

Drupal 精华教程集 1

目 录

第一部分 外观主题

1. Drupal 打造中国化主题系列教程.....	3
2. http://www.comeongame.com/ 的设计流程.....	7
3. Drupal 模板建议	8
4. page.tpl.php 的图形化显示.....	10
5. Drupal6 的模板文件详解	11
6. drupal6.x 中创建新的主题模版和区域(Regions)	13
7. Block (区块) 管理.....	15
8. 如何为 Drupal 网站创建一个 Magazine Style 的首页	19
9. 对一个 Drupal 主题的剖析.....	26
10. 主题的.info 文件.....	28
11. Drupal 6 开发用版型	31
12. 7 个步骤让您创建专属的 Drupal 版型	31
13. Drupal Theme --- 按照内容类型设计页面布局.....	31
14. garland 主题页面分析与修改详解	32
15. ZEN theme 主题修改成果小结.....	38
16. 小技巧--使用 firebug 帮忙定制 theme.....	39
17. Drupal 实现国内门户网站的风格	41

第二部分 CCK VIEWS 模块使用

1. Views2.0 从入门到精通.....	42
2. CCK & Views 模块使用小结.....	45
3. 使用 Drupal 6 Views Module 系列(一)	70
4. CCK 的介绍及使用	78
5. drupal 的 views 学习.....	81
6. drupal 教程 views 模块教程 (显示同一类型的文章)	83
7. 使用 views 创建一个最新文章列表区块	85
8. 综合首页 VIEWS 开发问题	

9. Drupal Views2 模块使用指南.....	87
10. 首頁=Panels2+Views2	89
11. cck contemplate theming 技巧.....	94
12. CCK 图像字段:Imagefield	97
13. 实例 BlueSnow 的工作平台(项目管理)	98
14. Views 2 教程 :创建一个显示相关网页的区块.....	103
15. VIEWS 實作公告+分類.....	109
16. Views 2 與 Tab	117
17. Drupal 项目实战 :公司员工订餐系统.....	120
18. Drupal 模块-CCK 和 Views 的使用.....	126
19. Drupal 6 中添加 "相关链接" 的方法.....	126
20. 用 CCK 模组做一个 "站内相关连结" 的栏位	131
21. Drupal 综合实战 : CCK+VIEWS = CMS.....	133

第三部分 其它备用模块

1. Panels 2.X 教學 (基礎應用篇)	140
2. Panels 的使用.....	149
3. 利用 Panels 模組來實做 tab 效果的 block.....	151
4. drupal 模块教程-pathauto 模块	154
5. drupal 教程 qiucktabs 模块教程.....	159
6. Quicktabs 模块的使用小技巧.....	161
7. Views Slideshow	161
8. DRUPAL 需要的模组	162

第四部分 案例分析

1. 使用 drupal 建的芥菜种书房	181
2. drupal 版山寨"开心网"	183
3. Popular Science Magazine 架站案例分享	185
4. DRUPAL 官方网站几个案例分析	188

第五部分 技巧分享

1. 编写提示	188
---------------	-----

2. drupal 开发建站系统的建议 189

3. drupal 建站常用小技巧 191

4.有用的资源 (含网址及视频教程) 191

Drupal 打造中国化主题系列教程



Drupal 制作主题，主要有 2 个部分是比较有难度，一个就是静态页面的制作，要熟悉 W3C 的标准，熟悉 div+CSS，熟悉各种网页制作工具，在一个就是由静态页面转化成 Drupal 主题，即将内容放置到 Drupal 的区块当中；我们在这里，从拿到静态页面以后开始讲述，首先要分析一下这一个主题的版式和制作流程：

(1) 主题分为前后台两个部分，我使用的是 drupal 经典主题 garland 更改得到我的主题的，这样我只覆写前台的主题，后台的样式依然采用的 garland，

因为后台都是自己看，适用默认的 garland 无论是标准还是大小，都是比较适合做管理界面的。

Drupaluser.cn 的前台主题主要是首页，列表页面，内容显示页面和其他静态页面，也就是静态页面只需要做 4 个页面就 OK。

首页我们使用 page-front.tpl.php 文件覆写

列表页面我们使用 page-taxonomy.tpl.php 文件覆写

内容显示我们使用 page-story.tpl.php 和 node-story.tpl.php 来覆写

其他的静态页面我们使用 page-page.tpl.php 和 node-page.tpl.php 来覆写

至于为什么能采用上述方式，请参阅 <http://drupaluser.cn/category/theme> 中的相关**模板覆写的文章**，采用上述方式是因为这样可以不用修改 `garland` 的默认模板，不会影响到后台模板的样式，这样看起来，比较像前后台相分离，这样也比较符合中国人使用 CMS 前后台相分离的习惯

(2) 流程方面，首先要在 `info` 文件中添加区域，然后将一部分静态的内容做成区块，直接写在 `block` 文件中，例如 <http://drupaluser.cn> 中最下面

的版权声明和友情链接，都是直接书写 `html` 到区块中的方式，然后将 4 个 `html` 页面布局拆分，使用 `views` 的要做 `views` 的 `tpl` 文件，其他的相应

制作上面所述的 4 个文件的 `tpl` 文件就 OK，我们下面将一步一步详细说明。

创建区域

我们首先分析一下 `Drupal` 使用者中文社区[<http://drupaluser.cn>]的主题应该划分为哪一些区域，首页分为上、中、下三个大区域，右边有可以分为上下两个大块区域，这样划分，也是为了照顾美工方面比较容易用 `css` 控制 `div` 的 `float` 所以我们定义了

```
regions[top] = Top
regions[left_region1] = Left Region1
regions[left_region2] = Left Region2
regions[right_region1] = Right Region1
regions[right_region2] = Right Region2
```

你或许问我，为什么不使用 `garland` 定制好的左右区域呢，如果我把区块放置在 `garland` 的左边，那么在后台也将会显示那个区块，这样是显示在几个页面还可以通过设置区块的显示位置来控制，而一旦多了，比较乱，所以我们对前台的左右区域重新订制，当然你仍然可以只用 `garland` 的主题左右区域，我为了以后好扩展，使用了此方法。

对于其他的列表页和内容页面，我定义了一个右侧的区域，这个主要考虑以后会放置 `AD`，所以起了个名字叫做 `AD region`，呵呵，目前主要放置的是最新文章的一个 `views` 列表

```
regions[ad_region] = AD Region
```

提醒：在 `drupal6` 中，如果你自定义了区域，则需要把默认的 5 个区域也写上，否则是写你自己定义的区域，它不会显示默认的 5 个区域，如下：

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

在下一讲当中，我们介绍添加几个静态的区块

静态区块

对于一些不经常变化，需要硬编码而且需要多次调用的内容，我采用了直接将其 `html` 代码放入 `block` 中，然后将 `block` 放置在 `region` 中就 OK

比如 `Drupal` 使用者中文社区[<http://drupaluser.cn>]最上面的一个区块，就是硬编码放上去的，包含了最上面的一个导航和下面的 `Logo` 部分

导航“管理”--“站点创建”--“区块”；选择添加区块；在正文中将静态 `html` 页面中的代码写入即可，这里要提示的是，需要启用 `php` 代码模块，因为在这里的路径，需要使用 `php` 代码写入 `drupal` 的变量，输入格式当然需要使用 `php code` 格式；

提示：在区块中使用 php code 非常危险，如果 php 代码输入错误，有可能整个站点将不能访问；笔者就因为粗心，搞错过一个字母，而不得已修改数据库才删除错误的，请各位谨慎使用。

页面可见性设置中，设置“在特定页面上显示区块”，这里填入你要将此区块显示在那一个页面上

在这里解释一点，你这样添加到页面的区块，drupal 会使用 block.tpl.php 模板显示区块的标签和 css，所以我们在 page 中并没有使用 \$style 样式

添加 css，而是手动添加的 css 样式；即便是有 drupal 默认的 div 标签包着，也不会影响到我们静态页中做好的样式，请大家放心使用就 OK

下一节，我们将自定义 page-contenttype.tpl.php 和 node-contenttype.tpl.php 的页面

创建 tpl 文件

Drupal 使用者中文社区[<http://drupaluser.cn>]的 tpl 文件主要是以下几个：首页我们使用 page-front.tpl.php 文件覆写列表页面我们使用 page-taxonomy.tpl.php 文件覆写；内容显示我们使用 page-story.tpl.php 和 node-story.tpl.php 来覆写其他的静态页面我们使用 page-page.tpl.php 和 node-page.tpl.php 来覆写

其中 page-story.tpl.php 和 page-page.tpl.php 不是默认就会起作用，比较常见的有两种方法可以让他起作用：

一种是 drupal 的风格，在 template.php 中覆写函数实现，这种方法老葛的 zhupou 上面有介绍，在此不再赘述了

另一种是比较容易理解的风格，可以在 page.tpl.php 上面写一个判断，让他根据内容类型自己选择模板文件，

如以下写法：

```
<?php
if($node->type == 'page' && arg(2) != 'edit'&& arg(2) != 'delete') { include 'page-page.tpl.php'; }
else if($node->type == 'story' && arg(2) != 'edit'&& arg(2) != 'delete') { include 'page-story.tpl.php'; }

else{ //include page.tpl.php 中的原来的函数。}
?>
```

这样可以和覆写函数得到一样的效果，这样做，只是为了不改动 garland 默认的 page.tpl.php。

这里面的样式就可以自己写了，我截取一段分析如下：

page-front.tpl.php 中的

(1)输出了自定义的 css，而没有用 \$style，也没有在 info 文件中加入这两个样式

```
<link rel="stylesheet" type="text/css" href="<?php print base_path().path_to_theme();?>/css/layout.css">
```

```
<link rel="stylesheet" type="text/css" href="<?php print base_path().path_to_theme();?>/css/main.css">
```

(2)输出了几个区域中的内容

```
<div id="mainleft">
  <div class="partA">
    <?php print $left_region1 ;?>
    <div class="c"></div>
  </div>
  <div class="partB">
    <?php print $left_region2 ;?>
    <div class="c"></div>
  </div>
  <div class="partC">
```

```
<?php print $left_region3 ;?>
```

```
</div>
```

```
</div>
```

(3)输出了导航菜单

```
<div id="menu">
```

```
<?php print theme('links', $primary_links) ?>
```

```
</div>
```

其他的 page 页面类似，非常简单，id class 写好了用 css 控制样式

在下一讲当中，我将介绍一下我的 views 列表的创建；这是中国化主题的关键之处

创建 views 列表

我估计大家感觉 Drupal 使用者[<http://drupaluser.cn>] 站点最中国化的地方应该是 四个文章列表了吧，呵呵国内 CMS 的一个特点就是做列表了，这一点比较符合国人的习惯，这一讲当中，将介绍一下 views 生成文章列表。详细的 views 使用，请参看 <http://drupaluser.cn> 站内的相关文章，在此只是大概说一下主要的和需要注意的地方

(1) 进入 views，点击“添加”，

views name: views 区块的名称，比如 Drupal_theme_block

views type : 选择“节点”；

(2) 进入 views 的详细配置页面，

Fields 项目中 选择“节点：标题”和“节点：post date”，即列表包含了两项，即标题和发布时间，不要忘记标题关联到节点的 url。

过滤器 项目 就是选择你要生成那一个分类的内容了，这里选择了 Drupal 主题制作分类的文章，其他根据自己的需要添加

Sort criteria : 排序，一般我都是有两个排序，首先按照是否置顶，其次是发布时间

basic setting 中有几个需要注意的地方

style 最好选择列表方式

Row style: 最好选择 Fields 方式

这样点击下面的预览看看是不是自己想要的文章列表，这样 功能方面我们就 OK 了，

下一讲，我们将介绍一下如何覆写 views 输出的样式，做成 [Drupaluser.cn](http://drupaluser.cn) 首页那样的文章区块

覆写 views 主题

本章节只是大概介绍一下 Drupal 使用者中文社区[<http://drupaluser.cn>]首页当中 下面四个 views 列表区块的覆写方式更详细的介绍，请期待 Views2.0 入门教程

在根据上一章节当中创建好的 views 中，在 views 列表页面的操作框右侧选择“block”，点击 add display

这样在 basic settings 中的 Theme infomation 中，会看到类似以下四个文件的覆写：

(1) 覆写全局输出，整个 views 布局，如果你的 views 是一个页面的话，通过覆写这个文件，可以改变页面的布局

* **Display output:** views-view.tpl.php, views-view--Drupal-Using-block.tpl.php, views-view--default.tpl.php,

views-view--Drupal-Using-block--default.tpl.php

(2) 覆写 views 列表，即可以通过次文件覆写掉 views 的列表的样式，我就是覆写的这一个文件

* **Style output:** views-view-list.tpl.php, views-view-list--Drupal-Using-block.tpl.php, views-view-list--default.tpl.php,

views-view-list--Drupal-Using-block--default.tpl.php

(3) 覆写 views 列表一行的样式

* **Row style output:** views-view-fields.tpl.php, views-view-fields--Drupal-Using-block.tpl.php, views-view-fields--default.tpl.php, views-view-fields--Drupal-Using-block--default.tpl.php

(4) 覆写 views 列表中每一个字段的样式, 这里只输出了一个 title, 所以只有一个。

* **Field 节点: 标题 (ID: title):** views-view-field.tpl.php, views-view-field--title.tpl.php, views-view-field--Drupal-Using-block.tpl.php, views-view-field--Drupal-Using-block--title.tpl.php, views-view-field--default.tpl.php, views-view-field--default--title.tpl.php, views-view-field--Drupal-Using-block--default.tpl.php, views-view-field--Drupal-Using-block--default--title.tpl.php

以上列出的文件是优先级越来越高, 你当前使用的文件会被高亮显示出来:

我的 views-view-list--Drupal-Using-block.tpl.php 文件代码如下:

```
<div id="zhengwu">
  <div class="zhw_head">
    <span class="pro_title">Drupal 使用</span><span class="pro_more"><a href="<?php print
base_path()?>category/drupalusing"></a></span></div>
  <div class="zhw_list">
    <?php foreach ($rows as $id => $row): ?>
      <div class="item_zhw">
        <span class="icon_zhw"></span>
        <span class="zhw_title"><?php print $row; ?></span>
      </div>
    <?php endforeach; ?>
  </div>
</div>
```

以上代码中, <?php print \$row; ?> \$row 即 views 列表中的每一行标题, 通过 foreach 将其循环输出

其次, 我的每一行标题, 我使用了优先级最低的那个文件覆写, 这样, 整站中 views 输出标题的时候, 就会使用没有任何 views 标签包裹, 看着比较舒服

```
views-view-fields.tpl.php
<?php foreach ($fields as $id => $field): ?>
  <?php print $field->content; ?>
<?php endforeach; ?>
```

----- END-----

<http://www.comeongame.com/> 的设计流程

这次我用了 6.x 设计我的 game 搜集站, 主要是因为 image field 可以有 link to node, 并且图片和

swf 文件的上传都采用 用户 ID/年月...这个方式, 避免了大量文件都传到一个文件夹下的情况.... 以后站大了, 这样做文件管理也好些...

第一: 主题设计

关于主题设计主要是采用了复写, 在这里感谢 drupaluser.cn 提供给我的帮助, 使我终于脱离了 panels 模块, 可以随意设计自己的排版!

page-front.tpl.php 复写首页

page-taxonomy.tpl.php 复写列表页

page-flashnode.tpl.php 和 node-flashnode.tpl.php 复写 flash 内页

复写 page-flashnode.tpl.php 注意....直接复写并不会成功, 参照

<http://drupaluser.cn/html/mytheme/2009-01-01/332.html>

这里主要是采用 drupal 的复写机制... 首先在.info 文件里面建立自己的 region, like this:

```
regions[left] = Left sidebar
```

```
regions[right] = Right sidebar
```

```
regions[content] = Content
```

```
regions[header] = Header
```

```
regions[footer] = Footer
```

然后你可以用 css+div 写自己的模板, 将 regions 放在你想放的地方... regions 不能放在 node 上... 如果想法似乎要搞 template.php, 我也尝试过, 成功了! (见: 添加 regions 到 node.tpl.php

(<http://drupalchina.org/node/3804>))

关于 flash 游戏页面的设计, \$content 变量被我具体化成很多变量, 主要是我想按照自己的 div 排版安排位置...这里我主要用 content templete 去找变量...然后同样道理用 div+css 结合变量去设计 layout....这样可以方便随意设计出自己的排版.....为了有很好的 seo, meta keywords 和 description 部分我也加入了部分变量输出.....

第二: 模块的使用

因为我的站是 flash 小游戏搜集站, 所以主要依靠 flash node 的功能....输出用 view2+cck, 然后还有投票的功能我用了 vote up and down....你也需要 cck 的 image field 去输出游戏的截图, 用 pathauto 去定义了下 node 的 url,形式是...../分类名/文章标题, 分类用 Taxonomy

主要有 6 大类, 分别在首页输出 block,关于 view2 的主题制作也是复写....用 css+div 控制下就可以了

大概就是这么个流程.... 希望高手指点....

***** END*****

Drupal 模板建议

建议只有和基模板放在同一个目录下面时,才能工作.换句话说,为了让 comment-blog.tpl.php 正常工作,你需要把 comment.tpl.php 也放到同一个目录下.

下面列出默认建议., 如果你想定制的话,可参看使用模板建议一页.

block-[region|[module]-delta].tpl.php

基模板: block.tpl.php

默认建议及顺序:

1. `block-module-delta.tpl.php`

2. `block-module.tpl.php`

3. `block-region.tpl.php`

"module"为模块名称,而"delta"为模块分配给该区块的内部 id.例如, "`block-user-1.tpl.php`"将用于默认用户导航区块,因为该区块由 `user` 模块创建,且内部 id 为 1. "region"将对特定区域产生作用.

`comment-[type].tpl.php`

基模板: `comment.tpl.php`

默认建议为 `comment-type.tpl.php`,它用于特定节点类型的评论格式,以区别站内的其它评论.与 `node-[type].tpl.php` 类似,但是它用于评论.

`comment-wrapper-[type].tpl.php`

基模板: `comment-wrapper.tpl.php`

与前者类似,但是用于包装器(wrapper)模板.

`forums-[[container|topic]-forumID].tpl.php`

基模板: `forums.tpl.php`

默认模板建议及顺序。

对于论坛容器

1. `forums-containers-forumID.tpl.php`

2. `forums-forumID.tpl.php`

3. `forums-containers.tpl.php`

对于论坛话题:

1. `forums-topics-forumID.tpl.php`

2. `forums-forumID.tpl.php`

3. `forums-topics.tpl.php`

`maintenance-page-[offline].tpl.php`

基模板: `maintenance-page.tpl.php`

当数据库不可用时,应用该模板建议.用来为用户展示一个不带错误信息的页面.首先需要设置维护页面的主题化.

`node-[type].tpl.php`

基模板: `node.tpl.php`

节点类型,例如"`node-story.tpl.php`", "`node-blog.tpl.php`",等等.

`page-[front|internal/path].tpl.php`

基模板: `page.tpl.php`

可以有无数个建议.具有优先级的为首页(front page).剩下的都是基于当前页面的内部路径.不要将内部路径和路径别名混淆了,这里不能使用路径别名.记住,通常使用 `pathauto.module` 来设置路径别名.

可通过"Administrator > Site configuration > Site information"来设置首页.对于设置好的首页,将会为其使用"`page-front.tpl.php`"模板.

下面为模板建议文件,根据内部路径,越特殊的次序越靠前.如果系统为当前页面找到了一个模板建议的话,就不会再调用位于它后面的模板建议了.例如,对于 `http://www.example.com/node/1/edit`,将会有下面的建议可用:

1. `page-node-edit.tpl.php`

2. `page-node-1.tpl.php`

3. `page-node.tpl.php`

4. `page.tpl.php`

`poll-results-[block].tpl.php`

基模板: `poll-results.tpl.php`

生成投票结果的主体函数，可供节点和区块共同使用。默认是用于节点的，但是模板建议使得可以用在区块区域中。这个建议是默认的，它位于"modules/poll/poll-results-block.tpl.php"。

poll-vote-[block].tpl.php

基模板: poll-vote.tpl.php

与 poll-results-[block].tpl.php 类似，但用来生成投票表单。你必须自己为其提供模板，以让其生效。

poll-bar-[block].tpl.php

基模板: poll-bar.tpl.php

与 poll-vote-[block].tpl.php 一样，但是用来生成单个的 bars(细长条纹)。

profile-wrapper-[field].tpl.php

基模板: profile-wrapper.tpl.php

这个 profile 包装器模板，是在浏览会员列表页面时使用的。当浏览特定的字段时，在模板建议中需要使用字段名称。例如 <http://drupal.org/profile/country/Belgium> 使用的模板为"suggest profile-wrappercountry.tpl.php"。

search-results-[searchType].tpl.php

基模板: search-results.tpl.php

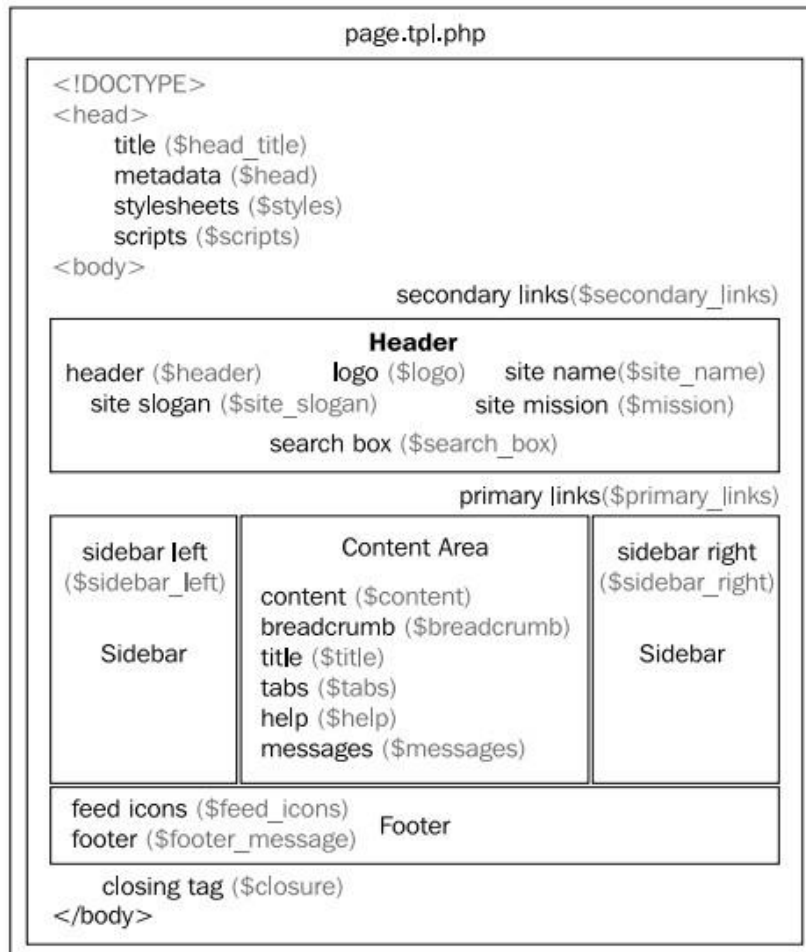
search-results.tpl.php 是搜索结果的默认包装器。根据搜索类型的不同，使用不同的建议。例如，对于"example.com/search/node/Search+Term"，将使用"search-results-node.tpl.php"。而对于"example.com/search/user/bob"，则使用"search-results-user.tpl.php"。模块可以扩展搜索类型，并为其添加更多的建议。

search-result-[searchType].tpl.php

基模板: search-result.tpl.php 与前者相同，但是用于单个的搜索结果。

END

page.tpl.php 的图形化显示



上图展示了 `page.tpl.php` 的输出的各个部分和变量的信息！

Drupal6 的模板文件详解

Drupal6 的模板文件详解--page.tpl.php

变量名	描述
<code>\$breadcrumb</code>	网站的层级导航链接，如“首页>管理>模块”。
<code>\$closure</code>	这个变量内主要包含一些页面使用的 JavaScript 等客户端脚本，它们通常在 <code></body></code> 标记结整前被载入。也就是当整个页面加载完时，再加载客记脚本。
<code>\$content</code>	页面的 HTML 代码
<code>\$footer_message</code>	在 <code>admi/settings</code> 页面定义的“底部信息”。
<code>\$head</code>	一些模块生成的 HTML 代码，在 <code><head></code> 标签中输出。
<code>\$head_title</code>	页面的标题，会在 <code><title></code> 标签中输出。它的值是由 <code>\$title</code> ， <code>\$title_name</code> 和 <code>\$site_name</code> 以及 <code>\$site_slogan</code> 组成的。
<code>\$help</code>	帮助文字。
<code>\$language</code>	页面的语言编码。例如英语就是 <code>en</code> 。
<code>\$layout</code>	这一变量允许定义外观的不同类型的风格，而变量 <code>\$layout</code> 的值依赖于启用的工具条

(sidebars) 的数量。可能的值包括: none, left,right, and both.

\$logo 站点图标的地址。

\$messages 给用户的状态和错误信息

\$primary_links 一个包含了一级链接的数组。在 Administer->Site building->Menus 中定义它们。通常

\$secondary_links 通过函数 theme('links')来定制输出的样式

\$search_box 返回搜索表单的 HTML。当管理员在启用的主题中的主题配置页面禁止展示搜索时, 或者搜索模块禁用时, \$site_slogan 为空。

\$secondary_links 一个包含了二级链接的数组。在 Administer->Site building->Menus 中定义它们。通常

\$secondary_links 通过函数 theme('links')来定制输出的样式。

\$sidebar_left 返回左边工具条的 HTML, 包含了属于该区域的所有区块的 HTML。

\$sidebar_right 返回右边工具条的 HTML, 包含了属于该区域的所有区块的 HTML。

\$site_name 站点的名称。在 Administer->Site configuration->Site information 中设置。当管理员在启用的主题中的主题配置页面禁止展示标语时, \$site_name 为空。

\$site_slogan 站点的标语。在 Administer->Site configuration->Site information 中设置。当管理员在启用的主题中的主题配置页面禁止展示标语时, \$site_slogan 为空。

\$styles 返回连接到页面需要的 CSS 文件的 HTML。CSS 文件通过 drupal_add_css()添加到 \$styles 中去。

\$tabs 返回用于为节点展示诸如 View/Edit 的标签的 HTML。

\$title 主内容的标题, 与 \$head_title 不同。当察看一个单独节点视图页面时, \$title 就是节点的标题。当常看 Drupal 的管理员页面时, \$title 通常有菜单项来设置, 菜单项对应于查看的页面。

Drupal6 的模板文件详解--node.tpl.php

变量名	描述
\$content	节点的主体部分, 如果是一个分页显示的结果时, 它为摘要 (teaser) 部分。
\$date	节点被创建的格式化日期。
\$links	与节点相关的链接, 例如“read more” 或者“add comment.”模块通过实现 hook_link()来添加额外的链接。
\$teaser	布尔值, 用来决定是否展示 teaser。当它为假时, 意味着节点采用 body 方式展示, 为真时, 表示以 teaser 方式展示。
\$name	创建该页面的用户名, 连接到他的个人主页。
\$node	节点对象。
\$node->body	整个 node 的内容
\$node->changed	本节点的最后修改日期的 UNIX 时间戳
\$node->created	本节点的创建日期的 UNIX 时间戳
\$node->nid	节点的 nid, 和数据库中的字段相对应。
\$node->teaser	节点的摘要
\$node->title	节点的标题
\$node->type	节点的内容类型 (content type), 如 story, blog, forum 等。
\$node->uid	创建此节点的用户 uid
\$node->name	创建此节点的用户 username
\$node_url	该节点的持久化的 URI。
\$picture	用户图像的 HTML, 如果启用了图像并且设置了用户图像。
\$submitted	“Submitted by”文本。管理员可以配置这一信息的展示, 在一个基于单个节点类型的主题配置页面配置。
\$terms	与该节点相关的分类单词的 HTML。每一个单词都指向他自己的分类单词页面。

<code>\$title:</code>	节点的标题。当在多个节点列表的页面这里有个链接指向该节点的主体视图。
<code>\$sticky</code>	置顶标记。如果为真，那么这个节点的内容就会一直在列表的上方。
<code>\$taxonomy</code>	由节点的分类词语构成的一个数组
<code>\$page</code>	当为 <code>body</code> 视图方式时，它为真，当为 <code>teaser</code> 视图方式时，它为假。
<code>\$id</code>	节点的 ID
<code>\$is_front</code>	是否为首页的标志

Drupal6 的模板文件详解--comment.tpl.php

变量名	描述
<code>\$author</code>	评论者的名称，并可以链接到用户的个人信息页。
<code>\$comment</code>	评论的对象，是在 <code>theme_comment</code> 函数中传递的。
<code>\$comment->subject</code>	评论的标题
<code>\$comment->comment</code>	评论的内容
<code>\$comment->name</code>	评论者的名称
<code>\$comment->timestamp</code>	评论的时间戳（并不是日期格式，而是 UNIX 的时间戳，需要使用 <code>date()</code> 函数进行转换）
<code>\$comment->uid</code>	评论者的 UID
<code>\$comment->new</code>	判断此评论是否被当前的用户浏览过。如果否，那么此评论对此用户而言，就是新评论，会在评论标题旁显示“new”字样。
<code>\$content</code>	评论的内容，和 <code>\$comment->comment</code> 内容相同。
<code>\$date</code>	格式化过的发布日期
<code>\$links</code>	操作链接，如“回复”、“编辑”和“删除”等。
<code>\$new</code>	与 <code>\$comment->new</code> 相同
<code>\$picture</code>	如果设为开启状态，并且发布者具有有效的头像，那么这个变量的内容就是用户的头像，并有一个链接可跳转至用户的个人信息页。
<code>\$submitted</code>	多语言的“Submitted by user_link on date.”
<code>\$title</code>	评论的标题，带有链接。
<code>\$id</code>	评论的 ID
<code>\$zebra</code>	奇偶标志位，用于交替样式的显示
<code>\$is_front</code>	是否在首页上显示

下面是 PHPTemplate 的默认的 `comment.tpl`

Drupal6 的模板文件详解--box.tpl.php

变量名	描述
<code>\$content</code>	块内容
<code>\$region</code>	块显示的屏幕位置。可以是 <code>main</code> , <code>left</code> 或 <code>right</code> 。默认是 <code>main</code> 。
<code>\$title</code>	标题
<code>\$id</code> 本块 (box)	所在的区块 (block) 的 ID 号
<code>\$zebra</code>	序号的奇或偶。可以通过这个变量来生成交替的样式效果。
<code>\$is_front</code>	判断本块是否在首页上

Drupal6 的模板文件详解--block.tpl.php

变量名	描述
<code>\$block</code>	区块对象
<code>\$block->content</code>	区块内容
<code>\$block->module</code>	生成此区块的模块 (module) 名
<code>\$block->region</code>	区块所在的区域 (region)。这个区域可以是在 <code>.info</code> 中定义的；
<code>\$block->subject</code>	区块的标题
<code>\$block->delta</code>	区块的序号。这个是在前面讲过的，用模块生成区块时的序号。

<code>\$is_front</code>	使用此变量可以判断是此区块是否在首页<front>上
<code>\$id</code>	区块在本页中 ID 号
<code>\$block_id</code>	区块在所有区块中的 ID 号

drupal6.x 中创建新的主题模版和区域(Regions)

本文分成 2 部分来讲解的主题和区域。内容的流程和介绍只代表我个人的理解和收集，一些想法和操作并不完全正确。只是为了给新接触 drupal 的朋友一点提示。如果读者的英文没问题请访问 drupal.org 阅读相关的英文文档。

一、创建新的主题模版

Drupal 6.x 系统自带的模版位于网站根目录 `themes` 目录中，Drupal 鼓励用户把非系统自带的模版放在 `\sites\all\themes` 目录中，并且不要修改系统自带的模版。好了在了解 Drupal 6.x 的模版存放的位置，我们就开始。

复制 `\sites\all\themes` 目录中 `garland` 目录到 `\sites\all\themes` 目录中（如果没有就创建它）

重命名 `\sites\all\themes` 目录中 `garland` 目录为：`test`

以下内容如没有特殊说明 `test` 目录即为 `\sites\all\themes` 目录中 `test` 目录

打开 `test` 目录，将 `garland.info` 重命名为：`test.info`，并编辑 `test.info` 文件

将：`name = Garland` 修改为 `name = test` 并保存

接下来到“管理--站点构建--主题”，其 URL 为：`/admin/build/themes`

你会发现刚才创建的 `test` 主题 已经出现在 主题列表中了，我们在 `test` 主题栏 选中“启用”和“默认”，然后点击“保存配置”

至此就完成了创建新的主题模版 `test`，并把它设置为站点的默认模版

二、增加新的区域(Regions)

Drupal 6.x 把一些模版相关的信息和参数设置都放在 `.info` 文件中，如：主题的名字、模版引擎类型、兼容的 Drupal 版本... 等等

在“创建新的主题模版”章节中我们编辑了 `test.info` 文件，将：`name = Garland` 修改为 `name = test`，请注意在 Drupal 5.x 中模版的名字是取决于主题模版的目录，而在 Drupal 6.x 中，模版的名字取决于 `test.info` 文件 `name = 模版名字`

Drupal 6.x 默认有 5 个区域(Regions):

`Left`, `right`, `content`, `header`, `footer`

在模版 `page.tpl.php` 文件我们可以使用：`$left`, `$right`, `$content`, `$header`, `$footer`

注意在 Drupal 5.x 是：`$sidebar_left`, `$sidebar_right`, `$content`, `$header`, `$footer_message`

Drupal 6.x 的 `$left`, `$right` 相等与 Drupal 5.x：`$sidebar_left`, `$sidebar_right`

Drupal 6.x 中 `$footer` 是新增的。

Drupal 6.x 中也有 `$footer_message` 变量，它的作用是打印显示“管理--站点设置--站点信息”的“页脚信息”文本框的内容：

如果我们需要增加一个变量名为 `$content_top` 新的区域(Regions) 也需要在 `test.info` 设置

在行 `stylesheets[print][] = print.css` 下新增下面代码

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

保存 `test.info` 文件

其中这些是系统默认区域(Regions)，我们需要把这些也添加进去

```
regions[left] = Left sidebar
```

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

下面这行是我们新增的区域(Regions)

```
regions[content_top] = Content top
```

接下来我们打开 page.tpl.php 文件, 将刚才新增的区域(Regions)添加到模版中找到:

```
<div class="clear-block">
  <?php print $content ?>
</div>
```

修改为:

```
<div class="clear-block">
  <!-- content_top -->
  <?php print $content_top ?>
  <!-- content_top end -->
  <?php print $content ?>
</div>
```

保存 page.tpl.php 文件。(注意: 如果添加了中文字符, 保存时记得把编码改成 UTF-8)

当然想放到那里看个人意愿和需要了, 我们这里只是演示。

接下来我们需要主题注册 (theme registry)

到“管理--站点设置--性能”其 URL 为: /admin/settings/performance

点击“清除缓存数据”按钮 来清除数据库缓存。

然后我们到“管理--站点构建--区块”其 URL 为: /admin/build/block

正常情况下我们会看到刚才新增的 Content top 区域(Regions), 你可以尝试一点操作:

将页脚区域的区块 Powered by Drupal 移动到 Content top 区域(Regions)

然后记得点击“保存区块”按钮。

最后我上传 page.tpl.php 和 test.info 文件, 希望对你有所提示帮助。

Block (区块) 管理

Drupal 通过 Block 来控制页面元素的显示, 例如左边栏, 右边栏, 头部, 页脚, 内容区域显示的内容。请大家先看一下这个页面: <http://www.alexacn.org/>, 左边栏显示教程链接等, 右边栏显示多个链接, 底部显示版权信息, 而顶部和内容区域没有使用区块的内容。下面我们就告诉大家如何通过区块来管理这些内容的显示以及如何自定义区块。

当你浏览 drupal 页面时, 左右边栏通常有一些导航条或提供一些额外信息的栏目, eg: 最新发布的文章, 最新评论, 导航. 甚至你还可以将一个节点的内容放在里面.....它可以分布在一个 Drupal 站点的边栏上或其它地方。这些位置是由你的模板或叫做主题事先设置好了的 regions(区域). 供你放置区块的位置通常是(header, left-sidebar, content, right-sidebar, footer), 许多区块是通过不同的 Drupal 模块动态生成的, 但你也可以自己添加任意个模块。

下面我来分二部分说说区块的使用, 一个是简单的使用, 即使用现成的区块, 二是定制自己的区块

A: 简单的使用

首先找到区块的位置, drupal 6 后的版本位置是(administer----->site building----->blocks), 打开 blocks, 出现的界面如下:



说明:在区块位置(region)下面,你可以使用鼠标拖拽来显示区块的显示顺序,没有启用的模块内会包含有一些模块生成的区块,若你在模块内启用了 **Throttle(节流)**,在配置位置的右边将会出现节流按钮,你可以选择将其勾上或不选,作用是:当你的站点访问量超过一定值时(你在 Throttle 内可设置),勾上的区块,将会被关闭。进行了一些操作后记得点击下面的保存按钮,否则无法生效,接下来我们来看看怎么配置区块,它可以使你更好的利用区块,可以使你设置什么用户,或在那些页面下显示.下面我来截个图说明



B:定制区块

定制区块可分为简单的定制和复杂的定制,后面的定制应该能算高人了,哈哈.....

简单定制:首先进入设置区块的位置,如下图:



点击进入后出现的图片如下,下面我们来做些设置.如下图:



说明:区块的 body 部分实在是强大,你可以和 Drupal 内的分类结合起来将其做成一个站点导航,可以将其作成你想要的东西,只要你能够用 HTML 或 PHP 将其编写出来.

复杂定制:

使用区块最高境界了,就是不依靠模板默认提供的几个位置来显示区块.这可能需要了解关于 Drupal 的更多知识,比如:模板,以及内部函数等.....

下面我来做个在每个内容下面添加个区块.

首先进入你所使用的模板文件内,找到.info 文件,drupal 5.x 不是在这里,加入代码

1. regions [left] = Left sidebar
2. regions[right] = Right sidebar
3. regions[content] = Content
4. regions[header] = Header
5. regions[footer] = Footer
6. regions[belowPost] = Below Posts

说明: Left sidebar

regions[right] = Right sidebar

regions[content] = Content

regions[header] = Header

regions[footer] = Footer

是默认的,当你添加上新的 regions 后,它将覆盖了模板提供的默认 regions,所以若.info 文件内没有上叙的,就应该加上.现在我添加了一个新的 regions,

再在 node.tpl.php 内添加代码:

1. `<?php print $belowPost ?>`
2. 在`<?php print $content ?>`后面

再在 template.php 内加入代码

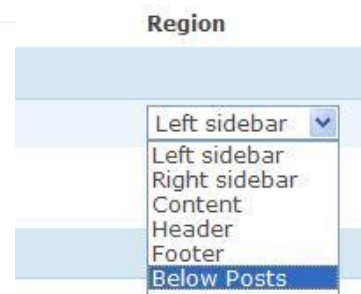
1. `function phptemplate_preprocess_node(&$vars) {`
2. `if (!$vars['teaser']){`
3. `foreach (array('belowPost') as $region) {`
4. `$vars[$region] = theme('blocks', $region);`
5. `}`
6. `}`
7. `}`

看看我测试的效果:图片如下:



区域内 regions 的图片如下:

区块的作用大吧!你想怎么弄就怎么弄,想让在里面显示什么就显示什么!另外,需要注意的是,当你自己添加个 regions 时,记得在 site



configuration----->performance----->Clear cached data,清除下模板的缓存,regions 才能起作用.
后天再来讲下加那些代码的作用

区块定制总结:

总体来说可以分为三步:

第一步:在模板文件.info 中定义一个新的 regions.格式是:regions[内部名称] = 显示在区块内的名称.eg:regions[footer] = Footer,(Drupal 是通过在模板内使用 regions 来控制显示区块的,所以需要定义),记得加上默认的那些 regions.(如果没有定义的话,系统将会使用默认值,若添加新的 regions,将使原来的失效,所以需要加上).

第二步:由于.info 在主题内存在缓存,所以添加后你需要去 Administer -> Site configuration -> Performance 清除缓存.

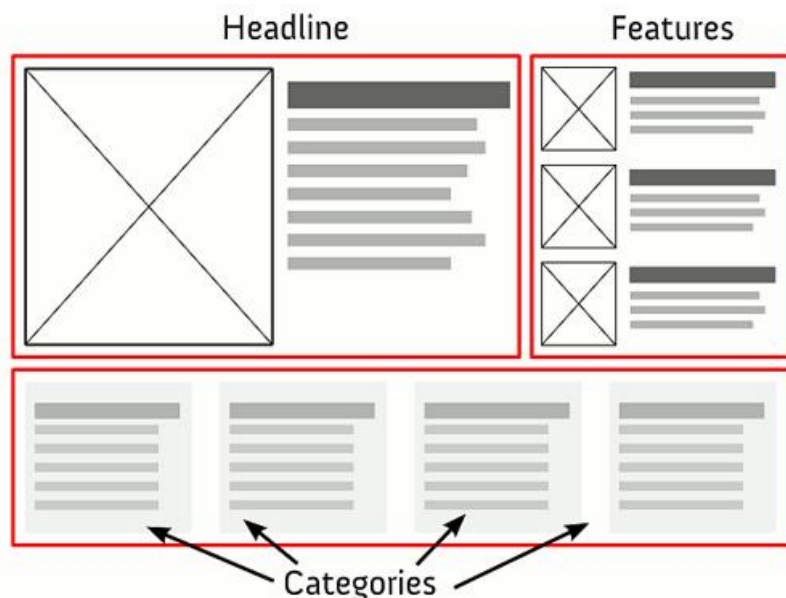
第三步:我们需要将新的 regions 放在模板内,并来显示他们.若需要在页面内加入区块,则只需要在 page.tpl.php 内在你想要它出现的位置添加上<?php print \$内部名称 ?> 由于内部名称能够自动在 page.tpl.php 模板内自动转换成 region 变量,所以你无须再做其他操作.若是加在其它地方,你需要在 template.php 内使用 function 函数来处理变量,因为它不能够动态生成,

-----END-----

如何为 Drupal 网站创建一个 Magazine Style 的首页

杂志样式首页 (Magazine Style Homepage) 是一种应用很广泛的首页样式,它能很好的组织与展示大量的内容,尤其是图文混排的内容,这种样式我们在各种在线杂志、门户网站等等上面看到过很多,在 CMS 中,我们也见到过很多 WordPress 的 Magazine Style Theme,但是我们确很少看到有 Drupal 使用的 Magazine Sytle 主题,这使得很多刚刚接触 Drupal 的人认为 Drupal 还没有 WordPress 的定制能力强。

本文为大家简略介绍一下,如何为 Drupal 创建一个 Magazine Style 的主题首页,这个首页有三个内容显示区块,左上区域为一个“首页头条”区块,在这个区块中,将显示一篇文章的摘要、文章的缩略图以及标题,这个标题是指向到文章页面的链接;右上区块为“推荐阅读”区块,在这个区块中,将显示被推荐到首页显示的文章的标题列表,每一个标题后带有该文章的缩略图;下方为“分类文章”区块,我们将四个分类的文章标题列表在这个区块中以文章分类为基础分开显示。如下图所示:



本文章对读者的主要要求有:

- 使用过 Drupal 一段时间,并且看过 Drupal 默认的主题的源代码(推荐)
- 了解如何安装 Drupal 模块与主题
- 了解一些基本的 PHP、CSS 以及 xHTML 知识

在本文章正式开始之前我们首先请大家安装好下列 Drupal 模块:

- [CCK \(Content Construction Kit\)](#)
- [FileField](#)
- [ImageAPI](#)
- [ImageField](#)
- [ImageCache](#)
- [Views](#)

我们在撰写本文时使用的是 [Drupal 6.10 版](#)，使用的主题是 [Framework theme](#)。

使用 ImageCache 创建三个图片自动缩放动作

按下表所示信息创建三个 ImageCache Preset:

ImageCache Preset Dimensions

Namespace	Width	Height	Actions
Large	640px	640px	Scale And Crop
Medium	350px	350px	Scale And Crop
Thumbnail	100px	100px	Scale And Crop

ImageCache 设置完成之后，我们现在可以通过 CCK 为 Story 内容类型增加一个图片字段，当我们在撰写文章时，如果作者通过此字段上传了图片，那么 ImageCache 将自动对图片进行缩放与裁切。

使用文章分类来使事情简单化

我们可以在网上随意搜索到很多 WordPress 的免费与收费的 Magazine Style Theme，我们稍稍研究 WordPress 的这些主题就可以很简单的发现他们都是通过一个十分简单的方法来读取首页内容的，那就是“分类”，我们现在要实现相同的功能，所以也可以使用相同的方法。

首先我们创建一个词汇表，名为 Tags（标签），将其设置成为“Tags”，这样我们就有了和 WordPress 或者 Movable Type 等博客程序一样的标签系统，将标签类型中的“Story”前面的复选框选中，点击保存；然后我们添加一个名称为“Category”（分类）的词汇表，将其设置成为“Multiple Select”（多选），然后将“Story”添加到这个词汇表中，这样我就有了博客系统中的分类系统。接着我们在 Category 中添加几个词汇（项目）：

1. **Headline**（首页头条）：所有属于这个分类的文章都将在首页头条区域显示，当然，我们可以设置最多显示几篇；
2. **Features**（推荐阅读）：所有属于这个分类的文章都将在首页的推荐阅读区块显示；
3. 其它分类：
 - 网站新闻
 - 生活纪实
 - 学习日记
 - 建站手记
 -

注册与创建区块（Create Regions）

到现在为止，网站、内容、模块与主题的基本配置已经完成，现在我们需要往更深层次学习了，首先，我们进入网站主题的文件夹（我们使用的是 Framework），之后将 page.tpl.php 文件复制一个复本并将复本名称修改为 page-front.tpl.php，Drupal 将会自动使用这个模块文件来生成网站的首页。

下面这一段代码是在 Drupal 模板文件中调用某个区块所使用的：

```
if ($the_region) { print '
'; print $the_region; print '
'; }
```

上面代码中的 \$the_region 变量可以代表任何内容，它是在模板的“Info”文件中注册的，其注册的格式为：regions[the_region] = My Region。其中的 the_region 是用来被机器识别的字段，而 My Region 是前面机器字段所代表的区块的

人类识别用的名称，如下我注册两个名称为 Headline 和 Features 的区域，它们的机器识别字段分别为 headline 和 features。

```
regions[headline] = Headline
regions[features] = Features
```

这个时候你并不能查看你的首页，因为上面注册了两个新的区块，这样一来，把 Drupal 默认注册的一些区块都给覆盖了，现在我们需要重新将 Drupal 默认注册的区块重新注册一次，如下：

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

在区块注册这一步骤中的最后一步是将 Headline 和 Features 这两个区块添加到模板中，我们打开 page-front.tpl.php 文件，找到文件中的主区块（如果你使用的是 Framework 主题，那么就选择 id 为 center 的 div 为主区块），将这个盒子中原来的代码全部删除，再将下列代码添加到主区块中去：

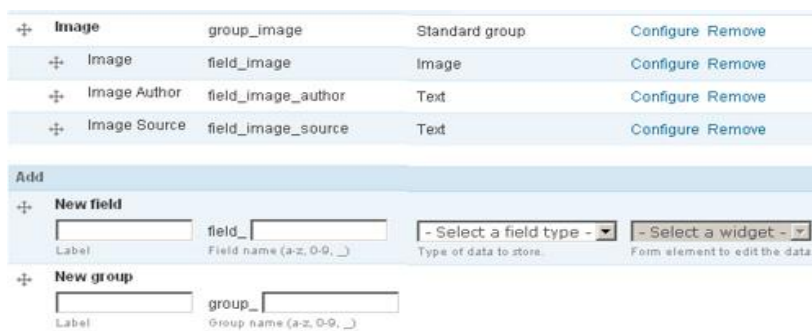
```
print '<div class="main">';
if ($headline) { print '<div class="region">';
print $headline; print '</div>'; }
if ($features) { print '<div class="region">';
print $features; print '</div>'; } print '</div>';
```

为了在后面的工作中更加容易的样式化这块区域，我在这里为这个区块添加了一个 CSS 类：main，为 Headline 和 Features 区块都添加了一个 CSS 类：region。

通过 CCK 扩展 Story 内容类型

你想让 Story 类型与 ImageCache 模块的功能联系到一起，首先你需要使用 CCK 为 Story 添加额外的字段，至少需要一个“Image”（图片）字段，如果你还想为图片添加一些版权信息，你还可以添加图片作者字段或者图片来源字段，现在，请按下表为 Story 添加三个字段：

Label	Field Name (prefixed field_)	Type of Data to Store	Form element to Edit Data
Image	image	Image	Image
Image Author	image_author	Text	Text Field
Image Source	image_source	Text	Text Field



添加这三个额外的 Field 的时候没有什么过多需要设置的，仅仅只需要注意的就是，添加 Image 字段的时候，我们需要为上传的图片选择一个保存目录，默认是使用网站文件根目录，为了管理与整理方便，我们在这里设置一个新的目录“images”，这样一来，所有上传的文件都会被保存到根文件目录的子目录“images”中。

创建 Views

Views 与 CCK 的配合使用可以说是 Drupal 应用的核心，所以，一般有 CCK 就一定会用 Views，下面我们就要开始为我们的 Magazine Style Frontpage 创建 Views 了。如果你不想自己一步一步添加，你只需要直接本文下方的导入文件直接导入到您的网站上去即可，不过我还是建议你一步一步自己动手来做。

Headline View

Headline View 是用来展示你认为最重要的文章的，在这个区块中，它将显示一个固定大小的图片，还有文章的标题和摘要，文章标题链接到文章的页面，现在我们来添加 Headline View。进入 View 添加页面（[Site Building » Views » Add](#)），View 的名称为“Headline”，介绍你可以填写任何东西，一般就是介绍这个 View 的作用等，Tag 就像文章的 Tag 一样，可以方便我们在网站有大量的 View 的时候查找特定的一个 View，填写好了之后点击 Next（下一步）：

View name:

This is the unique name of the view. It must contain only alphanumeric characters a theming template names for this view. If overriding a module provided view, the name must be the same as the module's name.

View description:

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

View tag:

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

View type:

Node
Nodes are a Drupal site's primary content.

Aggregator item
Aggregator items are imported from external RSS and Atom news feeds.

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.

Access log
Stores site access information.

The view type is the primary table for which information is being retrieved. The view type is set and cannot be changed.

进入下一页之后，首先在左上角的下拉菜单中选择 Block，然后点击添加一个 Block Display，之后按下面的信息来设置你的这个 Display 的基本设置（Basic Settings）：

- Name: Headline
- Title: None
- Style: Unformatted
- Row style: Fields
- Use AJAX: No
- Use pager: No
- Items to display: 1
- More link: No

- Distinct: No
- Access: Unrestricted
- Exposed form in block: No
- Header: None
- Footer: None
- Empty text: None
- Theme: Information

之后为这个 Display 添加 Fields，点击 Fields 右侧的“+”号，按下表来添加三个 Fields：

Field	Notes
Content: Image: Image (field_image)	Label: none Format: medium image linked to node
Node: Teaser	No additional configuration required
Node: Title	Check Link this field to its node

之后，再按下表添加 Filters

Field	Notes
Node: Published	True
Taxonomy: Term	Vocabulary: Categories Selection Type: Drop down Operator: Is one of Select terms from vocabulary Categories: Headline

如下图所示为添加好之后的界面：



之后别忘记点击保存按钮了。

Features View

Features 区块将显示最新三篇被标记为 Features 的文章，每一篇文章都有自己的缩略图和摘要。现在再添加一个名称为 Features 的 View，类型继续为 Node（节点），并按下面几个表格进行设置：

Basic settings

- Name: Block
- Title: Features
- Style: List
- Row style: Fields
- Use AJAX: No
- Use pager: No
- Items to display: 3
- More link: No
- Distinct: No
- Access: Unrestricted
- Exposed form in block: No
- Header: None
- Footer: None
- Empty text: None
- Theme: Information

Fields

Field	Notes
Content: Image: Image (field_image)	Label: none Format: thumbnail image linked to node
Node: Teaser	No additional configuration required
Node: Title	Check Link this field to its node

Sort criteria

Field	Notes
Node: Post date	asc

Filters

Field	Notes
Node: Published	True
Taxonomy: Term	Vocabulary: Categories Selection Type: Drop down Operator: Is one of Select terms from vocabulary Categories: Featured

Posts by Category

这个 View 与前面的两个 View 有一些不一样，它会展示相关分类的最新十篇文章标题列表，每一个标题又链接到文章页面，它与前面两个 View 最大的不同点在于，你不需要将这个 View 添加到哪个区块中，你需要直接在 FrontPage 模板文件中添加相关的 calls 来调用相关的内容，添加一个新的 Node 类型的 View，你只需要使用 Default (默认) 的 Display 即可。

Basic settings

- Name: Defaults
- Title: None
- Style: List
- Row style: Fields

- Use AJAX: No
- Use pager: No
- Items to display: 5
- More link: No
- Distinct: No
- Access: Unrestricted
- Exposed form in block: No
- Header: None
- Footer: None
- Empty text: None
- Theme: Information

Arguments

Field	Notes
	Action to take if argument is not present: Display all values
Taxonomy: Term	Validator:<Basic Validation>
	Action to take if argument does not validate: Hide view/Page not found (404)

Fields

Field	Notes
Node: Title Check Link	Link this field to its node

Filters

Field	Notes
Node: Published	True

Views Templates

如果前面的内容你都亲手操作并完成得相当漂亮,那么现在可以进入更深入的层次了,我们在这一小节里面将制作 Views Templates (Views 模板),用来定制内容的输出。Views 有很好的模板支持机制,在你没有为特定的 View 创建模板时, View 将调用主题默认的模板,但是当你为 View 创建的特定的模板时,这个 View 将使用你创建的模板来生成 View 的内容。

Headline Template

```
<div class="headline-summary"> <div class="imageframe"><?php print
$fields['field_image_fid']->content; ?></div> <h2>Headline</h2> <h3><?php print
$fields['title']->content; ?></h3> <div class="teaser"> <?php print $fields['teaser']->content; ?> </div>
</div>
```

将上面的这一段代码复制到名为 views-view-fields--Headline--block.tpl.php 的文件中,再将这个文件复制到你的主题文件夹里。这个文件会被 Views 模块自动调用并分配到 Headline View 的输出过程中。

Features Template

```
<div class="featured-summary"> <div class="imageframe"><?php print
$fields['field_image_fid']->content; ?></div> <h4><?php print $fields['title']->content; ?></h4> <div
class="teaser"> <?php print $fields['teaser']->content; ?> </div> </div>
```

将上面的这一段代码复制到名为 views-view-fields--Features--block.tpl.php 的模板文件中,再将此文件复制到主题文件夹,现在你已经有了完全由你控制输出的 Headline 和 Features 区块。

Categories Template

这里是本小节的一个难点,但是也是很有用的一点,所以,在这里大家认真的学习,因为在这里,你将需要在 page-front.tpl.php 文件中添加一些代码:

```
<div id="categories" class="region"> <?php /* Enter the ID of the vocabulary which terms you want to show
as headlines */ $vocabulary_id = "1"; /* Enter the IDs of any terms that should not be displayed */
$terms_to_exclude = array(1, 6, 7, 271); /*Enter the name of the view that should show the nodes */ $view_name
= "posts_by_category"; foreach(taxonomy_get_tree($vocabulary_id,0,-1,1) as $value) { if
( !in_array( $value->tid, $terms_to_exclude ) ) { print t("<div class=\"category-list
$value->description\">\n<h3\" . $value->name . \"</h3\""); print views_embed_view($view_name, 'default',
$value->name); print t("</div\""); } } ?> </div>
```

将上面这些代码复制到 page-front.tpl.php 文件中主内容区块里，即 Features 区块的下方，这一个循环会向 Posts_by_Category View 发出请求，根据 Vocabulary 来生成内容。

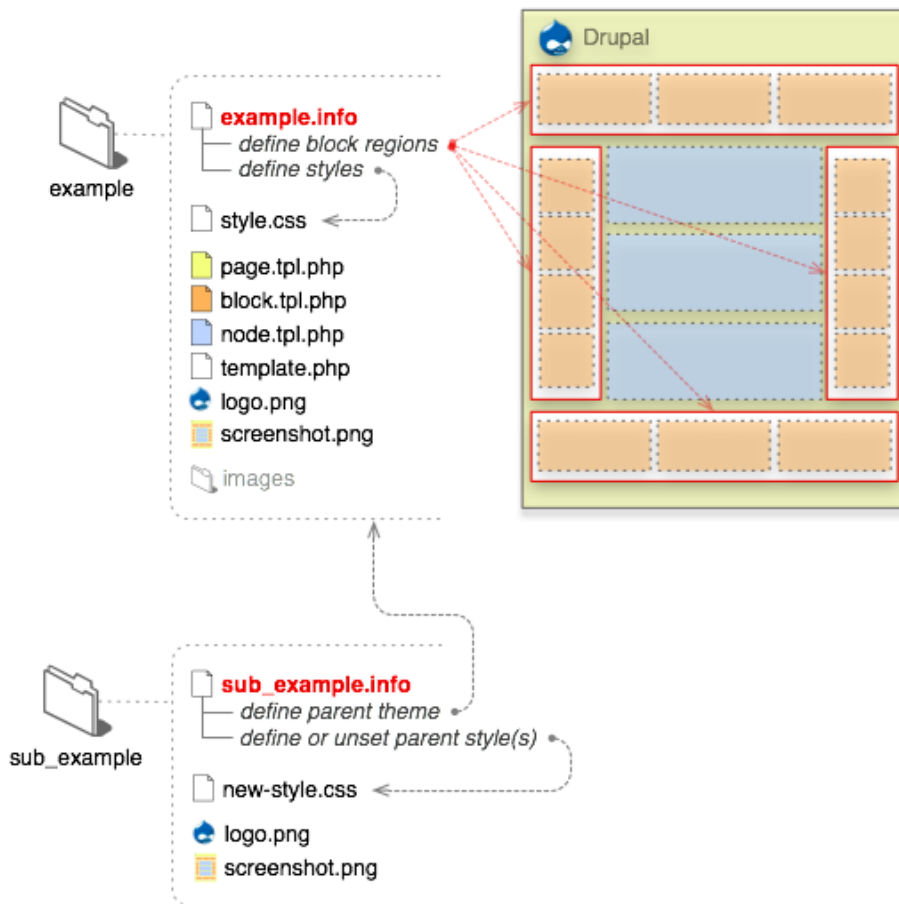
样式化

如果你完成按照我们的要求来做，那么现在你可以发布一些文章来测试了，如果没有出错，首页将会按我们的要求输入一个 Headline 和一个 Features，并且会将存在内容的几个 Category 输出，每一个分类一个区块，每一个区块内都有文章标题列表，现在要做的工作就是对这些内容进行样式化。样式化在这里不属于 Drupal 的专业内容，所以本文不作探讨，如果你不知道如何样式化这些内容，请去查阅与 CSS 相关的文章或者教程。

*****END*****

对一个 Drupal 主题的剖析

phptemplate theme:



.info (必须)

".info"文件是必须的,当 Drupal 寻找你的主题时,要用到这个文件.在这里可以定义:元数据,样式表, [JavaScripts](#).(区块)区域, 以及其它.而其它文件则是可选的.

观,那该怎么办呢?使用子主题,就可以在各个自站点中分享共同的设计资源。子站点相关的修改放在子主题里,而共享的设计资源则可以放在基主题中,这样一旦修改了之后,就可以应用到所有的子站点中了。如果仔细规划,那么就具有无限的可能。

为了声明一个父主题或者“基主题”(“base theme”),那么在子主题的.info 文件中设置下面一项,其中“ themeName”(“主题名”)为父主题在 Drupal 内部的名字:

```
base theme = themeName
```

下面的将被继承:

- 所有在父主题中定义的样式表(<http://drupal.org/node/171209>),但可以进行选择 (<http://drupal.org/node/171209#styles-override-parent>),所以这是可以控制的。
- 所有在父主题中定义的 **JavaScripts**。
- 所有的模板文件 (.tpl.php)。
- "template.php"文件中定义的所有东西。包括覆写的主题函数,预处理器,或者其它东西。每个子主题都包含它自己的 template.php 文件和其父主题的 template.php 文件。
- 如果子主题中.info 文件和父主题的设置相同的话,父主题的截图(Screen shot)也是可以继承的。

下面的不能被继承:

- 主题的 logo.png。不包括上传的 logo,因为这是总被用到的。
 - .info 文件中的一些设置。包括区域。如果你没有使用默认区域的话,那么子主题的"page.tpl.php"文件中用到的区域,一定要在.info 文件中定义过。每个基主题和子主题都可以拥有自己的区域设置。
 - 在高级主题设置 (<http://drupal.org/node/177868>) 中, "theme-settings.php"文件中的所有东西。
- 注意,子主题可以放在基主题目录的外面,也可放在里面。而在 Drupal6 以前的版本中,它们只能放在父主题目录的里面。

END

主题的.info 文件

Drupal 可以识别下面所列的键。如果.info 文件没有设置的话,Drupal 将为其使用默认值 (<http://drupal.org/node/171206>)。可参看核心主题中的例子 (<http://drupal.org/node/171205#example#example>)。

- **name** ! (名字)
- **description** * 描述
- **screenshot** 截图
- **version** * 版本
- **core** ! 内核
- **engine** * 引擎
- **base theme** 基主题
- **regions** 区域
- **features** 特性
- **stylesheets** 样式表
- **scripts** 脚本
- **php** php

name (required) 名字(必须)

这是用户可读的名字,与主题的 Drupal 内部名字可以分开单独进行设置.这在这里,字符的限制则很少.

```
name = Un tema nombre de fantasia
```

description (recommended) 描述(推荐)

主题的简短描述.你可以在页面"Administer > Site building > themes"看到主题的描述.

description = Tableless multi-column theme designed for blogs.

screenshot 截图

截图键时可选的,它告诉 Drupal 主题的缩略图在哪里,在选择主题页面(admin/build/themes)里用到了缩略图.如果.info 文件中忽略了该键,那么 Drupal 就会使用主题目录下面的"screenshot.png"文件.

只有当你的缩略图不叫"screenshot.png",或者你不想把它放到你主题的根目录(比如, screenshot = images/screenshot.png)下面时,才使用该键.

screenshot = screenshot.png

version (recommended) 版本 (推荐)

当发布一个新的版本时, drupal.org 会自动为其添加一个版本号.当你为 Drupal 贡献主题时,你可以忽略该值.如果你的主题没有放到 drupal.org 上的话,你可以为你的主题指定任意一个版本号.

version = 1.0

core (required) 内核 (必须)

从 Drupal 6.x 开始,模块和主题.info 文件都必须指明它们兼容的 Drupal 内核主版本号.这里设的值将与 DRUPAL_CORE_COMPATIBILITY 常量相比较.如果不匹配的话,那么主题将被禁用.

core = 6.xdrupal.org 的打包脚本,将根据每个发布版本的 Drupal 内核兼容性设置,自动设置该值.所以从 drupal.org 下载下来的主题,设置总是正确的.然而,对于直接通过 CVS 部署的 Drupal 站点来说,如果你将这一修改提交到你主题.info 文件中去的话,将会很有帮助.它也能够非常方便的帮用户指出,主题兼容 CVS 的 HEAD 中的哪些内核版本.

engine (recommended) 引擎 (推荐)

主题引擎,供主题使用.如果没有提供引擎的话,那么主题就是独立的,比如,实现一个".theme"文件.大多数主题都使用"phptemplate"作为默认引擎.

PHPTemplate 负责查找主题用到的主题函数和模板.只有当你理解你在做什么的时候,你才可以忽略这一设置.

engine = phptemplate

base theme 基主题

子主题可以声明一个基主题.这允许主题的继承,也就是说基主题中的资源将被传递下来并在子主题中使用.子主题可以声明别的子主题作为其基主题,也就是允许多重继承的存在.基主题的名字为其在 Drupal 内部的名字.下面是 Garland 的子主题 Minnelli 的相应设置.

base theme = garland 更多细节可参看子主题,它们的结构和继承 (<http://drupal.org/node/225125>).

regions 区域

我们这样定义主题中的区域,声明键'regions',紧跟着"[",接下来是内部名字,接着是"]",然后是一个等号,右边是用户可读的区域名字.例如, regions[theRegion] = The region name.

如果没有定义区域的话,那么使用下面的默认值.你可以根据自己的需要覆写这些值.

regions[left] = Left sidebarregions[right] = Right sidebarregions[content] = Contentregions[header] = Headerregions[footer] = Footer 更多细节可参看,"区块,内容和它们的区域"(<http://drupal.org/node/171224>).

features 特性

许多由主题控制输出的页面元素,可以在主题的配置页面启用或者禁用."features"键控制着出现在主题配置页面上的复选框.对于一个主题,如果你不想为其定义某个复选框时,着非常有用.为了删去某个复选框,只需要在"features"中将其删除即可.如果一个也没有定义的话,那么会输出所有默认的复选框.

下面的例子列出了所有由 features 键控制的元素.通过注释掉 primary_links 和 secondary_links 元素,那么站点管理员就不会看到这两个复选框了.

features[] = logofeatures[] = namefeatures[] = sloganfeatures[] = missionfeatures[] =
node_user_picturefeatures[] = comment_user_picturefeatures[] = searchfeatures[] = favicon; These
last two disabled by redefining the; above defaults with only the needed features.; features[] =
primary_links; features[] = secondary_links 更多信息参看“定制主题设置”(http://drupal.org/node/221905).
stylesheets 样式表

传统方式, 主题可自动的使用默认的 style.css, 并且可以在它们的 template.php 文件中通过调用 drupal_add_css()
来添加其它样式表。从 Drupal6 开始, 主题也可以通过.info 文件来添加样式表。

stylesheets[all][] = theStyle.css 更多信息可参看“样式表”一节 (http://drupal.org/node/171209)。

scripts 脚本

传统方式, 主题通过在 template.php 文件中调用 drupal_add_js()来添加 javascripts 脚本。从 Drupal6 开始, 主题
也可以通过.info 文件来添加 javascripts 了:

scripts[] = script.js 更多信息参看 JavaScript & jQuery 一节。

php

这个定义了主题支持的 PHP 最低版本。其默认值源自 DRUPAL_MINIMUM_PHP 常量, 它是 Drupal 内核所需要的 php
最低版本。对于一个新的版本, 如果需要的话, 可以对其进行重新定义。而对于大多数的主题, 都不应该添加这一项。

php = 4.3.3

.info 的默认值

下面是假定的默认值。如果它们没有被定义的话, 主题将自动使用这些值。

*注意: 这些默认值是作为一个组来共同起作用的。换句话说, 仅仅使用 regions[sub_header] = Sub-header 来覆写一个
区域的话, 将会造成其它默认区域的消失。为了使用它们必须对它们进行重新定义。对于"stylesheets"也是一样的。尽管
从技术的角度来看, 它不是一个组, 定义另外一个样式表, 除非你重新对"style.css"进行定义, 否则是找不到它的。*

regions

regions[left] = Left sidebar

regions[right] = Right sidebar

regions[content] = Content

regions[header] = Header

regions[footer] = Footer

features

features[] = logo

features[] = name

features[] = slogan

features[] = mission

features[] = node_user_picture

features[] = comment_user_picture

features[] = search

features[] = favicon

features[] = primary_links

features[] = secondary_links

stylesheets

stylesheets[all][] = style.css

scripts

```
scripts[] = script.js
```

screenshot

```
screenshot = screenshot.png
```

php (minimum support) (支持的最低版本)

[DRUPAL MINIMUM PHP](#) 是一个常量。它指出了运行 Drupal 所需 php 的最低版本。

```
php = DRUPAL_MINIMUM_PHP
```

Drupal 6 开发用版型

Drupal 6 開發用的版型，有 [Zen](#), [Basic](#), [Beginning](#), [Blueprint](#), [Framework](#), [Flexible](#), [Foundation](#), [Genesis](#)。

Zen: 使用這個版型，就一定要用 **subtheme**，也就是你自己新增一個 **theme** 資料夾，然後把要的東西從主 **theme** 資料夾複製進 **subtheme** 資料夾，很方便。並且 **div** 的層級很多，假設你的版型有許多圖，需要用 **css** 背景來實現(圓角框、多重底圖、圖形化邊框等等)，那就必須要用 **Zen**。

Foundation: 也是不錯的選擇。**foundation** 適合做較簡單、圖片少、乾淨整潔的版型。不是子版型。

Beginning&BeginningW2: 介於簡單和複雜的中間。非子版型，所以做法和 **foundation** 一樣。

Genises: ，也是子版型。

Clean: 看起來是簡單的小版型。

Framework: 和 **zen** 預設的 **classic** 有點像，沒使用過。

7 个步骤让您创建专属的 Drupal 版型

以下将介绍使用 Zen 来建立自己的 theme，zen theme 是一个很容易扩充子 theme 且广受欢迎的版型。

请您注意，**最好不要修改 zen/zen 档案夹下的档案**，请您依据以下的步骤操作。

0. 请您先下载 zen，并解压缩到 **sites/all/themes** 下

1. 将 **STARTERKIT** 档案夹复制到 **sites/all/themes** 下，并更改档案夹名称为您想要的名称：比如您命名 theme 为 **abcd**，那么请将 **sites/all/themes/zen/STARTERKIT** 数据夹复制出来，更改名称后，就会有 **sites/all/themes/abcd** 档案夹了。

2. 在您新建的目录中，请您将 **STARTERKIT.info** 改名为您目录名称。然后请您开始编辑 **.info** 档，将档案中所有 **STARTERKIT** 改为您的目录(版型)名称。例如：延续刚刚例子，我们将 **abcd/STARTERKIT.info** 改为 **abcd/abcd.info**。然后编辑 **abcd.info** 时，将 **STARTERKIT.css** 改为 **abcd.css**，更改”**name =**”的内容为”**ABCD**”，还有更改”**description =**”为”一个 zen 的子版型”

3. 接下来我们要决定 **layout** 了，**zen/zen** 中有两个 **css** 与 **layout** 有关，一个是 **zen/zen/layout-liquid.css** (变动的) 另一个是 **zen/zen/layout-fixed.css** (固定的)，请您选择需要的 **layout** 然后复制其档案到 **sites/all/themes/abcd** 档案夹下，并将档名改为 **layout.css**。

4. 将 **zen/zen/print.css** 复制到 **sites/all/themes/abcd** 档案夹下。

5. 将 **zen/zen/zen.css** 复制到 **sites/all/themes/abcd** 档案夹下并且更改档名为 **sites/all/themes/abcd/abcd.css**。

6. 编辑 **sites/all/themes/abcd** 档案夹下的 **template.php** 及 **theme-settings.php** 两个档案。请将两档案中所有”**STARTERKIT**”改为”**abcd**”

7. 登入网站后，请到**管理 > 网站建置 > 版型 (admin/build/themes)**中启用您刚刚新建的 **abcd** 版型。

Drupal Theme --- 按照内容类型设计页面布局

这篇文章介绍如果根据不同的内容类型 (**content types**) 进行页面布局 (**full page layout**)。比如说你想让你的 **blog** 和 **books** 拥有不同的外观。

步骤:

作为一个示例，下面一步一步演示一下如何为网站的 **blogs**, **books** 和首页 (**front page**) 拥有不同的页面布局。

首先:

1. 复制 page.tpl.php 文件一份副本, 将它重命名为 page-default.tpl.php
2. 再做几次相同的工作, 分别将其命名为"page-front.tpl.php", "page-blog.tpl.php" 和 "page-book.tpl.php" 等等
3. 使用编辑器, 修改每一个 tpl.php 文件, 按照你的要求进行修改页面布局
4. 将修改后的 tpl-type.php 文件(page-front.tpl.php, page-blog.tpl.php 等等)上传到你正在使用的 theme 目录下

然后:

1. 用下面这段代码替换 page.tpl.php 中的内容
2. 确保 theme 内有 page-default.tpl.php 文件
3. 上传 page.tpl.php 文件到你现在正在使用的 theme 目录内, 新的页面而已就会自动生效了

注:

1. 下面的代码片断可以按照新增加/删减的内容类型进行添加/删除。
2. page-default.tpl.php 文件将会是未指定特定内容类型的布局文件所默认的布局, 比如说, 按照前期设计要求完成了 theme 设计。后来可能根据新的需求又增加了新的内容类型, 那么它将会使用 page-default.tpl.php 的布局进行页面组织。

```
<?php
/**
 * This snippet loads up different page-type.tpl.php layout
 * files automatically. For use in a page.tpl.php file.
 *
 * This works with Drupal 4.5, Drupal 4.6 and Drupal 4.7
 */
if ($is_front) { /* check if it's the front page */
  include 'page-front.tpl.php'; /*load a custom front-page.tpl.php */
  return; }
if ($node->type == 'book') { /* check if it's a book page */
  include 'page-book.tpl.php'; /*load a page-book.tpl.php */
  return; }
if ($node->type == 'blog') { /* check if it's a blog node */
  include 'page-blog.tpl.php'; /*load page-blog.tpl.php */
  return; }
if ($node->type == 'image') { /* check if it's an image node */
  include 'page-image.tpl.php'; /*load page-image.tpl.php */
  return; }
if ($node->type == 'forum') { /* check if it's a forum node */
  include 'page-forum.tpl.php'; /*load page-forum.tpl.php */
  return; }
include 'page-default.tpl.php'; /*if none of the above applies, load the page-default.tpl.php */
return;
?>
*****
```

garland 主题页面分析与修改详解

让我们去掉我们不想要的内容, 包括把标题移到下面, 以及去掉"add new comment"链接, 还有"Mon, 10/20/2008 - 16:59 — admin", 以及黑体的标题, 正文, 就是显示的和上文的一样。现在让我们开始修改页面模板。

先到 sites\all 下面添加一个 themes 目录, 由于我们现在用的是官方的 garland 目录, 这是自带的, 最好不要直接修改它, 我们可以创建一个 garland 的子主题。不过在这里我们还是直接的在 garland 里面修改了。

直接就在 garland 目录下面修改了。拷贝 themes/garland 下面的 page.tpl.php, 并将复制品重命名为 page-front.tpl.php, 现在后者就负责首页的外观了, 但是现在我们还没有在里面做任何修改。打开 page-front.tpl.php, garland 的页面模板虽然丑陋, 但是却相当经典。以前我做页面模板时, 就是仿照这个, 通常 header 部分都是直接拷贝这里的。

```
<?php
// $Id: page.tpl.php,v 1.18 2008/01/24 09:42:53 goba Exp $
?><!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="<?php print $language->language ?>" lang="<?php print
    $language->language ?>" dir="<?php print $language->dir ?>">
  <head>
    <title><?php print $head_title ?></title>
    <?php print $head ?>
    <?php print $styles ?>
    <?php print $scripts ?>
    <!--[if lt IE 7]>
      <?php print phptemplate_get_ie_styles(); ?>
    <![endif]-->
  </head>
```

注意这里面的 php 语句, \$head_title, \$head, \$styles, \$scripts, 这些都是页面模板中的变量, 还有 \$language, 它现在成了一个对象变量了。具体的含义, 可参看本站的译文还有官方的手册。

紧接着的代码, 是网页最上面的一些 logo, 口号什么的,

```
<div id="header-region" class="clear-block"><?php print $header; ?></div>
```

```
<div id="wrapper">
```

```
<div id="container" class="clear-block">
```

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

```
<div id="logo-floater">
```

```
<?php
```

```
    // Prepare header
```

```
    $site_fields = array();
```

```
    if ($site_name) {
```

```
        $site_fields[] = check_plain($site_name);
```

```
    }
```

```
    if ($site_slogan) {
```

```
        $site_fields[] = check_plain($site_slogan);
```

```
    }
```

```
    $site_title = implode(' ', $site_fields);
```

```
    if ($site_fields) {
```

```
        $site_fields[0] = '<span>'. $site_fields[0] .'
```

```
    }
```

```

$site_html = implode(' ', $site_fields);

if ($logo || $site_title) {
    print '<h1><a href="'. check_url($front_page) .'

```

这些我们也不用管。

下面的这段是一级链接，和二级链接的输出。

```

<?php if (isset($primary_links)) : ?>
    <?php print theme('links', $primary_links, array('class' => 'links primary-links')) ?>
<?php endif; ?>
<?php if (isset($secondary_links)) : ?>
    <?php print theme('links', $secondary_links, array('class' => 'links secondary-links')) ?>
<?php endif; ?>

</div> <!-- /header -->

```

再下面是左栏区域：

```

<?php if ($left): ?>
    <div id="sidebar-left" class="sidebar">
        <?php if ($search_box): ?><div class="block block-theme"><?php print
$search_box ?></div><?php endif; ?>
        <?php print $left ?>
    </div>
<?php endif; ?>

```

如果左栏区域中没有区块的话，这里的 **html** 片段就不用输出了。

我们所关心的到了：

```

<div id="center"><div id="squeeze"><div class="right-corner"><div class="left-corner">
    <?php print $breadcrumb; ?>
    <?php if ($mission): print '<div id="mission">'. $mission .'

```

```

        <?php print $content ?>
    </div>
    <?php print $feed_icons ?>
    <div id="footer"><?php print $footer_message . $footer ?></div>
</div></div></div></div> <!-- /.left-corner, /.right-corner, /#squeeze, /#center -->

```

这里面放的主内容，包括节点的标题，各种字段，属性。即便是在别的主题中，通常我也是直接拷贝这一整段代码，通常需要略作修改，比如这里，我们需要把标题删去，也就是删除以下内容：

```
<?php if ($title): print '<h2'. ($tabs ? ' class="with-tabs" : ''). .'>'. $title .'</h2>'; endif; ?>
```

这样“drupal 国内新闻”这一标题就不出现在顶部了，当然现在啥也没有了。

如何去掉“Mon, 10/20/2008 - 16:59 — admin”这样的信息呢？有不少初学者都这样问。是不是删除以下内容：

```
<?php if ($show_messages && $messages): print $messages; endif; ?>
```

就可以了。

我试了一下，不行，看来文章的作者，及发布日期不在这里控制。你可以逐个的删除这里的变量，来观察页面内容的变化。

其中<?php print \$content ?>的\$content变量，是由节点模板控制的，也就是node.tpl.php,这个是最一般的节点模板，有时候我们需要更具体的节点模板，这个将在接下来的文章中讲到。

再下面是右栏：

```

<?php if ($right): ?>
    <div id="sidebar-right" class="sidebar">
        <?php if (!$left && $search_box): ?><div class="block block-theme"><?php print
    $search_box ?></div><?php endif; ?>
        <?php print $right ?>
    </div>
<?php endif; ?>

</div> <!-- /container -->
</div>
<!-- /layout -->

```

这个和左栏一样，不过就是放在了右边罢了，左右是对成的。

最后面的这个变量一定要注意，

```

<?php print $closure ?>
    </body>
</html>

```

\$closure是干什么的，我们先别管，忘了这个变量，会犯莫名其妙的错误的，有一次，我同事制作主题，出现了错误，怎么都找不到，搞了半天，我让他从头开始，一个一个对照，看到底哪里少东西了，最后发现，忘记了这个变量<?php print \$closure ?>。当你制作主题的时候，出了问题，也应该看看是不是缺了这个变量。这个是经验之谈。

你可以为特定节点类型的节点单独的创建一个节点模板文件,对于我们的首页节点类型,如果要单独的定制它的外观的话,就需要创建一个node-front_page.tpl.php.创建的过程就是拷贝node.tpl.php,然后将其重命名。

我们先来看看这个garland的node.tpl.php,也就是还没有被修改的node-front_page.tpl.php.注意不要把node-front_page.tpl.php写成了“node-front-page.tpl.php”,因为我在写本文的时候就犯了这个错误,有时候,这

样的一个很小的错误就会浪费你半天时间。我遇到过这样的情况。换成了后者，就不管用，因为系统内部没有 `front-page` 这种节点类型，只有 `front_page`。

让我们先分析一下这个模板文件，最前面的部分，

```
<div id="node-<?php print $node->nid; ?>" class="node<?php if ($sticky) { print ' sticky'; } ?><?php if (!$status) { print ' node-unpublished'; } ?>">
```

```
<?php print $picture ?>
```

```
<?php if ($page == 0): ?>
```

```
  <h2><a href="<?php print $node_url ?>" title="<?php print $title ?>"><?php print $title ?></a></h2>
```

```
<?php endif; ?>
```

第一行是输出 `div` 的 `id` 和 `class` 的，根据节点的类型，状态的不同，其 `id` 和 `class` 也不相同。其次是输出图片的，接下来看看是不是个整节点，如果不是的话，也就是仅仅是一个摘要（特 `aser`），那么在节点标题上加个链接。

再往下看：

```
<?php if ($submitted): ?>
```

```
  <span class="submitted"><?php print $submitted; ?></span>
```

```
<?php endif; ?>
```

这段就是输出节点的提交信息的，包括节点的作者，提交日期等等，也就是前文所提到的，尚未解决的问题。“Mon, 10/20/2008 - 16:59 — admin”，如何去掉这一信息呢？我们把上段代码从节点中删除，就可以了。

接下来就是节点的正文了，由 `CCK` 创建的内容，也在这里直接输出。

```
<div class="content clear-block">
```

```
  <?php print $content ?>
```

```
</div>
```

我们一会再看看，如果控制输出 `CCK` 定义的字段。

最后的这段代码，用来输出节点的分类，以及其它链接的：

```
<div class="clear-block">
```

```
  <div class="meta">
```

```
    <?php if ($taxonomy): ?>
```

```
      <div class="terms"><?php print $terms ?></div>
```

```
    <?php endif; ?>
```

```
  </div>
```

```
  <?php if ($links): ?>
```

```
    <div class="links"><?php print $links; ?></div>
```

```
  <?php endif; ?>
```

```
</div>
```

```
</div>
```

你在 `zhupou.cn` 的每篇文章下面，基本上都可以看到这样的分类，比如 `druupal6.x`, `druupal5.x` 等等。你可以根据需要将这些信息放到正文的上方，或者下方。

这里面用的变量，都是节点模板中常用的变量，具体含义可参看 <http://drupal.org> 上的手册，以及我以前翻译的一些文章。最后提示一个技巧，在这个节点模板中，善于使用 `<?php print_r($var) ?>`。

我们在前面提到，在节点模板中，模板使用 `$content` 变量将正文，以及由 CCK 创建的字段一股脑的输了出来，怎么才能将其单独的输出来，放到定制的 `html` 中呢？

在文章的最后面，我们提到了 `print_r()` 这个函数的应用，这个函数在制作模板文件中，用处可大了，又专门的支持主题制作的 `drupal` 模块，不过我没有用过，我觉得 `print_r` 已经够用了。让我们在节点模板中，插入 `<?php print_r($node) ?>`，访问首页，发现里面的内容全乱套了。

```
stdClass Object ( [nid] => 8 [type] => front_page [language] => [uid] => 1 [status] => 1 [created] => 1224493166 [changed] => 1224493166 [comment] => 2 [promote] => 0 [moderate] => 0 [sticky] => 0 [tnid] => 0 [translate] => 0 [vid] => 8 [revision_uid] => 1 [title] => Drupal 国内新闻 [body] =>.....
```

这么多东西，怎么办，我们要做的就是直接在这里找到对应的变量，怎么找，先找内容阿。先把那一大串复制到一个文本文件中，然后慢慢的找，最后找到了

```
<?php //print_r($node->title) ?>
<?php //print_r($node->body) ?>
<?php //print_r($node->field_news[0]['value']) ?>
<?php //print_r($node->field_news_body[0]['value']) ?>
<?php //print_r($node->field_security[0]['value']) ?>
<?php //print_r($node->field_security_body[0]['value']) ?>
```

怎么找出来的？慢慢的试出来的，最后发现 `$node->field_news[0]['value']` 可以，很快其它的也一样，搞定。找到了这些具体的字段，我们就可以将其放到我们的 `html` 片段中了。放置的过程中，发现不能使用 `$node->body`，因为同样输出所有的正文加上所有的 CCK 字段。这里应为使用 `<?php print $node->content['body']['#value'] ?>`，注意这里的 `value` 前面有个“#”号。我咋知道的？试了好几遍才试出来。最开始学习 `drupal` 的时候，都是这样试错试出来的。这样我们就可以将

```
<div class="content clear-block">
    <?php print $content ?>
</div>
```

替换为了

```
<div class="content clear-block">
    <div><h2><?php print $node->title ?></h2> <?php print $node->content['body']['#value'] ?> </div>
    <div><h2><?php print $node->field_news[0]['value'] ?></h2> <?php print
    $node->field_news_body[0]['value'] ?> </div>
    <div><h2><?php print $node->field_security[0]['value'] ?></h2> <?php print
    $node->field_security_body[0]['value'] ?> </div>
</div>
```

这样我们把 CCK 创建的节点类型就完整的输出出来了。最终首页节点模板的文件内容如下：

```
<?php
// $Id: node.tpl.php,v 1.5 2007/10/11 09:51:29 goba Exp $
?>
<?php //print_r($node) ?>
<?php //print_r($node->title) ?>
<?php //print_r($node->body) ?>
<?php //print_r($node->field_news[0]['value']) ?>
<?php //print_r($node->field_news_body[0]['value']) ?>
<?php //print_r($node->field_security[0]['value']) ?>
<?php //print_r($node->field_security_body[0]['value']) ?>
```

```

<div id="node-<?php print $node->nid; ?>" class="node<?php if ($sticky) { print ' sticky'; } ?><?php if
(! $status) { print ' node-unpublished'; } ?>">

<?php print $picture ?>
<?php if ($page == 0): ?>
    <h2><a href="<?php print $node_url ?>" title="<?php print $title ?>"><?php print $title ?></a></h2>
<?php endif; ?>

    <div class="content clear-block">
        <div><h2><?php print $node->title ?></h2> <?php print $node->content['body']['#value'] ?> </div>
        <div><h2><?php print $node->field_news[0]['value'] ?></h2> <?php print
        $node->field_news_body[0]['value'] ?> </div>
        <div><h2><?php print $node->field_security[0]['value'] ?></h2> <?php print
        $node->field_security_body[0]['value'] ?> </div>
        <?php //print $content ?>
    </div>

    <div class="clear-block">
        <div class="meta">
            <?php if ($taxonomy): ?>
                <div class="terms"><?php print $terms ?></div>
            <?php endif;?>
        </div>

        <?php if ($links): ?>
            <div class="links"><?php print $links; ?></div>
        <?php endif; ?>
    </div>

</div>

```

还差最后一个问题没有解决，那就是如何去掉“Add new comment”这一链接，其实这个很简单，那就是编辑该节点，在编辑页面的“comment settings”部分选中“Disabled”就可以了。好了最终的效果图如下：



你可能说，怎么还是这么丑，将 3 个 div 排成什么样的样式，是美工的活，CSS 的东西我不太懂，在这里就不多讲了。讲的就是这个流程，如何把 CCK 的字段提取出来，如何将 CCK 创建的节点设为首页。有点累了，歇会先。
××××××××END××××××××××××××××××××××××

ZEN theme 主题修改成果小结

- 1) 首先，你可以找一个 drupal 主题做为你的原始主题，原始主题越简单越好修改。你也可以自己创建一个，当然，逻辑还得按 drupal 来，这么一想就没有必要了，何况简单的比方说：css 的三栏自适应分布都是些固定的技术，你的新发明了也许并不一定比这些固定套路管用。所以，第一步，找个主题来修改吧，我用的 zen.觉得不错。
- 2) 你得规划好你网站主要栏目的架构，打个比方说，一个 blogs 的主页，你打算用什么样的布局呢，顶上有推荐，中间有图文，下面有内容的话，你可以试着用 33/34/33,40/40/20 的上层加上 zen 自带的 20/60/20 的底层，这样，你可能就需要三个区间。所以，先规划一下。我的网站我规划了 4 个主要的分栏。33/34/33,40/40/20,20/60/20,25/50/25,通过这三栏的组合，你可以得到 4*3 共 12 个分栏的样式。我想已经够用了吧。当然，考虑子菜单，我又加了一个 submenu 的 100%的分栏。这样，完全足够了。
- 3) 修改 INFO 文件，将你要加的这几个分栏加入内容 REGIN，可以自己考虑加些中文名比如“三列左”之类的助记。加入 REGIN 模块后，要记得修改 template 中相应的 css。
- 4) 修改 css 文件。主要是 layouts.css,copy 关于 20/60/20 分栏的内容，优化，改名，分配，这样，5 个分栏的 css 就完全了。当然，如果栏不存在什么的你需要考虑不要出现丑陋的空白，所以你的 layouts 文件对 margin,padinng 的考量一定要具体，或者你可以加上内容后再进行测试。
- 5) 修改 page.tpl.php 文件，将你的分栏加入你想要的页面。并不是每个页面都需要所有的分栏。所以，你自由组合吧，以 blogs 为例，你可以就需要 4 个分栏。把这 4 个分栏加入后，命名为 page-blog.tpl.php,你的任务就完成了。依次，你可以组建一些其它主栏的页面，如果你是以 page 的方式建立的这些内容，你可能需要命名的方式就是 page-node-111 这样的页面了。

6) 实际上, 每种类型的节点表现形式可能都不一样, 比如图片, 一般的就是图在上, 内容在下, 博客正好相反, 所以你得为你的不同的节点制造不同的布局, 所以你可能又会有一些新的东西, 比如 `node-forum.tpl.php,node-blog.tpl.php` 之类的文件。

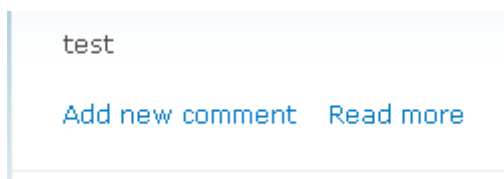
7)css 是算简单的技术, 只要你以前背历史的时候下了功夫练记性, 不过加上 IE6, IE7, firefox,maxon 这些自成标准的怪物, css 就什么都不是了。弄得 hack css 也成为了一门学问, 我就上周加这两天, 解决了 css 的一大堆问题, 比如 IE6 中的 border 经常消失, firefor 中的 margin 与 IE 中的不一样等等。反正怎么说呢, 可能 css 专家的意思就是说他失败的次数足够多了。

END

小技巧--使用 firebug 帮忙定制 theme

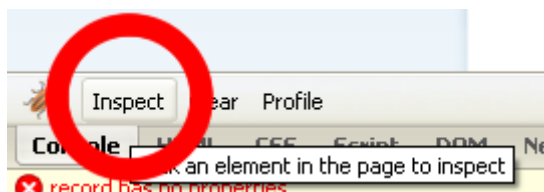
比如说我要在 "Read more" "Add new comment" 之前加个可爱的 icon

但因为"Read more" 和 "Add new comment" 畏使用不同的 icon (当然吧)
所以要看看从 css 上它们的 class 或 id 值有否特定
才可以用 css 定制不同的 icon
如不, 可能要 hack 核心或使用 js 来忙了

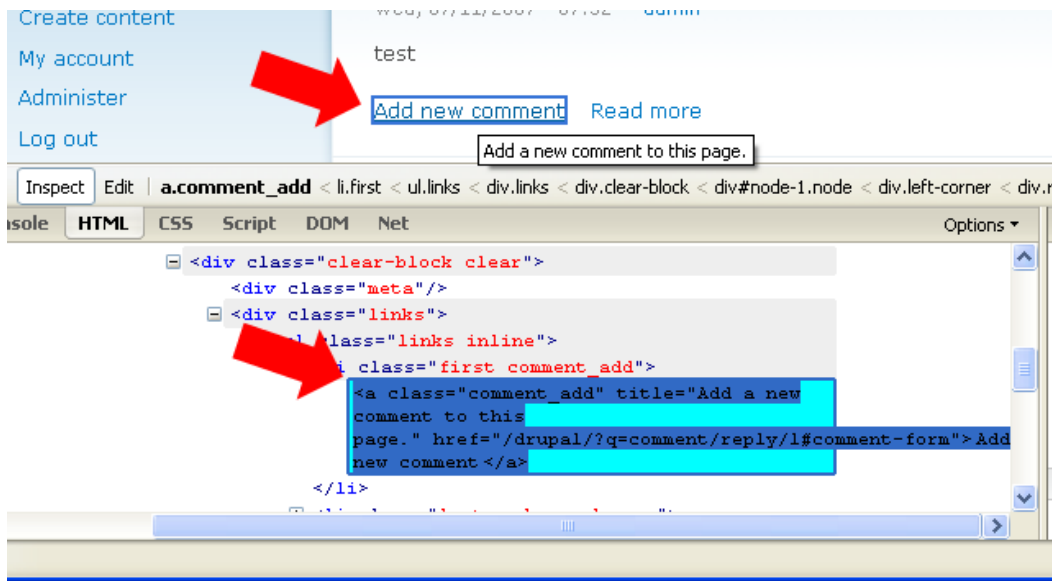


先打开 firebug (热键 F12)

按一下 inspect



指到 "Add new comment" 来查看它的 html



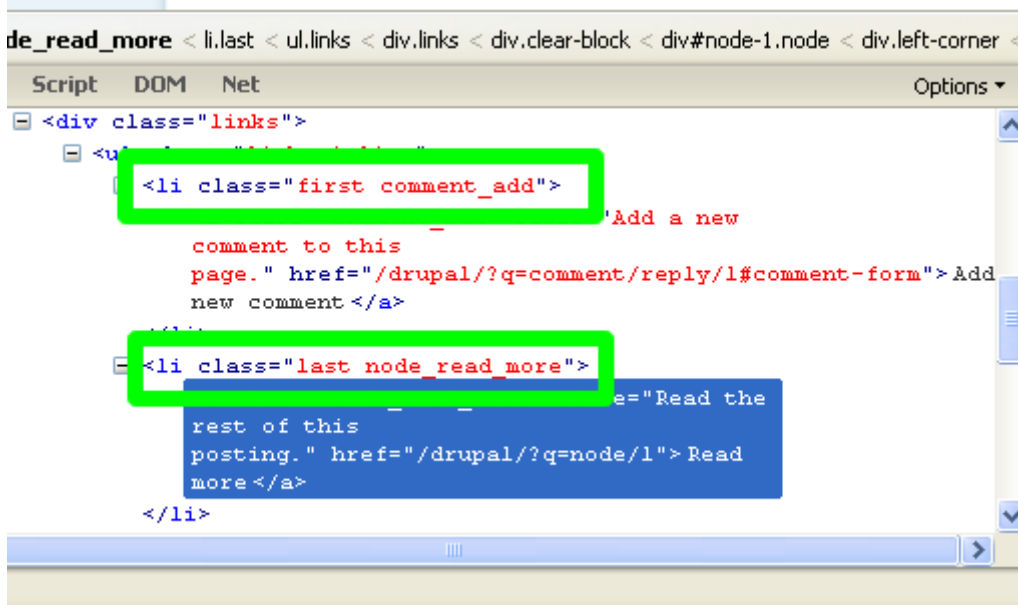
你可以看到 firebug 立即跳到 "Add new comment" 的 html 可以看到:

```
<li class="first comment_add">
  <a class="comment_add" title="Add a new comment to this page." .....
</li>
```

指到 "Read more" 可以看到:

```
<li class="last node_read_more">
  <a class="node_read_more" title="Read the rest of this posting." .....
</li>
```

可以看到它們各有一個自己的 class, 可以用 css 鎖定 tag 了



例:

```
.node_read_more{
  background-image:url(read_more.gif);
}
.comment_add{
```

```
bac.....  
}
```

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

Drupal 实现国内门户网站的风格

国内的门户网站大多有如下特点:

- * 首页往往会放上一个图片 Slide Show, 1, 2, 3, 4, 间隔的变换
- * 首页只列出文章的标题, 按日期排序, 不像 Drupal 默认的那样显示标题和摘要
- * 首页会划分为各个内容区块, 热点新闻, 行业新闻, 专题等等, 各管一摊

Drupal 默认安装后, 和以上描述的门户网站的风格差的有十万八千里, 很多人可能就是因此放弃对 Drupal 的继续研究, 认为洋人的东西不符合国人需求, 相反, DeDeCMS 之类的国产 CMS 默认安装就大致符合。

实际情况是, Drupal 做这样的门户站风格绰绰有余, 如下逐条说明:

- * Drupal 有大量 Image Slide Show 的模块, 随便找一个, 图片 Slide show 轻松搞定
- * 要实现文章列表显示标题的模式, 按如下步骤
 1. 下载 views 模块, 启用
 2. 进入后台, 进入 views, 新建一个 views
 3. 修改其中的 style, 然后可以到 fields 区块中点”+”, 选中 Node title, 更新后, 再指定: link to the node, 也就是说该标题指向文章的链接, 你也可以根据需要, 加上时间显示, 作者显示, 然后到 Sort 区块中指定一下排序方式, 一般按发布日期的倒序来排列.
- * 最后要实现内容区块, 可以下载 panels 模块, panels 可以根据你的需要把首页分割为各种区块模式, 三行三列, 两行两列, 随心所欲.

*****END*****

Views2.0 从入门到精通

在 Drupal6 中, 使用 Views2 模块, 还可以通过覆写 views 样式输出, 制作多种多样的列表样式, views 模块在多种场合显得非常方便, 比如, 显示一个分类下的 N 篇文章, 只输出文章标题, 时间。或者输出文章中上传的图片作为封面等等。

本系列的教程, 将着重和大家分享在 Drupal 6.x 中 Views 2 版本的基本应用, 利用基本的 views 操作, 逐渐深入 views 的使用, 满足自己的需求。

一、views 列表页面

进入 views 的列表页面, 发现 views2 有一个筛选的功能, 这也是为了我们在制作了大量的 views 列表以后, 方便我们快速定位自己要查找的 views 列表。如下图: (略)

我们分别来看一下, 各个选项的意思。

- (1) Storage: 我们可以理解为存储方式, 这是在 views2 中, 新增加的一个概念, 默认有三项, all, 即所有的方式, default, 即默认方式, 也就是在安装好 views 模块的时候, 自带的几个 views 列表, 第三项: overridden, 这是 views 子模式的意思, 在 views 中添加一个 page 以后, 还可以使用这个 views 添加一个 page 页面, 那么就是 overridden 了原来的 page, 就是这个 views 的子模式。
- (2) Type: 这个比较容易理解, 就是制作 views 的时候, 选择的类型 (View type), 是 Node, 还是 File, Node revision, Term, User. 默认就是这 5 种了。
- (3) Tag: 这个是在添加 views 的时候, 填写的 View tag 选项, 也是为了方便查找而设置的。
- (4) Display: 即制作 views 生成的样式, 默认有 block, page, feed, feed 样式是 Drupal6 中新增加的一个方式。
- (5) Sort by 和 Order 是用于排序的, 大家都明白。

在过滤框的下面, 是我们所创建的 views 和安装好 views 模块后默认的几个 views 列表样式, 可以针对 view 做一下操作。

点击“edit”：编辑当前的 views 列表，“Export”将这个 views 导出，“Clone”，复制这个 views，“Delete”，就是将这个 views 删除掉了。

如果你的 views 没有被启用，比如一些自带的 views，则点击“Enable”启用这个 views 列表即可，如下：

从下一节开始，我们介绍 views 的使用和添加，编辑。

二、-添加 views

首先，我们添加一个 views 列表，在 views 的列表上方，点击“add”，填写 views 列表的基本信息，如下图（略）

其中，必须要填写有两项：

（1）views name，即该 views 列表的名称，这里要填写机器可读的名称，只能是数字，字母，下划线命名，由于后期要使用该名称覆写 views 主题输出，我建议，名字简洁又明了最好。

（2）views type，即该 views 的类型，是 views 过滤内容的分类，分为 node，file，node revision 和 term，user 五种。还有一项是 views description，这是对这个 views 的描述，会在 views 列表页面显示，用来提示用户这个 views 中遍历的是什么内容的。

填写完以后，点击“Next”将进入 views 的具体配置页面，我们下一节将介绍基本配置。

三、基本设置

1、views2.0 的布局：进入 views 看到的会是如下界面：（图略）

这一个图是从 views 自带的帮助文档中找到的，它已经对 views-ui 的各个区域功能作了简单的介绍，我们来看一下；我们可以把这个 ui 界面分成 4 个部分。

（1）最上面的一排，是用来对 views 进行一些导出，复制，查看操作的，

Export 将 views 导出

Clone 复制这个 views，如果你想建造基本上类似该 views 列表的 views，可以使用这个功能复制出来一个。

View "Page" 查看这个 views 列表在前台的显示效果。

（2）左边的一块区域，用来添加 views 的显示方式，这个显示方式我们在上一节当中已经提高过，可以利用一个 views 生成多个 views 列表，可以是 block，page，feed 格式的。

（3）右边的最大区域，是用来设置该 views，几乎全部的 views 设置都在这里了。

（4）最下面的一部分，有两个功能，一个是预览当前的 views 输出，这个只能预览输出的内容，格式排版有可能会和在前台的有些差别[主要是 css 可能后台没有加载]，另一个是显示 views 查询的性能和 sql 语句，方便调试程序用的，基本上要是没有编程基础的话，用处不是很大。

2、基本设置部分介绍

基本设置部分包括 View settings 和 Basic settings 两个部分，在这个全新的 views 界面的右侧大区域，用区块隔开了，这样的界面也是非常友好的。我们分别来看一下，这两个部分的内容。

（1）View settings

设置 Tag 的选项，这里在添加 views 的时候，你就可以输入，如果要作出改变，就可以通过这个选项，主要是方便在 views 列表页面进行筛选。默认是 None。

（2）Basic settings

这就是正规的“基本设置”了，主要是下面几项

Name: 默认是 Defaults，就是这个 views 列表的名称；

Title:默认是 None，这个 views 的标题；

Change settings for this styleStyle: views 列表的显示方式，有四种：

Grid: 这样方式，有点类似于 Excle 表格，可以设置有多少行多少列，做出以下的效果：

List, 就是列表的样式，可以选择是 ul 还是 ol 格式

Table 表格样式

Unformatted 没有样式，一行一行的显示出来，用 div 包着。

Change settings for this styleRow style: views 列表中每一行的显示方式，有两种方式，

Fields, 默认方式，设置是否在一行显示，或者使用分隔符等。

Node，以节点的方式显示，可以选择显示标题，摘要，链接等。

Use AJAX: 是否适用 ajax 效果，主要是用在分页的时候，默认为 No

Use pager: 是否分页显示，分页有三种 Full pager Mini pager 和 No，默认为 No

Items to display: views 列表显示的条数，默认为 5；同时可以设置步长。

More link: 是否用 more 链接到页面，估计这个功能还没有完善好，这个功能我到现在也没发现如何使用，默认是 No

Distinct: 是否去掉重复的选项，默认为 No

Access: 查看这个 views 是否需要权限，可以根据用户组设置访问权限，默认是 Unrestricted

Exposed form in block: 默认是 No，这个选择以后，会在 block 或者 page 上方有一个过滤器，提供一些过滤得选项 用来快速定位节点。

Header: 设置 views 头部信息，默认为 None

Footer: 设置 views 底部信息，默认为 None

Empty text: 设置如果 views 查询没有结果会显示的内容，默认为 None

Theme: Information 这个就是大名鼎鼎的主题覆写功能，我们会在后面详细介绍。

四、选择 Fields

这一节我们介绍一下 Fields 选项的作用，Fields 用来添加 views 将输出哪一些内容，一般这是必须选择的，要不然就没有意义了，进入 views 中，如下图界面；（图略）

点击“+”号，就会出现下方的选择列表，上面有一个过滤器，方便用户分组显示要添加拿一些内容，比较常见的基本上都包含了，比如节点的名称，发布时间，评论次数，文章所在分类等。

比如我们选择了“node:title”项目，会出现下面的样子：

（1）Label，即列表显示的时候，着一个标题前面的标签，如果有文字写上的话，比如上图，则会显示成 Title : XXXX 的样子，我们一般将它清空即可。

（2）Exclude from display：如果选择了该项目，则这个标题则不会显示出来；一般不选择。

（3）Link this field to its node: 这一项是选择标题是否关联到文章的内容页面，一般会选择，即点击标题，会显示文章。在 Fields “+”号的右边，有一个调节按钮，点击以后，下面出现如图：

用鼠标拖动即可调节显示的顺序，这也是 Drupal6 中 views2 的一个新特性，所有的操作都是 Ajax 的，用户体验非常好。

如果选择其他的项目，操作基本上是类似的，遇到复杂的操作，我们后面将会继续介绍。

五、views 过滤器

说的直白一些，views 就是生成 sql 语句的一个工具，viewsd 中，Fields 就像是 sql 语句中 select 后面的东西，而 views 的过滤器，就像是 sql 语句中的 where 后面东西，也就是说，过滤器的作用就是制定查询的条件。

在 views Ui 界面中，如下图，就是设置过滤器的地方：

添加的方式，和 Fields 的添加类似，点击“+”号，会出现选择列表，如图：

这里就是选择 sql 语句中 where “字段名”= “***” 中的“字段名”，比如，我们选择了

“node: publish”，就是选择过滤是否发布得文章，如下图：

如果勾选了这个选项，即 publish=true，只过滤显示已经发布的文章，

如果选择了 term 以后，在下放的设置区域右边，有一个 Exposed，点击如下图：

这是设置是否有一个提供用户快捷过滤得选择，如果设置了 Unlock operator，则会在 views 列表的上方，会提供给用户一个类似于文章管理界面上方的一个过滤器；右侧和下侧则是对过滤器的一个设置，比较容易理解，这只是某些选项会有。

六、预览创建的 views 列表内容

在设置 views 的页面下方，有一个区域就是专门来显示预览内容的，如图：

Live preview 区块中分为两个部分，上面的区域可以用来填写一些参数，比如设置了 arguments，则填写相应的参数，而 display 是选择，预览那一个 views 列表，因为一个 views 可以创建多个列表，所以要在这里进行选择。

选择好 display 和 arguments 以后，点击右面的“preview”，则下面就会预览出内容了，这个功能还是非常适用的，在这里调整好大体的样式，然后可以通过覆写 views 模板文件来制作自己要的显示效果，比如，Drupal 使用者中文社区

[<http://drupaluser.cn>]的 views 列表就是这样做的。

=====END=====

CCK & Views 模块使用小结

CCK 和 Views 为什么如此重要呢？因为，CCK 和 Views 在从“内容”和“提取”两个方面给 Drupal 提供了更高的灵活性与适用性。在谈 CCK 和 Views 的具体使用之前，有必要了解其工作原理，这进一步需要我们理解 Drupal 的部分工作原理。

关于内容组织、提取与呈现的基本思路

组织

Drupal 最为特别也最重要的一个特性是，将所有内容（或者具体些叫文章）看作节点（Node），这些节点具有公共的和私有的特性。每个节点都有 Title，都有 Body，这些是相对显性的，它们就是这些内容我们能看到的部分；每个节点都是有人创建的（Author），创建都是有时间记录（Post date）的，等等，这些是节点的默认相对隐性部分。结合其他的功能模块，每个节点可以控制受这些功能模块影响的程度和方式，比如，是否接受用户评论。

对这些节点的特性进行单独的一个个设置是不合理的，因为可能我们需要面对成千上万的节点（内容）。那么对相似特性的节点进行分组分类，成为一个有效用的方式——这实际上是一个反推——Drupal 为我们提供的内容类型（Content types）正是这样一种分组设置 Node 特性的途径。

提取

作为最基本的使用，印象中我们进行提取的操作几乎没有——因为通过内容类型的设置，如何从内容记录（数据库）中调用数据已经被确定了。比如，在内容类型中我们可以设置 Body 为空，因而使内容编辑、显示时没有 Body 正文部分。

但是更进一步的，如果需要一些默认 Drupal 没有直接提供的内容形式时，就需要用到提取了。在 DrupalChina.org 中 east 有篇文章《不使用 views，如何自定义节点列表显示页面》，详细的介绍了“提取”数据的“手工”方式。

呈现

这个无需多说，主题（Theme）实现了 Drupal 内容的呈现方式。

CCK 功用

准备

前面提到，每个节点都拥有共性。在新安装 Drupal 后，默认提供了两种类型的内容，Page 和 Story。可以通过路径 admin/content/types/查看他们。Page 和 Story 提供了哪些可编辑内容？标题 Title，正文 Body。

假如，我想发布一篇关于某本本电脑的文章，希望包含电脑本本图片、本本配置表、本本特性清单，还有本本的价格、订购地址以及反馈邮件，甚至销售本本商家的地址，我该如何处理呢？

最简单的办法，装个 FCKeditor 一类的可视化编辑器，像在 Word 中一样，进行上述内容的录入、排版、保存、查看。有没有问题？当然没问题，但是也有很多问题（矛盾啊）。

在我们公司，很难找到一位可以保持前后同类 word 文档格式一致的同事。Word 编辑有太大的随意性，无法控制保证上述各项内容录入的有效性是否完整录入，也无法控制设置格式的一致性——可能本本 A 的报价单使用表格样式，本本 B 则使用“价格：¥XXX”的文本形式。使用 FCKeditor 的问题与此类似，更重要的是，因为 FCKeditor 编辑的文章内容是一个整体，想要细一些的控制——比如只允许注册用户查看本本商家的地址和反馈邮件——是无法办到的。

CCK 提供了另一种“复杂”的办法，你可以在“标题 Title”、“正文 Body”之外，为各内容类型提供各种字段（Field）级别的新内容条目，结合其他模块，可以进一步对这些字段进行控制，比如显示方式和访问方式（权限）。

扯的有些远，下面说说如何通过 CCK 设置一个本本类型的内容。开始之前请安装 CCK 模块以及部分 CCK 扩展模块，清单如下（其中加粗标注的部分模块是 CCK 自带的，它们应位于 sites/all/modules/cck/modules。关于扩展模块的组织方式请查阅其他文章）：

- cck
- **content_copy**
- **content_permissions**
- **fieldgroup**
- **number**
- **optionwidgets**
- **text**

- **userreference**
- **imagefield**
- **link**
- **email**
- **blockreference**
- **cck_blocks**
- **cck_fieldgroup_tabs**
- **cck_formatters**
- **commentreference**

CCK 模块类型

前面罗列了一些常用（实际是在 6.X 下可用 CCK 模块还不多）的 CCK 模块，有必要说一下这些模块的类型。总的来说，CCK 模块分为两类：字段类型类（链接、邮件地址等原生字段类型，xxx_reference 等引用字段类型）和辅助类（控件类型和显示模式等）。字段类型模块就是用来提供对应类型的字段（汗~~），如果稍有了解数据库的就能明白。

CCK 及其扩展模块提供原生字段如 **number** 数字、**text** 文本、**link** 链接、**email** 邮件地址、**imagefield** 图像等类型的，这些字段类型都按相应类型的保存实际数据到数据库中；

而引用类型的字段（大都包含 **reference** 字样）则可创建其他内容的引用，数据库中只建立一个映射。比如使用 **userreference**，可以创建一个对用户的引用，把它作为购买本本的联系人；

辅助类模块则用来提供更丰富易用的控件类型，比如 **optionwidgets** 为文本提供了下拉选择、文本框、自动文本框等等空间模式，用于构建更适合的录入、显示模式；

显示模式的模块，比如 **cck_fieldgroup_tabs**，可以为字段组（后面看图详细介绍）提供标签页的显示外观。**cck_formatters**，则可为数字指定小数位数的模式，很适合用在金额显示的场合中。**cck_blocks** 可以将 CCK 字段提供为一个块，可灵活的进行其他应用（区块应用我了解有限，可以查看一下其他文章）。

创建各种类型的 CCK 字段

新建一个内容类型

前面的废话中，大致说明了 CCK 存在的价值，也说明了 CCK 是必须结合内容类型（Content types）来使用的。因为我们要以本本为例，所以新建一个叫做本本的内容类型。这部分比较简单，大致罗列一下步骤即可（暂时使用英文版本）：

- 1、进入 `admin/content/types` 下，点击 **» Add a new content type** 新增一个内容类型；
- 2、进入 `admin/content/types/add` 下，需要输入的信息为：**Name:** 本本；**Type:** *benben*；**Title field label:** 名称；**Body field label:** 简介，其他暂时不管；
- 3、点击保存内容类型按钮，保存我们创建的本本内容类型；

保存完毕后我们位于 `admin/content/types` 下，如图所示：

Content types List Add content type Fields Export Import

The content type 本本 has been added.

Below is a list of all the content types on your site. All posts that exist on your site are instances of one of these content types.

Name	Type	Description	Operations
本本	benben		edit add field manage fields delete

» Add a new content type

操作（Operations）下面提供了几个可操作选项，其中增加字段（add field）与管理字段（manage fields）与我们的 CCK 直接相关，先增加字段。点击进入。

计划一下需要增加的字段及类型

兴冲冲的进来了，却不知道要填写什么...重新计划一下我们要增加的信息（字段）类型：

- 本本型号，文字类型的，比如我的 Acer TravelMate 3280，还可以更细；
- 本本外观，图片类型的，把本本的外观照贴上去；
- 本本产地，文本类型的，本本可能是台湾产的；
- 本本单价，数字类型的，比如 5000.00 元每台（我不能提供这个价格的 TM3280...）；
- 本本特性，文本类型的，比如什么蓝牙什么无线一类的，比较长的一些列表；
- 本本参数，文本类型的，一项一项的参数名称和参数数据；
- 本本评价，评论引用类型的，用户对某个本本的评价，可以加进来（抱歉，我还没有测试是先加还是后加，或者是自动引用，后面确认...）
- 生产厂商网址，网址类型的，提供生产厂商的网站地址；
- 意见反馈邮箱，邮箱类型的，提供一个邮箱地址可以让用户进行反馈方面的邮件沟通。为了避免垃圾邮件的骚扰，我们控制一下这个信息的访问权限——只有注册用户才能看见，当然你可以进行一些其他的权限设置；

当然还可以列更多，因为仅仅是个例子，也因为其他的问题，自己可以先去试试。

添加文本字段

我们已经在 `admin/content/node-type/benben/add_field` 页面下，如图所示录入下列信息：

本本 Edit Manage fields Display fields Add field Add group

Create new field

Field name: *
field_
The machine-readable name of the field. This name cannot be changed later! The name will be prefixed with 'field_' and can include lowercase unaccented letters, numbers, and underscores. The length of the name, including the prefix, is limited to no more than 32 characters.

Label: *

A human-readable name to be used as the label for this field in the 本本 content type.

Field type: *

The type of data you would like to store in the database with this field.

Continue

字段名称 (Field name) 是用于机器识读的名称，因为作为字段类型保存，一旦设定后就不能更改，并且只接受小写英文字符、数字和下划线。所有字段名称以 `field_` 开头，总的名称字符数不得大于 32 (含 `field_` 部分)；

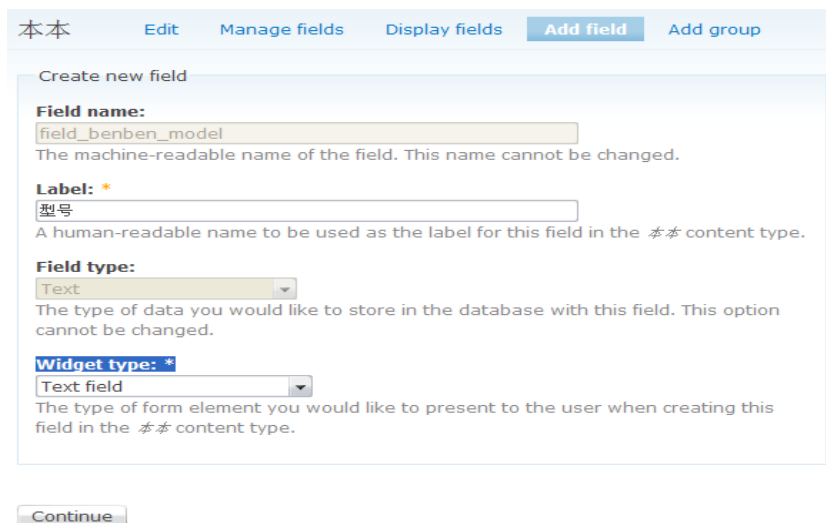
字段标签 (Label) 是显示在编辑、浏览时用户友好的名称，希望最终的页面中这一项是什么样子，可以在这里指定；

字段类型 (Field type) 是存储数据到数据库中的数据类型，说起来有些拗口。简单的说，记事本只能打字，小画家只能画图，我们要根据需要显示的内容选择合适的数据类型。前面已经罗列，“型号”适合使用文本方式保存；

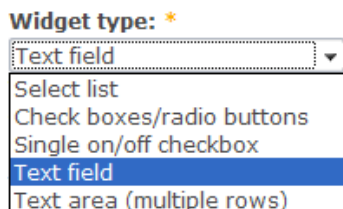
点击继续按钮进入下一步，关于字段类型 (Field type) 的其他选项后面陆续使用介绍。

添加文本字段

在这个页面中，我们需要设置该字段在创建时的控件类型 (Widget type)，如图所示，在这里还有机会更改字段标签：



安装上列 CCK 模块，我们使用文本类型时拥有如下类型的控件，我们选择单行文本框 (Text Field) 控件：



点击继续按钮到下一页 `admin/content/node-type/benben/fields/field_benben_model`，我们会在这里完成 `field_benben_model` 字段的全部设置。

这是字段“`field_benben_model`”的编辑页面，总共包含如下几个部分：

- **字段基本信息组 (basic information)**：这是我们前一步已经编辑完成的信息，包括
- 字段标签 (Label)
- 控件类型 (Widget type)
- 更改基本信息按钮 (Change basic information)：如果需要更改，点击这个按钮即可。
- **字段设置组 (settings)**：字段设置组仅影响用于本本这个内容类型的 `field_benben_model` 字段，包含如下选项：
 - 文本框大小 (Size of textfield)
 - 默认值 (Default value)
 - 显示分组 (Display in group)
 - 帮助文本 (Help text)
- **全局设置组 (Global settings)**：全局设置组会影响所有引用 `field_benben_model` 字段的内容类型，包含如下选项：
 - 必须 (Required)

- 值的数量 (Number of values)
- 文本处理方式 (Text processing)
- 最大长度 (Maximum length)
- 允许的值 (Allowed values)

说一下后两组的设置。**字段设置组 (settings)**

Field settings

These settings apply only to the 姓名 field as it appears in the 用户 content type.

Size of textfield: *

60

▼ Default value

姓名:

PHP code

Display in group:

<none>

Select a group, in which the field will be displayed on the editing form.

Help text:

Instructions to present to the user below this field on the editing form.

Allowed HTML tags: <a> <big> <code> <i> <ins> <pre> <q> <small> <sub> <sup> <tt> <p>

文本框大小 (Size of textfield): 根据我们编辑界面显示的需要, 可以为文本框指定宽度 (以字符数计算), 这个值不会影响最后网页显示的效果;

默认值 (Default value): 可以为字段指定默认值, 我们两种方式可以选择, 直接录入和通过 php 代码获得, php 代码具有更灵活强大的特性。需要注意的是, 如果在后面的允许的值 (Allowed values) 中进行了设置, 那么这里的默认值也只能在其中进行选择;

显示分组 (Display in group): 显示分组为多个字段在编辑、显示时提供了一种整理的手段。实质上是分组为组内的字段提供了一个 DIV 外框, 并为 DIV 预置了一些显示形式, 比如字段集、可折叠字段集、折叠的字段集或者标签页等。当然也可以通过 CSS 为这些 DIV 自行定义格式。现在还没有创建字段组, 我们后面会进行具体操作;

帮助文本 (Help text): 需要按照什么要求、什么方式录入字段内容, 可以在这里进行指定。帮助文本只出现在创建、编辑时的界面中, 不会出现在最终显示的页面内;

全局设置组 (Global settings)

Global settings

These settings apply to the 姓名 field in every content type in which it appears.

Required

Number of values:

1

Maximum number of values users can enter for this field. 'Unlimited' will provide an 'Add more' button so the users can add as many values as they like.

Warning! Changing this setting after data has been created could result in the loss of data!

Text processing:

Plain text

Filtered text (user selects input format)

Maximum length:

The maximum length of the field in characters. Leave blank for an unlimited size.

▼ Allowed values

Allowed values list:

The possible values this field can contain. Enter one value per line, in the format key|label. The key is the value that will be stored in the database, and it must match the field storage type (text). The label is optional, and the key will be used as the label if no label is specified.

Allowed HTML tags: <a> <big> <code> <i> <ins> <pre> <q> <small> <sub> <sup> <tt> <p>

PHP code

必须 (Required): 前面我们有提到一种假设——我们希望录入时不漏掉某些重要信息——这里就是实现这种假设的途径。通过选中必须选项, 操作员若提交时忘记录入该内容, 则会收到一个错误提示并要求纠正。对于本来说, 型号是必须的; **值的数量 (Number of values)**: 一个具体的本, 应该只有一个型号, 因此我们选择 1。但如果是本的靓照, 那可能会有很多, 比如 6 个角度的, 那可以选择 6。再或者, 可能不确定已有照片的数量, 由操作员自行决定, 那么选择无限; **文本处理方式 (Text processing)**: 文本类型的字段可以有两种方式来处理, 纯文本 (Plain text) 和可格式化的过滤文本 (Filtered text (user selects input format))。对于型号这类信息来说, 纯文本更适合。如果要想有段引用用户评价的内容, 那么使用 Filtered text 更为适合;

最大长度 (Maximum length): 本型号的字符数可能就 18 位, 我们不能允许 24 位的型号存在, 那么使用最大长度进行限制, 设置为 18, 超过该字符数的内容在提交时会收到错误提示。可惜没有最小长度的选项, 因为我们也不希望 2 位的错误型号存在, 暂行的替代方式是, 在帮助文本 (Help text) 中进行提示;

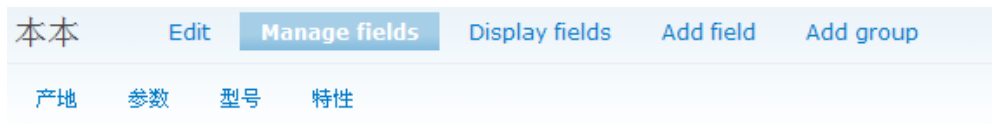
允许的值 (Allowed values): 如果要对输入内容进行限制, 比如只允许输入今年新发布 15 款本的型号, 那么可以在这个清单中录入。操作员如果录入了去年的型号或者对手的型号, 那么将收到错误提示并要求纠正。允许的值同样也提供 php 代码生成方式。

需要提醒的是, 同时提供普通方式和 php 生成方式取值的输入区域, 一旦使用了 php 代码, 那么它将覆盖普通方式的设置。关于这两种方式录入的细节要求, 可以阅读录入框下面的提示信息。

在上述步骤中, 我们创建了第一个字段 field_benben_model。点击了保存按钮我们会回到管理字段 (Manage fields) 页面下, 我们现在可以继续添加其他预计的文本类型字段。

- 本产地, 文本类型的;
- 本特性, 文本类型的;
- 本参数, 文本类型的;

文本类型的产地、特性、参数我们按照与型号基本相同的方式创建。稍有不同的是“特性、参数”, 我选择了多行文本框 (Text area(multiple rows)) 的控件形式 (参阅第一部分中的截图) 和“格式化”的文本处理方式——我想借用可视化编辑工具直接进行格式的设置。现在的管理字段界面看起来是这个样子。



Control the order of fields in the input form.

To change the order of a field, grab a drag-and-drop handle under the Label column and drag the field to a new location in the list. (Grab a handle by clicking and holding the mouse while hovering over a handle icon.) Remember that your changes will not be saved until you click the Save button at the bottom of the page.

Label	Name	Type	Operations
名称	Node module form.		
简介	Node module form.		
型号	field_benben_model	Text	Configure Remove
产地	field_benben_place	Text	Configure Remove
特性	field_benben_feature	Text	Configure Remove
参数	field_benben_parameter	Text	Configure Remove
Menu settings	Menu module form.		

现在我们来创建存储本外观的字段, 如前设定, 是采用图片类型的。在 CCK 扩展模块中, 6.x 版本下有两个可选途径: imagefield、imceimage。这两个模块各具特点, imagefield 具备丰富直观的设置界面, 而 imceimage 结合 imce 模块, 具备更强大的文件管理功能, 这个根据需要来选用。

供选择参考的更细节一些的差别是：
imagefield 特点是：

- imagefield 可以指定上传图片最小分辨率、最大分辨率，可以指定单个上传图片文件最大文件大小、每个文章中总计最大的上传文件大小，还可以预先设置 ALT 文本和 Title 文本；
- 结合 token 模块在设置字段时可以使用通配标记（token）为路径、ALT 文本和 Title 文本指定预置设置。这些设置对于操作员编辑时是不可更改的，因而可以更好的保证上传的图像文件按照预定的方式存储、标记；
- 但随之的问题是，每个上传的 imagefield 都是独立的记录，假如在多个文章中需要用到同一个图片，就需要多次上传，这是一个缺点。

imceimage 特点：

- imceimage 特性不多，大部分功能如上传、调整大小、生成缩略图等主要由 IMCE 来完成，唯一的，对应 imagefield 的缺点，imceimage 可以在配置的文件库中重复使用已有的图片，而不同重新上传。

对于这两个模块的选择，我的个人建议是，需要固定出现的图片、无需重复利用的图片采用 imagefield，比如本本的尺寸标记图；而可能需要重复利用的图片，使用 imceimage，比如本本制造商的 LOGO。

创建 imagefield 图像字段

大部分设置方式和文本类型的一致，没有特殊的地方抓图说明就是了，先是创建字段的第一步，如图：

本本 Edit Manage fields Display fields Add field Add group

Create new field

Field name: *
field_benben_images
The machine-readable name of the field. This name cannot be changed later! The name will be prefixed with 'field_' and can include lowercase unaccented letters, numbers, and underscores. The length of the name, including the prefix, is limited to no more than 32 characters.

Label: *
外观
A human-readable name to be used as the label for this field in the ~~##~~ content type.

Field type: *
Image
The type of data you would like to store in the database with this field.

-

指定控件类型，imagefield 只有 image 控件供选用。

本本 Edit Manage fields Display fields Add field Add group

Create new field

Field name:
field_benben_images
The machine-readable name of the field. This name cannot be changed.

Label: *
外观
A human-readable name to be used as the label for this field in the ~~##~~ content type.

Field type:
Image
The type of data you would like to store in the database with this field. This option cannot be changed.

Widget type: *
Image
The type of form element you would like to present to the user when creating this field in the ~~##~~ content type.

-

点击继续进入设置界面。图像类型的字段设置同样也是三组：基本信息、当前内容类型设置以及全局设置（关于当前内容类型设置和字段设置后面会有描述）。

当前内容类型设置各选项如下：

- 图像最小分辨率（Minimum resolution for Images）
- 图像最大分辨率（Maximum resolution for Images）
- 允许上传文件扩展名（Permitted upload file extensions）
- 显示分组（Display in group）
- 文件路径（File path）
- 每个文件最大上传大小（Maximum upload size per file）
- 每个节点最大上传大小（Maximum upload size per node）
- 帮助文本（Help text）
- 启用自定义 ALT 文本（Enable custom alternate text）
- 默认 ALT 文本（Default ALT text）
- 启用自定义 Title 文本（Enable custom title text）
- 默认 Title 文本（Default Title text）

其中显示分组和帮助文本域文本类型的一样，属于通用属性，不再描述。先看看下图中这些部分：

settings

These settings apply only to the 外域 field as it appears in the ## content type.

Minimum resolution for Images:
480
The minimum allowed image size expressed as WIDTHxHEIGHT (e.g. 640x480). Set to 0 for no restriction. If an image that is smaller than these dimensions is uploaded it will be rejected.

Maximum resolution for Images:
640
The maximum allowed image size expressed as WIDTHxHEIGHT (e.g. 640x480). Set to 0 for no restriction. If a larger image is uploaded, it will be resized to reflect the given width and height.

Permitted upload file extensions.:
jpg jpeg png gif
Extensions a user can upload to this field. Separate extensions with a space and do not include the leading dot.

图像最小分辨率、最大分辨率以 WIDTHxHEIGHT (例如 640x480)指定了允许上传、显示的图像尺寸，当小于指定尺寸时拒绝上传，大于指定尺寸时则自动调整尺寸至允许值。都设置为 0 则不做限制；允许上传文件扩展名指定了允许的文件类型，均以空格间隔的扩展名作为标记。

Path settings

File path:
benben\[uid\]\[site-date\
Optional subdirectory within the "files" directory where files will be stored. Do not include trailing slash.

Token	Replacement value
User tokens	
[user]	User's name
[user-raw]	User's unfiltered name. WARNING - raw user input.
[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.
[account-edit]	The URL the user's account editing page.
Global tokens	
[user-name]	The name of the currently logged in user.
[user-id]	The user ID of the currently logged in user.
[user-mail]	The email address of the currently logged in user.

文件路径确定了图片上传后保存于"files" 的相对位置，您可以直接输入路径指定，也可以采用下面提供的标记（Token）自动生成，关于标记（Token）的介绍，可以直接查看其说明信息。我的设置为：benben\[uid]\[site-date]，上传的图片将保存在"files" 文件夹下 benben\用户 ID\当前日期命名的文件夹内。

File size restrictions

Limits for the size of files that a user can upload. Note that these settings only apply to newly uploaded files, whereas existing files are not affected.

Maximum upload size per file:

Specify the size limit that applies to each file separately. Enter a value like "512" (bytes), "80K" (kilobytes) or "50M" (megabytes) in order to restrict the allowed file size. If you leave this empty the file sizes will be limited only by PHP's maximum post and file upload sizes.

Maximum upload size per node:

Specify the total size limit for all files in field on a given node. Enter a value like "512" (bytes), "80K" (kilobytes) or "50M" (megabytes) in order to restrict the total size of a node. Leave this empty if there should be no size restriction.

每个文件最大上传大小和每个节点最大上传大小，可以对上传文件大小进行限制，如果留空最大大小是由 php.ini 设置最大上传限制确定的。我这里不做设置。

ALT text settings

Enable custom alternate text
Enable user input alternate text for images.

Default ALT text:

This value will be used for alternate text by default.

Token	Replacement value
Global tokens	
[user-name]	The name of the currently logged in user.
[user-id]	The user ID of the currently logged in user.
[user-mail]	The email address of the currently logged in user.
[site-url]	The url of the current Drupal website.
[site-name]	The name of the current Drupal website.
[site-slogan]	The slogan of the current Drupal website.
[site-mission]	The mission of the current Drupal website.
[site-mail]	The contact email address for the current Drupal website.
[site-date]	The current date on the site's server.

当鼠标在图像上悬停片刻，会出现设置的 ALT 文本。我们在这里可以用两种方式来控制 ALT 文本的设置。启用自定义 ALT 文本，操作员创建图像时可以自定指定，默认 ALT 文本则使用标记（tokens）进行设置，用法同上。

Title text settings

Enable custom title text
Enable user input title text for images.

Default Title text:

This value will be used as the image title by default.

Token	Replacement value
Global tokens	
[user-name]	The name of the currently logged in user.
[user-id]	The user ID of the currently logged in user.
[user-mail]	The email address of the currently logged in user.
[site-url]	The url of the current Drupal website.
[site-name]	The name of the current Drupal website.
[site-slogan]	The slogan of the current Drupal website.
[site-mission]	The mission of the current Drupal website.
[site-mail]	The contact email address for the current Drupal website.
[site-date]	The current date on the site's server.

Ttile 文本设置方式与 ALT 文本类似，不再赘述。

全局设置各选项如下：

- 必须 (Required)
- 值数量 (Number of values)
- 默认列表值 (Default list value)
- 处理列表值的方式 (How should the list value be handled)

Global settings

These settings apply to the *标题* field in every content type in which it appears.

Required

Number of values:

1

Maximum number of values users can enter for this field. 'Unlimited' will provide an 'Add more' button so the users can add as many values as they like.

Warning! Changing this setting after data has been created could result in the loss of data!

Default list value: *

Listed

Hidden

The list option determines whether files are visible on node views. This will be used as the default value for the list option.

How should the list value be handled?: *

User Configurable. (Users will be able to set the list value per file.)

Enforce Default. (The default list value will be used for all files, and the list checkbox will not be displayed to users.)

必须、值数量，略过不表；

默认列表值，Drupal 为上传的图像提供了列表样式，即把图片附在文章末。这里你可以确定在这里是否使用该功能。选择罗列则出现在文末列表中，否则可选择隐藏；

处理列表值的方式，可以让用户为每个文件自行配置 (User Configurable)，也可以强制按照默认列表值来处理 (Enforce Default)。

增加图像类字段的方式说完了，还有四个类型的需要处理，先休息一下。

- 本本单价，数字类型的；
- 本本评价，评论引用类型的；
- 生产厂商网址，网址类型的；
- 意见反馈邮箱，邮箱类型的；

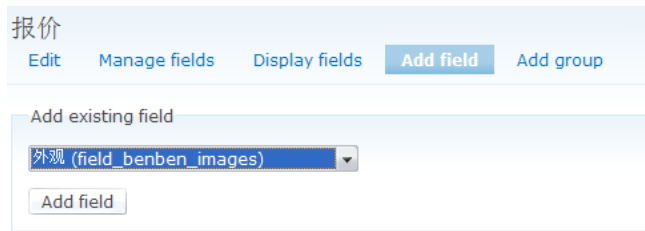
创建数字、引用、网址、邮箱类型的字段

通过文本、图像类型字段的设置，我们可以看到一些 CCK 字段类型的共同点：

- **1、**每个类型的字段都包含一组基本信息，即机器识读的“字段名称-Field Name”，用户识读的“标签-Label”，存储数据的“字段类型-Field type”以及处理数据录入的“控件类型-Widget type”。除标签、控件类型可以在创建后再次更改外，另外两项一旦设定就不能再变更，因此创建之前必须做好预期的设定。我个人的命名习惯是，通过下划线间隔，依次为内容类型、字段名、字段名分支。这会有一点例外，就是重复应用到多个内容类型的字段，不应采用这种命名方式；
- **2、**在针对具体内容类型私有设置中，则对字段“默认值-Default value”，字段的“显示分组-Display in group”以及“帮助文本-Help text”的信息进行设置，然后就是一些和字段类型相关的参数设置，比如图像（文件类）的存储路径、上传大小限制、命名规则等。对于这些与类型相关的设置选项，个人感觉主要在对这些类型的特点和应用需求的理解，比如，我们需要添加报价，作为应用于价格的数字类型字段，首先我想到的是保证价格的精确性，那就不能用整数类型了，再就是书写习惯，货币通常都是前缀货币符号，数字部分千位逗号分隔并保留两位

小数的，这些需求都要在 CCK 字段的设置过程成给予确定（下一步 CCK 字段显示会再描述）。直接阅读这些字段各项设置的说明即可弄明白用法和注意事项，因为我们不是做说明书，就不一一列举了。

- **3、全局设置组内的参数**，是应用到所有引用了该字段的内容类型中的设置。如是否是“必要字段-Required”，“字段数据的个数-Number of values”以及“允许的值-Allowed values”等。再就是不同字段类型独有的公用属性。每个类型的字段都包含一组针对某个发生引用内容类型的私有设置，以及一组用于所有引用发生引用内容类型的全局设置。这是因为，**在 CCK2 中，可在不同类型的内容中引用同一个字段**。通过这种方式可以降低重复维护多个字段的投入，也可以降低数据库表的复杂程度（个人看法）——虽然这个方法貌似存在一些问题（最最最后面，会发一些“牢骚”说说 CCK2、Views 的长短处）。在本例中，可以假设还存在一个“报价”内容类型，也需要一个本本的图片，我们可以如下图那样，仅需一次点击就添加这个字段：



添加完所有字段之后的管理字段界面如图，下一部分我们开始增加字段组（Add group）和字段显示（Display fields）的设置。



Control the order of fields in the input form.

To change the order of a field, grab a drag-and-drop handle under the Label column and drag the field to a new location in the list. (Grab a handle by clicking and holding the mouse while hovering over a handle icon.) Remember that your changes will not be saved until you click the Save button at the bottom of the page.

Label	Name	Type	Operations
名称	Node module form.		
简介	Node module form.		
型号	field_benben_model	Text	Configure Remove
产地	field_benben_place	Text	Configure Remove
特性	field_benben_feature	Text	Configure Remove
评价	field_benben_comment	Comment Reference	Configure Remove
邮箱	field_benben_email	Email	Configure Remove
网址	field_benben_website	Link	Configure Remove
外观	field_benben_images	Image	Configure Remove
单价	field_uprice	Float	Configure Remove
参数	field_benben_parameter	Text	Configure Remove

创建、分配字段分组和字段排序

字段分组的作用第一部分已经说过了，创建、分配都比较简单。基于 D6 的 Ajax 特性，大多数操作由鼠标拖动即可完成。

（相对于 5.x 手工指定权重的方式，D6 拖动模式方便了很多，但是因屏蔽了具体的权重数值，因此可能造成新增模块字段显示顺序失控的问题。如果不明白我说的这点没有关系，我还没有确定造成这种情况的原因）

点击增加分组（Add group）进入分组创建页面，依次指定标签即组名（Label）、编辑时表单样式（Style）、编辑时帮助文本（Help text）以及节点显示时分组描述（Description）即可。

有两点需要提示。第一，建议组名新建时使用英文名称，建成后修改为中文，因为使用中文时创建的机器友好组名会以以下划线替代，看上去有些怪异；第二，表单样式默认提供了三种类型即总是展开（always open）、可折叠（collapsible）和已折叠（collapsed），在加载 cck_fieldgroup_tabs 模块和 Tabs 模块后，还提供标签页样式。个人喜欢标签页样式，这样可以为节点编辑界面提供一个清爽、直观的环境。

在本例中，我们把之前创建的字段分为商务、技术两个部分，部分截图如下：

编辑字段分组的界面：

完成字段分组和字段顺序的界面：

产地	单价	参数	型号	外观	特性	网址	评价	邮箱
----	----	----	----	----	----	----	----	----

Control the order of fields in the input form.

To change the order of a field, grab a drag-and-drop handle under the Label column and drag the field to a new location in the list. (Grab a handle by clicking and holding the mouse while hovering over a handle icon.) Remember that your changes will not be saved until you click the Save button at the bottom of the page.

Label	Name	Type	Operations
+ 名称	Node module form.		
+ 简介	Node module form.		
+ 外观	field_benben_images	Image	Configure Remove
+ 技术信息	group_technology		Configure Remove
+ 型号	field_benben_model	Text	Configure Remove
+ 特性	field_benben_feature	Text	Configure Remove
+ 参数	field_benben_parameter	Text	Configure Remove
+ 商务信息	group_commercial		Configure Remove
+ 单价	field_uprice	Float	Configure Remove
+ 产地	field_benben_place	Text	Configure Remove
+ 评价	field_benben_comment	Comment Reference	Configure Remove
+ 邮箱	field_benben_email	Email	Configure Remove
+ 网址	field_benben_website	Link	Configure Remove

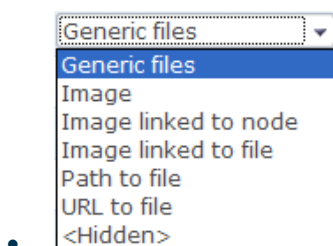
配置字段显示 (Display fields)

字段显示为字段最终的呈现样式提供了预处理功能（因为我们还可以进一步通过模板、CSS 进行这些字段外观的设定），在对外观没有特别需要的情况下，默认提供了样式可以满足大部分要求。

CCK2 的字段显示设置界面可将设置选项分为常规 (General)、高级 (Advanced) 两类。在常规页面中，我们可以指定字段标签的显示方式为 (位于上方、行内、隐藏)，我们也可以为摘要和完整节点页面分别指定类型相关的样式——这是非常有用的功能，在摘要模式下，我们完全可以隐藏重要或非重要字段，这可以获得相对简洁的页面效果（但我不能确定，这种隐藏是否可以降低数据库查询量，我没有对这点进行测试），也可以为摘要页面和完整页面赋予不同的 CSS 样式。

在高级页面中，我们可以为 RSS 条目、搜索索引以及搜索结果指定不同的字段样式，非常灵活，至于有什么好处或坏处，自己想想或者试试就知道了。

具体的，字段的显示与字段的类型相关，比如图像类型的字段，可以有如下几种显示方式：普通文件、图像、链接到节点的图像、链接到图像文件的图像、图像文件的路径、图像文件的 URL 地址，或者干脆隐藏。



文本类型、数值类型或者邮箱、评论、网址，都有自己可以设置的格式。我对本本的设置如图，您可以尝试一下其他的显示方式。

Field	Label	Teaser	Full node
外观	<Hidden>	Image	Image
技术信息	Above	fieldset	tabs
型号	Inline	Plain text	Default
特性	Inline	Comma-separated	Default
参数	Inline	Comma-separated	Default
商务信息	Above	fieldset	tabs
单价	Inline	9,999.99	9999
产地	Inline	Plain text	Default
评价	Inline	Title (no link)	Title (link)
邮箱	Inline	<Hidden>	Default email link
网址	Inline	<Hidden>	Title, as link (default)

基本上 CCK 的使用就是以上所述——还有最后一项，CCK 字段的权限控制。在 D5.X 版本下，CCK 的字段权限控制是一个独立模块，现在已经整合到 CCK2，并完全融入系统统一的权限设置界面中（位于 `admin/user/permissions`）。所有的字段都提供了查看 (View) 和编辑 (Edit) 两个权限，可以分配给不同的角色 (Role)。我想 CCK 权限与系统权限管理界面合二为一唯一的好处的提供了统一的通道，但是弊病也在于，现在 Drupal 的权限管理界面有些低能（抱歉，我还没有改善它的能力啊），十几个字段、几个角色最终组合在一起的“点击量”还是非常吓人的。所以有时候我也在幻想

CCK 拥有一个界面独立的权限设置场所...


扯远了，CCK 权限设置与其他模块、内容设置方式一致，就不再操作示范。在准备进入 Views 模块的使用介绍之前，我们来新建一个本本类型的节点，拿我最眼馋的 Macbook pro 来 YY 一下，看看我们的劳动成果。

Home

MacBook Pro [View](#) [Edit](#) [Outline](#) [Track](#)

Mon, 2008-09-15 23:13 — demo

[返回博客](#) [技术博客](#) [商务博客](#)



型号：MB133CH/A

最新 Intel 处理器，更大硬盘，更多内存空间和更多全新功能，全部汇集在小小的 2.99 厘米厚。全新 MacBook Pro 拥有台式机的性能，动力并高于普通，却完美保留台式机的强大传统。

设计——纯粹魅力，纯粹动力

MacBook Pro 采用 2.99 厘米的超薄精制外壳，新一代功能完备的锂电池供电。仅重 2.45 公斤，MacBook Pro 始终伴随你的脚步前进。

功能——Multi-Touch 触控板

全新 Multi-Touch 触控板，将令人惊叹的 Multi-Touch 技术引入 MacBook Pro 中。你只需单击，选想就成列表框，即可阅读文本，浏览照片或网页内容。

无线——随时随地，无线享受

MacBook Pro 拥有先进的内置无线技术，所以无论你身在何处或连接的任何网络，都能轻松无线连接。

Add new comment 7 reads

Home

MacBook Pro [View](#) [Edit](#) [Outline](#) [Track](#)

Mon, 2008-09-15 23:13 — demo

[技术博客](#) [商务博客](#) [返回博客](#)

特性：

扩展功能

- 一个 FireWire 400 端口，最多可支持 400 Mbps
- 一个 FireWire 800 端口，最多可支持 800 Mbps
- 15 英寸 MacBook Pro：两个 480-Mbps USB 2.0 端口
- 17 英寸 MacBook Pro：两个 480-Mbps USB 2.0 端口
- ExpressCard/34 插槽
- Kensington 锁接收孔

通讯功能

- 内置 AirPort Extreme 无线网卡功能 2（基于 IEEE 802.11n 增强标准），IEEE 802.11a/b/g 兼容 Wi-Fi 认证
- 内置蓝牙 2.1+EDR (Enhanced Data Rate)
- 内置 10/100/1000BASE-T 千兆以太网卡（内置 RJ-45 接头）

音频

- 内置立体声扬声器
- 内置麦克风阵列（位于左侧扬声器格栅下方）
- 此阵列支持输入/音频流输入/输出（迷你插孔）
- 此阵列支持输出/耳机输出/输入（迷你插孔）

参数：

	15.4-inch: 2.4 GHz (MB133CH/A)	15.4-inch: 2.5 GHz (MB134CH/A)	17-inch: 2.5 GHz (MB166CH/A)
显示屏	15.4 英寸 (对角线), 1440 x 900 分辨率, TFT 显示屏		17 英寸 (对角线), 1680 x 1050 分辨率, TFT 显示屏
处理器	2.4GHz Intel Core 2 Duo	2.5GHz Intel Core 2 Duo	
内存	2GB (两个 SO-DIMM) 667MHz DDR2 SDRAM (PC2-5300)		
PCI Express 图形芯片	配备 256MB GDDR3 SDRAM 显卡和双 DVI	配备 512MB GDDR3 SDRAM 显卡和双 DVI	配备 512MB GDDR3 SDRAM 显卡和双 DVI
硬盘 ³	200GB Serial ATA; 5400 rpm	250GB Serial ATA; 5400 rpm	
输入/输出	8 倍速 SuperDrive (DVD+R DL/DVD+RW/CD-RW)		
扩展功能	一个 FireWire 400 端口, 一个 FireWire 800 端口, 两个 USB 2.0 端口, 和 ExpressCard/34 插槽		

Home

MacBook Pro View Edit Outline Track

Mon, 2008-08-18 23:13 — demo

基本信息 技术信息 商务信息


售价: 17,998.00
产地: USA
邮箱: webmaster@apple.com.cn
网址: http://www.apple.com.cn/macbookpro/
Add new comment 7 reads

Home > MacBook Pro

MacBook Pro View Edit Outline Track

基本信息 技术信息 商务信息

名称: *
MacBook Pro

封面:


Description:
全新 MacBook Pro !

Alternate Text:
MacBook Pro

Title:
MacBook Pro

List

Replace:
 Choose...

Maximum Filesize: 2 MB
Allowed Extensions: .jpg .jpeg .png .gif

Upload Remove

缩略:
MB133C Split summary at cursor

简介:
全新 Intel 处理器, 更大容量, 更多内存选项和更多全新功能, 全部汇集在小小的 2.99 英寸屏, 全新 MacBook Pro 拥有台式机的性能, 助力并易于连接, 完美融合台式机的强大优势。
设计——强悍能力, 强悍助力
MacBook Pro 采用 2.99 英寸的高分辨率外壳, 新一代功能完备的超薄笔记本, 仅重 2.45 公斤, MacBook Pro 始终陪伴你的创意旅程。
功能——Multi-Touch 触控板
全新 Multi-Touch 触控板, 将令人惊叹的 Multi-Touch 技术引入 MacBook Pro 中, 你只需开会, 缩放或排列表格, 即可修改文本, 浏览并阅读网页。
外观——随时随地, 完美享受
MacBook Pro 拥有先进的内置无线技术, 所以无论你身在何处或连接哪种网络, 都能轻松完成连接。

Home > MacBook Pro

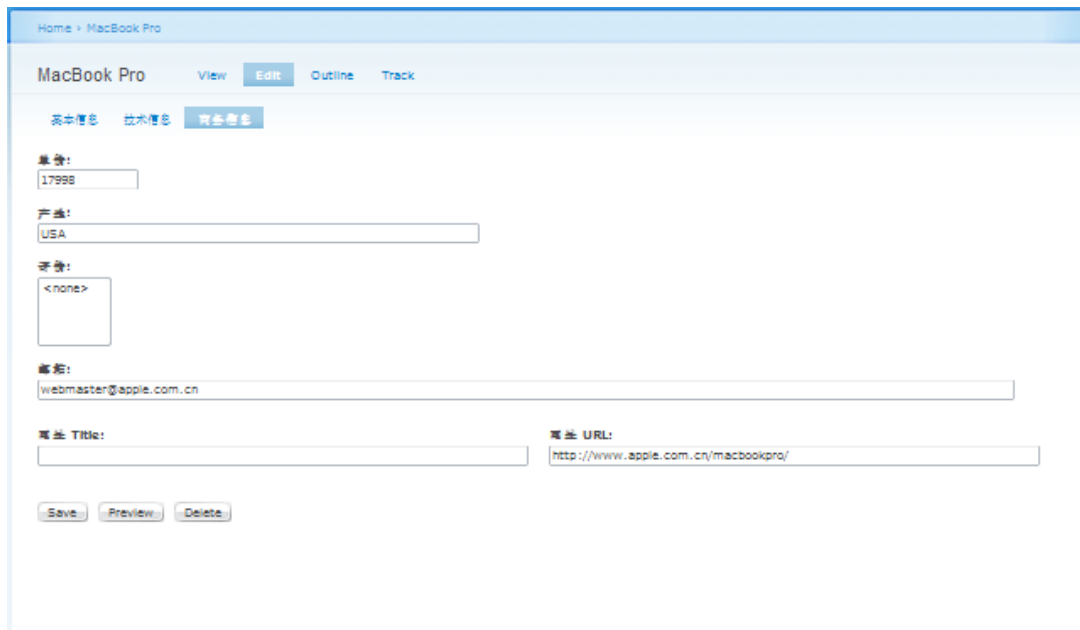
MacBook Pro View Edit Outline Track

基本信息 技术信息 商务信息

特性:
扩展功能
一个 FireWire 400 端口, 最多可支持 400 Mbps
一个 FireWire 800 端口, 最多可支持 800 Mbps
15 英寸 MacBook Pro: 两个 480-Mbps USB 2.0 端口
17 英寸 MacBook Pro: 两个 480-Mbps USB 2.0 端口
ExpressCard/34 插槽
Kensington 锁孔
连接功能
内置 AirPort Extreme 无线局域网 2 (基于 IEEE 802.11n 草案规范), IEEE 802.11a/b/g 兼容 Wi-Fi 认证
内置蓝牙 2.1+EDR (Enhanced Data Rate)
内置 10/100/1000BASE-T 千兆以太网 (内置 RJ-45 接口)

简介:
<table cellpadding="0" summary="MacBook Pro Configurations">
<tr class="header">
<th width="22%" class="left"> </th>
<th width="26%">15.4-inch: 2.4GHz
 (MB133C/A)</th>
<th width="26%">15.4-inch: 2.5GHz
 (MB134C/A)</th>
<th width="26%" class="right">17-inch: 2.5GHz
 (MB166C/A)</th>
</tr>
<tr class="odd">
<th class="left">显示屏</th>
<td colspan="3">15.4 英寸 / 对角线, 1440 x 900 分辨率, TFT 液晶</td>
</tr>
</table>

Save Preview Delete



贴图贴满，差不多就可以新开帖子开始 Views 了...

Views 的口水开场白

开始 Views 的总结不是很容易，与 CCK 不同，CCK 的应用反复在字段添加、编辑、设置之间，往返于多个页面，但其实过程是比较单一的，因为每个字段的处理都相对独立。Views，特别是 Views2，在一个界面中完成了所有的工作，各个参数之间相互关联，在我看来，难度确实要高于 CCK 很多...

但 Views 确实是让人惊喜的，特别对于我这样的程序盲。尽管如上所说它拥有太多的关联的参数，但 Views2 提供了尽量友好的方式，来进行相应功能的设置。

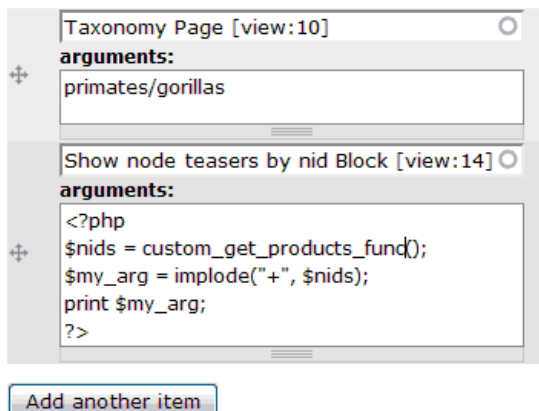
因为 Views 是一组查询，所以，根据不同的主表（primary table）Views 被分为节点 Node、评论 Comment、节点修订 Node revision、文件 File、术语 Term、用户 User 几个类型。您希望抽取站内某些注册的用户信息并显示出来，那么选择 Views 应主要考虑 User 类型。Views2 安装后提供了很多预置的 Views，可以启用、编辑查看相应的设置。Views 是一整套工具，因此除了上述核心内容之外，还提供了数据输入、输出的工具，控制缓存、预览等功能的设置，以及提供对 Views1 版本的转换。

后面将以创建具体的内容为例，为我们之前的本本创建一系列的“视图”，具体需求后面描述。

插播一项关于 View Reference 模块的信息

View Reference 是 CCK 模块，用来为节点提供 Views 的引用，就在今天（2008-Aug-30），View Reference 发布了 6.x-2.0 版本。现在，View Reference 终于增加了对 arguments 的支持（包括 php 代码模式的 arguments）。这意味着 Views 的应用灵活性再进一步。

View Reference:



将 Views 与 CCK 结合，是 Views 应用的重要方式之一。

我们将可能存在的需求罗列一下，然后尝试使用 Views 来完成这些需求，最后再来看看，Views 都为我们提供了什么。关于本本的一些扩展需求

通过 CCK 模块，我们创建了符合展现一个“本本”所需要的相关信息，但最终，它们都以一篇文章的方式表现了出来。我们的客户不会仅仅满足于这种形式的表现，那么我们可以考虑通过如下的形式来表现这些产品：

- 本期主打，这是比较有意思的一个功能，我们最新的本本比如 **Macbook pro**，或者按照某个规则指定的产品，将以独立的形式出现在首页顶部的某个位置，类似这个样子（抱歉，这不是 Macbook）：

Recommend Product



13.56MHz Contactless IDTECK Credential Smart Card Reader Series
SR10S/SR10SE/SR30S/
SRK101S/ISC80S/PRG2000S

- 13.56MHz (MIFARE) Contactless IDTECK Credential Smart Card Reader
- Up to 4" (10cm) Read Range
- ISC80S Smart Card Read Only

[▶ More](#)

- 分类产品，也是很典型的样式，比如我们按 **Macbook**、**MacBook air** 或者 **Macbook pro** 分类展现我们的新本本，前 5 款（或者更少/更多）将出现在这个分类产品列表里面，类似这个样式：

Hardware	Software	Application/Solution
 <ul style="list-style-type: none">• Biometrics• Standalone Controller• Access Control Panel• RF Reader / Card & Tag• ASK[EM] iPASS Series• Smart Card Reader• Long Range Reader & Tag• Accessories• Demo Kit	 <ul style="list-style-type: none">• Access Control Software• Time & Attendance Software• SDK	 <ul style="list-style-type: none">• Access Control• Time & Attendance• Car Parking Control• Total Security• CCTV• Case Study

- 产品列表，对于客户快速查阅本本的型号、价格和规格，这个模式可能更方便，没有参考图，类似最常见的 Excel 电子表单。

不好意思，拖的越来越久了...

需求分析——本期主打

从前面给出的本期主打样图来看，构成元素还是比较简单：一幅图片、三组文字（权且认为是简述、标题、摘要）以及一个 **More** 按钮。这些元素的关系也很简单，都是某主打产品节点信息的一部分。至于布局，按我的惯性理解，CSS 可以完成大部分的布局需求，所以最后应该是以使用 CSS 润饰收尾的。

知道了这部分内容的构成，主要思路就很简单了：

1、抽取节点的内容（字段）——》2、设置基本布局——》3、CSS 细节调整；

其中第 1、2 部分就是通过 Views 来实现的。

当然，我们还需要考虑其他的问题，才能最终满足实际应用，比如：

- 这项内容（本本）的状态，是否允许未发布的内容（本本）也显示出来？
- 要查看（访问）这项内容，是否需要某些权限的许可（比如访客是否可以直接看到——当然）？
- 再或者，以什么标准判定内容是“本期主打”？

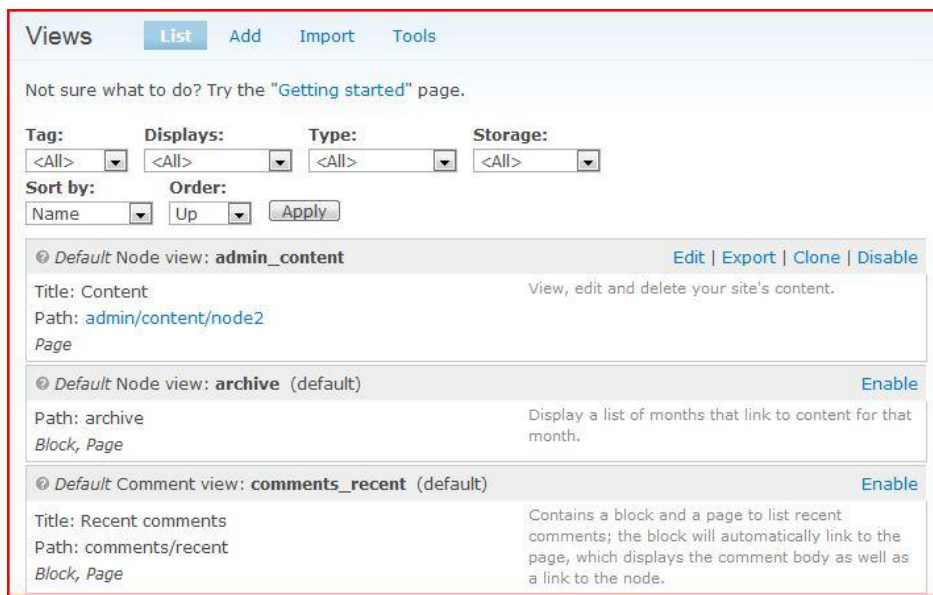
等等，这些方面的问题/需求，也大多能在 Views 中得到解决。

关于卖的够多了，下面开始进入具体的 Views 操作。

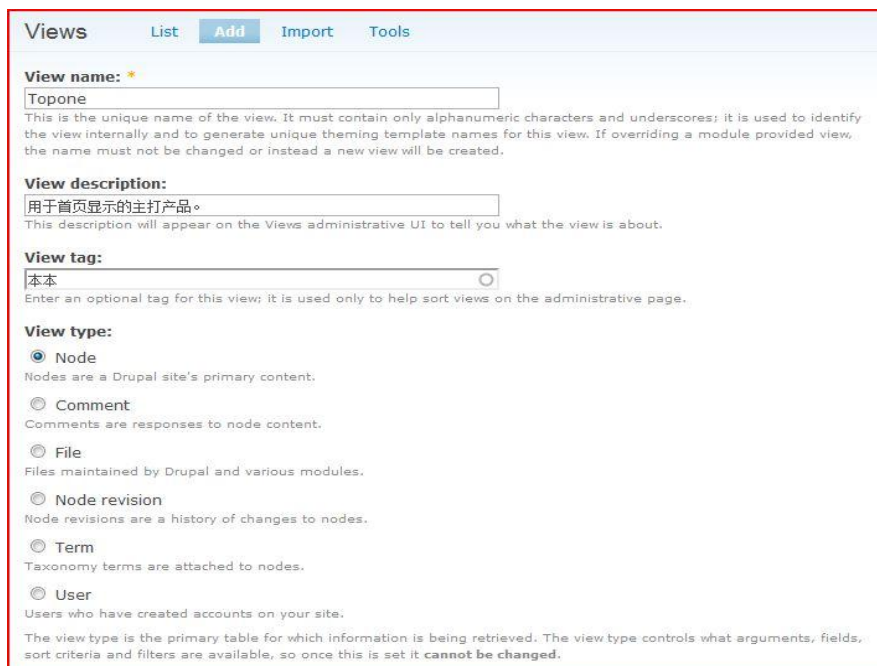
创建一个 Views（定制视图）

快速创建一个 Views

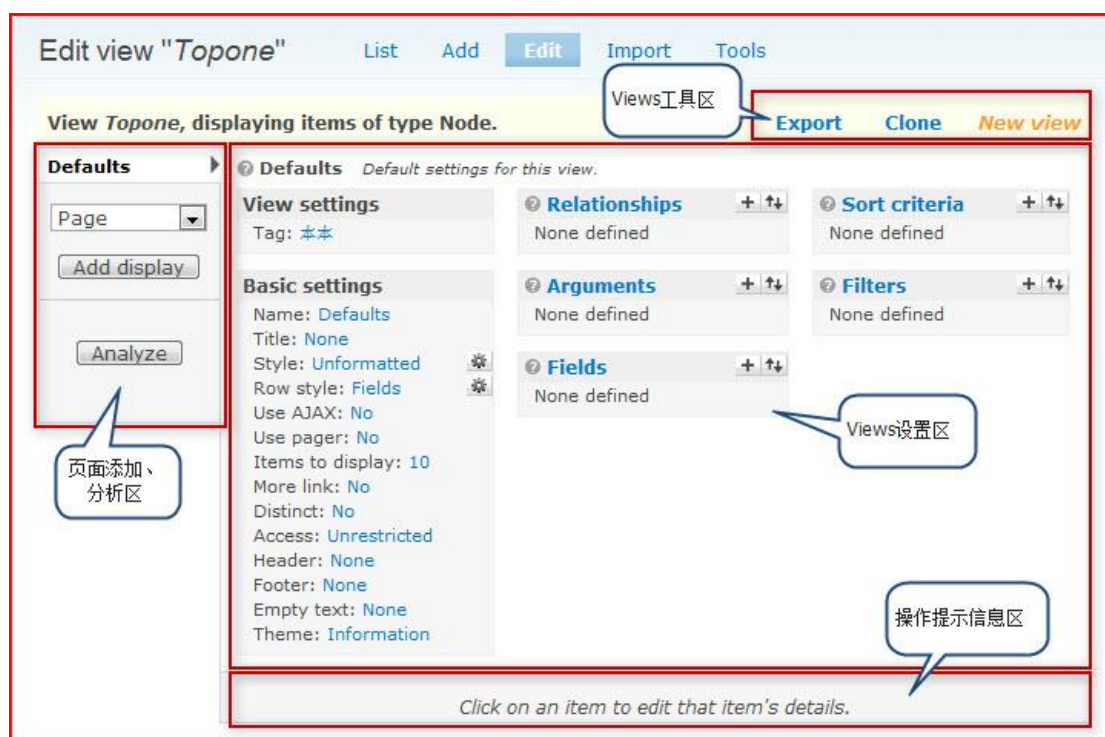
1. 进入 Administer > Site building > Views 页面（路径 admin/build/views）；



2. 点击 Add 添加一个 Views，如图设置之后点击继续；

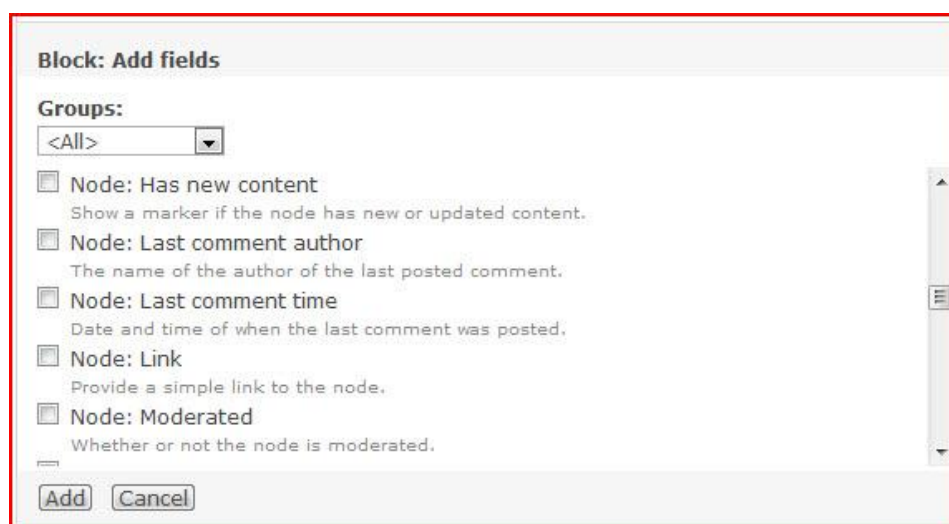


3. 进入 Views 编辑页面，各区域名称分别为如下所示，不求精确，仅进行辨识；



4. 在页面添加分析区下拉菜单处选择区块 **Block**，然后点击增加显示（Add display）；

5. 在 Views 设置区 Fields 部分点击右上角的十字图标，开始添加字段，Views 设置区下方将呈现如下设置界面；



6. 在字段组 Group 下拉框处，选择 Content，点选如下几项；

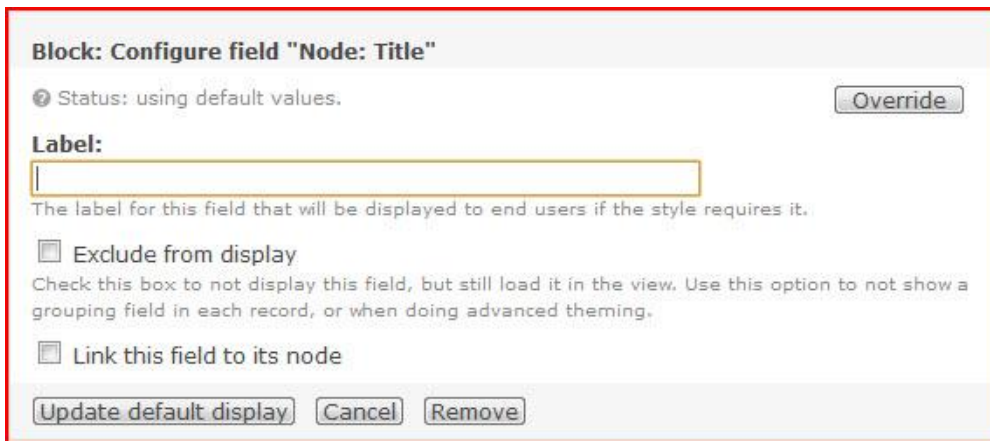
1. Content: Image: 外观 (field_benben_images);
2. Content: Text: 特性 (field_benben_feature);

如图所示：



7. 在字段组 Group 下拉框处，再切换至 Node，點選 Node: Title，然后点击添加 Add 按钮；
8. 紧接上步，将弹出一系列区块配置字段“Block: Configure field”的界面，一次设置以上步骤添加的字段并点击更新默认显示（Update default display），具体设施分别如下：

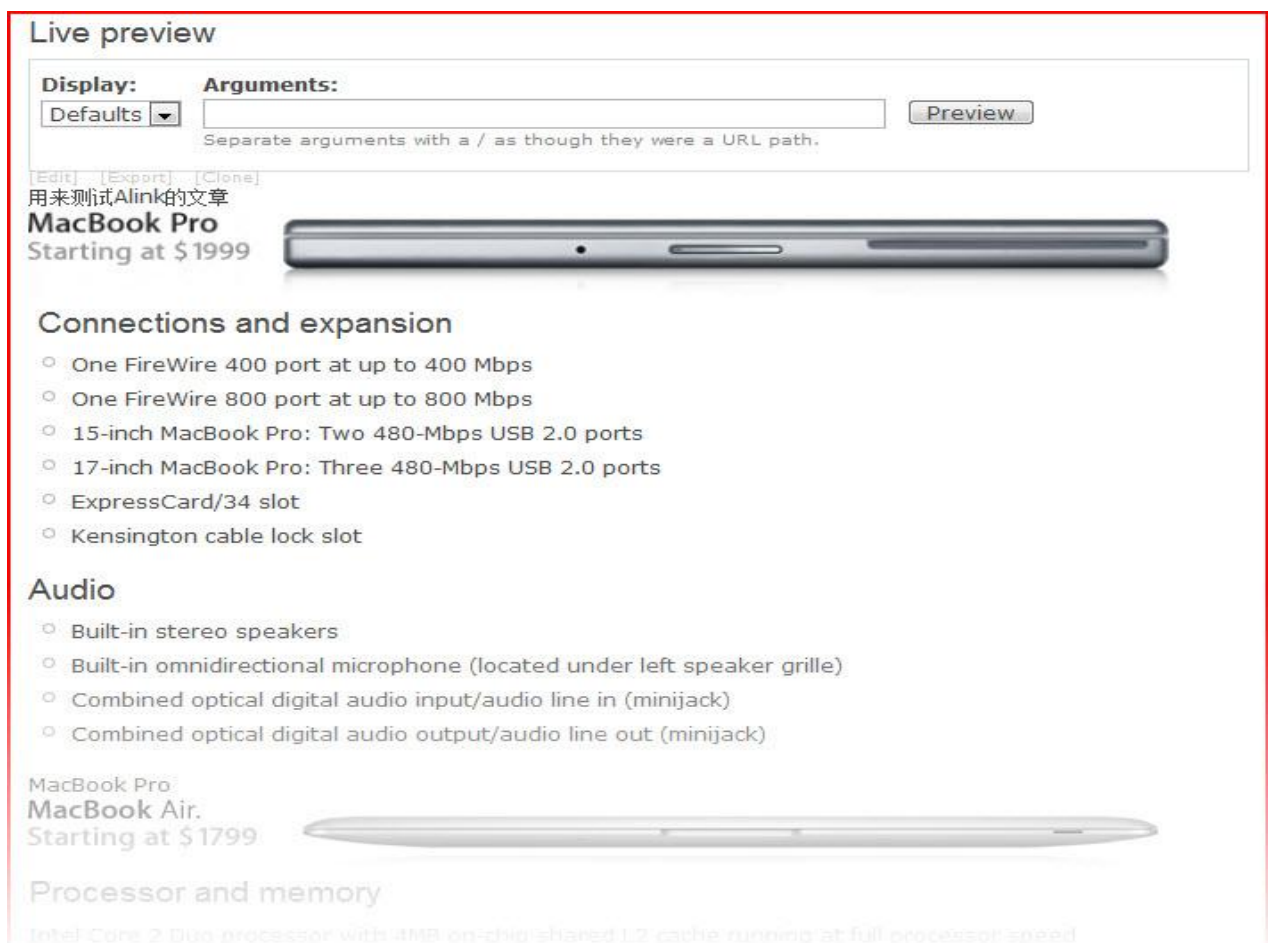




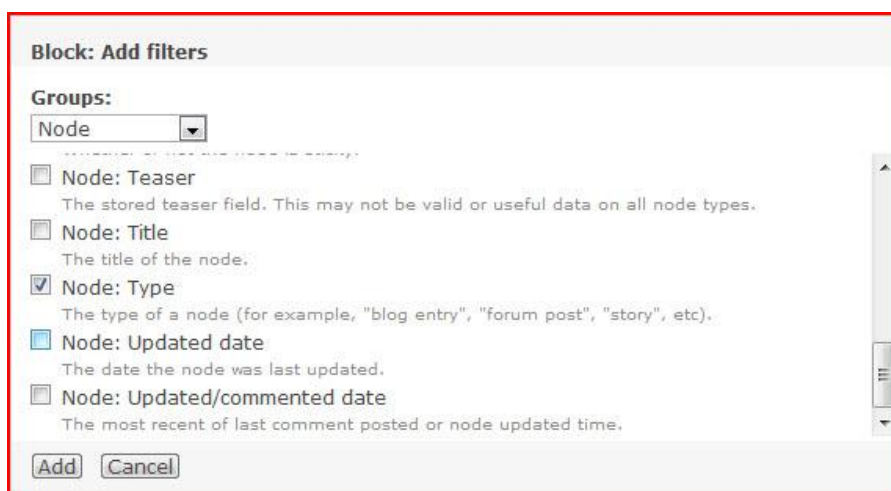
9. 分别对外观、特性、标题设置完毕后，Views 设置区 Fields 部分如图所示：



10. 注意下面的实时预览（Live preview）终于出现了内容，刚才一些红色的警告信息也消失了：



11. 继续在 Views 设置区 Filters 部分添加如下 Node: Type 内容（过程与刚才添加字段相似）：



Block: Add filters

Groups:
Node

- Node: Teaser
The stored teaser field. This may not be valid or useful data on all node types.
- Node: Title
The title of the node.
- Node: Type
The type of a node (for example, "blog entry", "forum post", "story", etc).
- Node: Updated date
The date the node was last updated.
- Node: Updated/commented date
The most recent of last comment posted or node updated time.

Add Cancel

12. 在紧接着的页面中按照如下设置进行配置，点击更新默认显示（Update default display）：



Block: Configure filter "Node: Type"

Status: using default values. Override

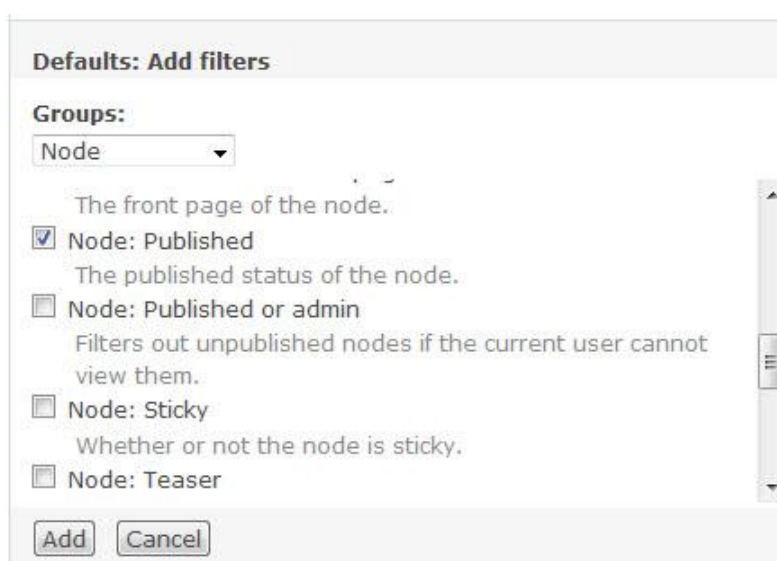
This item is currently not exposed. If you **expose** it, users will be able to change the filter as they view it. Expose

Operator: **Node type:**

- Is one of
- Is not one of
- 本本
- Page
- Story

Update default display Cancel Remove

13. 点击 Views 设置区下方的保存按钮保存设置，准备继续进一步调整——我们需要确保显示的产品是发布的，所以再增加一个节点发布状态（Node: Published）的过滤，并选择 Published 选项：



Defaults: Add filters

Groups:
Node

- The front page of the node.
- Node: Published
The published status of the node.
- Node: Published or admin
Filters out unpublished nodes if the current user cannot view them.
- Node: Sticky
Whether or not the node is sticky.
- Node: Teaser

Add Cancel

Defaults: Configure filter "Node: Published"

This item is currently not exposed. If you **expose** it, users will be able to change the filter as they view it.

Published

14. 我们还需要调整显示本本的数量，因为我们只需要 1 个型号显示在首页。点击基本设置（Basic settings）部分的显示条目数（Items to display），进行如下设置：

Defaults: Items to display

The number of items to display per page. Enter 0 for no limit.

Offset:

The number of items to skip. For example, if this field is 3, the first 3 items will be skipped and not displayed. Offset can not be used if items to display is 0; instead use a very large number there.

15. 问题来了，如何保证显示的这一个本本是我们的主打呢？可以有两种假设：

0. 所有最新发布的本本就是最新主打；
1. 最新发布的标记有主打的本本是最新主打；

第一种假设是比较牵强的，但是很好实现；第二种存在一个问题，那就是如何标记本本是否是主打产品。

这里我们只按照第一种假设为例进行示范。但是实现第二种的方法也比较简单，运用 CCK 增加一个本期主打字段，并将控件模式选择为选择框，然后再使用过滤器筛选就可以了。

但无论如何，这两种假设都需要一个顺序的问题，所以我们在排序规则（Sort criteria）处增加一项以发布时间为基准的排列，进行如下的设置：

Defaults: Add sort criteria

Groups:

- Node: Last comment time
Date and time of when the last comment was posted.
- Node: Moderated
Whether or not the node is moderated.
- Node: Nid
The node ID of the node.
- Node: Post date
The date the node was posted.
- Node: Promoted to front page

我们选择了节点发布日期 **Node: Post date**，当然也可以选择其他的一些参数。对于发布日期过滤器，设置排列顺序和精度两项参数：

Defaults: Configure sort criterion "Node: Post date"

Sort order:

Ascending

Descending

Granularity:

Second

Minute

Hour

Day

Month

Year

The granularity is the smallest unit to use when determining whether two dates are the same; for example, if the granularity is "Year" then all dates in 1999, regardless of when they fall in 1999, will be considered the same date.

16. 因为，我们创建的是区块 **Block**，还需要为 **Block** 指定一个名称，然后我们才能在页面中添加这个区块，点击 **Block settings** 的 **None** 添加一个管理描述名称：

Block settings

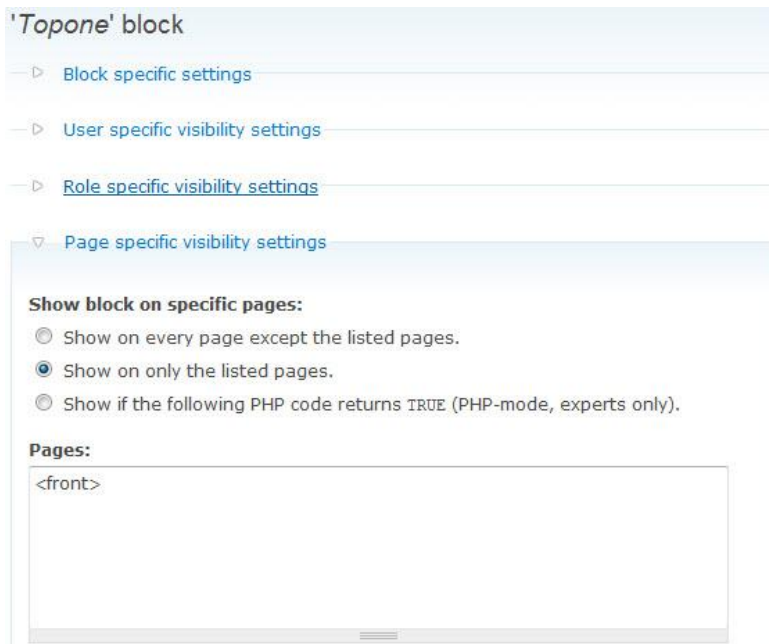
Admin: [None](#)

Block: Block admin description

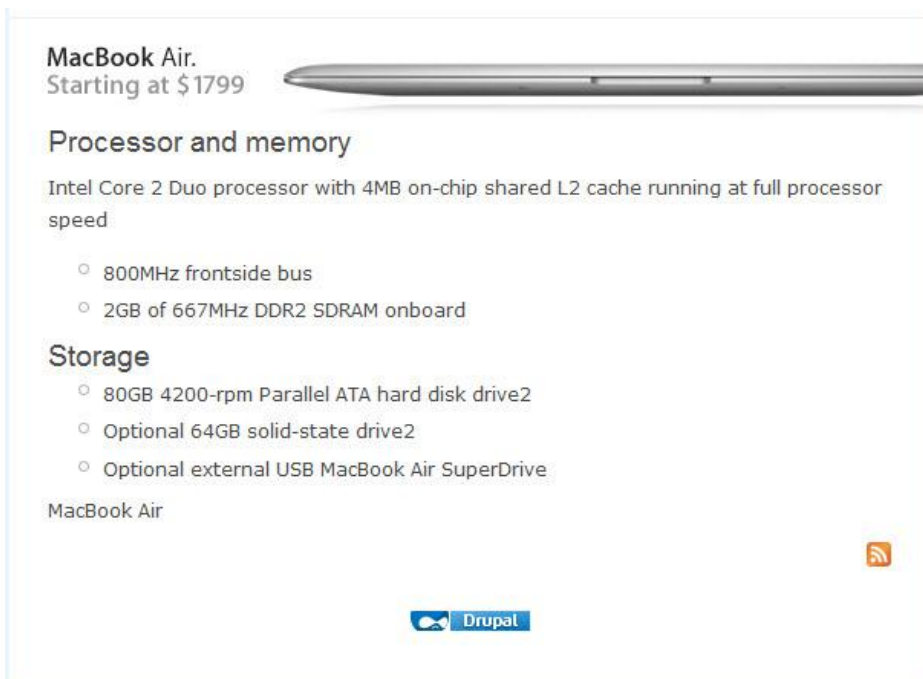
This will appear as the name of this block in administer >> site building >> blocks.

17. 点击保存按钮，然后我们去区块管理页面将我们的 **Views** 添加到需要的位置。比如首页的正文部分：

Block	Region	Operations
Left sidebar		
+ Navigation	Left sidebar ▼	configure
+ User login	Left sidebar ▼	configure
Right sidebar		
<i>No blocks in this region</i>		
Content		
+ Topone*	Content ▼	configure



18. 最终的首页效果如图:



19. 注意先别喷，我知道现在的结果距离开始的设定实在相差太远：图片很大而且撑破布局、居然显示在页面底部、文字格式内容超多等等等等。这涉及两个需要解决的问题：主题区域（Theme region）的设置以及 Views 主题的设置。主题区域的设置属于模板相关的讨论，建议参阅葛大翻译的《[Drupal 6.x 主题制作指南](#)》。我们下面来看看 Views 的主题如何设置。

Views 的主题设置

在我看来，Views2 这个版本主题设定的手段先进了许多，直接利用 Views 的设定，配合 CCS 可以完成很深很细的外观控制。点击基本设置中的“Theme: Information”，会显示如下的界面：

Theme: Information

Defaults: Theming information

This section lists all possible templates for the display plugin and for the style plugins, ordered roughly from the least specific to the most specific. The active template for each plugin -- which is the most specific template found on the system -- is highlighted in bold.

- **Display output:** **views-view.tpl.php**, views-view--Topone.tpl.php, views-view--default.tpl.php, views-view--Topone--default.tpl.php
- **Style output:** **views-view-unformatted.tpl.php**, views-view-unformatted--Topone.tpl.php, views-view-unformatted--default.tpl.php, views-view-unformatted--Topone--default.tpl.php
- **Row style output:** **views-view-fields.tpl.php**, views-view-fields--Topone.tpl.php, views-view-fields--default.tpl.php, views-view-fields--Topone--default.tpl.php
- **Field Content: Image:** 外观 (field_benben_images) (ID: field_benben_images_fid): **views-view-field.tpl.php**, views-view-field--Topone--field-benben-images-fid.tpl.php, views-view-field--default.tpl.php, views-view-field--default--field-benben-images-fid.tpl.php, views-view-field--Topone--default--field-benben-images-fid.tpl.php
- **Field Content: Text:** 特性 (field_benben_feature) (ID: field_benben_feature_value): **views-view-field.tpl.php**, views-view-field--Topone--field-benben-feature-value.tpl.php, views-view-field--default.tpl.php, views-view-field--default--field-benben-feature-value.tpl.php, views-view-field--Topone--default--field-benben-feature-value.tpl.php
- **Field Node: Title** (ID: title): **views-view-field.tpl.php**, views-view-field--Topone--title.tpl.php, views-view-field--default.tpl.php, views-view-field--default--title.tpl.php, views-view-field--Topone--default--title.tpl.php, views-view-field--Topone--default--title.tpl.php

Rescan template files

Important! When adding, removing, or renaming template files, it is necessary to make Drupal aware of the changes by making it rescan the files on your system. By clicking this button you clear Drupal's theme registry and thereby trigger this rescanning process. The highlighted templates above will then reflect the new state of your system.

-----END-----

使用 Drupal 6 Views Module 系列(一)

Drupal 是一套內容管理系統，本系列文章將以 View 模組實際示範網站建置，除展示其驚人的彈性之外，也希望能稍微降低其學習門檻，引領讀者們進入 Drupal 的世界。

假设我们现在想做出一个功能：让使用者建立歌词及歌手资料库，歌手及歌曲间需要建立 reference 关係(也许不是一对一)，建立起来后需要做不同的显示页面。这时我们就需要 Drupal 的外挂明星模组 CCK 及 Views 来帮助我们囉。

简单来说，CCK(Content Construction Kit)提供的服务就是让我们可以建立自订表单，并且可以在表单视需要添加各种资料输入栏位，包括一些基本的资料型态(ex: integer、text、date 等)、以及一些常见的表元素(ex: textfield、checkbox、radio 等)。最棒的是还可以建立资料参照(reference)。以前述例子来说，我们可以分别建立歌手与歌曲资料输入表单(两种内容类型)，然后将歌手的作品栏位定义成参照到歌曲这个内容类型即可。CCK 的工作到此完成。当资料库中渐渐累积了一些资料之后，我们会对系统预设的显示方式渐渐感到有所不足，这时就换到 Views 模组上场了。Views 模组目前最新的版本是搭配 Drupal 6 的 Views 2。

什麼是 Views? 一个 view 代表一份列举的内容，而 view 的输出叫做 Display。Display 可以显示在页面、区块、Feed、Attachment 或者任何外挂模组提供之 Display (ex: Date 行事曆)，且一个 view 可以有多个 displays。...」嗯...程式师写的说明文件果然是有看没有懂。这也就是笔者想要将自身学习的结果整理后用中文写下来的原因，希望这一系列的文章能给 Drupal 的中文使用者一些帮助。

开始使用 Views

用实际范例学习是快速入门的不二法宝，所以不囉唆，我们先建立两个内容类型分别如后，song 及 artist。第一个 song 的 title 栏位的 label 是「歌名」，body 栏位的 label 是「歌词」，CCK 栏位 field_song_type 之 label 为「类型」，data type

为 Text。栏位定义如下图：

Label	順序	名稱	類型
歌名	-5	Node module form.	
歌詞	-4	Node module form.	
類型	-3	field_song_type	Text

图表 1 栏位定义

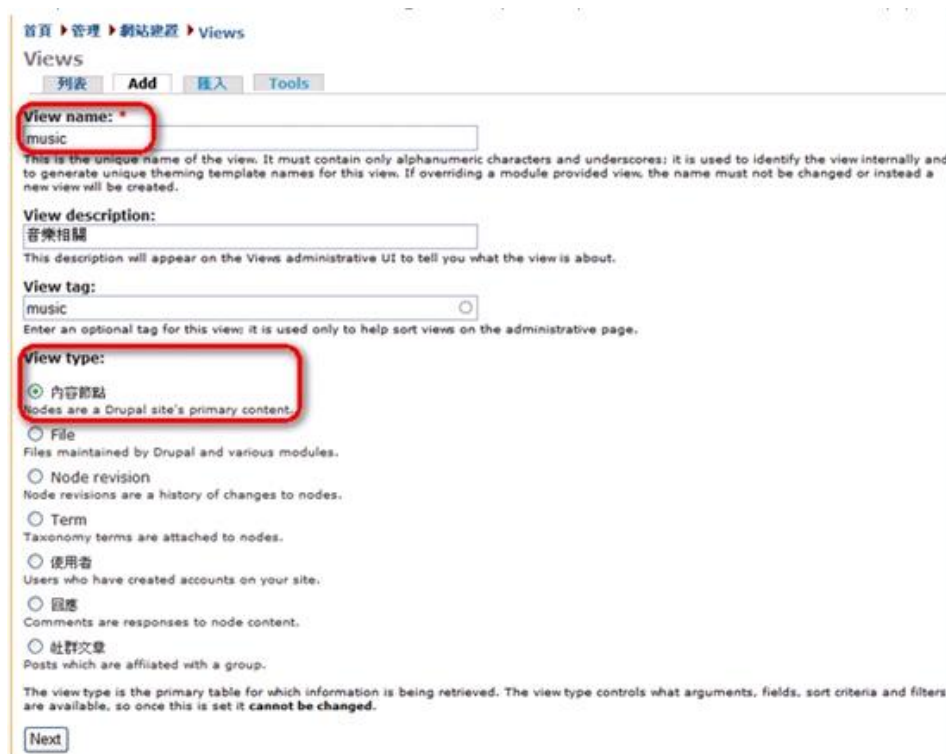
第 2 个内容类型 artist 的 title 栏位 label 是「歌手」，body 栏位的 label 是「歌手简介」；CCK 栏位 field_created_song 之 label 为「创作」，data type 为 Node reference，并与 song 这个内容类型产生关联。栏位定义如下图：

Label	順序	名稱	類型
歌手	-5	Node module form.	
歌手簡介	-4	Node module form.	
創作	-3	field_created_song	Node reference

图表 2 建立关联(Node Reference)

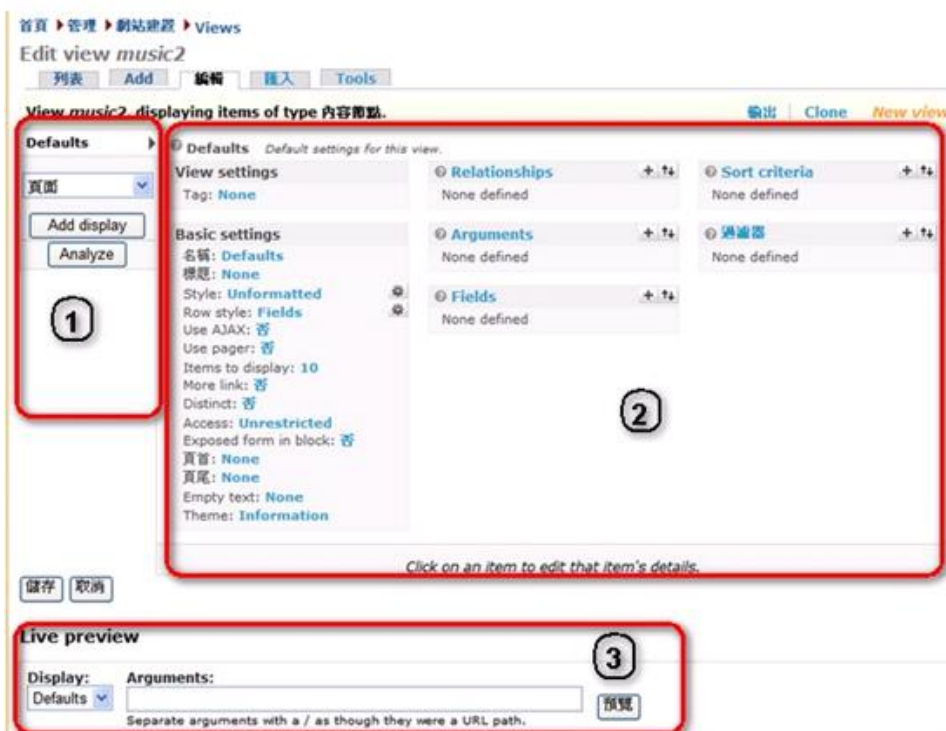
内容类型定义完成后，先建立几篇关于 song 的文章，然后再建立几个歌手的资料，歌手的创作栏位中可以单选或複选(也就是一个歌手可以有多个创作)。你可以到最新文章中检视刚才所发表的几篇文章。

现在请选择「管理→网站建置→Views」，按下 Add Tab 来建立一个新的 View 叫做 music，因为我们要产生显示的是内容相关的节点，故 View Type 请选择内容节点。



图表 3 建立 music View

按下 Next 后，即进入 configure 阶段。由于可设定的项目太多且複杂，我先将画面分成三大部分：① Add Display、②对应之 Display Settings、③预览。Views 2 的设定画面如下：



图表 4 configure View

在①中箭头标示目前指在 **Defaults display**，表示我们可以先在②做一些基本且共通的设定，比如标题或是欲显示的栏位 (Fields)，后续当然可以针对不同的 display 做修改。我们在 **Basic settings** 中设定标题为「我的音乐宝库」，并将 **Style** 设为表格，按下储存，此时会有 warning 说明尚未设定显示之栏位，所以请于 **Fields** 区块按下 **+** 就可以新增栏位，画面显示如下：



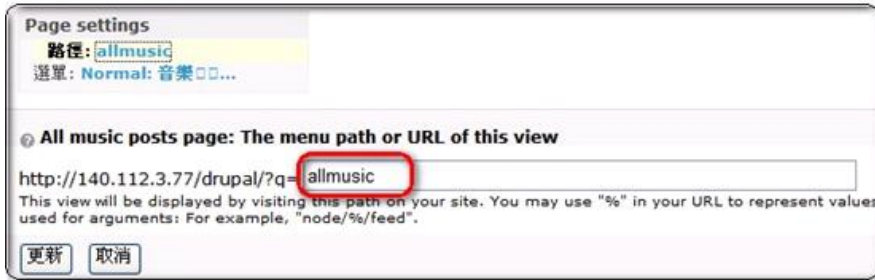
图表 5 加入栏位(Add fields)

在社群(group)的下拉式选单中，请选择内容节点，并勾选标题及类型，按下 **Add**，记得有变更任何设定时都要储存。一般来说，我们使用 **views module** 都倾向建立页面(page)或是区块(block)等基本的 display。

现在我们要製作一个名称为 **All music posts page** 的 display，先在①中下拉式选单选择 **页面**，然后按下 **Add display** 新增。前面提过每一个 display 可以有个别的设定，但预设会从 **defaults display** 先取出基本设定，然后，我们再依需求按下 **override button** 来新增或修改目前这个 display 设定。所以现在请按照下列顺序操作：

1. Basic settings→修改名称。

2. Page settings→ 设定路径。这是最重要的步骤，若无 path，此 display 无法建立。在同一个 view 中的不同 displays 可以设定相同的 path，因此若再搭配 Basic settings 中 Access 设定限制，我们可以让不同权限的使用者经由相同的 path 看到不同程度的内容。



图表 6 Page setting

3. Page settings→ 建立选单(若希望让 user 可以从网站导航中快速存取此 display 时)。



图表 7 Page Setting /建立选单

4. 若设定为 menu tab 类型，经由设定可以做出 tabs 的效果如下图(我们先将基本操作完成，下次再介绍此类进阶功能)：



5. 在②按下 Fields→按下 override→更新。此时针对 Fields 的所有增删及修改都不会动到 default display 的设置，且会出现 **+** (新增栏位)及 **↑ ↓** (栏位排序用)，所以按下标题作修改(勾选 Link this field to its node):

All music posts page: Configure field 內容節點: 標題

Status: using overridden values.

Label:
名稱

The label for this field that will be displayed to end users if the style requires it.

Exclude from display
Check this box to not display this field, but still load it in the view. Use this option to not record, or when doing advanced theming.

Link this field to its node

更新 取消 移除

图表 8 Link this field to its node

新增内容节点的 post date 栏位并设定如后，Date format 下拉式选单请选自订，Custom date format 文字输入格式请写 Y/m/d (输出格式会是 2009/02/05)，按下更新后储存。

All music posts page: Configure field 內容節點: Post date

Status: using overridden values.

Label:
張貼日期

The label for this field that will be displayed to end users if the style requires it.

Exclude from display
Check this box to not display this field, but still load it in the view. Use this option to not record, or when doing advanced theming.

Date format:
自訂

Custom date format:
Y/m/d

If "Custom", see the PHP docs for date formats. If "Time ago" this is the the number of d defaults to two.

更新 取消 移除

图表 9 Custom date format

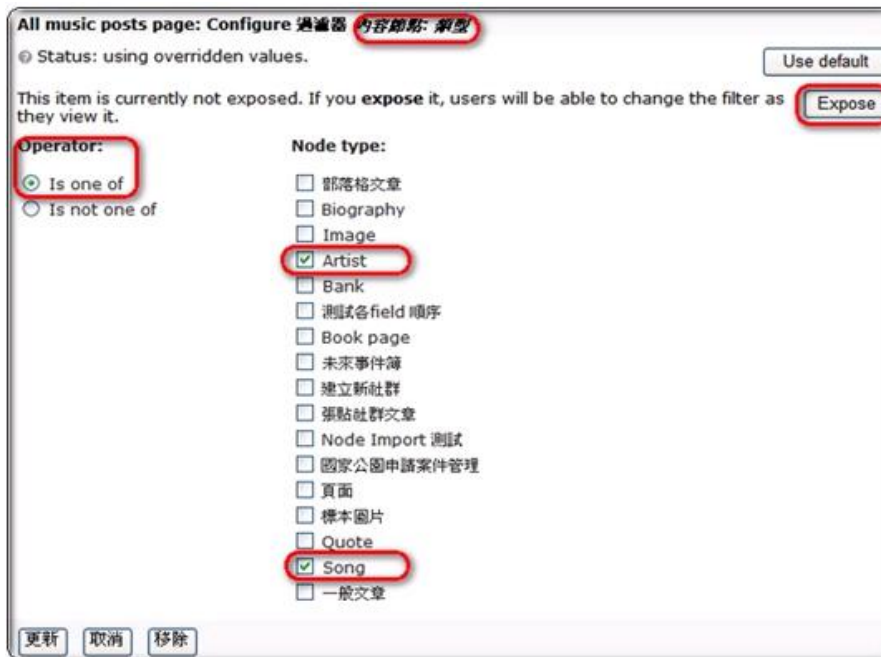
最后 Fields 设定会成为如右图示:

Fields + ↑ ↓

- 內容節點: 標題 名稱
- 內容節點: 類型 類型
- 內容節點: Post date 張貼日期

图表 10 Fields

6. 在②按下过滤器(filter)→按下 override→更新→按下过滤器上的 + 新增一个内容节点: 类型，选择只显示 Artist 及 Song 这 2 个内容类型的文章。设定方式如下:



图表 11 内容型态过滤

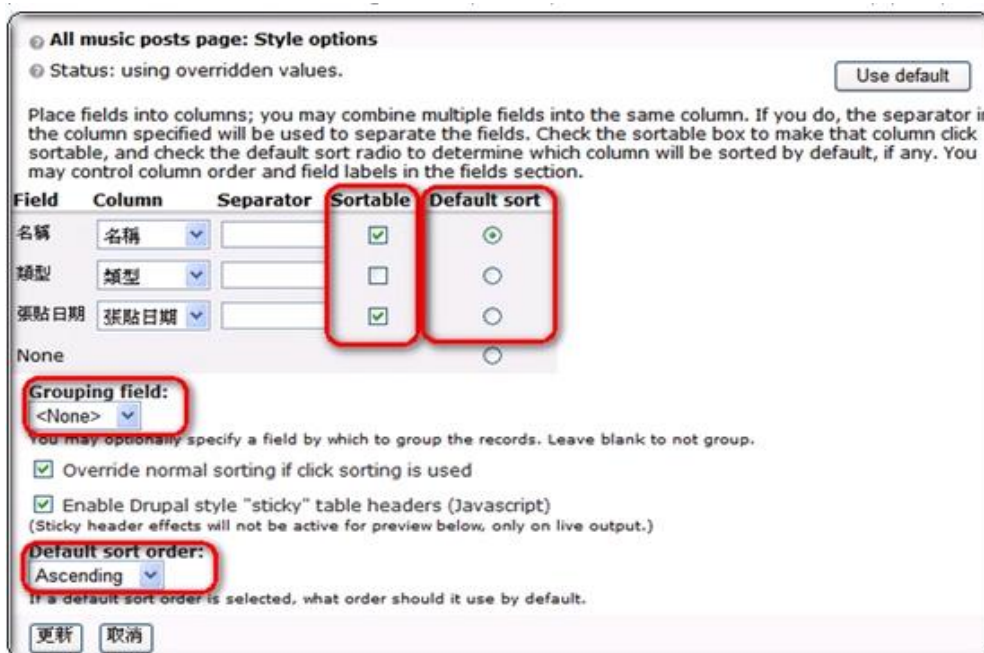
7. Expose 也是一个进阶的功能，提供 user 自行在网页上筛选欲显示的内容类型文章。

8. 再回到 Basic settings→设定 style→按下 override→更新。现在我们要针对表格做设定，所以请按下「Style: 表格」右侧的 Change settings for this style button。



图表 12 Basic settings / Style

9. 在 Style: 表格设定画面中，请于 **Sortable column** 勾选「名称」及「张贴日期」，并设定 **Default Sort** 栏位是「名称」。**Grouping field** 设定是将某个 field 设为 group by 的依据，这个 field 会以 table header 方式显示，相关的资料会显示于 header 下。在此我们不设任何 Grouping field，而是以一个完整 table 显示。做完设定请按下更新。



图表 13 Style options

10. Basic settings→Use pager 设为是→Items per page 设为 10。这个设定会形成分页的效果，亦即每一页只显示 10 笔资料，最后请按下储存，这个 display 的设定才算完成。

11. 在 Basic settings 中，我们再做些说明

11.1 **Access: authenticated user** 是指所有认证过的 user 经由 URL path 或者导览选单皆可看到目前这个 display。因此若要限制部分使用者浏览，可以经由此设定所允许的角色(Role)或权限。

All music posts page: Access options

Role:

- anonymous user
- authenticated user**
- 國家公園
- 小學生聯盟
- 第三組
- 臺大計中

Only the checked roles will be able to access this display. Note that users with "access all views" can see any view, regardless of role.

图表 14 存取设定

11.2 页首、页尾及 **Empty text** 则是做文字说明，比如筛选出来无任何资料显示时，让 user 看到些说明，而不是只有空白的网页。

11.3 **Theme: Information** 会显示此 view 所使用的 templates 档桉，所以我们可以针对需求修改这些档桉，档桉内容是 CSS 及 PHP 程式码，当然这也是属于进阶的功能。

12. 终于要说到画面③**Live preview**，请在 Display 下拉式选单中选取 **All music posts page**，并按下预览：画面的上半部是预览结果，下半部则是 SQL Query 语法及执行的时间。这部分的资讯可以提供程式师判断目前设定是否会耗去太多系统资源，可以再做修正。

Live preview

Display:
All music posts page

Arguments:

Separate arguments with a / as though they were a URL path.

名稱	類型	張貼日期
Beautiful	Song	2009/01/22
CRANBERRIES	Artist	2008/12/25
Dreams	Song	2008/12/25
Emotions	Song	2008/12/24
Hero	Song	2008/12/24
So Cold In Ireland	Song	2008/12/25
True Color	Song	2008/12/24
We Belong Together	Song	2008/12/24
七里香	Song	2008/12/24
周杰倫	Artist	2008/12/24

1 2 下一頁 > 最後一頁 >>

Query

```
SELECT node.nid AS nid,
       node.title AS node_title,
       node.type AS node_type,
       node.created AS node_created
FROM   node node
WHERE  node.type in ('artist', 'song')
ORDER BY node_title ASC
```

標題 我的音樂寶庫
路徑 allmusic
Query build time 16.3 ms
Query execute time 0.83 ms
View render time 46.35 ms

13. 现在来看执行结果:

请登录 drupal, 在画面左边的导览会出现「音乐宝库」的 menu link, 按下后画面中间即为 **All music posts page** 的 display。使用者可以 click「名称」或「张贴日期」来重新排列资料序, 下端则有分页的连结。若想要看详细内容, 只需点击歌曲或歌手名称即进入该篇文章。另外, 使用者也可以直接于网址列 type 路径 <http://yoursite.com/drupal/?q=allmusic>, 也会得到相同显示。



What else can we do next?

先回到 music view 编辑画面, 在区域①的下拉式选单中至少还有 **Feed**、**区块(Block)**及**附加档档(Attachment)**还没加入; 而在区域②中的 **Arguments**, **Relationships** 及 **Sort Criteria** 也都是 None defined。由此看来我们的 Views 还有将近三分之二的功能没开发, 我们将在本系列(二)中为大家介绍这些强大的功能及使用方式, 敬请期待。最后设定页面之画面如下:



CCK 的介绍及使用

下面我将简单向大家介绍些 CCK 和分享下一个简单使用的案例，让大家更好的熟悉 CCK。

CCK 模块，是 Content Creation Kit 的缩写。主要用于为站点建立一个新的特定用途的内容类型的模块组。可以很方便的通过 web 页面来创建设定你想要使用的内容类型，及其呈现的方式。比如像下面这些例子：

l 电影：电影标题，导演，制作公司，发布时间，星级等等；

l 会议：会议主题，开始时间，结束时间，组织者，讨论点，地点等等；

l 书籍：书名，作者，出版时间，出版社等等。

CCK 模块除了内建的字段类型外，还有很多可以自定义内容类型的字段类型，日期，email，图片，运算，星级评定，幻灯片等。使用这些特定字段类型，你就可以创建自己想要的。

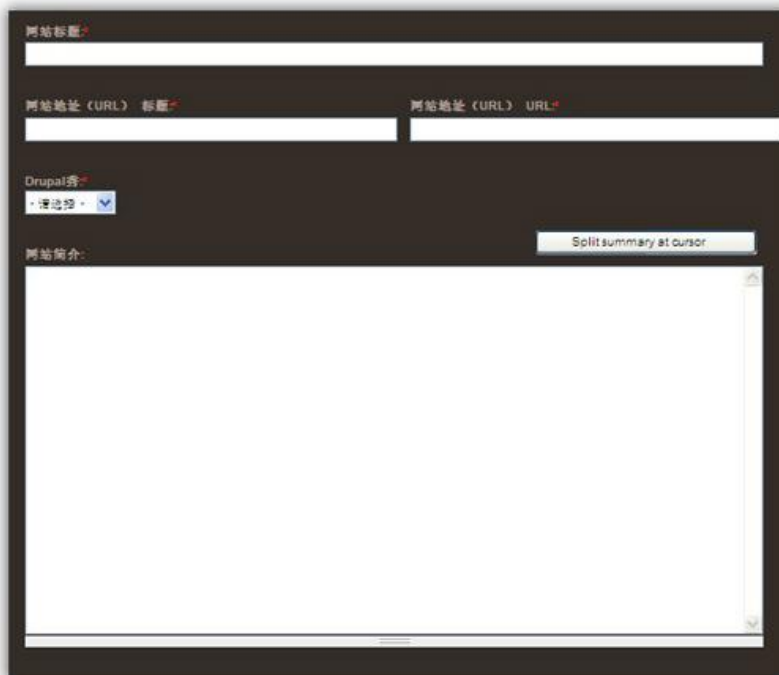
简单的介绍完后，是不是已经感到 CCK 模块的强大。下面我就分享一个小小的使用案例：“Drupal 秀”内容类型的创建。

显然，Drupal 秀的用途就是秀下你的 Drupal 站点了。只需要用户按照内容格式的输入，就可以在本站点上来秀下你的 Drupal 网站。

效果展示：<http://www.cmstheme.org/node/add/drupal>

这是一个很简使用，Drupal 秀的需求是：

Drupal 秀：网站标题，Url 标题，URL，网站简介。如图界面：



如图： Drupal 秀提交界面

首先，安装 CCK 模块和 CCK field 模块。

CCK 下载地址：<http://drupal.org/project/cck>

安装 CCK 模块与其他模块安装是一样。此模块包有一系列的模块，可以单独的启用，如下截图：

应用	名称	版本	描述
			Allows administrators to define new content types. Content Copy (disabled), Content Permissions (disabled), Cache (disabled), Date Copy (disabled), Paragraph (disabled), FileField (enabled), ImageField (disabled), Link (enabled), Node Reference (enabled), Number (enabled), Option Widgets (enabled), Text (enabled), User Reference (enabled), FileField Meta (disabled), FileField Tokens (enabled), 禁止再
<input checked="" type="checkbox"/>	Content	6.x-2.1	
<input type="checkbox"/>	Content Copy	6.x-2.1	Enables ability to import/export field definitions. 依赖于: Content (enabled)
<input type="checkbox"/>	Content Permissions	6.x-2.1	Set field-level permissions for CCK fields. 依赖于: Content (enabled)
<input type="checkbox"/>	Fieldgroup	6.x-2.1	Create display groups for CCK fields. 依赖于: Content (enabled)
<input checked="" type="checkbox"/>	FileField	6.x-3.0-alpha6	Defines a file field type. 依赖于: Content (enabled)
<input type="checkbox"/>	FileField Meta	6.x-3.0-alpha6	Add metadata gathering and storage to FileField. 依赖于: FileField (enabled), field missing, Content (enabled)
<input checked="" type="checkbox"/>	FileField Tokens	6.x-3.0-alpha6	Token integration for FileField. 依赖于: FileField (enabled), Content (enabled)
<input type="checkbox"/>	ImageField	6.x-3.0-alpha4	Defines an Image field type. 依赖于: Content (enabled), FileField (enabled), ImageAPI (disabled)
<input checked="" type="checkbox"/>	Link	6.x-2.5	Defines simple link field types. 依赖于: Content (enabled)
<input checked="" type="checkbox"/>	Node Reference	6.x-2.1	Defines a field type for referencing one node from another. 依赖于: Content (enabled), Text (enabled), Option Widgets (enabled)
<input checked="" type="checkbox"/>	Number	6.x-2.1	Defines numeric field types. 依赖于: Content (enabled)
<input checked="" type="checkbox"/>	Option Widgets	6.x-2.1	Defines selection, check box and radio button widgets for text and numeric fields. 依赖于: Content (enabled) Node Reference (enabled), User Reference (enabled), 禁止再
<input checked="" type="checkbox"/>	Text	6.x-2.1	Defines simple text field types. 依赖于: Content (enabled) Node Reference (enabled), User Reference (enabled), 禁止再
<input checked="" type="checkbox"/>	User Reference	6.x-2.1	Defines a field type for referencing a user from a node. 依赖于: Content (enabled), Text (enabled), Option Widgets (enabled)

如图: Cck 模块组

选择自己需要的模块进行启用, 如图。

点击“内容管理”内容类型”界面, 点击添加新的内容类型, 如下图设置(可根据自己的需要进行设置):

名称:

 此内容类型的用户可读名称。此名称会显示在 建设内容页面的内容类型列表中。如果使用英文, 建议此名称用大写字母开头, 并且仅使用字母、数字和空格。这个名称不能与已有的内容类型重复。

类型:

 此内容类型的机器名称。此字符串将会用来构建 建设内容页面的内容类型的URL。此字符串应该只包含小写字母、数字和下划线。在构建 建设内容页面的URL时, 下划线“_”会被自动转换为横杠“-”。这个名称不得与其它内容类型名称重复。

描述:

关于此内容类型的简要描述。此文本将在 建设内容页面中显示为列表的一部分。

[提交表单设置](#)

[进阶设置](#)

[Image Attach settings](#)

[评论设置](#)

[保存内容类型](#) [删除内容类型](#)

如图 3

接着点击“提交表单设置”，将原来的设置替换成如图所示：



如图 4

其他选项默认即可，点击保存。这样，一个雏形的 Drupal 秀内容类型，就有了。下边接着将其设置完整吧。

点击“manage fields”，进入设置界面。其实，现在在这里只需要添加网站地址（URL）字段，其他的默认就可以了。如图设置



点击保存，进入下一界面，设置可根据自己的需求来定。这里我使用默认的即可。点击保存后，将其调整下顺序，如图所示：

Label	名称	类型	操作
✚	网站标题	Node module form.	
✚	网站地址 (URL)	field_url Link	设置 删除
✚	分类	Taxonomy module form.	
✚	网站简介	Node module form.	
✚	附件	Upload module form.	
✚	菜单设置	Menu module form.	

如图 6

Ok，到这 Drupal 秀的内容类型就建成了。只要熟悉了 CCK 功能设置，使用起来会大有帮助的。这里只是简单的介绍了下，以后会有更详细的说明。

-----END-----

drupal 的 views 学习

一：简介（这里好像是 VIEWS1 版本的）

views 的核心功能就是帮助你改变 node 的组织与显示模式。举几个可能会用到 views 的例子：

你希望首页的内容以不同的顺序显示。譬如不是按照从新到旧的顺序，而是按照“热门”程度（浏览次数，评论数目）；

你希望改变 taxonomy/term 的展现方式。当你点开某个 term 的时候，你看到的是这个 term 下内容的按照张贴顺序的列表，譬如你想改变这个列表的显示方式。默认的显示方式是“题目加摘要”，但是你希望它是以 table 的形式一目了然的显示所有必要的信息；

你想用 /tracker 来跟踪最新的帖子，但是只希望某些类型的帖子显示（譬如，你的 CMS 站是销售某种产品的，你只希望在 tracker 中显示最新添加的产品，而不包括论坛的讨论贴）

你想在区块（block）中显示某个分类下的最新的帖子；

你想为用户现实“未读的论坛帖子”；

你希望有一个类似“按月归档”的区块，可以方便访客按照时间来查询帖子；

使用了 cck + views 之后，再加上一些简单的主题与 css 的技巧，基本上没有 drupal 仿造不出来的网站，也就是说你的毛坯房也可以改造为希尔顿。

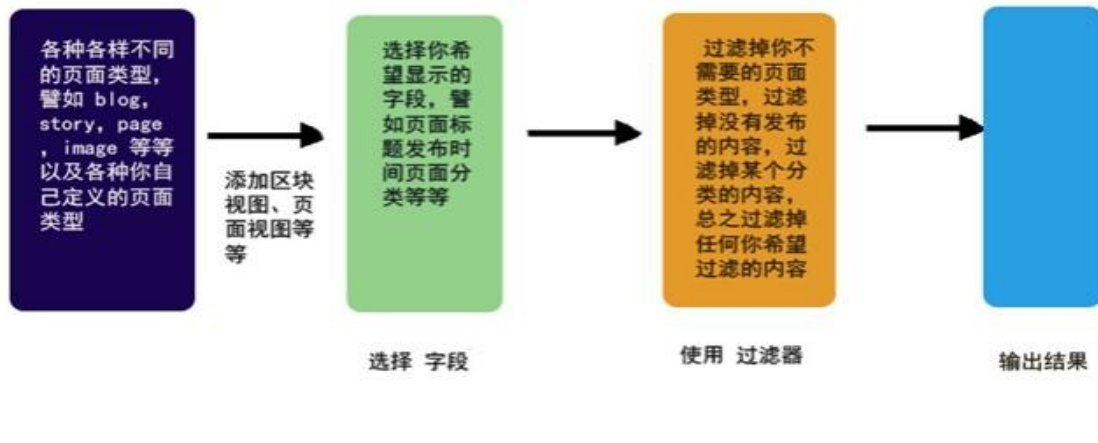
二、概述

views 模块自带了四个常用的视图，分别是：“最新评论”，“首页”，“分类词”和“最新文章”。

如果你输出的是页面，则在 URL 下显示的就是你的页面的地址，注意，它可以覆盖已有的页面，譬如例子中的 taxonomy/term/\$arg，这里的 \$arg 是一个 argument，这里我定义的是 term 的 id，这样它就可以覆盖所有 taxonomy/term 下的显示方式，类似个通配符，把诸如 taxonomy/term/1；taxonomy/term/2 等等都按照我定义的方式来显示。

三、解剖

Views 的工作流程可以用下图表示：



在明白这个流程后，让我们来解剖一个 Views 的实例，使用的样本是 Views 模块自带的 Tracker 视图。Drupal 的 Tracker 通常是 Tracker 模块提供，可以通过 yoursite.com/tracker 来跟踪网站最新的内容。但是有了 Views 模块，我们可以在不启动 Tracker 模块的情况下实现 Tracker 的效果。

在视图（教程中的 views 暂译为“视图”）的管理界面，找到 Default Views 是 tracker，title 是 Recent Post 的那个视图，点击后面的“add”，进入编辑状态。

Basic Information 项下，设置名称，访问权限和描述。名称的设置参加教程（二），访问权限和描述按照惯例。

Page 项下第一个选项 Provide Page View，当你希望输出的结果是页面是勾选这个选项；第二个选项 URL，则是设置访问这个页面的地址，理论上讲，你可以设置任何地址，但是记住，需要使用相对地址，并且首尾不要“/”。譬如你想把访问的地址定义在 <http://mysite.com/new>，那么这儿，你只需要填写 new 就可以了。这里，因为是在模仿 Tracker 的效果，所以 URL 的地址也设置为 tracker。注意，如果 mysite.com/tracker 这个地址是系统已有的（譬如你启动了

tracker 模块), 那么你定义的视图会覆盖系统已有的显示方式。另外, 这个 URL 可以使用通配符变量, 也就是 \$arg, 这个变量的定义在下面讲。再往下, View Type, 提供了五种常用的视图显示方式: 全文, 摘要, 表格显示, 列表显示, 和显示 RSS。Tracker 模块提供的默认显示方式就是 表格显示, 所以这里我们也选择 Table View。再接下来的 Title, 就是这个页面的标题, 这里我们的定义为 Recent Posts, 注意这个标题也可以使用 \$arg, 譬如你定义的是一个分类的显示页面, 你希望这个页面的标题是分类的名称。再往下的, use pager 这个选项决定是否使用分页显示, 当结果比较多的时候, 建议选择这个选项, Breadcrumb trail should not include "Home" 这个选项决定你页面左上方路径的现实方式, 是否包括 "home"; Node per page, 如果选择了分页是, 这里可以定义每页显示多少个结果; Header 和 Footer, 如果需要可以输入相应的页首和页脚信息; Empty Text 是你定义的 Views 返回的结果为 0 时显示的内容, 譬如你定义了某个 term 的视图, 但是这个 term 下没有文章, 那么你可以在这里定义返回的结果。一般使用默认就可以了。menu 选项, 第一个选项 Provide Menu 决定是否为此页面定义一个菜单, 要和紧接的 Provide Menu as Tab 配合使用。如果选择提供菜单, 但是不作为 Tab, 那么这个页面的连接就会在 Drupal 的主菜单系统中显示, 也就是在边栏里的导航菜单中显示; 如果选择作为 Tab, 这不在边栏中显示, 而在页面的上方作为标签显示。如果作为标签显示, 那么 Make Default Menu Tab 这个选项则是确定是否把现在这个视图作为这个路径地址的默认视图, 因为存在标签, 所以一个基本路径地址可能有不同页面, 譬如 tracker 模块提供的页面中, 上面有“最新文章”和“我的最新文章”两个标签, 默认的显示“最新文章”这个标签, 关于如何设置多 tab 显示的方式, 我们以后还要讲到(在我们的第一个实例里), 这里从略。Tab Weight 用来调整 tab 的顺序; Menu Title 定义这个页面在系统菜单中的名称, 如果什么都不填则默认值为页面的名称。Block 项这里什么都不用设置, 因为我们定义的是一个页面而不是一个 block。block 的定义方式和 page 差不多, 不同的是输出的结果是 block 而不是页面。

再往下就是最有趣的字段, \$arg 和 过滤器设置了。我们下次再说。

Field 项是 views 模块的最终要部分, 直接控制着输出视图的显示方式。一个典型的 tracker 页面由以下部分组成:

页面类型 标题 作者 回复数目 最新回复时间

可以参看 Drupal China 的 tracker 范例。所以, 在 field 项里, 我们需要做的就是按照需求选取字段(field)。第一个要选取的就是 页面类型。在 Add Field 菜单中找到 Node: Type 点击添加, Node: Type 就出现在 field 项下了, 然后我们要做一些设置, Label 就是这一项在列表中的显示名称, 填入“页面类型”(英文版的 views 默认的是 Type) 即可。Sortable 是设定是否按照这个字段排序, 这里选择 no, 如果选择 yes, 那么在 Default Sort 那里要选择默认的排序方式。

同样的, 我们可以确定第二个“标题”字段, 添加 Node: Title 即可, 在 Handler 设定下, 我们选择 with updated mark, 这样如果内容更新, 标题上就会显示“new”或者“新”的标示。Option 设置为“as link”, 这样就可以生成链接了, 注意, 不是每个字段(field)都会有这样的 option, 譬如 CCK 有一个辅助模块 ImageField, 在 views 中并没有 as link 的选项, 非常不方便, 必须通过 patch 才能达到目的。

添加完所有的 field, 工程就完成了大半了, 每个字段的最右边都有操作的图标, 你可以方便的上移或者下移以及删除这个字段。另外要注意的是, 这里所有的 sortable 选项全部都是 No, 这点对于网站的优化很重要, 如果某个字段允许排序, 那么访客在点击它的时候, 事实上是对整个网站的内容进行排序, 这样不仅慢而且浪费网站资源, 但是如果不让字段排序, 如何能做到显示“最新”文章呢? 这个马上就要说到。

Field 项下是 Argument 项, 在这里, 不需要什么设置。

再向下是 Filter 项, 它的作用是过滤掉你不想显示的内容, 譬如这里, 我们只希望发布的内容显示, 所以我们添加 Node: Published 在 Operator 中选择 equals (等于), value 中选择 yes。在你设计的任何试图中, 通常都要添加这个 filter, 不然没有发表的内容也会被显示出来。注意, 在 filter 的右端还有一个 expose 选项, 如果选择这个, 那么这个 filter 就会变成 exposed filter, 也就是这个 filter 是“暴露”给访客的, 访客可以使用这个 filter 来过滤内容。

Filters 项下面就是 exposed filters 项, 这里, 你可以定义任何你向让访客使用的过滤器。

最后一项是 Sort Criteria, 也就是排序标准, 在这里我们可以设置默认的排序方式(这里设置排序的好处就是可以不把某个字段的属性设置为可排序, 正如上文所述)。添加“Comment: Last Comment Date”, order 设置为降序, option 设置为 normal 即可, 其他的 option 规定了时间选择的最小单位。

OK, 到这里, 整个 tracker 部分就讲完了, 点击 mysite.com/tracker 是不是出现了最新文章列表? 呵呵, 不过仔细

一看，发现和 tracker 模块提供的 tracker 界面相比还了少些什么，对了，就是“所有最新文章”和“我的最新文章”这两个标签。我们现在用 views 做出的 tracker 只有所有的最新文章。如何在同一个页面添加不同的标签呢？下次再说。

继续我们解剖 tracker 视图的教程。我们知道 Tracker 模块提供的 tracker 视图，除了能显示所有的最新文章外，还能显示已登录用户的所有文章，这两项内容在 tracker 页面下的两个标签 (Tab) 中显示。其实，views 模块可以方便帮助我们制作标签。首先我们创建“所有最新文章”标签下的视图。这个与我们在上两个教程中讨论的 tracker 视图的设置基本相同，但是需要做一个小的更改，首先把 url 改为 tracker/all；同时，按照习惯，把这个 views 的名称命名为 tracker_all。然后在 menu 项下勾选：provide menu；provide menu as tab；make default menu tab。这是在告诉 views 模块，这个视图提供菜单，菜单输出为标签，把这个标签当作默认显示的标签。

然后我们需要创建“我的文章”这个标签，首先 url 设定为 tracker/mine（这个其实随意，你把它设定为 tracker/mypost 也可以），名称命名为 tracker_mine，同样勾选 provide menu；provide menu as tab，但是不要勾选 make default tab，因为只能有一个默认的显示。此外，如果想控制两个标签的先后顺序，可以通过 tab weight 来设定。Tab title 填写为“My Posts”。因为这个视图是只显示现在登录的这个用户的，所以我们要过滤掉其它的作者的文章，这就需要设定 filter 了，在 filter 那里新添加一个 Node: Author is Current User 把它设定为“Is Equal to “currently logged in user”即可。然后保存。工作完成。

现在通过 yoursite.com/tracker 来访问你的视图，你就可以看到两个标签页面了，呵呵，很方便吧！

区块与 argument 的使用

通过利用 views 模块制作“按月归档”的页面和区块 (block) 来讨论一下 argument 的使用。Drupal 其实有按月归档这个模块，叫做 Monthly Archive，是 Mars 做的，不过用 views 模块实现这样的功能也不难。我的 blog 的侧边栏就是一个实例，点击“blog 归档”下面的“更多”就可以进入按月归档的页面，看到所有 blog 内容的档案。

开工。在 views 的管理界面下点击“add”标签，添加新的视图。名称是 Archive，给所有的访客访问的权限，描述随意。然后“页面”项下选择 provide page view，提供页面视图，url 按照惯例是 archive。view type 随意，我这里选择的是习惯的 teaser list，标题“按月归档”。使用 pager（分页）。其它的默认即可。

然后在“区块”项下选择 provide block，提供区块。view type 考虑到是在侧边栏现实，所以选择 List view（列表显示）。区块的标题“Blog 归档”，你想继续叫“按月归档”也行。一个区块现实的页面链接数，5 个。然后选择 [more] link。

Fields 项下，只需要添加一个字段 (field)，Node: Title。Option 设为 As Link，不可 sort。这里定义的是最基层的视图的显示，譬如这个链接下的显示：<http://www.kzeng.info/archive/200704>。

过滤器项下，第一个过滤器，Node: Published 只选择发布的内容；第二个过滤器，Node: Type 只选择 Blog entry，当然你可以按照你的需要选择显示的类型。

Sort 项下选择 Node: Created Time 按照降序排列。

最后让我们来看 argument 项，这个 argument 实际上提供了按月归档的列表，是高一层的视图，这里选择 Node: Posted Month + Year，然后 summary, sorted descending，标题“按月归档”。搞定。简单的说，这个 argument 帮助生成了子页面的链接，非常的方便。

按月归档的页面可以在 yoursite.com/archive 下访问，而相应的区块需要在区块管理中启用。你可以适当的更改 CSS 来美化这个列表。

-----END-----

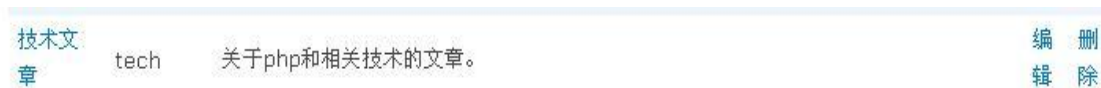
drupal 教程 views 模块教程（显示同一类型的文章）



我想点击“技术文章”的时候，把我的心得体会的列表都能显示出来。这个用到了 views 模块（这里好像是 views1）。具体的操作如下：

1、先创建一个路径为 tech 的内容类型（管理》内容管理》内容类型）。

图片效果如下：



2、在上面创建的内容类型下建立一篇文章（为了看显示效果）

3、进入 views 模块（管理》站点创建》views），选择“add”，准备添加内容

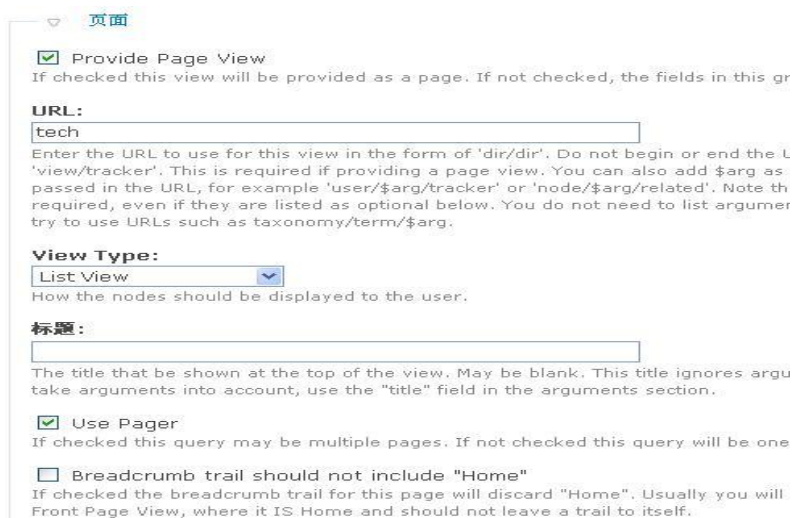
图片效果：



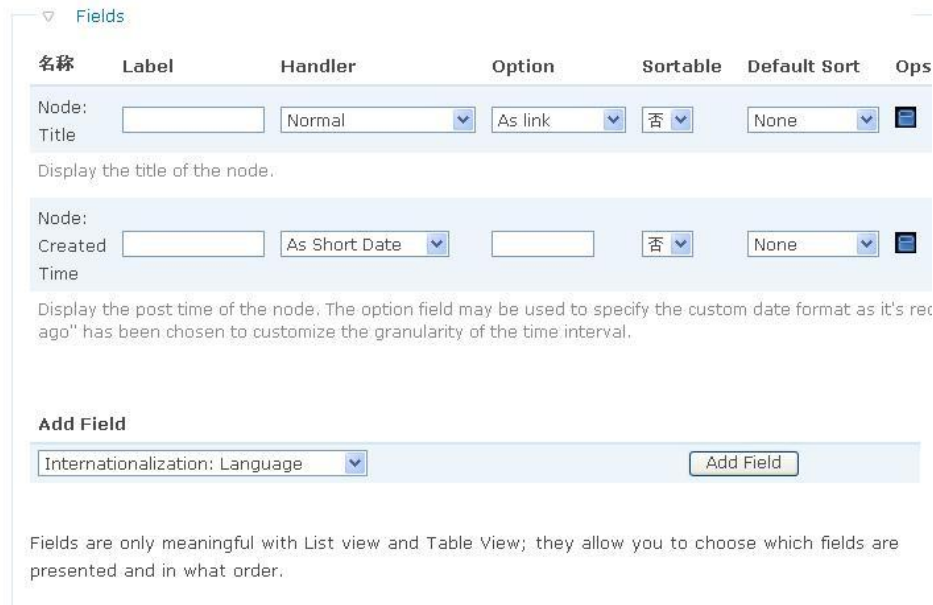
1) 在“Basic Information”选项卡先添加名称“tech”

2) 在“页面”选项卡里，选中“Provide Page View”，表示将有一个页面来显示；添加“URL”为 tech, 这是创建技术文章内容类型时设置的路径；“view type”为“list view”，列表显示页面。

效果如图：



3) 在“fields”选项卡选择显示的内容，选择“Node: Title”，展示 node 的标题



4) 在“过滤器”选项卡选择需要显示的“node:type”, 这里选择“tech”这个新创建的 node 类型



5) 在“Sort Criteria”选项卡选择需要排序的条件, 这里选择“created time”, 表示按照创建时间排序



经过以上步骤, 保存。

再点击查看“技术文章”, 就是你原来添加的那篇文章了。

=====END=====

使用 views 创建一个最新文章列表区块

使用 drupal 的 Views 模块，可以非常容易的创建一个最新文章列表区块。略加变通，你就可以创建最新评论列表区块，如果安装了评价模块(rating module)的话，你也可以按照分数高低来对节点排序。这里所讲的是用于 Drupal5.X。在 Drupal 6.X 可能会有所不同。

步骤 1 – 安装 Views

下载 Views 模块。按照指示来安装它。

步骤 2 – 区块标题

决定这个区块的名称。

步骤 3 – 节点数目

决定你想在区块中放置几个节点。你也可能需要决定要放置哪些信息。在我们这个 drupal 简明教程中，我们将仅仅列出节点标题，当然你也可以把 teaser (摘要) 也列出来。要展示的单个信息的总量将会影响到你的节点数目，如果只列出标题的话，展示的节点数目就会多一些。

步骤 4 – 开始使用 Views

登录到你 drupal 站点的后台管理界面，导航到 Site Building > Views。

步骤 5 – 区块基本信息

当你来到 Views 的管理页面以互，点击添加。然后输入基本信息-名称，描述，和访问权限。名称必须是字母数字字符，外加下划线可用，不能包含其它特殊字符。

步骤 5 – 基本的区块设置

创建区块是非常容易的。在当前页面，你往下拉，直到你看到带有超链接的"Block"。点击它就会显示出区块部分。

在区块部分的顶部，你会看到一个复选框"Provide Block"，选中它。这意味着 Views 将为你创建一个区块。

现在，我们得告诉 Views 如何展示我们的输出。你有多个选项，但是在这里，我们选用"list View"。这是由于我们仅仅想列出节点标题。如果你想列出更多信息的话，你可以使用"Teaser List"来给出标题和 teaser。所以，在"List Type"上我们将选择"List View"。

最后，你需要指出列表中节点的数目。在"Nodes per Block"中填入你想要的数字。如果你使用的是"Denver"主题的话，要将列表放在顶部区域的话，你可能选择 5 个。如果想放在左栏或者右栏的话，你是以使用 10 个或者更多。这也依赖于你内容的更新速度。

步骤 7 – 选择字段

完成了上面的步骤以后，你需要告诉 views 你想展示哪些字段。在"Block"部分下面就是"Fields"部分。点击"Fields"来打开它。然后选择你要添加的字段。然后点击"Fields"按钮。在我们的这个例子中，我们想选择节点的标题。所以我们选择"Node : Title"。如果你愿意的话，你可以为它加个标签，比如"Newest Posts" 或者"Latest News"。

步骤 8 – 过滤

你可能不想在最近发布中展示所有的东西。你可以对它进行限制。为此，你需要找到页面的"Filters"部分。点击"Filters"打开它，这样就显示出来了过滤选项。在这里，你可以选择在区块中展示节点类型。在"Add Filter"中选择"Node : Type"，然后点击"Add Filter"按钮。操作符你需要选择"Is One Of"，接下来，你需要在"Value"下拉列表中选出要用的节点类型。

步骤 9 – 保存 View。做完以后，保存你的 View。

步骤 10 – 展示区块

导航到 Site Building > Blocks。在这里你将看到所有的区块。找到你最新创建的，并为它选择要放置的区域。根据你主题的不同，你的区域可多可少。

指定了区域以后，设置重量 (weight)。重量越大，区块在区域中所处的位置越往后。

设置完重量以后，配置区块。你可以决定哪些人可以看到区块，以及区块展示在哪些页面。一个放置最新文章列表最好的地方是首页。为了实现这一点，你需要找出要这些页面，选择"Show only on selected pages"选项，然后把你的页面输入到文本框中。

当你为这个区块完成了所有的配置以后，保存区块。

注意：你可以使用这种方法以其它的方式展示你的内容。你可以展示评价最高的内容或者最近的评论。

你也可能像使用一个特定的页面,来列出你的评价最高的或者最新的 25 篇文章.有专门的模块能帮你实现这一点.你可以使用"Insert View" 或者"Viewfield"将 views 放到页面中。

***** END *****

综合首页 VIEWS 开发问题

drupal 默认的是 q=node,内容是罗列了各个 node 摘要信息。

那么比如我要定制首页, 框架如下:

area1 | area2

area3 | area4

是一个 2*2 的块, 每个块呢, 列出 5 个 node 的标题, 并且可以点击。

请问用什么方案实现比较理想?

用 views 模块可以

专门写个 page-front.tpl.php 文件,然后用区块定义

首先定义 4 个 region (区域) 。

然后用 views (相当于 sql 语句) 生产出 5 个 node 的列表的 block。

再然后在把 block(就是块), 放到那 4 个区域。

结束。

views 可以将内容做成区块显示, 但是布局要看 css 功底了

Drupal Views2 模块使用指南

View 界面说明



首先看到的是 views 列表页面, 列出所有创建的 views 规则. 安装以后, 系统已经默认创建了很多规则, 处于未启用状态, 点击右侧的 Enable 启用该 View 规则。

系统默认的有:

archive (按月的文章列表)

backlinks 返回链接

comments_recent 最新评论

frontpage

glossary 以词汇表的形式索引所有内容

popular 热门文章

taxonomy_term

tracker

利用这些默认设置的 views 规则我们已经

可以实现很多功能了, 例如热门文章, 最新评论, archive 等等. 很多人抱怨 Drupal 安装以后功能太少, 还没有 wordpress 功能多, 安装 Views 以后, 利用这些基本可以做一个大的 Blog 出来了。

创建新的 view

下面重点介绍如何创建新的 views 规则。

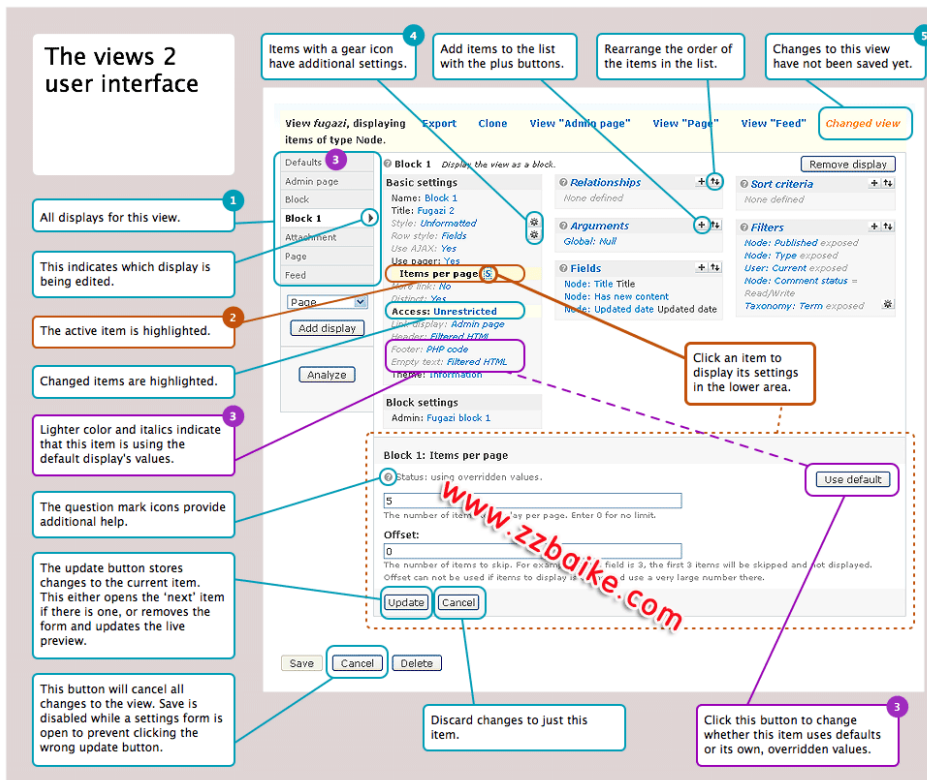
1. 点击 ADD 标签开始添加新的 views，界面如下图：



View type: 也就是在前面 views 列表的图中看到的节点 评论等，指这个 views 对应什么类型的，例如设置为评论那么调用出来的就是评论里面的内容。其实也就是设置你的这些查询获取的是什么类型的内容。

2. 设置好以后点击 next 进入 view 设置界面：

为了方便，按照官方教程的图片来说明。此图片都对每一项内容进行了编号，方便我们说明。



第一部分，defaults 显示的是此 view 的默认设置。在下拉列表中选择显示形式，然后点击 Add Display 就可以增加表现形式，也就是你的这个 view 以什么形式表现出来，可以设置多种表现形式。最常用的是页面和 block（区块）。页面就

是 view 显示在一个页面上，通过这个页面的路径就可以访问，而区块则增加一个新的区块，在区块设置中可以任意调用到任何地方。其他显示类型还有附件，feed。

右侧的查询条件设置有点乱，等我整理好再发。如果大概能看懂英语的话，可以随意编辑测试几次，很快就会明白都是什么意思。

*****END*****

首頁=Panels2+Views2

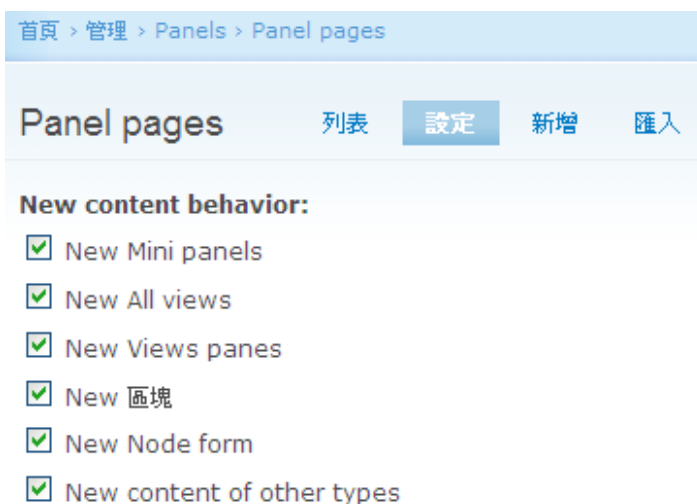
1. 首先，請您先確認一下是否已經安裝完成了 Views 2 與 Panels 2 的模組，並且已經啟動運行了。
2. 接下來就是到 **Views 的管理介面**(admin/build/views)將兩個等等我們會用到的 views 給啟用，如圖所圈出的。



3. 啟用以後，請到 Panels 的 Panel Pages 頁面(admin/panels/panel-page)



4. 在我們新增 panel 以前，我們需要先確認 Panels 模組是能夠"瞧"見我們將要添加的內容。因此請您點選畫面中上方的"設定(Settings)"按鈕。
5. 然後呢，請您勾選畫面中所有的選項，就好像...



6. 勾選完成後，請給他"儲存"起來。
7. 現在我們可以開始新增一個 panel 了，請您點擊畫面中"新增(Add)"按鈕來新增一個 panel。

8. 首先您看到的畫面是選擇佈局(layout)，請您選擇彈性佈局(Flexible)，這個佈局能讓我們靈活的修改。



9. 然後來到設定的頁面，請參考下圖的內容進行設定：
設定好後，請您點擊"Save and proceed"進行下一步。



10. 來到 panel layout 設定了。看到畫面時是不是讓那麼一堆設定給鎮住了？其實我們現在只需要修改一些地方，其他的我們暫時將他當做空氣。現在我們需要兩個欄位（一左一右），其中，左邊的欄位佔了 80%，右邊 20%。如下的設定畫面。

Rows:

The number of rows this layout can have.

Row 1

Columns:

Width 1:

Column titles, separated by commas:

11. 如上圖，我們修改了 Rows:3->1, Columns:1->2, Column titles,:Top->左欄,右欄。修改完成後存(Save)起來，我們還需要進行比例的修改。

Rows:

 The number of rows this layout can have.

Row 1

Columns: <input type="text" value="2"/>	Width 1: <input type="text" value="80"/>	Width 2: <input type="text" value="20"/>	Column titles, separated by commas: <input type="text" value="左欄, 右欄"/>
---	--	--	---

12. 如紅框框所示，我們將左右兩邊的欄位比例調為 80:20。修改完畢後，請您按下"Save and proceed"，儲存並進行下一步。
13. 現在來到了 Advanced 的設定頁面。在這裡我們只需要做兩個動作：勾選 Views 將產生「選單」(Provide Menu) 以及標籤次序(Tab Weight):-10，編輯完成我們依舊是按下"Save and proceed"，儲存並進行下一步。

選單

Views 將產生「選單」
 If checked this panel be given a menu entry in the Drupal menu system. If not checked the data in this group will be ignored.

以標籤方式使用「選單」類型 Views
 If checked this panel's menu entry will be provided as a tab rather than in the main menu system.

標籤次序:

 當設定完標籤選單時，可設定排序次序，由左到右，由小到大。

施比受更有福 It is more blessed to give than to receive
 本網站內容由 Thomas Fan 撰寫，以 創用CC 姓名標示-非商業性 2.5 中國 授權條款釋出。

14. 接下來您會到 Context 的設定頁面，因為在這個範例中我們不涉及到參數方面的設定，所以請直接按下"Save and proceed"，儲存並進行下一步。
15. 好了，現在我可以開始添加 pane 到這個 panel 中了，首先，請您先在左欄的方框中找到"+"的按鈕。

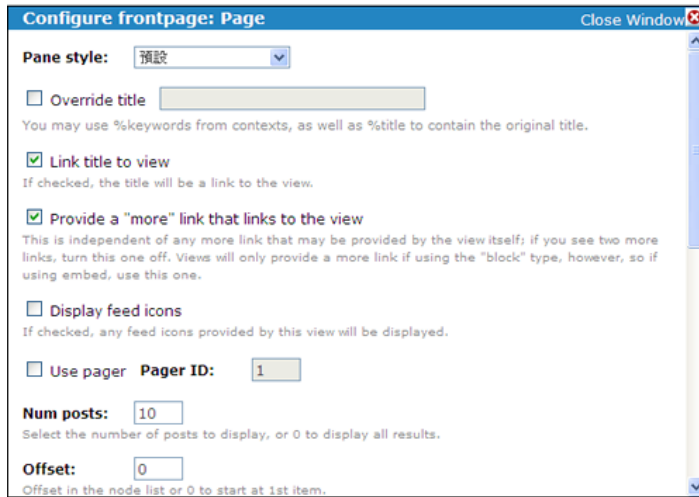
← 左欄

施比受更有福 It is more blessed to give than to receive
 本網站內容由 Thomas Fan 撰寫，以 創用CC 姓名標示-非商業性 2.5 中國 授權條款釋出。

16. 在接下來的對話框中，我們將選擇"Views 管理"下面的 frontpage page。這也就是我們剛剛啟動的那個 view。

- date_browser: Page
- drupal_links: 頁面
- drupal_links: Defaults
- frontpage: Defaults
- frontpage: Feed
- frontpage: Page
- glossary: Attachment

17. 在接下來的畫面中，我們只需要勾選-'Link title to view','Provide a "more" link that links to the view'兩選項。勾好後卷到最下面按下'Add pane'。



Configure frontpage: Page Close Window

Pane style: 預設

Override title

You may use %keywords from contexts, as well as %title to contain the original title.

Link title to view
If checked, the title will be a link to the view.

Provide a "more" link that links to the view
This is independent of any more link that may be provided by the view itself; if you see two more links, turn this one off. Views will only provide a more link if using the "block" type, however, so if using embed, use this one.

Display feed icons
If checked, any feed icons provided by this view will be displayed.

Use pager **Pager ID:**

Num posts:
Select the number of posts to display, or 0 to display all results.

Offset:
Offset in the node list or 0 to start at 1st item.

18. 左欄的 pane 我們算是添加完畢了，現在我們到右欄添加 pane,作法與左欄一樣，按下"+"按鈕來添加 pane。
19. 這次我們將要添加'新會員(Who's new)', 請如圖所示的添加。



20. 接下來的畫面我們只需要直接按下"Add pane"就行。

21. 緊接著，我們在右欄在新增一個"線上使用者(Who's online)"。



22. 然後畫面一樣我們只需要點下'Add pane'就行了。

23. 添加了兩個 pane 之後，我們在添加一個在右欄。這次我們選擇 Views 管理下面的 archive:Block



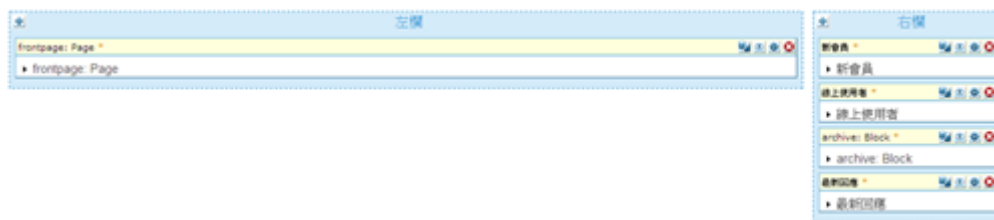
24. 然後的畫面中，請勾選 'Link title to view'後，在按下最下面的'Add pane'按鈕。



25. 最後，請再右欄添加一個 pane，這次我們選擇'最新回應(Recent comments)'。



26. 這個 pane 我們也不需要進行任何設定。按下'Add pane'的按鈕後，您將看到



27. 按下"Save"後，Panels 算是完成。

28. 因為我們打算讓他這個 Panel page 成為首頁，因此我們必須到網站設定>網站資訊 (admin/settings/site-information)頁面的最下方，將"預設的網站首頁"從"node"給為我們剛剛產生的 panel page "front"。

預設的網站首頁：*

http://localhost/drupal/ front

以此相對網址作為網站首頁來呈現內容。如果不確定的話，請填入"node"。

29. 儲存設定之後，就大功告成了。



=====END+=====

cck contemplate theming 技巧

注 1: 以下假设你已经对 cck, template 和 views 有基本使用的认识

注 2: 以下的 template 指 node-contenttype.tpl.php (或名为 content type template),

可以对特定的 content type 作版面修改, 非指 template.php

先针对 cck theming,

cck 自己只应该当作一种资料集合的储存

各种 field 模组(imagefield, link, email...) 都只为了丰富资料集合的种类