个字段。



Removed field *url* from *Link*.

Add fields and groups to the content type, and arrange them on content display and input forms.  
Note: Installing the [Advanced help](#) module will let you access more and better help.

Label	Name	Type	Operations
+	Title	Node module form.	
+	Menu settings	Menu module form.	
+	Short Description	Node module form.	

**Add**

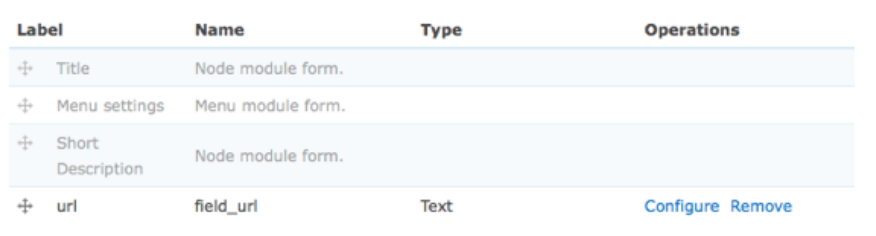
+

**New field**

url      field\_ url      Text      Text field

Label      Field name (a-z, 0-9, ...)      Type of data to store.      Form element to edit the data.

保存。然后会出现另一个页面，采用默认设置就可以了，所以只需点击“Save field settings”保存字段设置。完成后 Link 内容类型应该下面的内容。



Label	Name	Type	Operations
+	Title	Node module form.	
+	Menu settings	Menu module form.	
+	Short Description	Node module form.	
+	url	field_url      Text	Configure Remove

## Views

接下来我们为 Link 内容类型建立一个 view。

Views 可以让页面变得非常复杂，但是在我们这个例子中，我们会保持简单，因为我们只需要显示链接而已。

访问 <http://localhost/admin/build/views>，点击“Add”添加一个 Views。

**View name:** \*

Links

This is the unique name of the view. It must contain only alphanumeric characters and underscores; it is used to identify the view internally and to generate unique theming template names for this view. If overriding a module provided view, the name must not be changed or instead a new view will be created.

**View description:**

A list of links

This description will appear on the Views administrative UI to tell you what the view is about.

**View tag:**

Enter an optional tag for this view; it is used only to help sort views on the administrative page.

**View type:**

Node  
Nodes are a Drupal site's primary content.

Comment  
Comments are responses to node content.

File  
Files maintained by Drupal and various modules.

Node revision  
Node revisions are a history of changes to nodes.

Term  
Taxonomy terms are attached to nodes.

User  
Users who have created accounts on your site.

The view type is the primary table for which information is being retrieved. The view type controls what arguments, fields, sort criteria and filters are available, so once this is set it **cannot be changed**.

[Next](#)



接下来的页面比较复杂，通过这个例子让你更熟悉 Views。

首先，我们希望 Link 显示为一个完整的页面，所以添加页面显示。

**Defaults** ▶ Defaults *Default settings for this view.*

Page

**View settings**  
Tag: None

**Basic settings**  
Name: Defaults  
Title: None  
Style: Unformatted   
Row style: Fields   
Use AJAX: No  
Use pager: No  
Items to display: 10  
More link: No  
Distinct: No  
Access: Unrestricted  
Exposed form in block: No  
Header: None  
Footer: None  
Empty text: None  
Theme: Information

**Relationships** + ↑↓  
None defined

**Sort criteria** + ↑↓  
None defined

**Arguments** + ↑↓  
None defined

**Filters** + ↑↓  
None defined

**Fields** + ↑↓  
None defined

下一步我们必须作一些设置。将 view 的名字和标题设置为 Links，设置“Row Style”为 Node，并选择显示 teaser 和 links。

**Links: How should each row in this view be styled**

Status: using default values.

Fields

Node

You may also adjust the [settings](#) for the currently selected row style by clicking on the icon.

**Links: Row style options**

Status: using default values. Override

Display only teaser

Display links

Display node comments

Update default display Cancel

请记得每次都点击一下“Update Default Display”

设置 path(路径)为“links”，这样这个 view 将显示在 http://localhost/links

为这个 view 设置 menu(菜单)，选择“Normal menu entry”。标题为“Links”，Menu 选择“Primary Links”。

**Links: Menu item entry**

**Type:**

No menu entry

Normal menu entry

Menu tab

Default menu tab

**Title:**

Links

If set to normal or tab, enter the text to use for the menu item.

**Menu:**

Primary links

Insert item into an available menu.

**Weight:**

0

The lower the weight the higher/further left

现在 Basic settings 应是为下面这样的。

**Basic settings**

**Name:** Links

**Title:** Links

**Style:** Unformatted

**Row style:** Node

Use AJAX: No

Use pager: No

Items to display: 10

Distinct: No

Access: Unrestricted

Exposed form in block: No

Header: None

Footer: None

Empty text: None

Theme: Information

**Page settings**

**Path:** links

**Menu:** Normal: Links

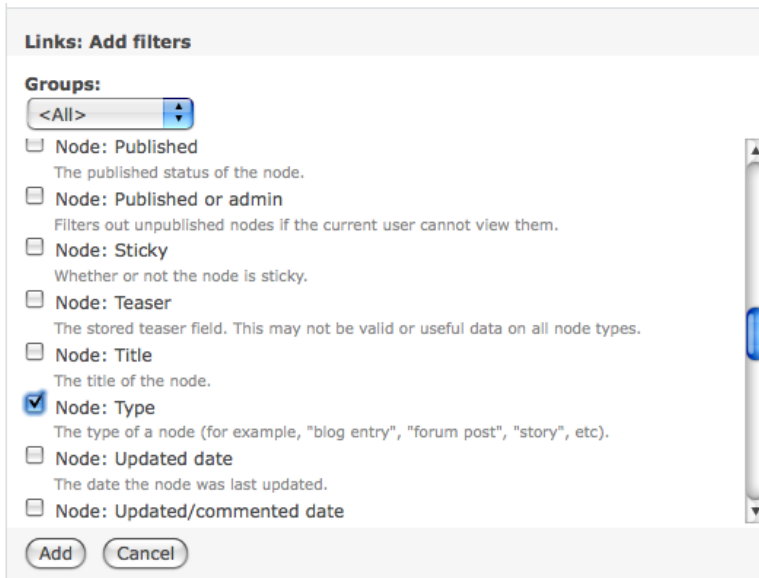
完成上面的步骤后我们需要添加一个 filter（过滤器），以只显示类型为“link”的节点。

点击过滤器上面的“+”

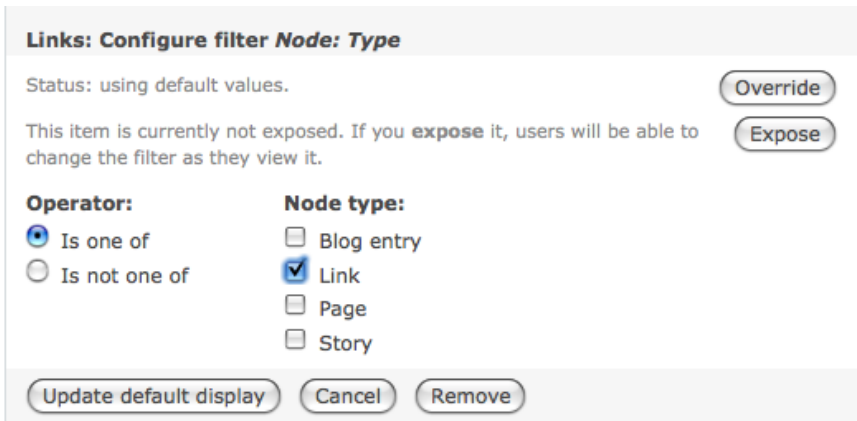
**Filters** + ↑↓

None defined

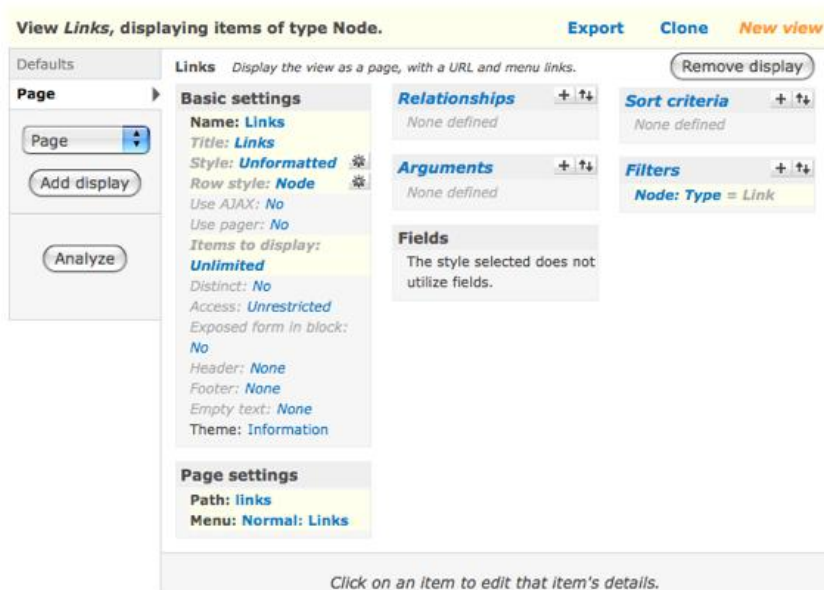
向下滚动，看到“Node:Type”后选定并添加。



选择节点类型为“Link”



到这一步，所有的设置应该是下面图中那样。



保存这个 view，以后我们还会要回来修改。

\*\*\*\*\*END\*\*\*\*\*

## Panels 2.X 教學 (基礎應用篇)

我將教學分為四部分 (A, B, C, D), 每一部分都附圖片, 看文字外亦請留意圖片內容 (想讓大家看清楚點, 圖片較大, 抱歉!!)

在示範中, 我使用了:

Drupal 5.14

Panels 5.x-2.0-rc1a

\*此教學的理論適用於 Panel 2.x 任何版本

\*由於 Panels 不穩定, 如果你使用的版本不一樣, 效果可能會有差別

\*Drupal 6 中, 你會發現錯誤, 或者找不到我提及的東西

**教學內容:**

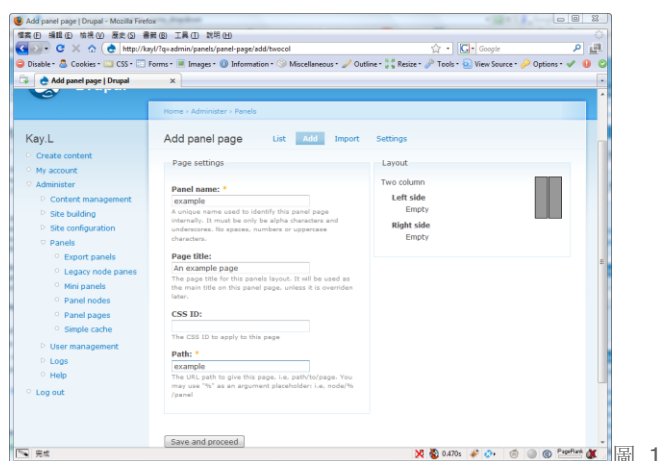
- A. 建立最簡單的 Panels 頁面
- B. 應用 ConTEXT Arguments 的 Panels
- C. 在不同的 Content type 顯示不一樣的内容
- D. 限制只在某些 Content type 應用 Panels

在教程示範中, 沒有提及的設定, 直接略過就可以

### A. 建立最簡單的 Panels 頁面

在這部分, 你會學到建立最簡單的 Panels 頁面

#### 1. 增加一個 Two column Panel



### 1-1. 瀏覽"?q=example", 你應該看見空白頁面

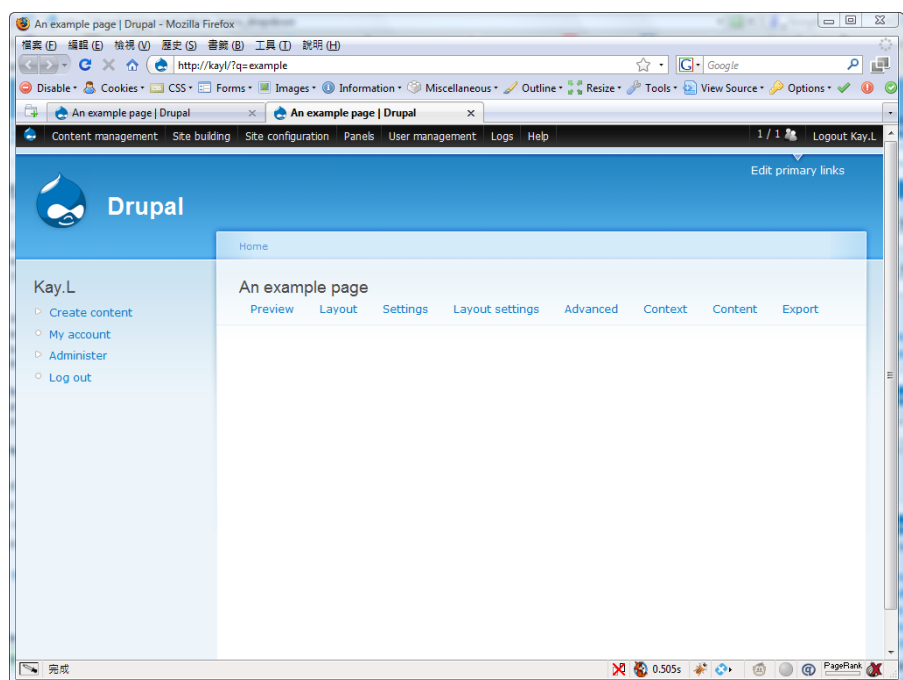


圖 1-1

### 1-2. 在 CONTENT 隨便增加一些東西

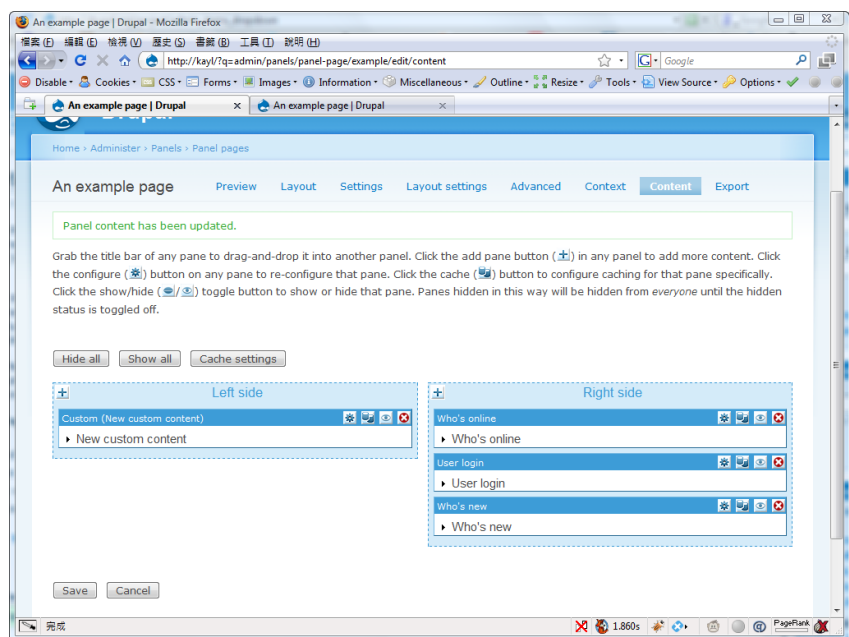


圖 1-2

### 1-3. 再瀏覽"?q=example", 你應該看見增加的內容了

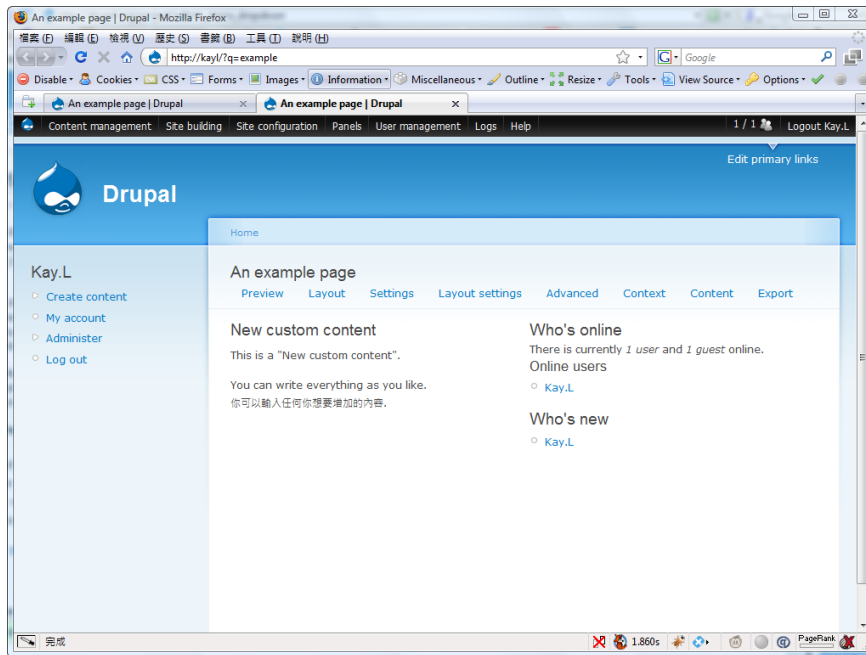


圖 1-3

以上就是最簡單的自定頁面。

### B. 應用 ConTEXT Arguments 的 Panels

你會學到如何應用 ConTEXT 中的 Arguments, 這是令 Panels 達到 Panels everywhere 的第一步及重要的一步

#### 2. 增加一個 Two column Panel. Path 設定是: node/%

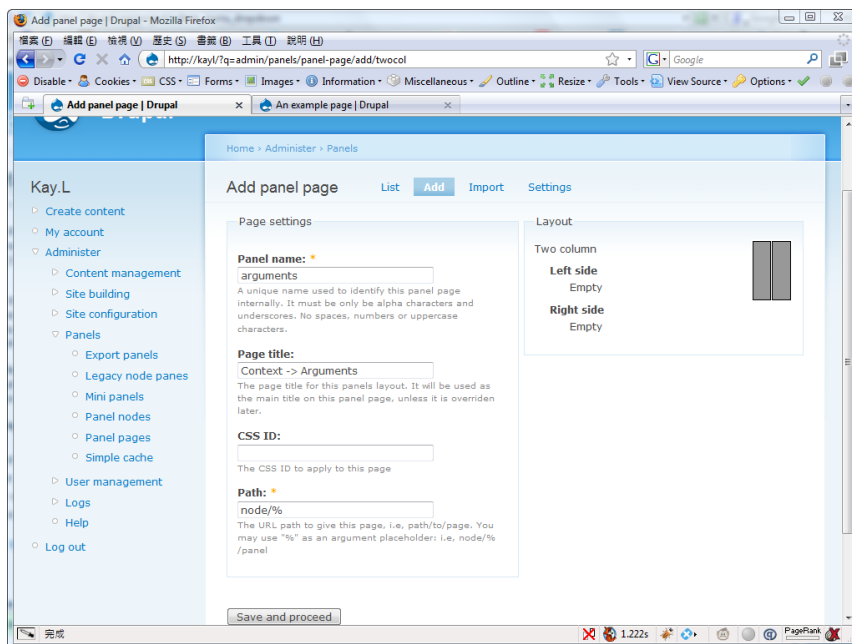


圖 2



## 2-1. ConTEXT 增加 Arguments-> Node ID

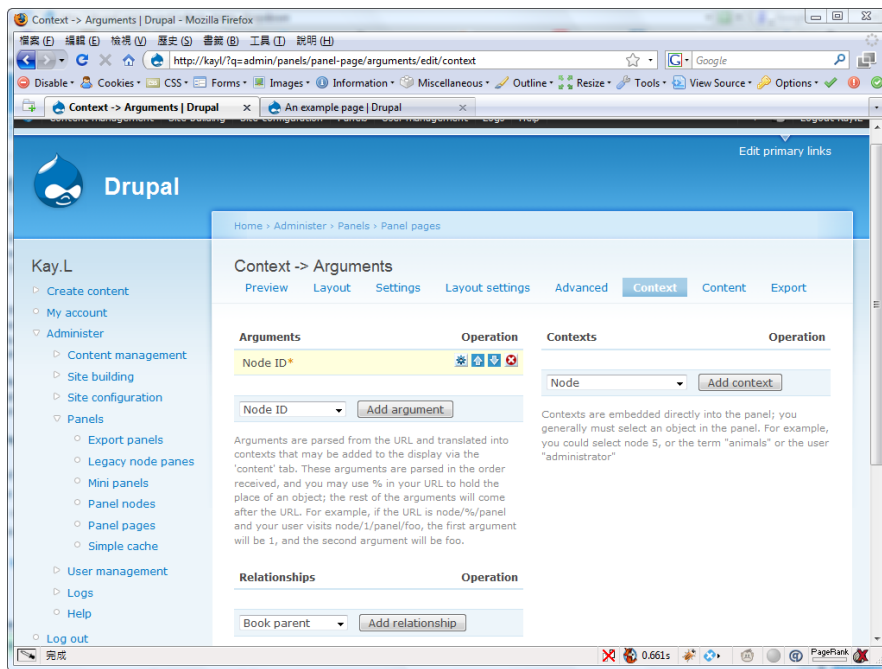


圖 2-1

## 2-2. 在 Content, 增加 Node content 欄下的 Node content

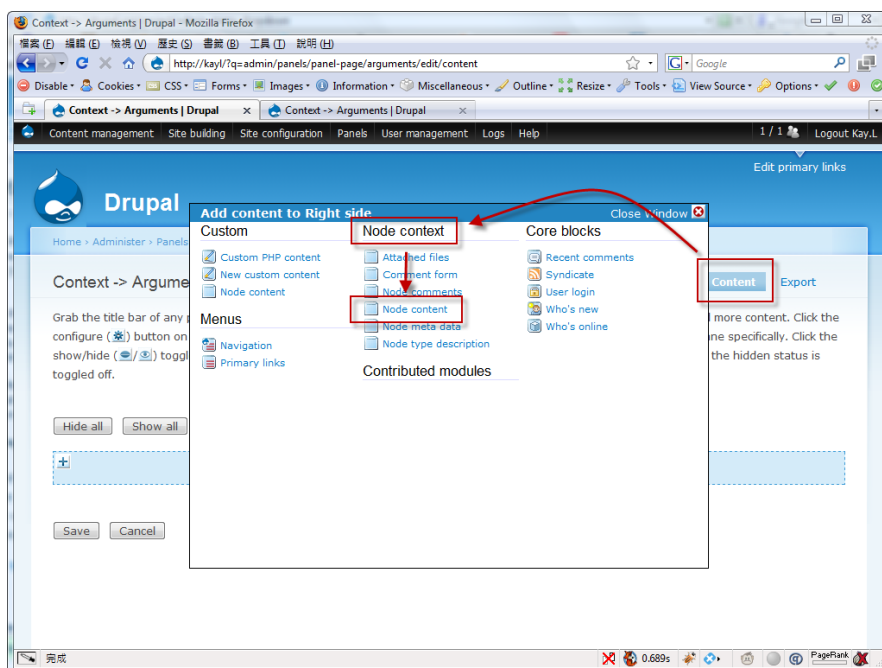


圖 2-2

### 2-3. 再增加一個 Custom 欄下的 Node content

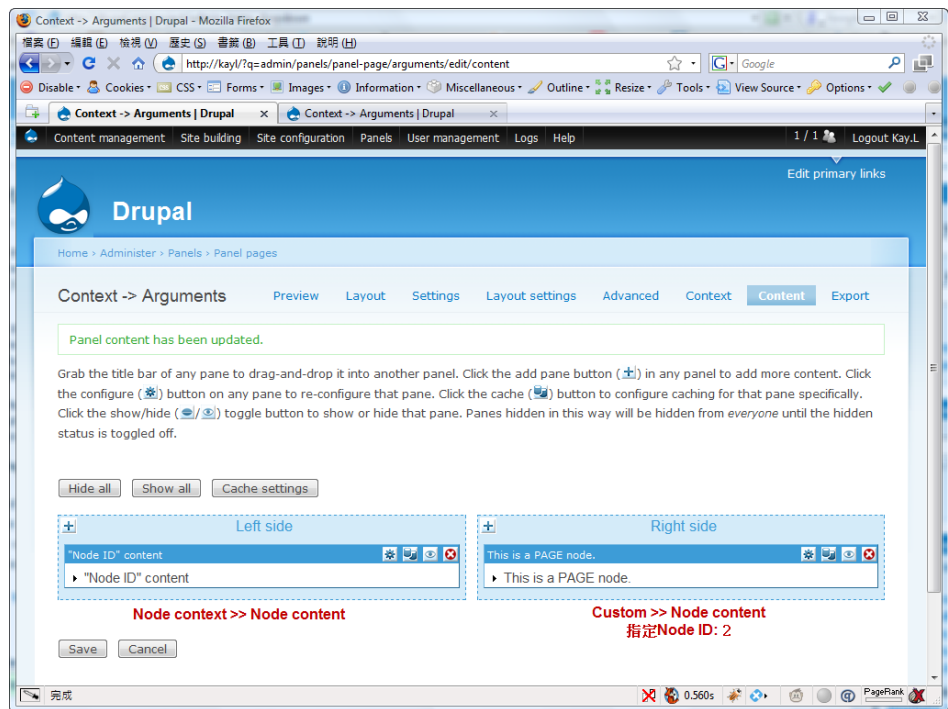


圖 2-3

在這裏，請大家特別注意，Node content 及 Custom 欄下都有 Node content，但兩者是有不同分別的 !!

2-4. 你可以清楚看見，左面是根據 URL 決定顯示什麼內容，而右面是在上一步中指定的 Node ID 內容

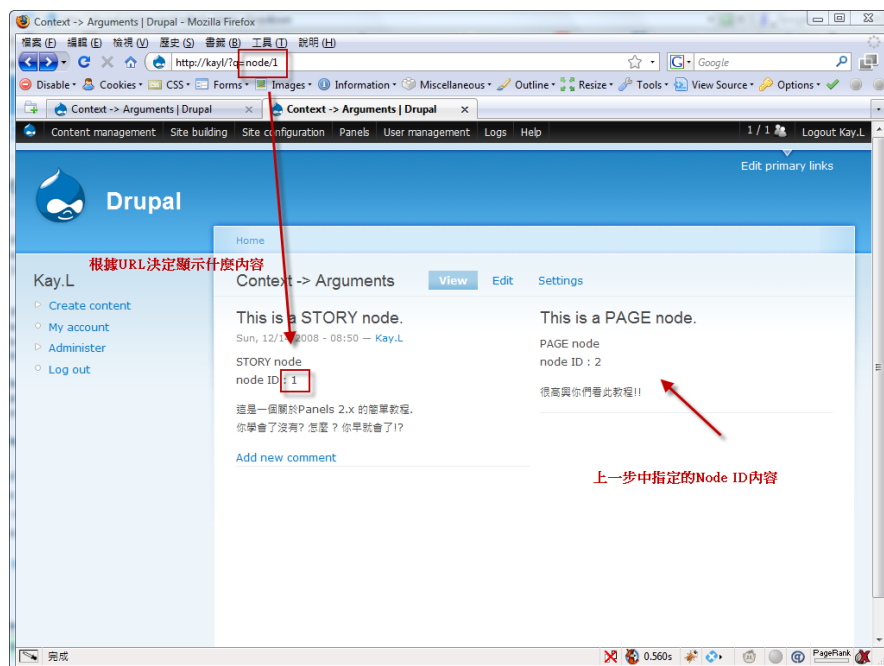


圖 2-4

2-5. 現在，我們回到 ConTEXT 中增加一個 Arguments-> Node add form

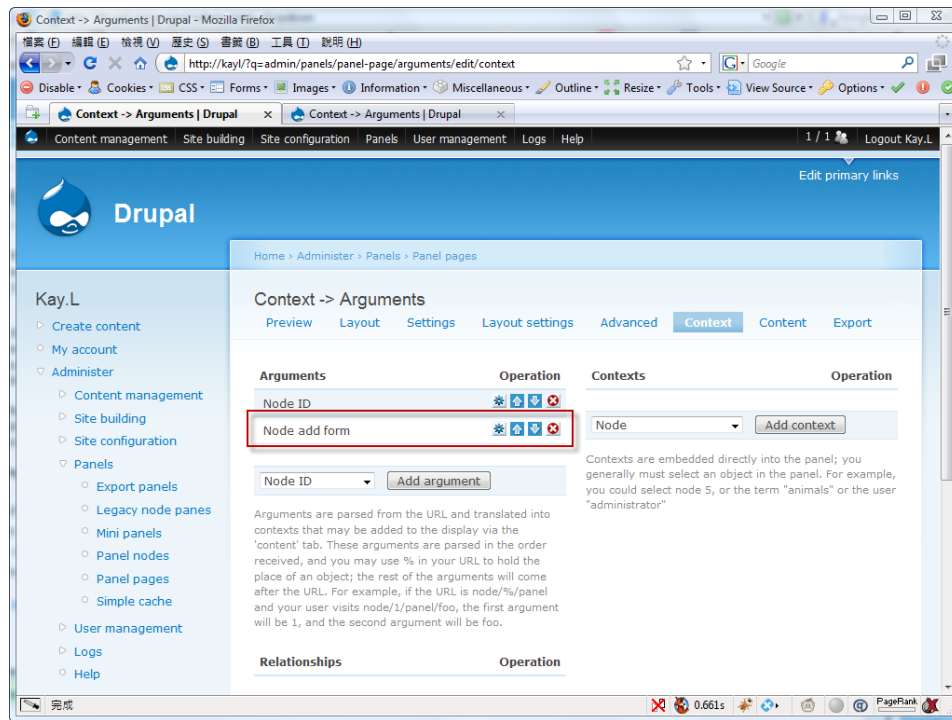


圖 2-5

2-6. 再回到 Content, 當你增加東西時，你會發時多了 Form 一欄，我們如圖增加一個 General Form

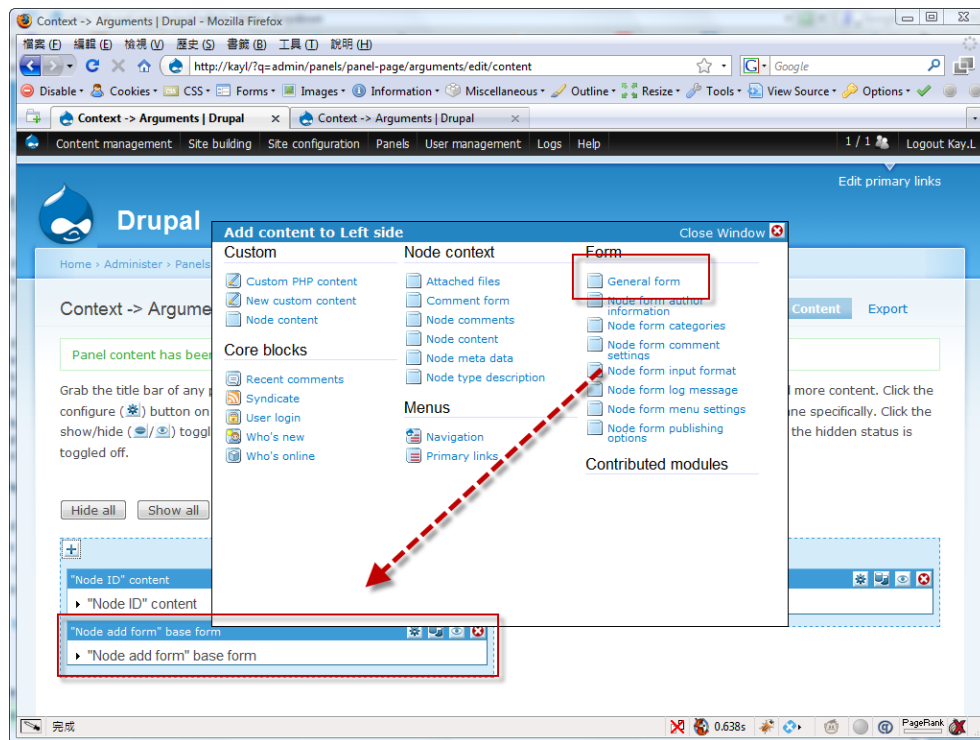


圖 2-6

2-7. 瀏覽"?q=node/1"看看，怎麼了 !! 那不是 ADD FORM !!!

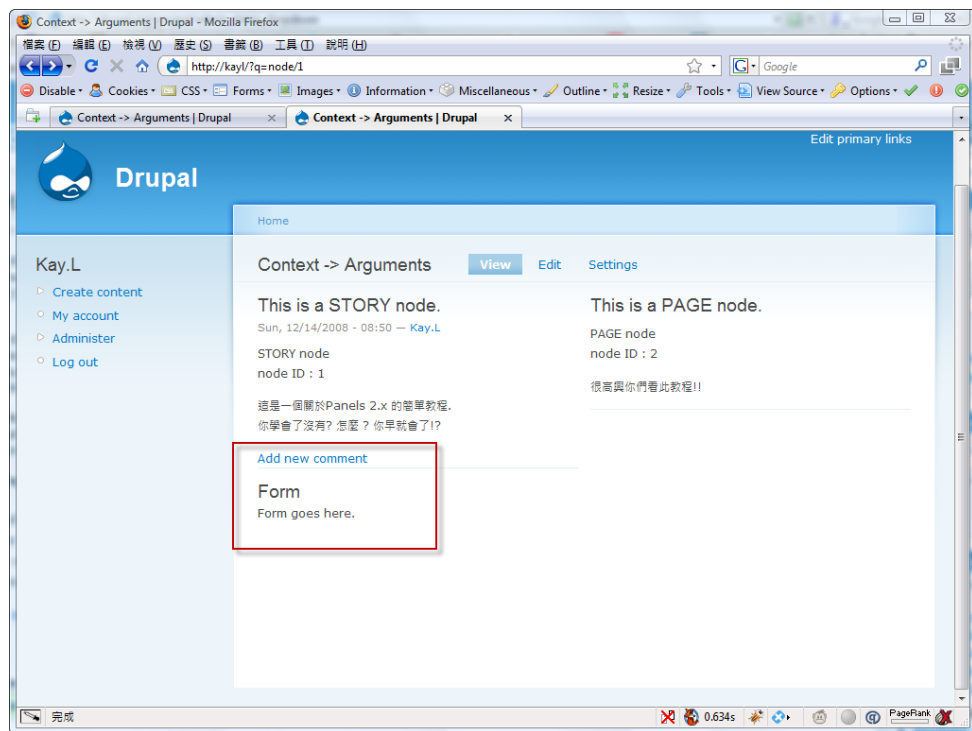


圖 2-7

2-8. 對!! 因為我們瀏覽了錯誤的 URL, 正確應該是"?q=node/1/page", "?q=node/1/story".

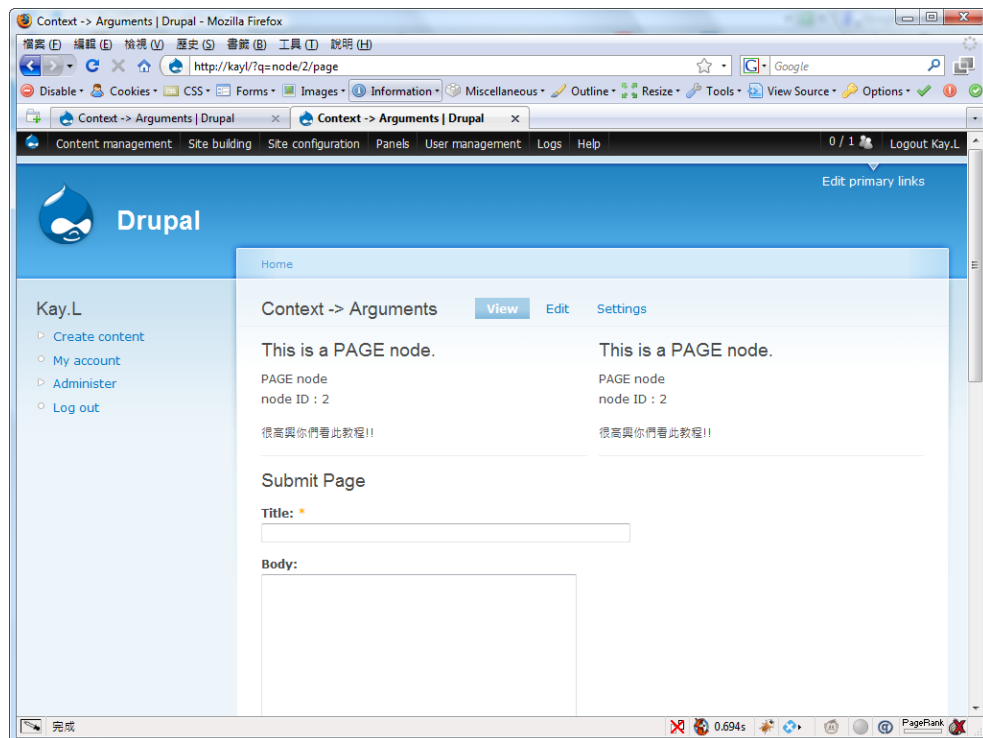


圖 2-8

(抱歉，這張圖片我截了"?q=node/2/page", 所以 add form 上的內容跟圖 2-7 有差別)

聰明的你當然發現了：

Node ID: 在 URL 中就是 node 的 id

Node add form: 就是 Content type

TIPS:

- Setting 中的 Path: node/%

"?q=node/1/page" = "?q=node/Argument 1/Argument 2"

- Setting 中的 Path: node/

"?q=node/1/page" = "?q=node/Argument 1/Argument 2"

- Setting 中的 Path: node/%/sth/%

"?q=node/Argument 1/sth/Argument 2"

至於 conTEXT 中的 Contexts 是什麼？

我想亦不用像上面這樣說了。就是設置指定的東西，而非跟隨 URL 而變化。

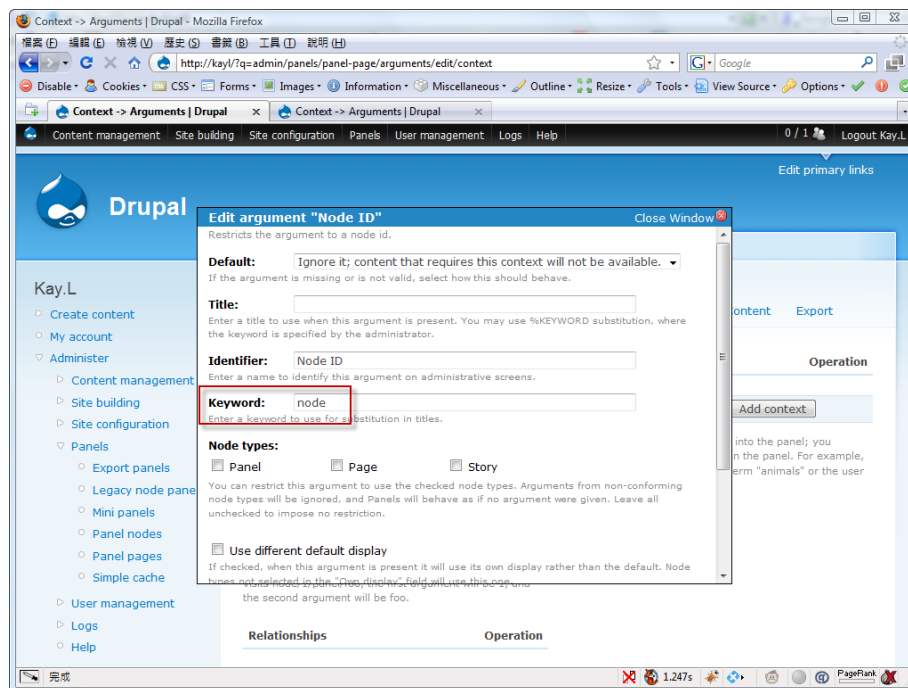
2-9. 好了，到現在你會發現瀏覽"node/1", "node/2"... 標題都一樣的 !!

怎麼解決 ??

其實很簡單，我們可以在 title 輸入 "%node"

是不是解決了？但我還知道你們會問：%node 是怎麼來的？

其實就是 KEYWORD，看圖



2-9

### C. 在不同的 Content type 顯示不一樣的内容

#### 3. 回到 conTEXT, 如圖設定

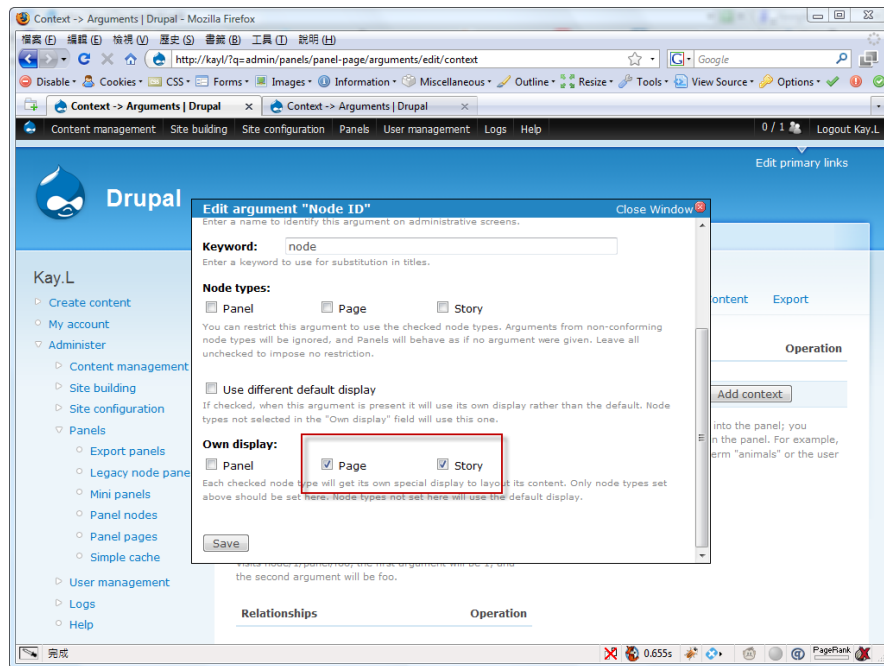


圖 3

3-1. 現在你會在 Layout 及 Content 中看見圖中畫上直線的選項, 隨意設定, 就能做到不同的 Content type 有不同的顯示

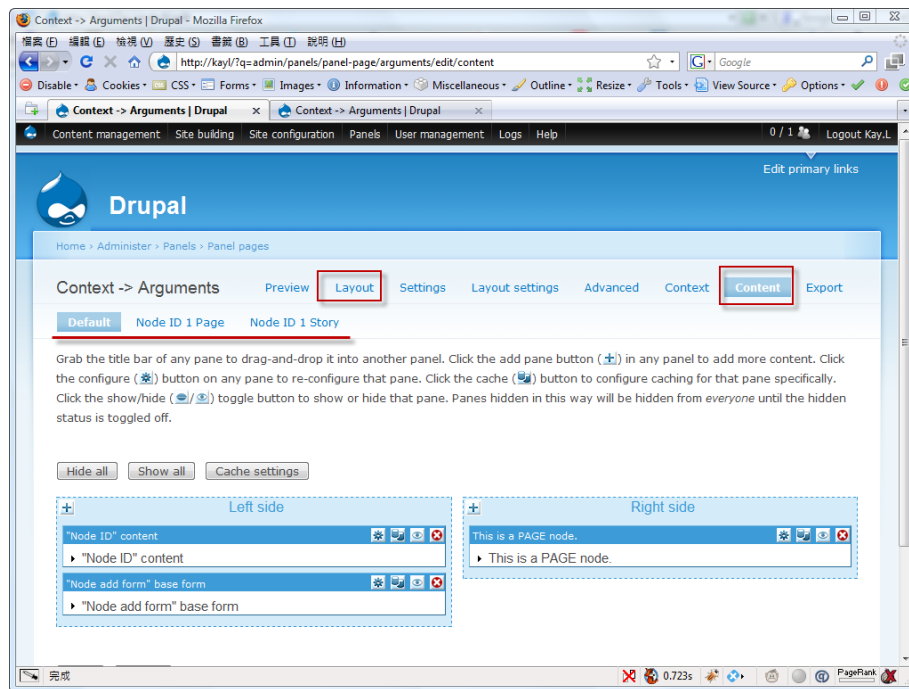


圖 3-1

## D. 限制只在某些 Content type 應用 Panels

### 4. 回到 conTEXT, 如圖設定

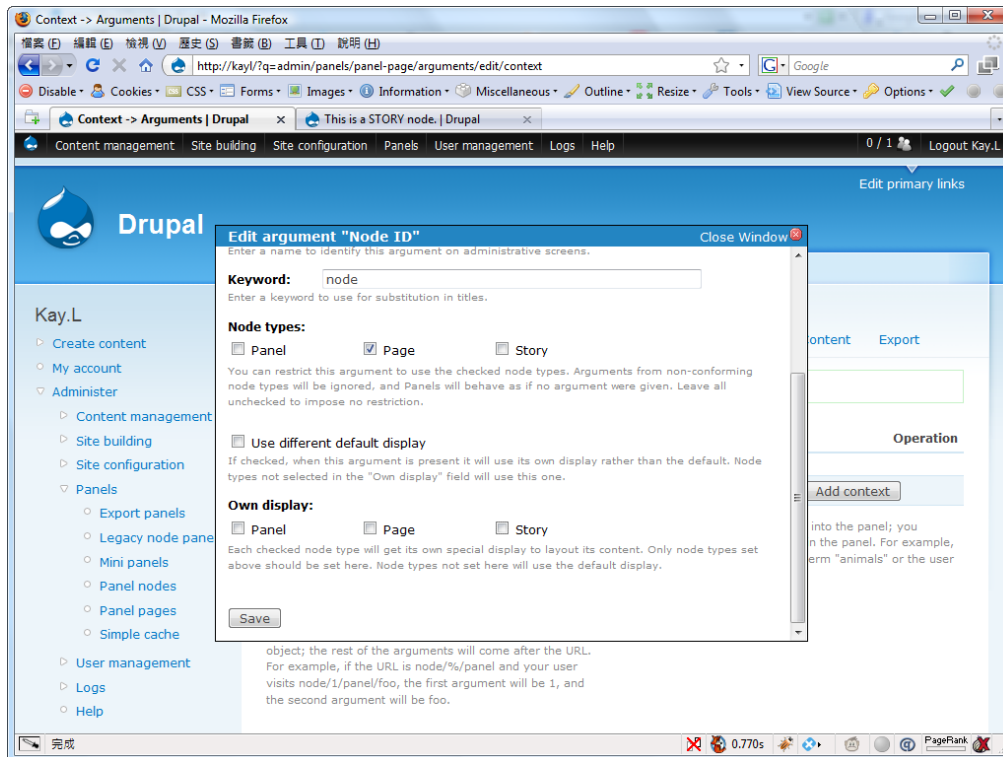


圖 4

現在你會發現只有 content type 是 page 的 node 才會執行 Panels

\*\*\*\*\*END\*\*\*\*\*

## Panels 的使用

Panels 的確是一個非常強大的模組,它的功能在於可以將多頁面, blocks 放在一個自定的網址裏在大型的入門網站,常見複雜的排版,讓用戶可以用最短的時間,拿到最多的資訊

但 Drupal 的頁面只分為 header, left, center, right, footer 不足以應付複雜的排版需要要解決的辦法正是 panels 這個模組

Panels 內建已經有多種常見的排版

如 3-cols, 又有圖示提示,一目了然 詳細的操作也非常簡單

在設定頁的底部有一 add content 先選定要新增的 area 再新增 block, 或 views, 或 node 等



但這次的重點是要說明如何自定一個排版方式

這樣大家都可以自定一個有個性的首頁了 Panels 將數種排版方式名為 layouts

你可以在 panels 的資料夾內找到一個 layouts 資料夾

而一種排版(layout)內有三個檔案,都要放在 layouts/ 內

.css

.inc

.png

假定將要新增的排版名為 **joe**,那三個檔應名為

joe.css

joe.inc

joe.png

joe.png 是一個圖檔 顯示這個排版的樣式,給使用者一個概念,這個排版是什麼樣子的

joe.inc 內有兩個函數:

```
<?php
/**
 * 初始化用的函數, 定義各種參數
 */
function panels_joe_panels_layouts() { //panels_排版名_panels_layouts()
    $items['joe'] = array(
        'module' => 'panels',
        'title' => t('Joe'), //標題
        'icon' => 'layouts/joe.png', //圖示檔名
        'theme' => 'panels_joe', //panels_排版名
        'css' => 'layouts/joe.css', //css 的路徑
        'content areas' => array('left' => t('Left side'), 'middle' => t('Middle column'), 'right' => t('Right
side')), //給每一個區域一個名字, '代碼名'=>t('顯示用的名')
    );

    return $items;
}

/**
 * 顯示用的函數
 */
function theme_panels_joe($id, $content) {
    if ($id) {
        $idstr = " id=' $id' ";
    }
}

/**
 * 排版用的 html tags
 */
$output = <<<EOT
<div class="panel-joe" $idstr>
    <div class="panel-col-first">
        <div>$content[left]</div>
    </div>

    <div class="panel-col">
        <div>$content[middle]</div>
```



```

</div>

<div class="panel-col-last">
  <div>{$content[right]}</div>
</div>
</div>
<br class="panel-clearer" />
EOT;
  return $output;
}
?>

```

初始化的函數只要跟著改 要留意的是 **theme** 函數  
 它定義顯示的方式, 代碼, **id**, **class** 等  
**\$content[left]** 代表用戶設定的 **left** 內的 **block** 或 **views**  
 然後用 **class**, **id** 加上 **css** 的控制 做出無限可能的排版  
 實際用起來難度都不是太大

Joe,

\*\*\*\*\*END\*\*\*\*\*

## 利用 Panels 模組來實做 tab 效果的 block

目的:

在首頁建立一個 tab 效果的 block, 而裡面是幾個獨立的 views, 效果如下, 或參照本站首頁.



安裝步驟:

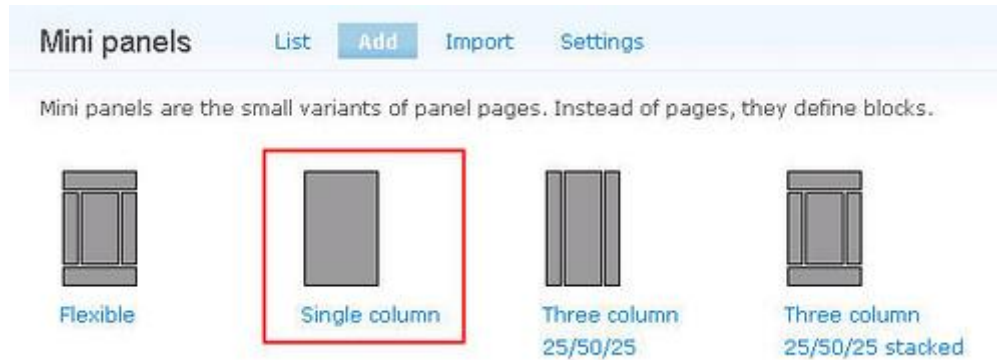
1. 安裝這些模組

- Panels 的 Mini panels
- Panels 的 Panels
- Panels 的 Views panes (這個是因為我要把 views 當成 panels 的內容來源才需要裝的)
- Tabs panel style

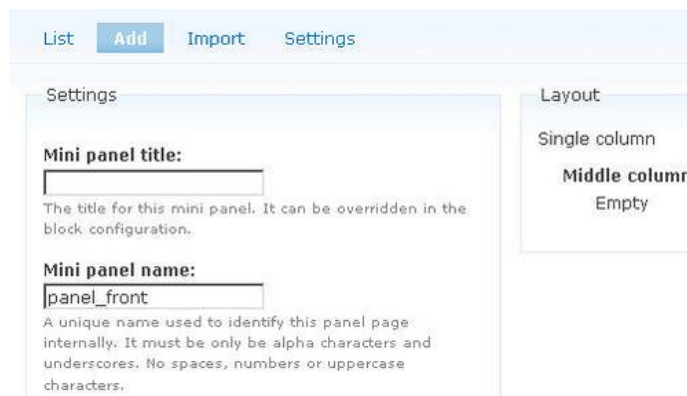
他是有建議也把 Panels page 裝起來, 不過在這個例子用不到, 就先不裝了.

2. 開啟適當權限

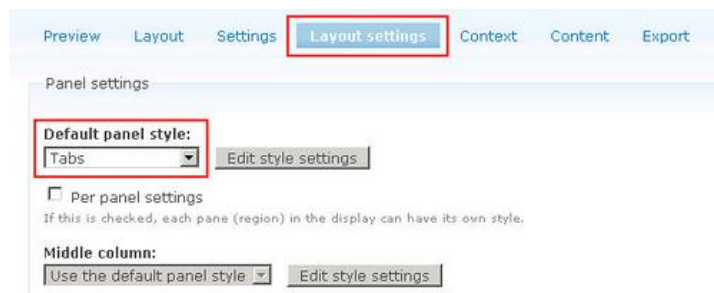
3. 執行功能表 Panels -> Mini panels -> Add, 選擇 Single column



4. 輸入 Mini panel name, 取個自己喜歡的名字, 按下 next



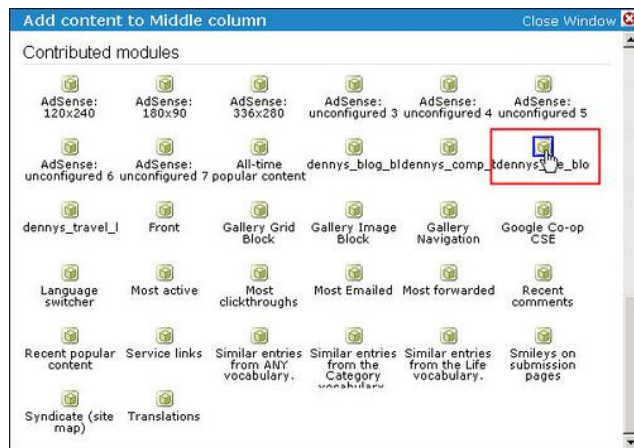
5. 把 Layout 設定改成 "Tabs"



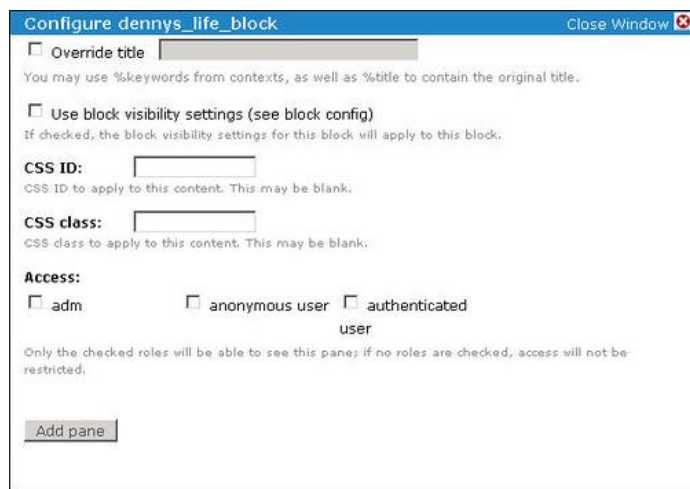
6. Double click Content 這個 tab, 並在 "+" 上面按一下以新增一個 mini panel



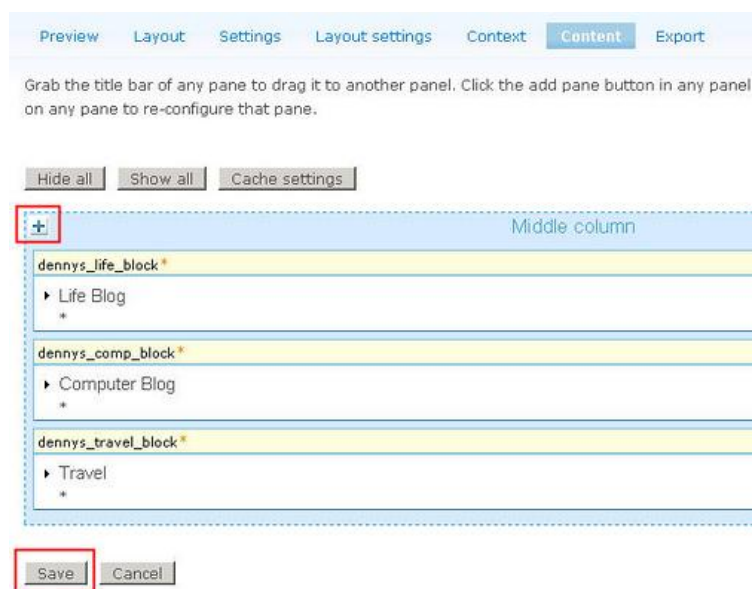
7. 選擇你要放的內容, 我是選了一個之前設定好的 view



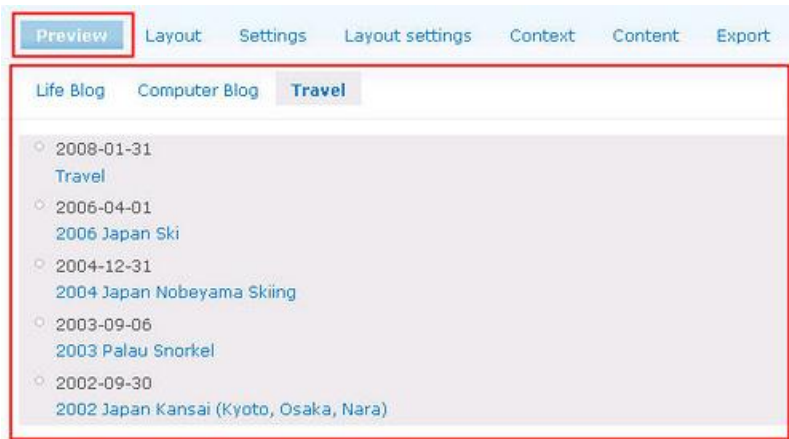
8. 這裡可以指定 title, CSS, 權限等等. 設定好之後按 Add pane 即可



9. 依照上面順序, 依序增加 3 個 pane



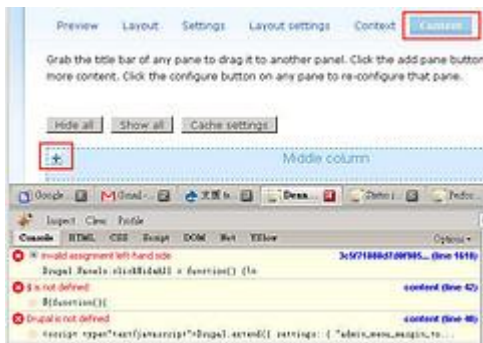
10. 最後按下 Preview, 就可以看到結果了



11. 這時候在 block 就可以看到一個 Mini panel: "Panel\_front" 的 block 可以使用了, 把他放到首頁就可以了.

**注意事項:**

1. 可和 i18n 模組搭配使用
2. 請勿打開 JavaScript Aggregator, 會出現如下錯誤, 已回報 (<http://drupal.org/node/230524>)



可以在 Exclude from js aggregation 設定不處理這些 JavaScript

```

/sites/all/modules/panels_tabs/panels_tabs.js
/sites/all/modules/panels/js/checkboxes.js
/sites/all/modules/panels/js/display_editor.js
/sites/all/modules/panels/js/edit_settings.js
/sites/all/modules/panels/js/layout.js
/sites/all/modules/panels/js/list.js
/sites/all/modules/panels/js/modal_forms.js
/sites/all/modules/panels/js/panels.js
/sites/all/modules/panels/js/lib/dimensions.js
/sites/all/modules/panels/js/lib/form.js
/sites/all/modules/panels/js/lib/mc.js

```

\*\*\*\*\*END\*\*\*\*\*

## drupal 模块教程-pathauto 模块

pathauto 是一个极其灵活的 url 别名模块

### 一、模块安装

1. 将 pathauto 源文件拷贝至服务器 modules 目录下
2. 进入后台管理页面 admin/build/modules, 开启 pathauto 模块

## 二、配置权限

进入权限页面：`admin/user/access#module-pathauto`

	系统管理员	内容管理员	会员	游客
administer pathauto 管理 pathauto	√			

## 三、模块设置

进入模块设置页面：`admin/settings/pathauto`

### 1、General settings（常规设置）

A、Verbose（详细）：`Display alias changes (except during bulk updates)`

（显示 alias 改变，除非大量升级期间）

B、Separator（分隔符）：`Character used to separate words in titles. This will replace any spaces and punctuation characters.`（标题中用来分割单词的符号。将代替空格和标点符号。）

C、Character case（大小写字符）：

a、Leave case the same as source token values.（保持源样式）

b、Change to lower case（改变为小写字符）

D、Maximum alias length（最大 alias 长度）：`Maximum length of aliases to generate. 100 is recommended. See Pathauto help for details.`（Alias 的最大长度，推荐为 100，详情请查看帮助。）

E、Maximum component length（最大组件长度）：`Maximum text length of any component in the alias (e.g., [title]). 100 is recommended. See Pathauto help for details.`（组件的最大长度，推荐为 100，详情查看帮助。）

F、Maximum number of objects to alias in a bulk update（大量升级时最大 alias 改写）：`Maximum number of objects of a given type which should be aliased during a bulk update. The default is 50 and the recommended number depends on the speed of your server. If bulk updates time out or result in a white screen then reduce the number.`

（当一种类型 alias 需要大量升级时，最大允许多少个 alias 被执行。默认为 50，你可以根据服务器来设置数量，当超时或出现白屏时调整数量大小。）

G、Update action（升级设置）：`What should pathauto do when updating an existing content item which already has an alias?`（当升级一个已存在的 alias 时的处理方法。）

a、Do nothing. Leave the old alias intact.（保留原有 alias 样式）

b、Create a new alias. Leave the existing alias functioning.

（创建新的 alias，保留已存在的 alias 设置。）

c、Create a new alias. Delete the old alias.（创建新 alias，删除旧的。）

H、Transliterate prior to creating alias（）：`When a pattern includes certain characters (such as those with accents) should Pathauto attempt to transliterate them into the ASCII-96 alphabet? Transliteration is determined by the i18n-ascii.txt file in the Pathauto directory. This option is disabled on your site because you do not have an i18n-ascii.txt file in the Pathauto directory.`

（）

I、Reduce strings to letters and numbers from ASCII-96（使用 ASCII-96 格式减少字符）：`Filters the new alias to only letters and numbers found in the ASCII-96 set.`

（使用那些 ASCII-96 中存在的字符和数字来过滤新的 alias。）

J、Strings to Remove (取消的字符): Words to strip out of the URL alias, separated by commas. Do not place punctuation in here and do not use WYSIWYG editors on this field. (在 url alias 中去掉的字符, 使用逗号分离, 不要将标点放在最前面, 也不要使用所见即所得编辑器。)

2、Punctuation settings (标点设置)

- A、Double quotes (双引号)
- B、Single quotes (apostrophe) (单引号 (省略号))
- C、Back tick (后引号)
- D、Comma (逗号)
- E、Period (句号)
- F、Hyphen (连字符)
- G、Underscore (下划线)
- H、Colon (冒号)
- I、Semicolon (分号)
- J、Pipe (|)
- K、Left curly bracket (左大括号)
- L、Left square bracket (左中括号)
- M、Right curly bracket (右大括号)
- N、Right square bracket (右中括号)
- O、Plus (加号)
- P、Equal (等于)
- Q、Asterisk (乘号)
- R、Ampersand (&号)
- S、Percent (百分号)
- T、Caret (^)
- U、Dollar (美元符号)
- V、Hash (#号)
- W、At (@符)
- X、Exclamation (!感叹号)
- Y、Tilde (~)
- Z、Left parenthesis (左括号)
- AA、right parenthesis (右括号)
- BB、Question mark (问号)
- CC、Less than (<小于号)
- DD、Greater than (>大于号)
- EE、Back slash (\)

3、Blog path settings (博客路径设置) — 其他设置相同

- A、Pattern for blog page paths (博客页面路径样式)
- B、Bulk generate aliases for blogs that are not aliased (设置未设置 alias 的博客路径)
- C、Create feed aliases (创建 feed 路径)

\*\*\*\*\*END\*\*\*\*\*

Drupal-pathauto 模块教程

pathauto 模块是自动产生路径别名的模块, 装上这个模块以后, 你可以节省为每个文章添加别名的劳动。

下面介绍这个模块的使用教程。

首先看 pathauto 的整体配置图如下:

## Pathauto

- ▷ 常规设置
- ▷ Punctuation settings
- ▷ Blog path settings
- ▷ Forum path settings
- ▷ Node path settings
- ▷ Category path settings
- ▷ User path settings

保存配置

恢复到默认值

### 1、常规设置:

▼ 常规设置

Verbose  
Display alias changes (except during bulk updates).

**Separator:**  
  
Character used to separate words in titles. This will replace any space

**Character case:**

Leave case the same as source token values.

Change to lower case

**Maximum alias length:**  
  
Maximum length of aliases to generate. 100 is recommended. See [Pa](#)

**Maximum component length:**  
  
Maximum text length of any component in the alias (e.g., [title]). 100

**Maximum number of objects to alias in a bulk update:**  
  
Maximum number of objects of a given type which should be aliased of your server. If bulk updates "time out" or result in a "white screen"

**Update action:**

Do nothing. Leave the old alias intact.

Create a new alias. Leave the existing alias functioning.

Create a new alias. Delete the old alias.

What should pathauto do when updating an existing content item which already has an alias?

Transliterate prior to creating alias  
When a pattern includes certain characters (such as those with accents) should be transliterated to ASCII. This is determined by the i18n-ascii.txt file in the Pathauto directory. **This option is disabled by default.**

Reduce strings to letters and numbers from ASCII-96  
Filters the new alias to only letters and numbers found in the ASCII-96 set.

**Strings to Remove:**

Words to strip out of the URL alias, separated by commas. Do not place punctuation

常规设置包含很多需要设置的选项。

verbose:这个目前还没搞太清楚什么意思。

Separator: 当遇到空格或者特殊字符时,用这个分隔符来代替。

Character case: 分两个选项,第一个是维持原状,第二个是都变成小写(这个主要针对英文)

Maximum alias length: 别名的最大长度,推荐是 100

Maximum component length: 组成别名的文本的最大长度,推荐是 100

Maximum number of objects to alias in a bulk update: 当批量更新别名时,最大的更新数目,推荐是 50。

update action:更新时产生别名的的方式。有三种,第一是维持原状,第二创建一个新的别名,但是留着原来别名的一些功能,第三创建一个新的别名,但是原来的别名删除。

## 2、punctuation settings (特殊符号的设置方式)



Punctuation settings

Double quotes " :  
Remove

Single quotes (apostrophe) ' :  
Remove

Back tick ` :  
Remove

Comma , :  
Remove

Period . :  
Remove

Hyphen - :  
Remove

Underscore \_ :  
Remove

Colon ::  
Remove

double quotes " :有三个选项,第一"remove"就是删除这个双引号,第二 replace by separator, 就是让前面"常规设置"里设置的 Separator 来代替这个双引号,第三, no action(do not replace)维持原状不变。

下面的所有的特殊标点符号的设置跟上面的说明类似。

## 3、blog path settings(blog 的别名路径的设置)



Blog path settings

Pattern for blog page paths:  
blogs/[user-raw]

Replacement patterns

Bulk generate aliases for blogs that are not aliased  
Generate aliases for all existing blog pages which do not already have alia

Internal feed alias text (leave blank to disable):  
feed  
The text to use for aliases for RSS feeds. Examples are "feed" and "/0/feed"



```

Replacement patterns
Use -raw replacements for text to avoid problems with t

[user]
  User's name

[user-raw]
  User's unfiltered name. WARNING - raw user inp

[uid]
  User's ID

[mail]
  User's email address

[reg-date]
  User's registration date

[reg-since]
  Days since the user registered

[log-date]
  User's last login date

[log-since]
  Days since the user's last login

[date-in-tz]
  The current date in the user's timezone

[account-url]
  The URL of the user's profile page.

```

pattern for blog page paths:这个用于设置 blog 别名路径的模式。

"blogs/[user-raw]"这个值表示别名的路径是 blogs/经过过滤的用户名，这种模式。

后面的变量值[user-raw]可以用下面 replacement patterns 里的值来代替。

bulk generate aliases for blogs that are not aliased:这个选项选中的作用就是给那些没有别名的 blog 页面，批量产生别名。

internal feed alias text(leave blank to disable):内部 rss 种子需要用的别名形式。

#### 4、forum path settings(forum 的别名路径设置)

用法参考上面的 blog 路径的设置

#### 5、node path settings(node 的别名路径设置)

用法也参考上面的 blog 路径的设置

#### 6、category path settings(category 的别名路径设置)

用法也参考上面的 blog 路径的设置

#### 7、user path settings(user 的别名路径设置)

用法也参考上面的 blog 路径的设置

\*\*\*\*\*END\*\*\*\*\*

### drupal 教程 qiucktabs 模块教程【原创】

首先简单介绍一下 qiucktabs 模块,这个模块的主要作用就是制作出很多样式各异的标签切换效果。当然这个模块的局限性是，这些样式各异的标签效果，只能以 block 的形式显示出来。

使用教程如下：

- 1、在《站点创建》模块里启用 quicktabs 模块。
- 2、在《站点配置》Quick Tabs 里面对 tabs 的样式进行设置。

如下图 1，一下就能看出，这里需要你选择标签想要显示的样式效果。



图 1

Show block titles within Quick Tabs blocks

**Choose how your views should be built within**

- Embed
- Block
- Page

**Choose an effect for your tabs:**

- None
- Slide
- Fade

图 2

图 2“Show block titles within Quick Tabs blocks”的含义是在创建的 quick tabs 的区块里显示区块的标题。

“Choose how your views should be built within Quick Tabs blocks: ”

选择你的 quicktabs 模块是以什么方式建立。有三种选择，第一种是插入方式，第二种是区块，第三种是页面的方式。

“Choose an effect for your tabs: ”选择 quicktabs 模块在标签切换时的效果。也有三种选择，第一是切换时没有效果，第二是切换时滑动的效果，第三是切换时逐渐出现内容的效果。这三种方式你可以自己去测试验证。

3、上面进行完 quicktabs 模块的配置完，下面就要创建 quicktabs 模块了。

在《内容创建》Quick Tabs 里创建 quicktabs 模块实例。点击“New QT block”来创建第一个实例。

如图 3 所示，



“Block Title”用于显示创建的 block 的标题。

“tab title”用于显示创建的标签的名字，

“权重”指的是用于创建的标签的显示顺序，权重越大，标签显示越靠后。

“内容”就是标签里面要显示的内容，目前只能选择 block 类型的内容。

“Create more tabs:”下面的选择框可以选择添加更多的标签，选一个数字以后，点击“creat more tabs”会出现多几个标签，然后继续添加，添加完以后就提交。

提交完以后就进入了 站点创建》区块 设置里。因为此时刚才添加的

quicktabs 实例已经生成为一个 block 了。这时候选择刚才添加的“Block Title”对应的 block 名字，然后设置 block 的顺序就可以了。

## Quicktabs 模块的使用小技巧

Thu, 2008-11-13 20:19 — g089h515r806

Quicktabs 模块很好用，类似的还有 magic tabs 模块。都是用于 tab 切换的。使用这个模块，一是 jason 使用过，我看到他有一篇文章介绍使用心得，另外就是这个模块在 drupalmodules.com 网站的排名很高。所以我决定用它来实现 tab 之间的切换。

Quicktabs 默认是使用鼠标 click 事件，来触动 tab 之间的切换的，客户提出来了需求，那就是鼠标以上就可以了。我说很简单，很好实现。确实很好实现，在 quicktabs\js 中的 quicktabs.js 文件中，有这样一行 jQuery 代码：

```
$(this).bind('click', quicktabsClick);
```

把这一行 jQuery 代码改为

```
$(this).bind('mouseover', quicktabsClick);
```

就可以了。

改完以后，果然实现了客户的要求。但是客户很快就发现了一个 bug，那就是不小心点击的时候页面跑到了上面。怎么办？客户说，用户点击鼠标的时候，也要差生同样的效果。实现起来也很简单。那就是两个事件都加上就可以了：

```
$(this).bind('click', quicktabsClick);
```

```
$(this).bind('mouseover', quicktabsClick);
```

测试了一下，满足了要求，而且问题也解决了。客户很满意。

```
8*****
```

## Views Slideshow

Views Slideshow can be used to create a slideshow of any content (not just images) that can appear in a View. Powered by jQuery, it is heavily customizable: you may choose slideshow settings for each View you create.

這真是一個很酷的模組!!!

這個模組能在 Views 中增加一種 Style - Sildeshow(幻燈片)效果，因此，您可以任意編輯、組合、顯示圖面，文字... 等內容，而且可以 page, block 方式呈現出來。

**安裝說明：**

安裝的方式其實很簡單，您只要下載模組然後解壓縮到 **sites/all/modules/** 目錄下，然後到**管理 > 網站建置 > 模組 (admin/build/modules)** 下啟用此模組。

模組下載連結：

[http://drupal.org/project/views\\_slideshow](http://drupal.org/project/views_slideshow)

```
*****END*****
```

## DRUPAL 需要的模組

已經用上，或者即將用上，先收藏備用。

### AdSense

模組介紹：

這個 Google AdSense 模組讓你可以在網站上顯示 Google 的 AdSense 廣告，藉此賺點小錢。

此模組可以設定廣告的類型（文字/圖片）、顏色（組合），並支援多種廣告格式與廣告 channel 設定，還可以針對特定使用者角色隱藏廣告（例如付費使用者），或是在某些頁面上不顯示廣告。

特別的是，它還可將特定節點類型所獲得的廣告收入，拆分給作者，或是介紹網友點選廣告的使用者。

連結網址：<http://drupal.org/project/adsense>

### A-dvertisement

Advertisement 模組（用字典查這個英文意思是廣告、宣傳的意思 - 名詞）

簡稱 A-d 模組

在繁體站上找到的 Banner 模組沒有 5.x 版的但是竟然自己也沒有想到英文的 drupal 網站上已經有另一套廣告模組了!!所以找不到廣告模組的可以用看看這個模組

連結網址：<http://drupal.org/project/ad>

### Akismet

模組介紹：

這個模組讓網站可以使用 Akismet 的網路服務，減少垃圾廣告對你網站的攻擊。

Akismet 可以說是目前 Drupal 與 WordPress 上較有效阻止垃圾廣告的方法。要使用此服務，必須先於 Wordpress 上面免費註冊得到一個 key。

連結網址：<http://drupal.org/project/akismet>

### audio

模組介紹：

可以上傳聲音/音樂檔案到網站上，讓網友串流播放或是下載收聽。每一個檔案都有一個自己的節點，並會自動讀取/呈現檔案裡的 ID3 資訊。網站上可以列出最新的檔案，所產生的 RSS feed 會加上多媒體資訊，例如類型、長度、作者等，可作為 podcast 之用。

連結網址：<http://drupal.org/project/audio>

### Backup and Migrate 備份與搬移模組

模組介紹：

這個模組提供簡易的備份或資料庫搬移功能，備份資料可儲存在網站上，或下載儲存。

支援壓縮格式檔案（GZip, Bzip, Zip... 主機有提供的話）的備份與回存、定期備份以及手動回存。

另可自由篩選不必要的資料表格或資料內容備份。

連結網址：

[http://drupal.org/project/backup\\_migrate](http://drupal.org/project/backup_migrate)

### Banner Rotor Module

在區塊裡顯示可輪替的圖片，可作為內容或商品的廣告推薦。

連結網址：<http://drupal.org/project/rotor>

### Block Bar

模組介紹：

可以建立多個用來含括其他區塊的母區塊，可以設定為展開或收合子區塊，並可以記住現有的開啟或關閉狀態。也可以限定母區塊在同一時間只能開啟一個子區塊。

連結網址：

<http://drupal.org/node/36373>

## captcha

模組介紹：

在使用者註冊/回應時，加入「圖形檢核碼」的機制，防止有人使用惡意程式大量註冊。

另有「加法版」的模式可供選擇（會問你  $12+5=?$  這類的問題），避免 accessibility 的問題。

由 lee 於 2007/05/28 新增以下內容：

此模組須搭配 Text Image 模組方可顯示圖形檢核碼

Text Image 連結網址：<http://drupal.org/project/textimage>

更新：by Hipfox 2007/07/23

Official releases: 5.x-3.0-rc1 2007-Jul-20 354.38 KB

這個版不再相依其他模組，包括 textimage, form store 在內，依靠自帶的功能便能安裝。

**過濾的機制有三種：**1. 算術問題 2. 圖形檢核碼 3. 文字

其中「圖形檢核碼」的部分，已附一些字型可供使用，並且可以調整適合顯示的文/數字，可把容易混淆的文字，按照需求自行定義。文字過濾機制，可亂數產生或自行預設 20 組以上的字串。

另外改良前一版必需依靠 form store 才能對其他表單加上 CAPTCHA 的不便，模組本身有個選項可以在網站中看得到的表單，為其附加 CAPTCHA。

模組的設定位置：使用者管理 -> CAPTCHA

之前的中文翻譯檔，因為本版進化不少，不太適用，必須重翻。

更新：by Dennys 2008/02/11

補充說明一下 arbitrary forms 的使用。在 CAPTCHA 3 之後，預設是可以設定 comment, contact 以及 user 相關的幾個 form 要使用 CAPTCHA。至於其他的地方或模組，譬如 forward 或 guestbook 之類的，可以把 "Add CAPTCHA administration links to forms" 選項打開，然後用有 administer CAPTCHA settings 權限的帳號進到那個頁面，就會出現一個 "Place a challenge here for untrusted users." 的連結，點下去之後，在 CAPTCHA 的設定頁面 (admin/user/captcha) 就會出現剛剛指定的新 form 了。

還有就是從 2.x 升級上來的，似乎預設只有打開 comment 的 CAPTCHA，如果要加上 contact 的 CAPTCHA 也是一樣用上面的方法設定就可以了。

另外就是，模組的名字已經改成全大寫的 CAPTCHA，請參考 <http://groups.drupal.org/node/5834>

連結網址：<http://drupal.org/project/captcha>

## CAPTCHA Pack

模組介紹：

包括多種檢核碼(CAPTCHA)工具，可與 CAPTCHA 模組一起運用，以滿足不同需求（例如頻寬、運算、近用等限制）。

功能包括：

- 強化算數檢核碼：" $2 * ? = 6$ "
- 強化文字檢核碼：克漏字、同義字等
- CSS 檢核碼
- ASCII art 檢核碼
- 隨機運用以上各種類型的檢核碼

連結網址：

[http://drupal.org/project/captcha\\_pack](http://drupal.org/project/captcha_pack)

## Content Construction Kit (CCK)

模組介紹:

用來建立各種特定用途的內容類型的模組，透過瀏覽器設定你想要使用的內容欄位，就能自訂內容類型，包括呈現的方式。除了內建的文字、選單、數字等欄位之外，還可加掛許多模組，提供運算、日期、email、圖片、連結等功能，以增添其功用。

連結網址:

<http://drupal.org/project/cck>

## Content Templates (Contemplate)

模組介紹:

可以很容易地調整每個內容類型的摘要 (teaser) 和全文的顯示欄位，它會把可使用的物件列出來，只要在上面按一下，就會把對應的程式碼加到版型裡，並可手動作細部調整。適合與 CCK 合用。

連結網址:

<http://drupal.org/project/contemplate>

## Control Panel

模組介紹:

提供圖示化的管理面板。管理者可以從選單裡指定控制面板的來源，並可設定子面板 (選單) 的層級數，也可以在區塊裡設置管理面板。圖示化的管理介面，會讓初學者感覺非常友善，也可以隨己意更換圖示。

連結網址:

<http://drupal.org/node/32400>

## Copyright

模組介紹:

可以設定網站整體和單篇內容的著作權/授權聲明。管理者可以設定可選用的著作權種類，呈現在區塊或頁尾裡。手冊頁面 (Book) 可以選擇性地設定所有子頁面為同一著作權聲明，並讓每一份手冊或區域有其自己的著作權聲明。

連結網址:

<http://drupal.org/project/copyright>

## Counter

模組介紹:

提供簡單好用的計數器功能，包括：網站計數器 獨立使用者統計

- 註冊/未註冊使用者統計
- 已發表/未發表文章統計
- 顯示來訪者 IP 位址
- 報表：來訪者 IP、存取時間/頁面

圖示:



連結網址:

<http://drupal.org/project/counter>

### CustomError

模組介紹:

讓網站管理者可以直接設定 404/403 的內容, 可以使用 HTML 語法, 不需要另外建立網頁, 頁面裡也就不會顯示不必要的作者和日期資訊。

連結網址:

<http://drupal.org/project/customerror>

### Dashboard

模組介紹:

提供一個簡單的節點類型, 可以作為一種靜態頁面, 以 header/footer+兩欄的方式呈現資訊。我覺得此模組一個另類的功能在於, 提供了 `dashboard_get_block('forum',1)` 這種 function, 可以很方便地在頁面裡顯示特定區塊的內容。

連結網址:

<http://drupal.org/project/dashboard>

### Demonstration site (Sandbox)

模組介紹:

這個模組可以幫助你對 Drupal 網站製作一或多個備份映象檔, 可定期排程或手動操作還原映象記錄的狀態。

因此, 對於瘋狂的模組測試者, 或是有提供公共展示的網站, 要還原之前的設定狀態會方便許多。

連結網址:

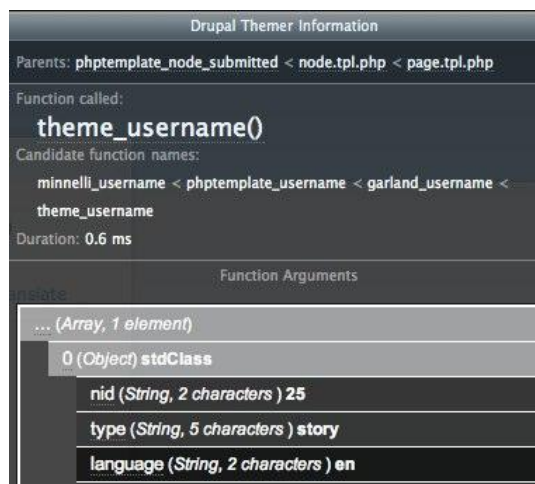
<http://drupal.org/project/demo>

### Devel

模組介紹:

提供模組與版型開發的工具, 包括:

- 提供資料庫查詢結果列表
- 列印陣列內容
- 版型開發工具 (Firebug for Drupal!!! D6 only)。範例。
- 產生測試用資料, 例如內容、回應、分類、使用者等
- 內容存取摘要



連結網址:

<http://drupal.org/project/devel>

### **Dynamic Logo**

模組介紹:

能夠讓管理者上傳一個以上的 logo 而且能夠在特定的時間顯示某個 logo

連結網址:

[http://drupal.org/project/dynamic\\_logo](http://drupal.org/project/dynamic_logo)

### **Flickr**

模組介紹:

提供一個過濾器，可以在文章裡插入 Flickr 裡的照片和相簿，或是建立照片和相簿的區塊。

採用的輸入格式為:

[flickr-photo:id=230452326,size=s]

[flickr-photoset:id=72157594262419167,size=m]

\* s - small square 75x75

\* t - thumbnail, 100 on longest side

\* m - small, 240 on longest side

\* - - medium, 500 on longest side

\* b - large, 1024 on longest side

\* o - original image

連結網址:

<http://drupal.org/project/flickr>

### **Form Block**

模組介紹:

讓建立內容的表單可以顯示在區塊裡，很適合與 Panels 模組一起使用。

連結網址:

<http://drupal.org/project/formblock>

### **Front Page**

模組介紹:

Front Page 可以讓你指定並編輯網站的形象頁 (splash page, 通常是那種大而不當的 flash 動畫...), 不需要套用預設的版型和樣式, 也可以針對已登入會員和匿名使用者, 呈現不同的頁面。

這個首頁設定, 跟 Drupal 原本的首頁設定是不同的。它只會在剛進入網站時出現, Drupal 網站裡其他的首頁連結 (例如 path 裡的 home), 都會導到正常的首頁, 而不是一再地播放形象頁。

連結網址:

<http://drupal.org/project/front>

### **Gallery**

模組介紹:

此模組是將 Drupal 和網路相簿系統 gallery2 (<http://gallery.menalto.com>) 做整合, 可以設定區塊(block)隨機顯示照片。但要注意原本相簿的名稱不能取 gallery, 會跟 Drupal 衝到。

連結網址:

<http://drupal.org/project/gallery>

### **Google Translate**



模組介紹:

利用 Google 所提供的翻譯功能，為網站提供不同語言的瀏覽介面與內容翻譯（當然，可讀性仍待改善）。

示範網站: <http://drupal.owahab.com/5x/>

連結網址:

<http://drupal.org/project/gtrans>

## Header image

模組介紹:

可以在特定頁面顯示特定圖片，例如網站的各個單元首頁上方的視覺圖示。可以在同一頁面的多個地方同時顯示不同圖示。

範例網站:

<http://link3.nl/>

連結網址:

<http://drupal.org/project/headerimage>

## Image Caption

模組介紹:

使用 JQuery 動態地把圖說加到圖片旁。可以與 FCKEditor 或 TinyMCE 整合運用。

使用範例:

```

```

範例圖片:



連結網址:

[http://drupal.org/project/image\\_caption](http://drupal.org/project/image_caption)

## Insert View

模組介紹:

Insert Views 提供一個 filter，讓你可以在發表文章時用簡單的語法插入既有的 view

語法範例:

**[view:name of view]** 插入該 view 篩選出的 node

**[view:name of view=x]** 插入 x 則該 view 篩選出的 node

當然，views 需要先安裝好才能運作

連結網址:

[http://drupal.org/project/insert\\_view](http://drupal.org/project/insert_view)

## Internationalization

模組介紹:

提供多國語言網站的建置。i18n 包含多種模組，支援一個多國語言網站的建制。

1. 多國語言的 vocabulary 分類表，對應的語言顯示對應的分類
2. 多國語言的 block 區塊，對應的語言顯示對應的 block
3. 多國語言的 node 內容，對應的語言顯示對應的 node
4. 多國語言的 menu 選單，對應的語言，顯示對應的 menu
5. 多國語言的 variable 變數，對應的語言，顯示對應的變數（如 site slogan, site name 等變數）。
6. 多國語言的切換，提供一個預設的 block 區塊，讓人方便切換不同語言

連結網址:

<http://drupal.org/project/i18n>

### **jQuery FontSizer**

模組介紹:

即時變換 字體大小

連結網址:

[http://drupal.org/project/jquery\\_fontSizer](http://drupal.org/project/jquery_fontSizer)

### **Legal**

模組介紹:

在註冊時顯示網站的使用條款，使用者必須接受使用條款，才能完成註冊程序。

連結網址:

<http://drupal.org/project/Legal>

### **logo tool**

模組介紹:

更換版型 logo 圖檔的模組

可以設定隨機或定時更換也可以指定特定網頁使用特定 logo

強烈推薦

連結網址:

<http://drupal.org/project/logotool>

### **Menu Block Split**

模組介紹:

可以把選單分成兩個區塊，第一個區塊是第一層選單，第二個區塊是第二層及其以下的選單

範例網站:

[http://menu\\_block\\_split.demo.garrigos.cat/](http://menu_block_split.demo.garrigos.cat/)

連結網址:

[http://drupal.org/project/menu\\_block\\_split](http://drupal.org/project/menu_block_split)

### **Meta tags**

模組介紹:

可以在每篇文章、view 或 panel 裡加上 meta tags（關鍵字、說明等等），對於 SEO 有幫助。

連結網址:

<http://drupal.org/project/nodewords>

**Module Installer**（建议本地测试用）

模組介紹:

安裝此模組後，只要在管理介面輸入想要安裝的模組的檔案網址，就可以直接下載並安裝模組。

連結網址:

[http://drupal.org/project/module\\_installer](http://drupal.org/project/module_installer)

## Mollom

模組介紹:

### 1. 簡介

Dries Buytaert (Drupal 創始人) 和 Benjamin Schrauwen 在幾個月的努力後，發佈了 [Mollom](#)——自動化的內容監控軟體。Mollom 結合了最新的 spam 過濾和 [CAPTCHA](#)，目的是希望大幅減少網站的清潔工作、讓使用者產生的內容維持高品質。

如果你對 Mollom 有興趣，請閱讀 [How Mollom works](#) 頁面，和 [Mollom FAQ](#) 瞭解更多細節。

[Mollom 模組](#) 目前已有 Drupal 5.x 和 Drupal 6.x 版本。如果你是開發人員，想要利用 Mollom API 建立 plug-in，請參見 [開發文件](#)。除了 Drupal 以外，還提供 Java、Ruby、Python、PHP5 等版本。目前，Mollom 是免費的，未來則會有企業版。

### 2. 原理

Mollom 可以分析使用者所產生的內容的品質。這些包括評論、聯絡資訊表單、部落格、論壇主題等等。Mollom 可以決定這個內容究竟是不想要的(spam)、還是想要的(ham)。所有允許使用者貢獻內容、張貼評論的網站，都可能被不當的、甚至非法的訊息淹沒，其中大部分都是由 spam 機器人(spambot)所產生的。Mollom 可以在他們張貼之前，就先監控所有內容。

使用 Mollom 的網站將需要檢查的內容送至 mollom.com，然後 Mollom 會判斷這是 spam 還是 ham。如果 Mollom 無法判別，就會回報「不確定」，然後請求 Mollom 的 CAPTCHA 伺服器，讓使用者回答視覺或聽覺的 CAPTCHA 問題。多虧有「不確定」分類和 CAPTCHA，Mollom 可以避免將合法內容誤判為 spam。結合文字分類和 CAPTCHA 技術有兩個優點:

1. 可以有效減少誤判，因為這些內容不可能來自合法使用者；
2. 因為 CAPTCHA 很少詢問合法使用者（目前只有約 2% 的人類使用者會遇到 CAPTCHA 問題），可以讓網站更容易親近，同時改善整體品質。

Mollom 的關鍵特色之一是，所有參與的網站都可以回報漏網之魚的 spam，Mollom 整合這些資訊，從中學習，以避免將來的濫用。

### 3. 站外連結

- CenterNetworks » [Mollom Just Gave Me Back An Hour Per Day](#)
- SilenceWolf » [Mollom: 博客屏蔽垃圾留言的新选择](#)

### 4. 內容來源

- Drupal.org » [Mollom, my content monitoring startup](#)
- mollom.com » [How Mollom works](#)

連結網址:

<http://drupal.org/project/mollom>

## Moodle Single Signon

模組介紹:

(這個好像教育單位比較會用到)

Moodle 是一個免費 開放原碼的課程內容管理系統

Moodle Single Signon 能夠 shared cookie, and maintains a database that moodle can read.

Uses moodle lazy account creation.

連結網址:

<http://drupal.org/project/moodlesso>

## Nice Menu

模組介紹：

輕易地製作出下拉式導覽選單，包括水平和垂直（選單向左拉或向右拉）兩種形式。

選單項目是從原本的選單而來的，你只要指定哪個選單要採用此模組所提供的功能，就可以達成此效果。



連結網址：[http://drupal.org/project/nice\\_menu](http://drupal.org/project/nice_menu)

## Nodetype

模組介紹：

可以在編輯內容時，改變其內容類型。

連結網址：<http://drupal.org/project/nodetype>

## OG 模組（社群站架設時候使用）

## Openads

模組介紹：

將開源碼廣告系統 [OpenX](#) 整合到 Drupal，可依據角色顯示廣告、顯示在區塊或版型裡。

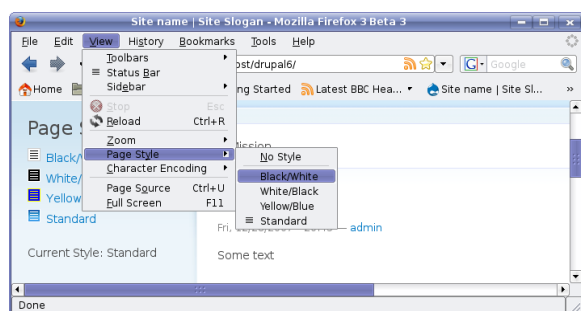
連結網址：<http://drupal.org/project/openads>

## Page Style

模組介紹：

在區塊裡提供一個切換瀏覽樣式的工具，以提供更好的網站近用(Accessibility)。可以與切換版型模組(Switchtheme)合併使用。

範例網站：<http://www.zwahlendesign.ch/drupal6/> 圖示：



連結網址:

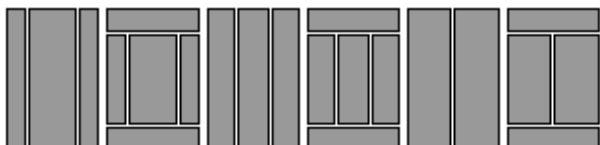
<http://drupal.org/project/pagestyle>

## Panels

模組介紹:

可以建立客製化的頁面，每個頁面裡可以分成多個區域，可以挑選想要顯示的文章、區塊、views 等元件呈現在各區域裡。只要有一點 HTML/CSS 技術，就可以很方便的設計複雜的版型，很適合用來作為網站首頁的呈現，跳脫原本一欄到底的方式。

預設至少有這些版型:



連結網址:<http://drupal.org/project/panels>

## Pathauto

模組介紹:

自動為節點產生網址別名，可針對不同的內容類型（節點、分類、使用者），設定不同的網址格式。

例如：為所有內容都加上.html 的副檔名；或是將使用者的網址以 user name 呈現，而非編號。



連結網址:<http://drupal.org/project/pathauto>

## Picasa

模組介紹:

利用 Google 的 Picasa API，讓使用者在網站上顯示他們的 Picasa 相簿和照片。

範例網站:

<http://www.kevinbridges.org/picasa>

連結網址:

<http://drupal.org/project/picasa>

## Qedit (Batch and mass operation for nodes)

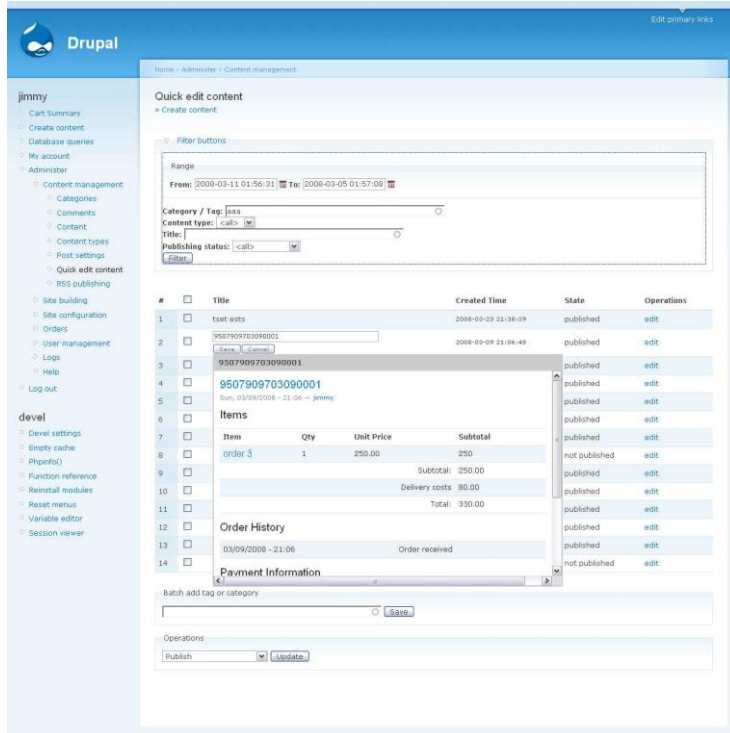
模組介紹:

Quick Edit 是一個取代 Drupal 原有管理 content 的介面。

重要的特色有:

- \* 可以以類型、標籤、分類、標題、和日期等條件篩選 node，然後再進行批次編輯
- \* 可以直接在管理介面中編輯標題和日期 (just click title or date, then appear textfield)
- \* 可以直接預覽文章內容，只要點擊標題列
- \* 跟 drupal 的管理介面一樣，可以批次修改文章的屬性，但除此之外，亦可在此介面批次為 node 增加 tag 和分類

有任何 bug，歡迎到英文官網上 report new bug 給 Jimmy



連結網址：

<http://drupal.org/project/qedit>

## Quick Tabs

模組介紹：

可以建立最多達 6 個頁籤的區塊，用來呈現多樣資訊。同一網頁可以擺放多個多頁籤區塊。頁籤的內容可來自區塊或 views，並可使用一個 views 的多種 arguments 來建立多個頁籤。

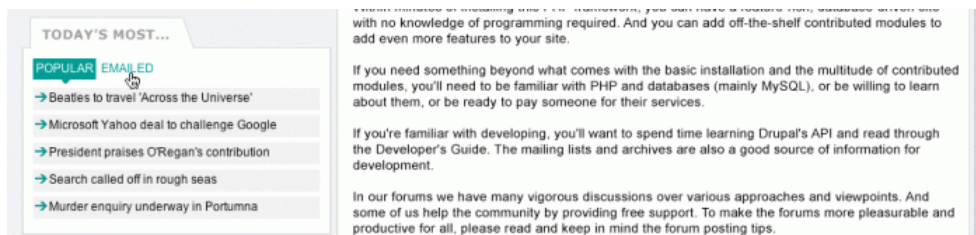
範例網址：

<http://www.katbailey.net/demo/>

相關討論

<http://drupal taiwan.org/forum/20080211/1869>

圖示：



連結網址：

<http://drupal.org/project/quicktabs>

## Search Cloud Generator

模組介紹:

替熱門搜尋字串產生 Tag Cloud, 同一個詞被搜尋越多次則 Tag 的字體會越大  
需啟用 Search Module

使用方法:

在[網站設定] -> [Search Index Cloud]:

可以作細部設定, 不過預設值就可以用了, 所以不改也可以

在[網站建置] -> [區塊]:

將 [Search Index Cloud] 區塊放在適當的位置

隨便使用搜尋功能搜尋某個詞, 就可以看到產生 Tag Cloud Block 有東西產生

實作說明:

這是一個蠻單純的模組

searchcloud.module 主要實做 hook\_search() 和 hook\_block()

searchcloud\_search():

當執行搜尋功能時會被執行, 會將搜尋詞加入 searchcloud table 或更新搜尋次數

searchcloud\_block():

用來產生 Tag Cloud View

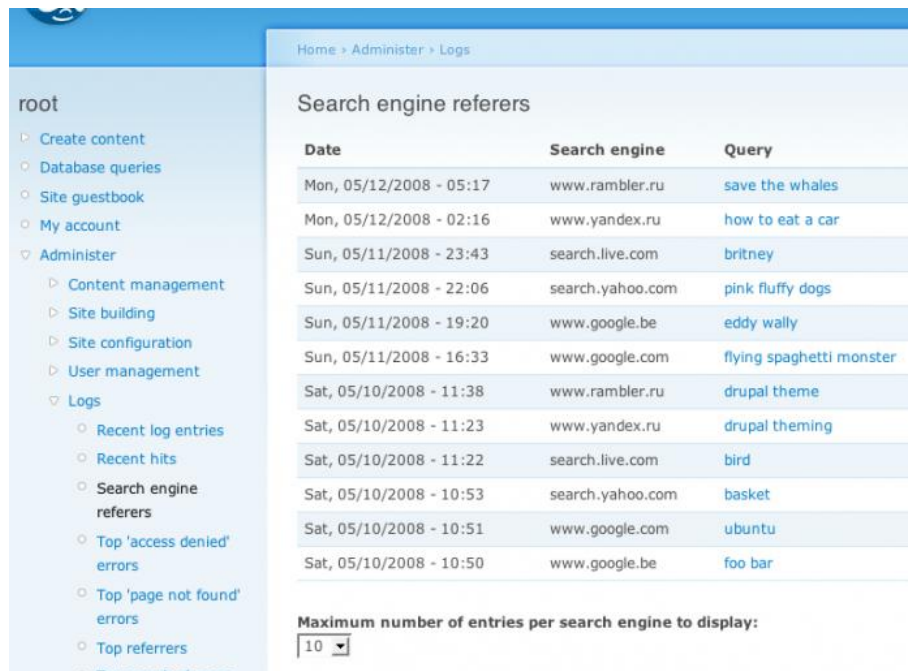
連結網址:

<http://drupal.org/project/searchcloud>

## Search Engine Referers

模組介紹: 列出訪客透過那些搜尋引擎與關鍵字來到你的網站。

圖示:



Date	Search engine	Query
Mon, 05/12/2008 - 05:17	www.rambler.ru	save the whales
Mon, 05/12/2008 - 02:16	www.yandex.ru	how to eat a car
Sun, 05/11/2008 - 23:43	search.live.com	britney
Sun, 05/11/2008 - 22:06	search.yahoo.com	pink fluffy dogs
Sun, 05/11/2008 - 19:20	www.google.be	eddy wally
Sun, 05/11/2008 - 16:33	www.google.com	flying spaghetti monster
Sat, 05/10/2008 - 11:38	www.rambler.ru	drupal theme
Sat, 05/10/2008 - 11:23	www.yandex.ru	drupal theming
Sat, 05/10/2008 - 11:22	search.live.com	bird
Sat, 05/10/2008 - 10:53	search.yahoo.com	basket
Sat, 05/10/2008 - 10:51	www.google.com	ubuntu
Sat, 05/10/2008 - 10:50	www.google.be	foo bar

Maximum number of entries per search engine to display:  
10

連結網址:

[http://drupal.org/project/search\\_engine\\_referers](http://drupal.org/project/search_engine_referers)

## SEO Checklist

模組介紹:

提供一個關於搜尋引擎最佳化 (SEO, Search Engine Optimization) 的檢查表, 包含多種模組與外部工具的運用, 讓你可以一步步調整網站, 讓搜尋引擎幫你帶來訪客。

連結網址:

[http://drupal.org/project/seo\\_checklist](http://drupal.org/project/seo_checklist)

## Shadowbox

模組介紹:



一個強大的線上媒體瀏覽器

支援

- \* Images (png, jpg, jpeg, gif, bmp)
- \* Video (avi, mpg, mpeg)
- \* Quicktime (dv, mov, moov, movie, mp4)
- \* Windows Media Player (asf, wm, wmv)
- \* Flash (swf, flv)
- \* IFrame (asp, aspx, cgi, cfm, htm, html, pdf, pl, php\*, phtml, rb, rhtml, shtml, txt, vbs)

看來功能非常強也非常炫

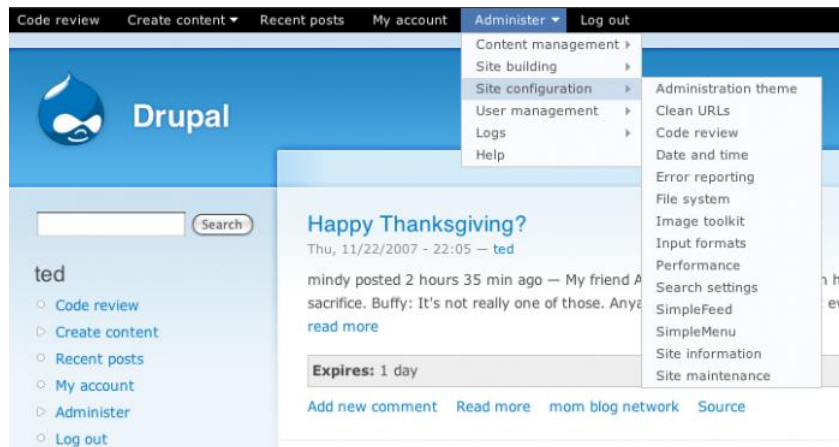
連結網址:

<http://drupal.org/project/shadowbox>

## SimpleMenu

模組介紹:

SimpleMenu 會在每個網頁的最上端顯示導覽選單 (menu bar), 它使用 JQuery 固定其位置, 並且與 IE 相容。非常適合用於難以放置管理區塊的版型。





連結網址:

<http://drupal.org/project/simplemenu>

### Site map

模組介紹:

提供一個"網站地圖", 讓使用者對於你的網站有個概括性的瞭解, 同時也會顯示所有部落格和分類的 RSS。

示範: <http://xdeb.org/sitemap>

連結網址:

[http://drupal.org/project/site\\_map](http://drupal.org/project/site_map)

### Sliced Menu

模組介紹:

Sliced Menu 可以把選單"切"成多個區塊, 每一層選單一個區塊, 然後就可以把各層選單的區塊, 依需要放在想要顯示的地方。例如把第一層選單放在 header, 第二層選單放在左欄。

連結網址:

<http://drupal.org/project/slicedmenu>

### Splash

模組介紹:

將內部/外部網址或文字作為前導頁(splash), 然後才會進入真正的網站首頁。主要功能包括:

- 設定每週、每日、僅一次或總是顯示。
- 可根據特殊狀況(角色、PHP 程式碼)來判定是否要顯示。
- 可以使用 ThickBox 來呈現, 並設定視窗大小。
- 可以隨機或循序選擇一系列前導頁。

連結網址:

<http://drupal.org/project/splash>

### Storm 專案管理 (HelpDesk, Tickets...)

模組介紹:

這個模組跟 casetracker 有點像, 卻又大大不同, 主要是用戶權限的劃分不一樣。在 IT 業界常拿來當客服專用的平台, 如 HelpDesk, Tickets 等。

casetracker 的每一個 case 或 project 如果沒有透過 content access 或 acl 等模組來處理瀏覽權限, 用戶是都可以看到每個 case 的資料, 但 Storm 已經把權限的劃分完整設計進來, 不用依靠其它模組。

特色:

- 可以自行調整或創造 tasks status, countries, currencies 之類的屬性。
- 創建公司、組織名稱或你定義的名稱。
- 每個專案可以有重的工作項目定義 (tasks)。
- 工作項目(task): 這是專案中的一個元件。
- ticket: 這真不到該翻譯成什麼哪! 傳票嗎? 好像要進法院的樣子。在提交 ticket 的時候, 可以和專案、工作項目或組織等關連。
- Timetrackings: 你可以為每個專案、組織、工作項目或 ticket 登記時間記錄。
- 為組織中添加聯絡人。
- 很棒的權限劃分, 可以規劃給組織或用戶來分享資料。
- 專案報表輸出。
- 為專案、工作項目等寫備忘。

- 知識分享。
- 會計帳單。

還有其它應用功能等著你來使用喔！

連結網址：

<http://drupal.org/project/storm>

## Subdomain

模組介紹：

可以把網站內容以另一個次網域來呈現，不僅看起來酷，對於 SERP/SEO 也有幫助。目前支援三種模式：

作者：

如果作者叫作 charlesc，他的文章就會出現在 charlesc.drupal taiwan.org

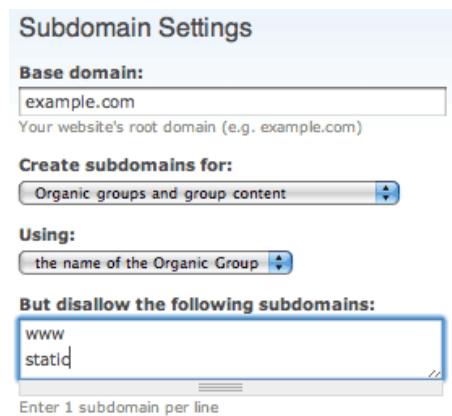
分類：

如果有個分類叫作 modules，那麼相關文章就會出現在 modules.drupal taiwan.org

Organic Group：

如果有個社群叫作 taipei drupaler，該社群的文章就會出現在 taipei-drupaler.drupal taiwan.org

這些僅是基本的用法，管理者可以進行客製化。需與 Pathauto 共用，並修改伺服器設定與 Drupal 的核心程式。



連結網址：

<http://drupal.org/project/subdomain>

## SWF Tools

一個 Flash Media Player 可線上播放 mp3 flv swf 等檔案

只要在建立內容時 將 mp3 flv swf 等上傳就可以

[教學影片](#)

連結網址：

<http://drupal.org/project/swftools>

## Tagadelic

模組介紹：

可以為 Drupal 裡的分類建立一個「[標籤雲](#)」的呈現方式，以字體大小來呈現分類資料的多寡，可依照權重或字母排序。

連結網址：

<http://drupal.org/project/tagadelic>

## Taxonomy Block

模組介紹：

根據分類 (taxonomy) 建立區塊，顯示該分類裡最新的文章列表。

連結網址:<http://drupal.org/node/15192>

## Taxonomy Defaults

模組介紹:

如果你想預設某個分類給新發表的內容，這著模組就是這樣的功能了。**Taxonomy Defaults** 在管理分類的地方，可以選擇已經建立好的分類給某些類型的內容當預設值。

實際上可以運用在有提供相簿的 **drupal** 中，然後又給使用者可以在內容中加入影像的情況。**image** 模組必須先新增好相簿，然後在新增影像類型時才能決定相簿的位置。所以如果在一般內容中開放可以上傳影像，那麼這個影像會變成孤兒，也就是變成沒有任何分類的影像類型，結果在相簿中就看不到這個影像。使用 **Taxonomy Defaults** 就可以讓沒有指定相簿的影像歸到某個相簿中，方便管理。

連結網址:[http://drupal.org/project/taxonomy\\_defaults](http://drupal.org/project/taxonomy_defaults)

## Taxonomy dhtml

模組介紹:

提供一個依照分類列出文章標題的網頁，點選分類後，會展開最新的數則文章標題。也提供一個區塊，列出所有分類及其文章數。

連結網址:[http://drupal.org/project/taxonomy\\_dhtml](http://drupal.org/project/taxonomy_dhtml)

## Taxonomy image

模組介紹:

讓網站管理者可以替分類設定圖片，顯示在版型裡。樹狀分類裡的次分類會使用其上層分類的同一個圖片，並可與 **imagecache**、**views** 合用。

範例網站: <http://kerneltrap.org/>

連結網址:

[http://drupal.org/project/taxonomy\\_image](http://drupal.org/project/taxonomy_image)

## Taxonomy Manager

模組介紹:

這個模組提供了一個有力的介面來管理分類詞彙。在一個動態的樹狀畫面中顯示一個分類詞，主要分類詞能夠展開其子分類詞的列表，或者收攏這個列表。

**Taxonomy Manager** 支持下列的操作:

- \* 大量刪除
- \* 大量增加新的子分類詞(**term**)
- \* 移動樹狀階層結構中的分類詞
- \* 合併分類詞
- \* 以向上和向下的箭頭，快速改變權重（並以 **AJAX** 的方式儲存）
- \* 以 **AJAX** 的方式編輯表單
- \* 簡易搜尋介面

使用心得: 非常棒的一個管理工具，特別適用於管理大量分類詞的情況。很希望以後所有有關權重的管理介面都做成這樣，經濟多了。

其選單放在「內容管理」中，不同於「分類」的連結，別搞錯了。

在 Localizer 中運作無誤，只是要記得分類詞中的語系要先改成預設語系，否則這個模組所進行的變更會無效（持續觀察中）。

連結網址：[http://drupal.org/project/Taxonomy\\_Manager](http://drupal.org/project/Taxonomy_Manager)

### Taxonomy Super Select

模組介紹：

選擇分類時，可以使用單選按鈕或勾選框，來取代原本的下拉選單。功能有：

- 多選分類會以勾選框來呈現。
- 單選分類則是變成單選按鈕。
- 多層分類會顯示為巢狀欄位組。
- 可選擇要用在哪些分類。
- 支援自由分類（free tagging）。

範例圖示：[http://www.northstudio.com/drupal/modules/taxonomy\\_super\\_select.html](http://www.northstudio.com/drupal/modules/taxonomy_super_select.html)

連結網址：

[http://drupal.org/project/taxonomy\\_super\\_select](http://drupal.org/project/taxonomy_super_select)

### Taxonomy Theme

模組介紹：

原本是用來為不同的內容分類（主分類/次分類）指定使用不同的版型。更進一步，則可以指定討論區和管理介面的版型，以及特定 Drupal 路徑的版型。

連結網址：

[http://drupal.org/project/taxonomy\\_theme](http://drupal.org/project/taxonomy_theme)

### Teaser Images

模組介紹：

自動地把文章摘要（teaser）裡的圖片，在呈現於首頁上時，轉換為縮圖呈現。對於固定寬度的或是多欄編排的版型來說，很實用。

連結網址：

[http://drupal.org/project/teaser\\_images](http://drupal.org/project/teaser_images)

### Teleport - Drupal 網站內的任意門

模組介紹：

多啦 A 夢來了！？它讓你穿梭在 Drupal 的網站內，按 Ctrl + M 就會蹦出一個對話窗口，就像 Quicksilver, Gnome Do 和其他作業系統桌面的 launchers 一樣，敲幾個你依稀記得的標題關鍵字、路徑...等，就會快捷列出一堆相似的選單，給你一點就中，找到想要的頁面，這也太神奇了吧~

連結網址：

<http://drupal.org/project/teleport>

### Terms of Use

模組介紹：

讓使用者在註冊時，必須先同意使用條款。

與 Legal 模組相比，此模組把使用條款保存在一個內容節點裡，好處是：

- 不另建資料表

- 可選用輸入格式
- 可使用修訂版本
- 可以加入到選單裡

圖示：

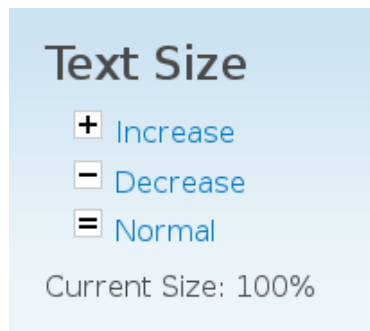
連結網址：

[http://drupal.org/project/terms\\_of\\_use](http://drupal.org/project/terms_of_use)

## Text Size

模組介紹：

可以在區塊裡加入調整字體大小的功能。



連結網址：

<http://www.zwahlendesign.ch/en/node/192>

## Theme Settings API

模組介紹：

Theme Settings 是一個模組，可以讓你方便地修改制式的"詳全文"、"加入回應"這種修辭，改成更親切、更符合你的網站風格的語氣。

Theme Settings 也是一個 API，可以讓版型設計師在版型設定頁裡，加上自訂的選項設定。

連結網址：

<http://drupal.org/project/themesettings>

## Update status

模組介紹：

一天檢查 [drupal.org](http://drupal.org) 一次，看看 Drupal 以及目前使用之模組是否有最新版本。需要 cron 來加以執行，所以請確定您有正確設定 cron，要不然無法得知最新資訊。

使用心得：會列出模組與 Drupal 的目前使用版本與最新版本資訊，並且提供最新版本之壓縮檔網址，直接下載。正式版的優先權會比 dev 版高，所以使用日期較新的 dev 版，反而會被通知要更新到正式版。

連結網址：

[http://drupal.org/project/update\\_status](http://drupal.org/project/update_status)

## Viewfield

模組介紹：

提供一個 cck 的 field，讓它可以插入既有的 views。

在發表新文章時，可以選擇要插入現有哪一個 view。

在顯示文章時，文章的那個欄位就會顯示該筆 view 所篩選出來的結果。

連結網址：

<http://drupal.org/project/viewfield>

## Views

模組介紹：

Views 模組提供了一個很有彈性的方式，讓網站管理者可以很容易地設計網站內容的呈現方式。它可以製做出網頁和區塊，並以表格、摘要、全文、RSS 等方式來呈現。

舉例來說，Views 模組可以輕易滿足以下需求：

- 你喜歡預設的首頁呈現方式，但是想要用不同的方式來排序。
- 你喜歡預設的分類呈現方式，但是你需要不同的排序方式，例如按照字首順序。
- 你使用最新文章（/tracker），但是想要改成只顯示某種內容類型。
- 你想要在區塊裡呈現某個內容類型的 5 篇最新文章。
- 你想要提供一個未讀的討論區文章的區塊/網頁。
- 你想要有一個類似 MT/Wordpress 的每月彙整文章的連結區塊，連結到不同月份的文章列表。

連結網址：

<http://drupal.org/project/views>

## Views Bonus Pack

模組介紹：

這個模組安裝後，會增強一些既有 views 的功能，Views Bonus Pack 1.0 除了 teaser、full node 又多了更多種 node 的排版呈現

1. 兩欄、三欄等多種版型呈現（需要 panels.module）
2. 方格版型的呈現、summary 版型的呈現（不需額外模組）
3. 多層分類呈現（需要 lineage.module）
4. 播放清單呈現（需要 audio.module 和額外的 flash player）

詳看官網說明

連結網址：

[http://drupal.org/project/views\\_bonus](http://drupal.org/project/views_bonus)

## Views carousel

模組介紹：

類似 [nodecarousel](#) jcarousel lite 之類的模組

教學 [screencast](#)

看了 screencast 感覺起來是同類模組中較容易設定的

連結網址:

<http://drupal.org/project/viewscarousel>

## Views Filter Block

模組介紹:

將既有的 views 產生在內文 (content) 區域裡頭的篩選控制器, 移到 block 裡頭, 並做較好的排版配置。

連結網址:

[http://drupal.org/project/views\\_filterblock](http://drupal.org/project/views_filterblock)

## views ticker

模組介紹:

安裝後 在 view 的 view type 裡會多出 Fade BBC Style Scroller(有垂直和水平)幾種呈現效果

[demo](#)

連結網址:

[http://drupal.org/project/views\\_ticker](http://drupal.org/project/views_ticker)

## XML Sitemap

模組介紹:

自動產生一個符合 [XML Sitemap](#) 的 XML 檔案, 網站管理者可以根據節點類型、回應數量、出版狀態等條件, 設定節點的重要性, 並可以把 XML 檔案自動上傳給 Google、Yahoo!、Ask、Windows Live, 讓你的網站內容更容易被搜尋引擎索引、被網友找到。

連結網址:

<http://drupal.org/project/xmlsitemap>

中文翻譯 PO 檔:

[gsitemap -zh-hant-4.7.po](#)

\*\*\*\*\*END\*\*\*\*\*

## 使用 drupal 建的芥菜种书房

最重要的排版使用了 Panels2 和 Views, 里面所有的功能都通过模块搭建, 所以用的模块较多, 除了前面所说的之外, 还有

ccount\_reminder,adsense,adsense\_injector,advanced\_forum,audio,blog\_addons,buddylist,captcha,contact\_attach,drupalit,taxonomy\_breadcrumb 等等, 一共用了 62 个扩展模块

补充一下, 区块部分我采用了 panels 中的 mini panel, 把所需要的 block 放在 mini panel 中, 然后来定义 mini panels 的出现位置等。

总得说来 panels2 非常强大, 对于不熟悉 coding 的建议多多使用,

嗯, 我基本没有写过代码。美工部分我请同事帮忙(CSS 也是他帮忙搞定)

首页部分的制作简单的说分为两大步骤，一个是用模块把功能都搭出来(样子就差不多了)，接下来的步骤就是请同事帮忙美化。

第二个步骤我也无法讲的更多，对于第一个步骤我说明一下。



页首页尾就不讲了(这是 Drupal 的基本哦，如果实在不熟悉可以到我的站点的留言簿上给我留言，可以详细交流)，主要看到页面主体部分：

菜单使用 nice\_menus 模块

接下来看到版式的第一层嵌套(从外往里看)：分为上下 2 层，欢迎信息和公告是第一层(称为 Level 1)，下面的内容是第二层(称为 Level 2)；

再下来看第二层嵌套：Level 1 分为欢迎信息(称为 Level 1-1)和公告(称为 Level 1-2)，其中 Level 1-1 是静态内容，通过添加 block 加入静态的 html 代码就可以(后来我同事帮我有些改成 div+css)，level 1-2 可以自定义公告 内容类型后用 views 生成个 block 出来用；

Level 2 比较关键，分为左边的核心内容页(称为 Level 2-1)和右边的边栏(称为 Level 2-2)，其中 Level 2-1 采用 panels 中 panel pages 选项下的 flexible 版式来做，分为 4 个 rows，第一个 row 分为 2 列，第二个 row 1 列，第三个 row 2 列，第四个 row 2 列，其中的内容都是通过 views 或者 block 引入；Level 2-2 采用 panels 中的 mini panel 实现，其中每一项的内容也是通过区块来引入。

差不多就是这样，还有不清楚可以给我留言





，Products 除了页首页尾外，页面分为两大块，一块是主页面（左边），另一块是 Block(右边)。

左边的主页面，我采用 Panels2 中的 panel pages，选择 Flexible，rows 为 2，提交后，在第一行分 2 列，第二行 1 列，比列(或者尺寸)根据需要来定。如果 Panels2 使用有困难的话请到芥菜种书房的留言簿给我留言(这样我会比较及时看到)，其它的主要就是在 Content 标签中加入每一部分的内容，这些内容都是用 Views 生成的，我这里分为书籍的 Views、礼品的 Views 和饮品的 views，一一加入然后 save 即可。

左边的区块类别是使用 Taxonomy List 模块来实现的，呈现方式是用 panels 的 mini panel。

希望能详细介绍一下 panels 配合其他模块制作成“最新博文”和“论坛新贴”这两个版块，并且只在首页上出现的过程。最新博文和论坛心贴都可以通过 views 中生成一个 block(可以不需要选择 page,看需要),然后在 panels 2 的 Content 标签中把最新博文和论坛新贴（名字可以在 panels 中自己定义）的区块(在 views 中找到)添加进来（panels 2 支持布局 and 排序，如果没有合适的默认布局就可以使用 flexible 布局，可自行定义几行，然后每行几列），剩下的就是 CSS 的问题了。

如果你有 drupal 基础的话应该比较容易懂了，一旦还有不明白的请留言。

对的，是采用 drupalit 来做的，使用很简单。

step1: 下载 drupalit 模块解压并放在 all/sites/modules 下

step2: 到 user access 下开启对应群组的使用权限

step3: 到 site configuration 下找到 drupalit，在其中做一些相应的设置

step4: 效果出来了，具体美化方面需要写一点 CSS

\*\*\*\*\*END\*\*\*\*\*

## 社区开发: drupal 版山寨"开心网"

使用 Drupal 的注册模块实现开心网的功能匿名用户 PK 网的首页

1,Drupal 使用用户名+Email 注册，开心网使用 Email 注册

2,Drupal 默认是记住登录状态的，在每个项目的 settings.php 文件中有 sessions 的设置，开心网是您可以选择是记住登录状态，这样，在不安全的机器上就可以选择不要记住登录状态，在关闭窗口时即可以删除本机的 session 信息。

3,忘记密码功能与用户注册功能稍后再提，基本上 Drupal 默认模块能够满足要求

4,首页的'我们每天都在进步着'可以理解为本更新，这个 Drupal 的 Project 项目有更新文件，可以用这个来做这个模块，不过有点大材小用了，直接设置一个 story 的分类中做个'版本更新'的 term，再在新建的 story 中列入本栏目，新建一个 views,默认显示为 list 列表即可实现开心网首页的这个功能。

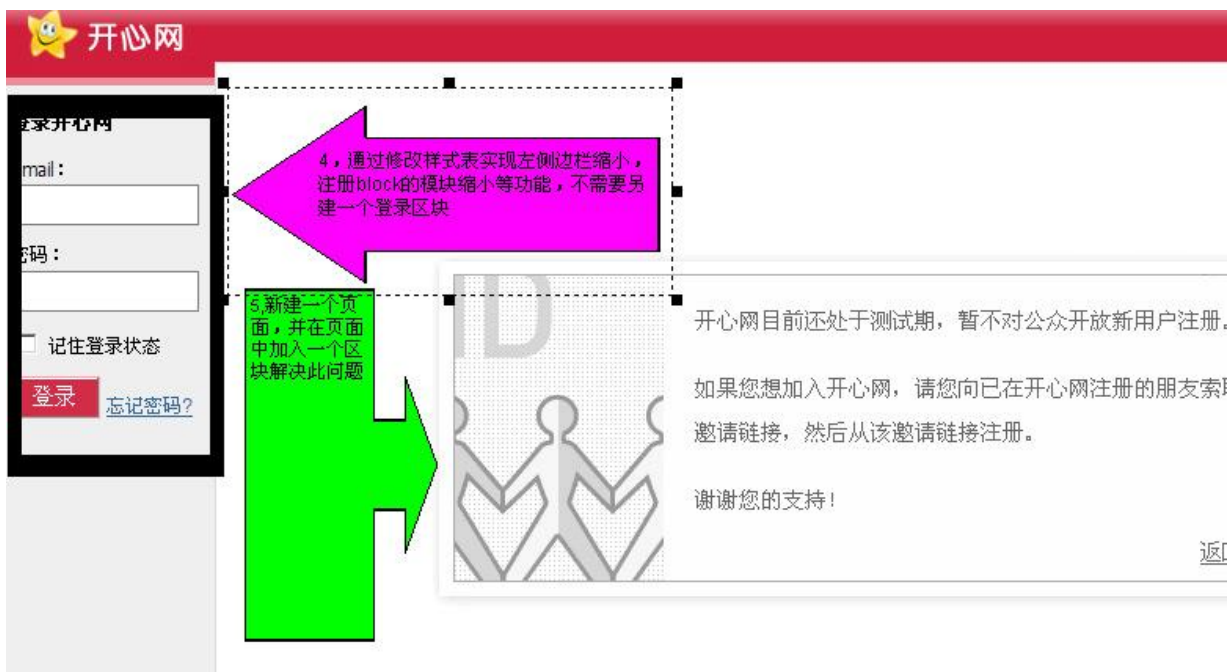
5,给我们提意见,可以直接修改 contact 模块的样式实现。



PK 网的匿名用户注册时会跳转到邀请码页面,登录时对开心网的山寨

1,开心网需要邀请码才能注册,所以在注册页面显示的内容是网站处于测试期,对比此,Drupal 的注册也不难,处理的方式是

- 1) 安装 invite 模块
- 2) 在用户设置页面将新用户的注册定义为如下图所示
- 3)在完成用户设置后,请至少加上 smtp 模块能够发邮件
- 4)对用户登录区块进行一些 css 的设置,因为使用的是 zen 模块,所以很轻易就可以按 sidebarleft 及登录区块的内容进行修改,不需要另外加此区块模块
- 5) 此部分内容可以直接新建页面并跳转到此部分解决



序曲-3:帮助

由 rainfu 于 周一, 2008-12-29 11:10 提交。

开心网对匿名用户有个帮助手册，能够让初次来的用户知道如何玩转这个网站，Drupal 天生就是写手册的强者，所以这个内容不需要山寨，直接写吧

- 1, 新建一个 book 页面，叫帮助手册
- 2, 雇用 pathauto 模块，将帮助页面的 url\_alias 的规则定为 help/[nid].htm
- 3, 在区块设置中将“手册导航”模块默认显示在 help/\* 中，不设置也可以，会自动显示的。但要启用
- 4, 添加一个新的区块在页面顶部，这个区块就是模仿开心网的顶部部分，

代码如下：

```
<?php
global $user;
?> <?php if (!$user->uid) : ?>
<div class="topmenu"><a href="/user/register">注册</a> | <a href="/user/login">登录</a> | <a
href="/help">帮助</a></div>
<?php else : ?>
<div class="topmenu"><a href="/invitefriend">邀请</a> | <a href="/user">帐户</a> | <a
href="/site/privacy">隐私</a> | <a href="/logout">退出</a></div>
<?php endif;?>
```



同事、家人保持更紧密的联络

的最新动态

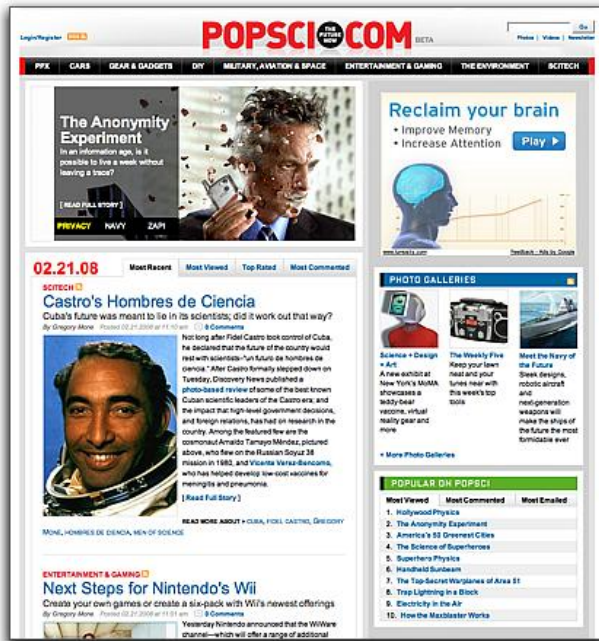
心情、快乐...

注册格式选择为 php code，雇用这个区块后对注册与非注册用户就显示不一样的顶部了。

5, 找找开心网的 theme 对 css 样式进行处理。大功告成，以后想到什么帮助直接在你页面点击添加子页面即可。

\*\*\*\*\*END\*\*\*\*\*

## Popular Science Magazine 建站案例分享



原文: <http://drupal.org/node/233090> )

這個分享就是很好的一個 case study, 說明 open source 怎麼幫助產品, 而需要怎樣的 know how/work 才有可能達成。內容的規劃應該如何做起?

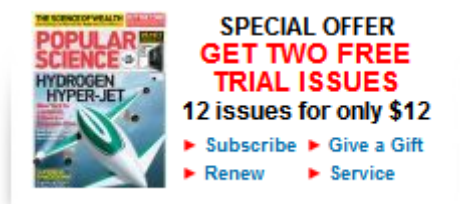
因此, PCS 找來的開發團隊 pingVision, 規劃了 7 種 content type 來達成網站的整合 他們的概念也是由 Drupal 的核心概念出發, 把各種規劃都整合進 node 和 CCK。

## A. 內容类型

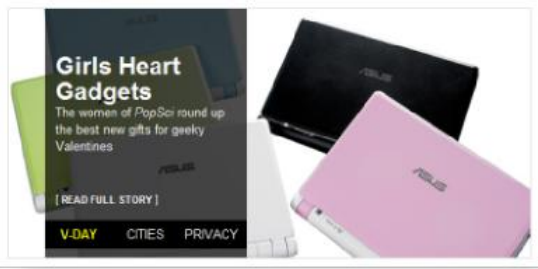
### 1. Articles(文章)



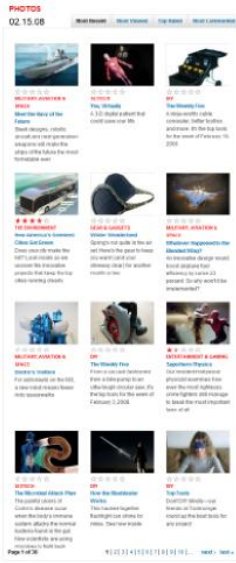
### 2. Current Issue(消息)



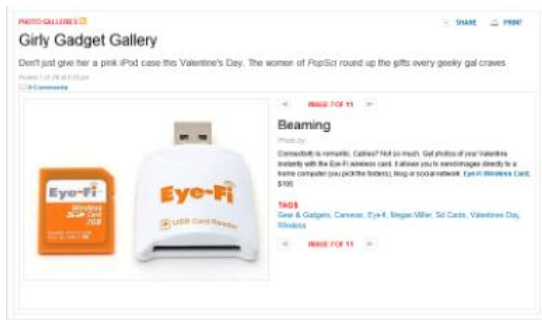
### 3. Featured Tout(特輯)



### 4. Images(圖片)



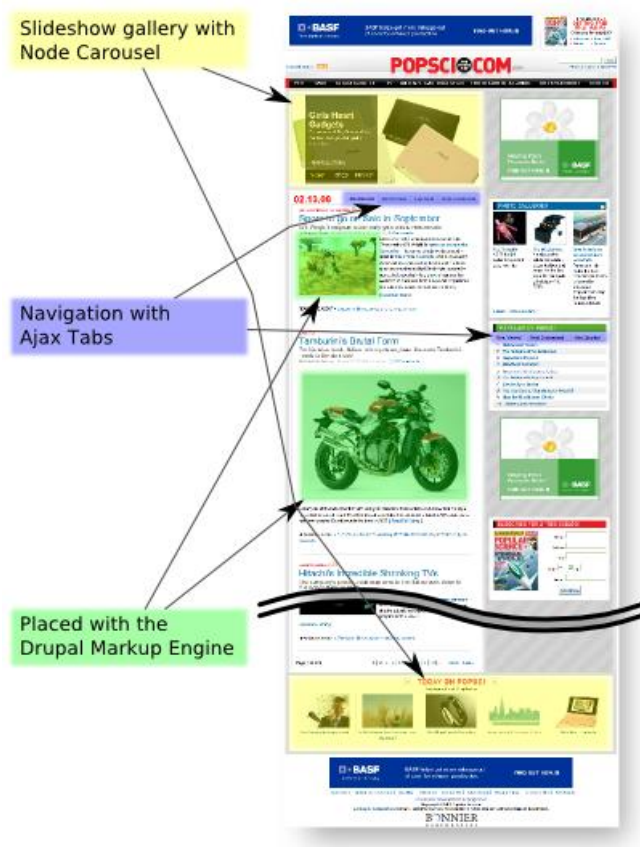
### 5. Photo Gallery(相簿)



### 6. User Video(影片張貼)

因此可以注意到，圖片和相簿都是個別的 node，可以透過 node reference 來整合而 articles 裡頭也有 reference 到圖片的欄位，可以把文章的標題 image 做起而 Current Issue 則規格化了最新消息的美化版本，善用既有的欄位工具，來讓編輯輕易產生好的最新消息最後 Featured Tout 則是為了他們貢獻的 node carousel module，專門用來產生 JQuery 滑動圖片的效果讓廣告或其他想要提昇的特輯，可以吸引使用者的目光這些難不難呢？包含 gallery 的頁面、image 的頁面，以及 Current Issue 的 layout，都需要花功夫但畢竟 Drupal 已經把後台整理好了，而 pingVision 的開發團隊真的妥當了利用這樣的概念。

因此整個線上雜誌的首頁，就可以安然操縱在編輯的手中了，有圖為證：



部份翻譯自：[Popular Science Magazine \(PopSci.com\) Case Study](#)

原文還有更詳細的 module listing

## DRUPAL 官方网站几个案例分析

聰明的 Eureka! Science News: <http://drupal.org/node/261340>

善用 panel/views 的 Spirit Library: <http://drupal.org/node/242993>

案例分析: SonyBMG 的 MyPlay: <http://drupal.org/node/241344>

新聞/社區網站 - HamptonRoads.com & PilotOnline.com <http://drupal.org/node/202403>

The Rake Magazine: 藝文雜誌 The Rake Magazine 的網站改用 Drupal，在網站架構、呈現、管理與開發上，採用了多個模組，並建立了 30 種內容類型、40 個以上的欄位、5 種版型、超過 70 個 .tpl.php、30 個 views、40 個 block。

想要知道他們是怎麼做到的嗎？請見：<http://drupal.org/node/191608>

## 编写提示

输入格式：

- **Filtered HTML:**
  - 自动将网址与电子邮件地址转变为链接。
  - 允许 HTML 标签: <a> <em> <strong> <cite> <code> <ul> <ol> <li> <dl> <dt> <dd>

本站允许 HTML 标签，虽然学习 HTML 可能比较困难，但学习如何使用一小部分基础的 HTML“标签”是很容易的。这个表格提供了本站可用的 HTML 标签的例子。

更多信息，请参见 W3C 组织的 [HTML](#) 或使用您喜欢的搜索引擎搜索其他解释 HTML 的站点。

**标签说明**

**您输入**

**您得到**

锚点(anchor)用来生成链接到其他页面的链接。	<code>&lt;a href="http://fluu.cn/blog"&gt;福祿志&lt;/a&gt;</code>	福祿志
强调	<code>&lt;em&gt;强调&lt;/em&gt;</code>	<i>强调</i>
强调	<code>&lt;strong&gt;强调&lt;/strong&gt;</code>	<b>强调</b>
引用	<code>&lt;cite&gt;引用&lt;/cite&gt;</code>	<i>引用</i>
编码化文本，用于显示程序源代码	<code>&lt;code&gt;编码化&lt;/code&gt;</code>	编码化
无序列表 - 使用 <code>&lt;li&gt;</code> 开始每个项目	<code>&lt;ul&gt; &lt;li&gt;第一项&lt;/li&gt; &lt;li&gt;第二项&lt;/li&gt; &lt;/ul&gt;</code>	<ul style="list-style-type: none"> <li>▪ 一项</li> <li>▪ 第二项</li> </ul>
顺序列表- 使用 <code>&lt;li&gt;</code> 来开始每个项目	<code>&lt;ol&gt; &lt;li&gt;第一项&lt;/li&gt; &lt;li&gt;第二项&lt;/li&gt; &lt;/ol&gt;</code>	<ol style="list-style-type: none"> <li>3. 第一项</li> <li>4. 第二项</li> </ol>
定义列表类似于其他 HTML 列表。 <code>&lt;dl&gt;</code> 开始定义列表， <code>&lt;dt&gt;</code> 开始定义列表项， <code>&lt;dd&gt;</code> 则开始列表描述。	<code>&lt;dl&gt; &lt;dt&gt;第一个术语&lt;/dt&gt; &lt;dd&gt;第一个定义&lt;/dd&gt; &lt;dt&gt;第二个术语&lt;/dt&gt; &lt;dd&gt;第二个定义&lt;/dd&gt; &lt;/dl&gt;</code>	第一个术语 第一个定义 第二个术语 第二个定义

许多特殊字符可以轻松的直接输入。

如果您碰到问题，尝试使用 HTML 字符实体。一个通常的例子，类似于 `&amp;` 代表一个与符号(&)。全部的列表参见 [HTML 实体](#)。某些可用字符包括：

**字符说明**

**您输入**

**您得到**

与标记(&)	<code>&amp;amp;</code>	<code>&amp;</code>
大于号	<code>&amp;gt;</code>	<code>&gt;</code>
小于号	<code>&amp;lt;</code>	<code>&lt;</code>
双引号	<code>&amp;quot;</code>	<code>"</code>

- 行和段被自动识别。`<br />` 分行、`<p>` 段落开始以及 `</p>` 段落结束标记会被自动插入。如果段落没有被识别出来，只要加入一对空行。

● **Full HTML:**

- 自动将网址与电子邮件地址转变为链接。
- 行和段被自动识别。`<br />` 分行、`<p>` 段落开始以及 `</p>` 段落结束标记会被自动插入。如果段落没有被识别出来，只要加入一对空行。

\*\*\*\*\*END\*\*\*\*\*

## drupal 开发建站系统的建议

开发过程的建议：（以下假设使用 drupal6 作为基础，进行开发制作）。

（1）根据要求制作效果图和静态 HTML 文件，明确站点的功能需求，书写需求文档，即使不将需求文档书写出来，也要对站点的功能需求有确定的规划，在这一步中，最重要的就是一定要将站点的所有功能需求明确好，以免造成在开发过程中，出现功能需求不明确的问题，影响开发进度。

（2）根据已经明确的功能需求，进行功能到模块的转换，即根据所列出的功能，寻找相应可以实现此功能的 drupal 核心模块或者第三方贡献模块，所以这一步更要求第一步中功能需求一定要明确好，这里建议，功能到模块的转换，尽量使用 drupal 模块去解决问题，而不是修改模块本身的实现来达到目的，比如，留言本的功能，在以往的实现过程中，是将 drupal 自身的评论功能进行了更改，这样，如果站点同时存在留言本和文章评论的需求，则实现起来较为困难。

（3）启用第二步中选择好的模块，实现站点的基本功能，在这里建议，尽量使用模块生成区块，而不是页面，以加强区块的重用性，方便站点的维护。[详细方法见后面的区块使用技巧建议当中。]

（4）整合页面，进站点行主题的制作。这里建议，主题制作中，尽量使用 drupal 自建的 api 函数去书写主体代码，减少大量书写 HTML 硬编码造成的维护困难。[详细方法见后面的主题制作建议当中。]

（5）最后一步，则是进行各种测试，包括将在 Windows 平台下 Apache + MySQL + PHP 环境下开发的程序移植到服务器 Linux 平台上，检验是否会出现异常的错误等，尤其注意，如果站点启用“简洁链接”以后，移植后是否会出现异常的无法访问的情况。

建议二： 开发技巧的建议

### （1）“区块”的使用技巧

在以往的网站开发过程，由于当时对概念的不熟悉和时间问题，并没有使用“区块”的概念和特性去制作，而采用了 HTML 硬编码的方式书写各个不同的区块代码，造成了维护和修改的复杂程度增加。

在自定义建站系统的过程中，建议如下方法，

第一步：通过.info 文件机制，根据制作好的效果图，定制出页面的“区域”，例如：

```
regions[left] = Left sidebar
regions[right] = Right sidebar
regions[content] = Content
regions[header] = Header
regions[footer] = Footer
```

即可将页面区域分成 left right content header footer 五个部分。

第二步：将网页中，各个不同的分类（例如，文章分类）的列表用 views 生成 block，而不要生成 page，这些 block 将在后台管理的区块页面里列出来，

第三步：通过拖动 block 到不同的“区域”，即可将不同的区块放在页面的不同的地方，并且，在 block 的配置中，可以配置 block 在某一个或者某一些 page 中显示。这样的方法，大大增强了站点区块的可重用性，不必，每个页面都需要书写同一个区块的硬编码了，另外，更改维护和扩展也相对以前的制作方案方便了许多。

### （2）Drupal 内置 api 函数的使用

在以往的开发过程中，由于是刚刚开始使用 drupal，对其 api 函数不是非常了解，所以开发过程中，大量使用 HTML 硬编码来书写诸如 css 加载，js 加载，图片的插入等代码，给维护和更改造成了一定的影响，现提出尽量使用 api 函数来书写 theme 代码的建议，例如：

要在 tpl 文件中插入 image，使用 theme('image',\$path,\$arg)即可插入图片，而无须使用 如此不方便维护的代码。

再比如，要在 tpl 文件中加入连接，使用 l(\$text, \$path, \$options = array()) 函数，而不是使用

<a href=" alt=">text</a>的方式书写连接，这样可以避免启用/关闭“简洁链接”还要修改连接的代码，加入/删除“? q=”等代码，为维护和更改提供了极大的便利。

在比如，t 函数书写的字符，可以被 drupal 很好的转译，安全性也可以保证，并且由 t 函数载入的字符可以生成多语言包，用于 drupal 多种语言。

### （3）主题制作中一些建议

在以往的开发过程中，我们使用 HTML 硬编码来书写 drupal 本身可以自动生成的变量，比如，导航连接我们使用



<ul><li></li>.....</ul>方式书写, 在以后的开发过程中, 我建议, 尽量采用 drupal 自己生成的变量来书写, 比如可以用

```
<?php print theme('links', $primary_links, array('class' => 'links primary-links')) ?>
```

输出导航, 可以避免长达 7 行的<ul>输出, 开发效率也将大大提高,

再例如,

```
<?php print $head ?> 输出头部信息
```

```
<?php print $styles ?>输出样式表
```

```
<?php print $scripts ?>输出脚本
```

```
<?php if ($show_messages && $messages): print $messages; endif; ?>输出错误信息, 等等
```

这样的变量, 可以使用 drupal 自建, 更改维护将更为方便, 尤其是在自定义建站系统中, 这样书写变量, 假设更改导航内容, 可以直接在后台更改即可, 不需修改 theme 模板中的代码。

所以, 尽量使用 drupal 自建的变量, 而不是硬编码, 则可以加快 theme 制作的效率, 为日后更改, 维护, 重用同样提供了较大的方便。

以上建议, 是在以往的开发过程中总结和与小组成员共同探讨时获得的经验, 希望可以在以后自定义建站系统的开发应用上能够起到一定的作用。

## drupal 建站的 50 个常用小技巧

### Drupal 管理和配制

1. 创建一个 HTML 帮助的 block,让它只出现在"node/\*/edit" and "node/add/\*"
2. 用 admin/user/roles 下的"edit permissions"去管理每一个具体内容的权限
3. 在 admin 进入网站时, 用 Firefox Web Developer Toolbar 去"Populate Form Fields" (不太明白什么意思)
4. 用 google 的"site:drupal.org {search terms}"格式去搜索 http://drupal.org 上的内容, 在 firefox 上建立 shortcut, 你可以用"dru [your search]"和 api.drupal.org/apis/{search\_term}格式去 searchdrupal.
5. 用 pathauto 给用户名设一个别名, 这样用户就无法改变自己的用户名了
6. 用 Pathauto 和 path redirect 给新的连接设一个别名重定向到旧的连接。这样搜索引擎就不会抛弃你了
7. 用 Global Redirect 模块确保首页以"/"结尾, 并且当前页没有别名
8. 当创建 PathAuto 模版的时候, 记住要唯一.你可以在你的 blog 或 news 里添加一个日期来确保唯一性
9. 创建一个自定义的 menu item,这样可以移动"my account"菜单
10. 用 Update\_status 模块和 cvs\_deploy &cvs tage 去管理网站
11. 用 Imagecache 去重设图片的大小
12. 用 contemplate 可以把任何带有图片字段类型的自定义 CCK 转换成 Photocast
13. 当你用 cck 创建类似 first name,last name 字段的时候, 可以考虑用 Automatic Node Titles 模块
14. 可以用 simplemenu 删除页面上不需要的 admin block
15. 当你升级的时候, 记着把网站设为 offline mode,以免让用户看到错误信息
16. 可以用 locale.module 改变页面上的一些文字。例如, 把"Forums"变成"Message Boards".检查 default.settings.php 页面来获得更多详细信息
17. 用 views + views\_fastsearch 可以构建 "section search" 或者高级搜索页面
18. 在 drupal.org 打开贡献区块, 它会带给你很多有用的连接。
19. 经常查看 http://drupal.org/planet ,会得到很多最新的开发信息。
20. 把你对于 drupal 和第三方模块的建议和 bug 提交给 issue queues,而不是发布在论坛中, 这样更容易引起开发者的注意。
21. 参考 http://drupal.org/node/10259, 学习如何提交问题级 issue queue, 包括如何提交一个好的报告等
22. 在 http://drupal.org/videocasts 上, 你可以找到很多有用的 videos 和各种各样的 subjects.
23. 在 http://drupal.org/security 上注册一下, 这样你可以保持得到最新的安全通知。
24. 在高级搜索中, 可以用"-"去掉你不想在结果中出现的词;如果只想在某一内容类型中搜索, 可以"type:forum" 或者 "type:image"等;如果想搜索短语, 可以用""包起来, 例如"Drupal rocks"

### 关于 Theming

25.你可以用 <http://drupal.org/project/zen> 等最简单的 theme 搭建一个站点，然后在你的设计师正式开始设计之前，向他指出哪点是你想要的结果。

26.当你做一个新的 theme 时，记着测试管理员的界面。

27.开发 javascript,css 等前端时，你应该知道 FireBug。

28.通过 `var_dump(get_defined_vars());` 来查看一个模版文件中哪些变量是可用的。

29.当你移除 Teaser 中的 read more link 时，通过 `hook_link_alter()` 来添加 "More..." link。

30.`format_interval()` 函数相当 cool,它可以输出 "X 分 Y 秒 之前"，或者相似的结果。

31.在 theme 中，替换页面里的动态 elements 来适应页面缓存。如果用户没有登陆，用绝对值如 "posted on {date} at {time}" 来代替 "{X} hours, {Y} seconds ago"

#### 关于 Views

32.用 `theme('view',{arguments})`,你可以在你的 theme 模块或.tpl.php 文件中的任何地方嵌入一个 view(你也可以用 `views_embed_view` 函数,详情请参考 <http://zhupou.cn/node/661>)。

