Drupal主题制作指南(v6)

译者：葛红儒，http://zhupou.cn 原文：http://drupal.org/theme-guide 欢迎来到主题制作的用户手册页面. 以下页面将涵盖隐藏在主题制作背后的核心理念,该理念适用于Drupal 6及更高版本. Drupal 5及以下版本的主题制作指南(http://drupal.org/node/509)依然可用.关于两者之间不同之处的概述,参看升级指南(http://drupal.org/update/theme). 本手册将在适当的地方说明,与以前的版本相比,所存在的一些分歧.这是从主题制作者的角度着眼的. 如果你正在开发一个模块，需要向表示层输出数据,那么请你参阅模块开发指南中的主题一节(http://drupal.org/node/165706).所有的输出都应该可以主题化的. 注意：对于文档贡献者.请不要在这里添加第3方模块或者主题相关的HOWTO用户指南.这里的信息都是关于内核的,只有在当需要的时候,我们加个链接,链到其它页面就可以了.

主题化介绍 主题化概况 • 对一个Drupal主题的剖析 • 主题的.info文件 • 覆写可主题化的输出 • 样式表 JavaScript & jQuery 区块,内容和它们的区域 • 自定义主题设置 对维护页面定制主题 • 为你的主题进行检修 主题编码习惯 主题截图指南 将你的主题添加到Drupal.org上去

主题化介绍

译者：葛红儒，http://zhupou.cn
原文：http://drupal.org/node/221881

Drupal的强大和灵活性是众所周知, 如果你想一下子就完全掌握它的话,这将是一个难于登天的事情.解决主题化问题的方式有很多,但是不是每种方式都是值得推荐的.掌握"Drupal 的方式"可以精简你的代码,从而使其更易维护. 如果您选择打破规则,走自己的路,那么首先了解"Drupal 的方式",将会增大你的成功机会。

这并不意味着, 为了制作主题,你必须全面的掌握Drupal.你只需要掌握完成任务所必需的技能就可以了,但是,你的站点的设计越复杂,你越需要了解drupal的主题机制.

本手册的目的，,在于揭示drupal主题制作(theming)的所有方面。一些地方比较难，适用于技术高手，而一些地方则比较容易，适用于初学者。在下面的部分中，我们将对内容进行展开，从每个主题开发者都应该熟悉的总体概况，到更具体的细节，有时，需要更多的技术解释。

在继续阅读本文以前，你需要了解一下的相关知识：
• 了解xHTML 和CSS
• 如果你的主题需要脚本支持的话，JavaScript 和jQuery也是必备的
• Drupal中所用到的术语（http://drupal.org/node/21951）

在某些情况下，会用到PHP的知识，但是基于纯CSS的主题，可以完全避免使用PHP。

根据你主题目标的不同，它可能非常简单，也可能非常复杂。Drupal是非常灵活的，所以你必须仔细的考虑你要做什么。你首先要考虑的是网站的需求。制作一个特定需求的主题，与制作一个通用的主题相比，要容易很多。

如果你碰到了一个难题，请阅读疑难解答（http://drupal.org/node/37156），或者在论坛的主题制作版面或者IRC @ #drupal-themes上向别人请教。请阅读“如何高效利用IRC”（http://drupal.org/node/108355）一文。

OG首页主题化

更多资源参看: http://drupal.org/node/307828

OG首页的主题化,我想修改一个OG首页的外观,但是不知道怎么实现,打算用panels,但是对这个模块还不是很熟,另外就是OG在Panels方面还不成熟.

OG首页就是一个节点类型的主题模板,因为一个小组就对应一个节点,从og\theme下面拷贝node-og-group.tpl.php到当前的主题目录下面,并不生效,因为我已经为该类型的节点创建俄一个模板,对该模板重命名,现在node-og-group.tpl.php起作用了.

其实我想要的就是控制content中的group post节点的列表，在网上找了半天，都没有找到答案，有人和我遇到了同样的问题，而且没有人解答。how to theme og home page?? http://drupal.org/node/344484 。

问题的关键点就是，这个$content把所有的东西都属出来，怎么分开显示呢？由于og首页的主题化是第一次实现，所以感觉有点神秘，因为它提供的默认模板node-og-group.tpl.php。

没有办法，只好print_r了，一个一个的找，最后找到了$node->content[view]['#value']，现在再看这个问题，其实很简单，只是刚开始自己把它想复杂化了。那就是使用group的节点类型就可以了，和普通的CCK节点类型的主题化完全一样。

相关联接：http://zhupou.cn

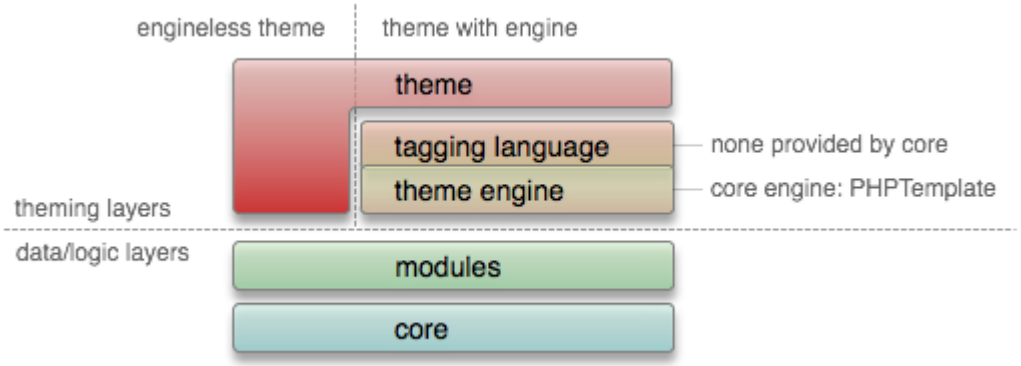
主题化概述

原文：http://drupal.org/node/171188
译者：葛红儒， http://zhupou.cn/

在软件开发中,将逻辑层和表示层分开,是很常见的.其中原因很多,最明显的原因是,后台的业务逻辑层所需要的技能,与前台表示层所需要的技能相比,有很大的不同.作为一个主题开发者,你可以在许多方面控制用到的数据,但是它局限于输出和显示.而只有Drupal内核模块和其他模块才用来负责输入.例如,一个模块可以实现一个带有默认外观的表单,来处理用户输入,并将其保存到数据库中.而Drupal主题(theme)的作用是仅仅用来重写默认外观.

在Drual中,这一抽象层是通过theme(主题)函数实现的.主题函数就是一个管道,将逻辑层与表示层连在一起.在主题引擎(theme engines)之上,有一个一个可选的中间层,用来选择标记语言比如PHPTAL或者Smarty.它还允许主题控制所有表示层的标识字体.而主题引擎像标记语言一样,都是可选的. PHPTemplate是默认的主题引擎.从名字我们就可以看出,它在xHTML中输出变量时,使用PHP作为标记语言.

从Drupal6开始，创建主题引擎的需求已被充分的淡化。



从上图可以看出，逻辑层和主题层(表示层)都可以为输出实现一个主题化的外观，（存在极个别例外情况）只有主题层才能对底层进行覆写。主题引擎（Theme engine）可以覆写内核模块和可选模块中的主题输出，而主题(图中最上方的theme一层)对下面的每一层都可以进行覆写。

注意，PHPTemplate引擎并没有覆写任何输出，但是其他主题引擎可以这样。也存在特殊情况，一个模块可以影响输出，或者对任何东西都可以覆写，但是他只用在非常特殊的情况下，而在大多数情况下，都遇不到这种情形。例如，devel的主题者模块(themer module)就属于这种特

例，它是用来帮助主题开发的。更多细节将在接下来的章节中展开。

如果你的主题是由CSS完全控制的，也就是纯CSS主题，那么这里所讲的大部分内容你都不需要关心，但是当需要修改标记文本时，理解如何找到输出的源头，对于你的工作来讲，是非常有帮助的。

• 注意，Drupal的内核和第3方模块，总是使用可主题化的函数和模板文件来输出表示层的文本的。千万不要在这些模块里面直接对主题进行修改，如果这样做的话，当你需要升级时，情况就会变得复杂起来。这样做就是所谓的“分支”("forking")。开源的强大之处在于，让开源社区来负责修正臭虫（bug）和添加新的特性。一旦你建立了一个分支，也就意味着你创建了一个封闭的系统，这样你就脱离了社区。Drupal提供了各种功能，用来覆写表示层。如果你必须要在模块中直接对主题进行修改的话，要么是你做错了，要么是你发现了一个臭虫。如果是后者的话，请填一个臭虫报告。如果提供一个补丁来修正该问题的话，那是最好不过的了。

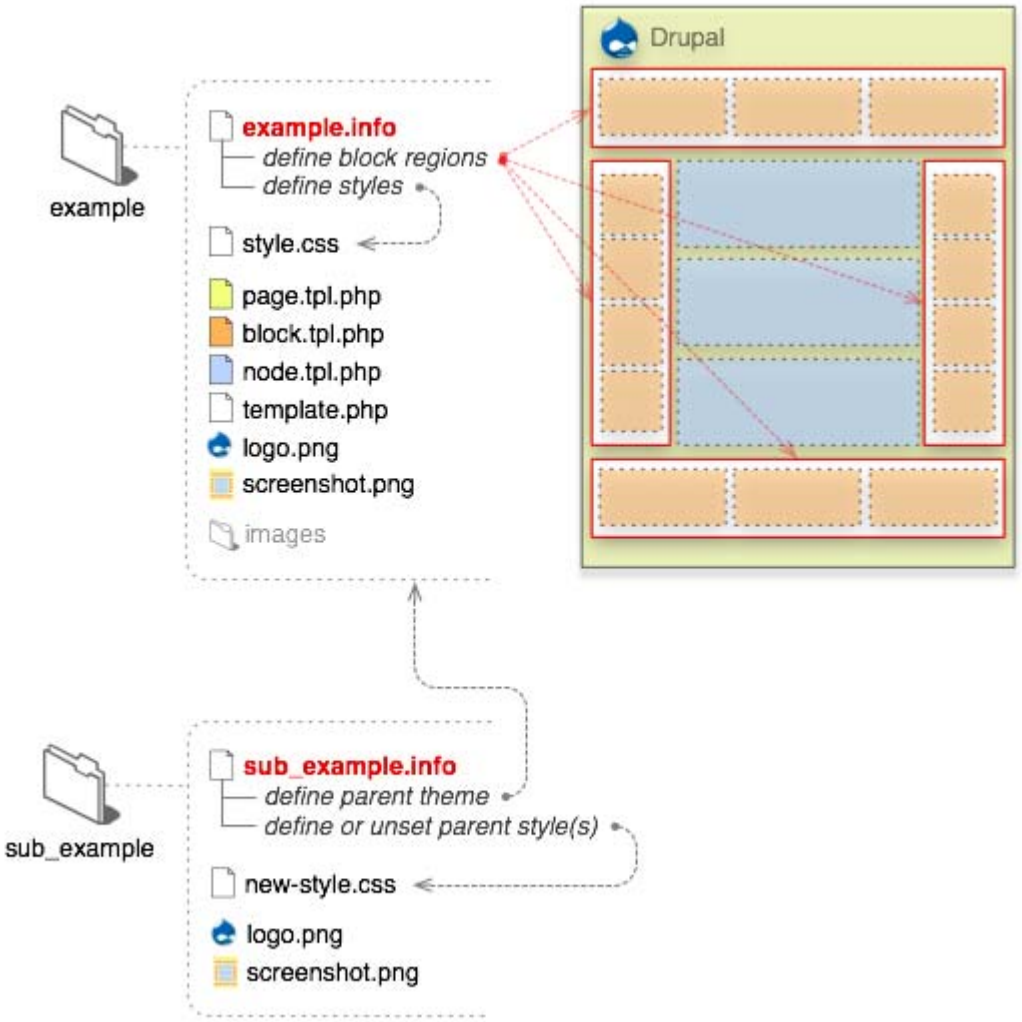
• 对于那些熟悉drupal以前版本的PHPTemplate引擎朋友来说，在Drupal6中，PHPTemplate引擎中的大部分函数都被移到了更底层的内核中。现在PHPTemplate仅仅用来 发现主题函数和该主题函数用到的模板文件。与其说它是一个主题引擎，不如说是一个主题帮助函数。PHPTemplate是最初由Adrian Rossouw为Drupal4.7编写的。Drupal6中的修改是由Earl Miles完成的。论坛中的一个帖子(http://drupal.org/node/7133)提供了最初创建引擎的原因，而在http://drupal.org/node/130987 则提供了Drupal6中的问题列表。

对一个Drupal主题的剖析

原文：http://drupal.org/node/171194

译者：葛红儒，http://zhupou.cn

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.

主题的内部名称也是由该文件衍生出来的.例如,如果它名为”drop.info”的话，那么在Drupal中，"drop"就是主题的名字。*而在Drupal5以及更早的版本中，Drupal使用主题文件夹名作为主题的名字。*

Info文件是Drupal6引入的新的特性。在Drupal5中，.info文件仅适用于模块。

模板文件(*.tpl.php*)http://drupal.org/node/190815

这些模板文件，都是由xHTML和PHP变量构成的。在一些情况下，它们也可以输出其它的数据类型——比如*xml rss*。每个.tpl.php文件负责特定一块数据的输出，有时，根据suggestions (建议),它可以处理多个.tpl.php文件。模板文件是可选的，如果在你的主题中不存在的话，那末就会使用默认输出。注意，在这些文件中，不要包含复杂的业务逻辑。一般情况下，只需要包含xHTML标签和PHP变量。在内核和可选模块所在的目录中，存在着一些模板文件。将它们拷贝到你的主题目录下面，Drupal将会使用主题目录下的模板文件来代替模块里面的。

注意：theme registry(主题注册表)对可用主题数据信息进行了缓存。当你在你的drupal主题中添加或者删除模板文件（或者主题函数）时，你需要重置主题注册表，以清空缓存。

template.php

可将输出时所用到的所有条件逻辑和数据处理，都放到template.php文件中。该文件不是必需的，但是它能使.tpl.php文件保持整洁，在这里，可以对.tpl.php文件中的PHP变量进行预处理。定制函数，覆写的主题函数或者其它对原始输出的外观定制函数，都可以放到这里。这个文件开头必须是PHP开始标签"<?php"，但是结束标签不是必须的，最好将其忽略。

Sub-themes（子主题）

在表面上，子主题和其它主题是一样的。唯一的区别是，它们继承了父主题的资源。为了创建一个子主题，在.info文件中必须包含"base theme"项。这样它就可以继承来自于父主题的资源了。继承可以是多重的；一个子主题可以是另一个子主题的父主题。对此没有进行限制，可以一直继承下去。

Drupal 5以及更低版本中，子主题需要位于父主题下面的子目录里面。Drupal6中则没有这一限制。

其它

- Logo和截图并不是必需的，但它是推荐使用的，特别是你想把你的主题贡献到Drupal资源库中时。截图应该展示主题的特色，比如包含管理页面和用户帐号设置，它可在设置了适当权限后来选择主题。
- 当你需要管理UI设置或者logo、搜索、使命（mission）等等以外的“特性”时，可以使用"theme-settings.php"文件。这是一个高级特性。更多信息，可参看用户手册的高级设置(http://drupal.org/node/177868)。
- 对于颜色模块(color module)的支持,需要一个"color"目录，里面放置"color.inc"文件，以及其它各种支持文件。

- *如果你想将你的主题基于核心主题，可以使用子主题(http://drupal.org/node/225125),或者拷贝主题并将其重命名。直接修改Garland或者Minnelli是非常非常不好的，这是由于安装和升级过程中都要用到他们。*
- 所有主题都应该放在*"sites/all/themes"目录下，以将其与核心文件区分开来。参看多站点安装(http://drupal.org/node/43816)，可以了解到所有可以放置主题的目录。*

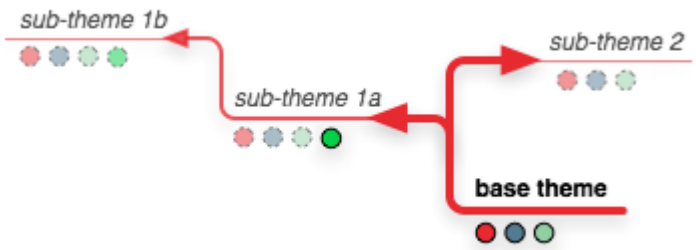
- 子主题，它们的结构和继承

子主题,它们的结构和继承

原文：http://drupal.org/node/225125

译者：葛红儒， **http://zhupou.cn**

子主题(**Sub-theme**)和其它主题一样,唯一不同之处在于:它们继承来自于父主题的资源.对于多重继承,则没有限制.一个子主题可以是另一个子主题的子主题,允许存在分支和层级结构,只要你觉得合适就可以了.这使得子主题具有巨大的潜力.



假如开始我们用基主题(base theme)勾画出主题的轮廓，那么在子主题中我们要做的就是描绘出所有的细节。接着，从同一基主题出发，创建另一个子主题分支，用来尝试另一种设计。如果你要建立的是多站点的话，但是你想要一个统一的外观，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。

主题的.info文件

原文: http://drupal.org/node/171205

译者:葛红儒 ,http://zhupou.cn

.info配置文件是在Drupal6中新赠的,每个主题都必须有一个.info文件.该文件应该放在你主题的目录下面.如果没有该文件的话,Drupal就找不到你的主题. .info文件的后缀名必须为".info".

主题在Drupal内部的名字源自于这个文件.例如,如果文件名为drop.info",那么在Drupal内部,主题名字就为"drop".名字里面不要包含奇怪的字符,这是由于在Drupal中，许多PHP函数都是以主题名打头的，所以主题名和函数名存在同样的限制。起始字符必须为alphabetic字母，不能包含空格，标点等字符。可以包含下划线，但是不能包含连字符。

数字字符也是允许的，但不能出现在首位。

注意：

- 警告！模块的内部名称如果与主题的内部名称重名的话，那么你的站点将不能工作。因为可能会造成同名函数的存在，这在*PHP中是非法的。每个安装了的部件（模块或者主题）都必须有一个唯一的名字。*
- *.info文件中的内容是缓存在数据库中的，所以对它的修改不会在Drupal中立马生效。（不要与主题注册表的缓存相混淆了。）为了清除缓存，须这样做：*
  1. 导航到*"Administer > Site configuration > Performance"，点击"clear"按钮。*
  2. 如果启用了*devel区块（安装了devel模块的话），点击"Empty cache"（“清空缓存”）链接。*
  3. 然后导航到主题选择页面*"Administer > Site building > Themes"。*

语法与INI（http://en.wikipedia.org/wiki/INI_file）文件类似。.info文件就是一个用来配置主题的的静态文本文件。文本文件的每一个行就是一个键值对（key-value），其中键位于左边，值位于右边，而中间则有一个等号。（例如：*key = value*）。分号是用来注释的。有些键使用了特殊的语法，带有中括号[],用来构建一列关联值，也就是我们常说的“数组”.如果你不熟悉数组的话，模仿Drupal默认.info文件中的例子，根据例子中的解释，完全可以依葫芦画瓢，得到自己的数组了。

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

主题的.info文件（1）

原文: http://drupal.org/node/171205

译者:葛红儒，http://zhupou.cn

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，模块和主题的.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的.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

Drupal核心主题中的.info文件例子

原文: http://drupal.org/node/171205
译者:葛红儒，http://zhupou.cn

Garland:

; $Id: garland.info,v 1.5 2007/07/01 23:27:32 goba Exp $name = Garlanddescription = Tableless, recolorable, multi-column, fluid width theme (default).version = VERSIONcore = 6.xengine = phptemplatestylesheets[all][] = style.cssstylesheets[print][] = print.css; Information added by drupal.org packaging script on 2008-02-13version = "6.0"project = "drupal"datestamp = "1202913006"
Minnelli sub-theme of Garland.:
; $Id: minnelli.info,v 1.7 2007/12/04 20:58:44 goba Exp $name = Minnellidescription = Tableless, recolorable, multi-column, fixed width theme.version = VERSIONcore = 6.xbase theme = garlandstylesheets[all][] = minnelli.css; Information added by drupal.org packaging script on 2008-02-13version = "6.0"project = "drupal"datestamp = "1202913006"
注意，从行“; Information added by drupal.org packaging script on 2008-02-13”开始直到结束，这都是由drupal.org的打包脚本自动添加的。你一定不要手工添加project(项目)和datestamp（时间戳）键。手工添加version(版本)键（第一个），那么就可以允许直接从CVS建立起来的站点使用你的主题了。

.info文件的默认值。

.info的默认值

原文：http://drupal.org/node/171206
译者：葛红儒,http://zhupou.cn

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

覆写可主题化的输出

原文：http://drupal.org/node/173880

译者：葛红儒, http://zhupou.cn,

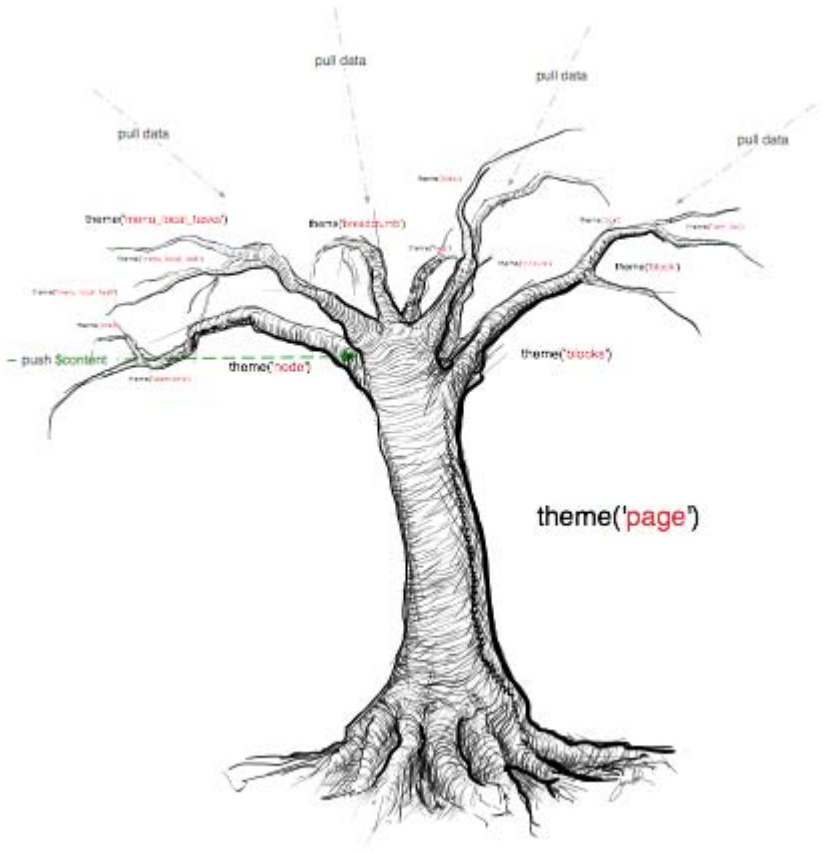
只有当你需要修改默认输出时,你才需要阅读本节.如果你的显示层是完全有CSS样式表负责的，那么可以略过本节。

覆写主题输出，需要掌握3个方面。首先，你需要知道源头在哪里，其次你要进行覆写，最后你需理解它的类型。

注意，Drupal使用主题注册表(theme registry)来缓存主题数据覆写完成后，你必须清空缓存。

1. 寻找源头：

寻找主题输出的源头，是比较困难的，这是由于主题系统的多层级结构造成的，使得源头可能出现在系统的各个地方。



仅用来演示用。

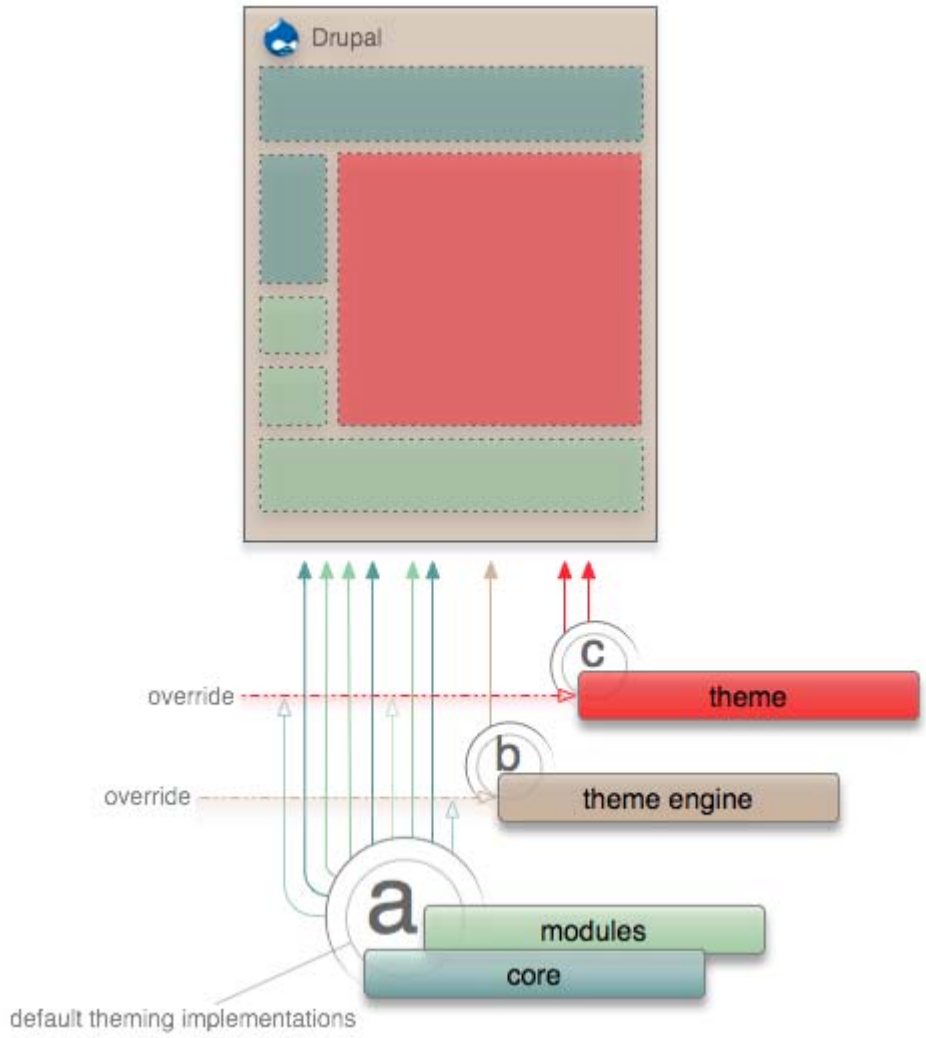
大多数页面元素一般是由theme('page')拉进来的，在显示导航信息以后，它们被放置在page.tpl.php模板中。页面元素包括主菜单、区块区域元素，区域中的区块等等。通常通过一个主题钩子来引用每个大块的主题化数据。

注意：主题钩子现在包括主题函数和模板两部分。系统中的许多钩子都与主题无关。我们这里提到的钩子都是特定于主题的。

现在使用devel模块(devel module)可以很容易的追踪特定内容的源头。devel模块包含一个主题工具，可以方便的查勘任何输出的源头，类型以及其它主题相关数据。参看示范screencast(http://drupal.org/node/209561)。由于技术方面的限制，它只能应用到Drupal6以及更高版本中。

2. 覆写机制：

覆写机制有一个特定的级联顺序，当然也包含一些特殊情况。Drupal内核和模块使用主题钩子提供了一个合理的默认输出。如果默认输出不符合主题的需求的话，那么就可以对其进行覆写，从而避免使用默认输出。这样，默认的将被放到一边去，我们不再管它，而将所有的特定修改放到所用的主题目录下面。不要直接修改Drupal内核和模块，只在主题目录下面修改。



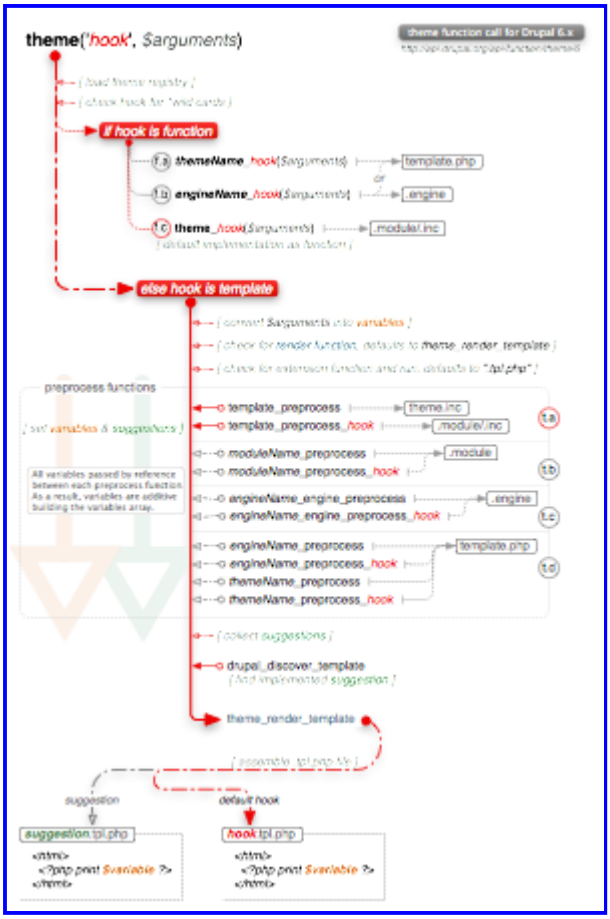
如果传统的覆写不能满足你的需要的话，你可以使用主题注册表。

注意：尽管在Drupal 6中PHPTemplate.engine还存在，但它已不再覆写主题函数了。在Drupal5中，它允许使用模板来处理部分主题钩子。而现在则不必要这样了。

函数VS模板

3.函数VS模板：

正如前面所说，实现特定的钩子有两种方式。通常的“函数”或者“模板”。根据要主题化的元素的特点，选用最合适的方式。内核和模块可选用任何一种方式来构建输出。而上面的主题层，可以使用同样的方式来覆写，或者改变选用的方式。



Links to PDF. Flow map for 5 also available for comparison.

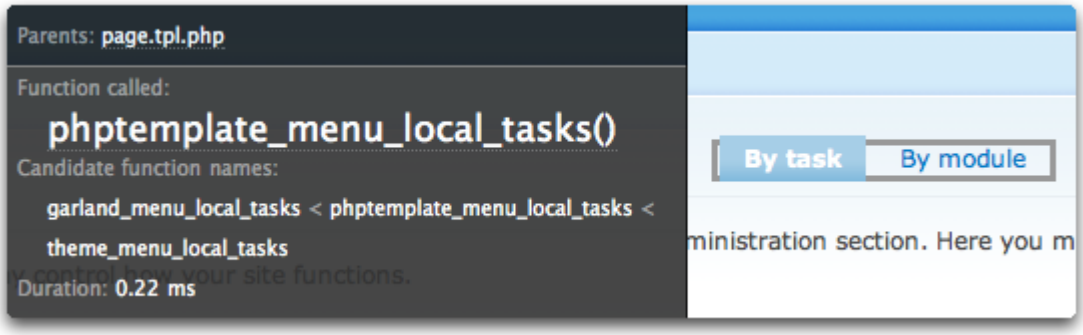
使用函数来实现主题钩子，具有速度上的优势。通常比使用模板的快5倍，但是对们对于涉及者来说，就比较困难，通常涉及者都比较熟悉直接的xHTML，而不是php函数。通常，我们不需要考虑这样一点，而是根据钩子的特点和在一个页面中被调用的次数，来决定要选择的方式。

注意：在Drupal5及以前版本，内核和模块只能使用函数来实现主题钩子。想要使用模板的话，必须使用PHPTemplate来覆写钩子，并在引擎层中进行转换。

下面是覆写的两个例子，这里使用了devel themer.

函数方式：

主题函数theme_menu_local_tasks，是一个用来输出一级和二级标签的简单函数。这里的主题钩子就是"menu_local_tasks"。为了覆写它，可将函数名中的前缀"theme"替换为你主题的名字或者所用主题引擎的名字。最好使用主题的名字，这样可以避免潜在的与子主题（http://drupal.org/node/225125）的命名冲突。



本例中，Garland是使用引擎名来覆写的。如果你的主题是基于Garland的子主题，那么你必须使用你的主题名了。

将下面的代码放到主题的template.php，清空主题注册表缓存，这样就覆写了默认输出。注意，将"drop"改为你主题的名字。

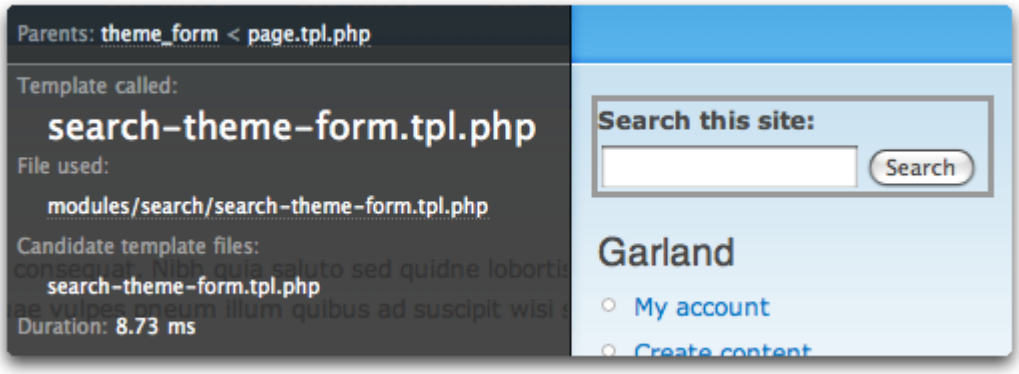
```
<?php
function drop_menu_local_tasks() {
  $output = '';

  if ($primary = menu_primary_local_tasks()) {
    $output .= "<ol class=\"tabs primary\">\n". $primary ."</ol>\n";
  }
  if ($secondary = menu_secondary_local_tasks()) {
    $output .= "<ol class=\"tabs secondary\">\n". $secondary ."</ol>\n";
  }

  return $output;
}
?>
```

这里所做的唯一修改就是将<ul>标签改为了<ol>。

在api.drupal.org你可以找到所有的主题函数。

模板方式：

如果默认是使用模板实现的，那么你只需要简单得将模板源文件拷贝到主题下面，然后清空主题注册表，就完成了覆写。下面为search-theme-form.tpl.php的一个例子。注意在这里，主题钩子就是"search_theme_form"，需要将连字符“-”替换为下划线“_”.



这就是你要做的。使用编辑器，打开拷贝的模板，对其进行修改。内核中的所有.tpl.php文件都带有注释。根据注释，你就可以做出具体的修改了。

注意：模板可以放在主题下面的任何目录中。这样便于管理，也避免了主题根目录下面的混乱。

相关页面：

- 为了定制模板中的变量，参看自页面“预处理函数”（http://drupal.org/node/223430）。
- 可以在子页面“核心模板和建议”（Core templates and suggestions）找到所有的主题模板。

将函数转化为模板

将一个主题函数转化为一个模板，开始是需要一点工作的，但一旦完成，便很好使用。如果你和设计者一同工作，那么转化为模板，将会使设计者更专注于设计，而不是编码。在内核中，已有了很多模板，而在将来的版本中，将有更多的主题函数转化为模板。第3方模块，为了与内核接轨，最好也使用模板。本部分是为那些还没有使用模板的主题钩子准备的。

Drupal以模板的方式识别主题钩子是自动完成的。下面是完成修改所需的所有必要条件：

- 模板名必须与主题钩子匹配。
- 主题钩子中的下划线必须改为连字符。
- 模板文件必须使用扩展名".tpl.php"。（根据主题引擎的不同，扩展名会有所不同）

假如主题函数为theme_user_signature。这里的主题钩子就是"user_signature"。创建一个名为"user-signature.tpl.php"的文件，清空注册表，就会告诉Drupal现在钩子已改为模板方式了。现在该文件中的内容将替代相应的函数。这里的难点是，设置模板文件中用到的变量，这可以通过预处理函数来完成。

需要注意的一些点：

- 可以在模板中直接放置代码，但这种方式不好。所有的复杂逻辑都应该从.tpl.php文件中分离出来。这样使得模板文件更干净，更便于管理。
- 这也有安全方面的考虑。将逻辑与显示分离，可以避免潜在的恶意用户来创建内容，从而减小CSS(cross-site scripting)攻击的机会.当让你的设计者处理模板文价时，所有的输出都应该非常干净，这样设计者就不用考虑安全问题了。
- 比较关于论坛的theming functions in 5和template conversions in 6。你可以使用这个作为例子，来进行两种方式之间的转换。
- 更多关于预处理函数的信息（http://drupal.org/node/223430）。

主题注册表

原文：http://drupal.org/node/173880
译者:葛红儒， http://zhupou.cn,
Drupal的主体注册表维护了主题钩子相关的缓存数据，包含主题钩子和如何处理它们的信息。
对于大多数主题开发者来说，都不需要直接与注册表打交道。只需要记住，当添加或者删除主题函数和模板时，要清空它。编辑已有的函数和模板时，则不需要清空。
清空主体注册表，有3方式：

1. 位于"Administer > Site configuration > Performance"的clear按钮。
2. 如果启用了devel区块(devel模块创建的)，点击"Empty cache"连接。
3. 使用API函数drupal_rebuild_theme_registry。

主题注册表是主题钩子相关信息的缓存数据，包括Drupal可用的主体钩子，勾子类型，即如何处理它们。在以前的版本中，所有的主题调用都是直接完成的。由于在底层需要进行大量的处理工作，而缓存可以加快这种处理，特别是对于模板而言。你主题用到的引擎应该为你自动的注册所有的主题钩子。
在一些特殊情况下，你需要直接与注册表打交道。如果你的主题需要注册一个新的钩子，而该钩子不在底层中（内核，模块，引擎）。比如说一些表单，内核或者模块没有明确对其主体化，而仅仅使用了默认的表单输出。

- 更多细节，参看子页面“特殊情况下的主题注册表”（The theme registry for special cases）。
- 不要将主题注册表与主题的.info文件混淆了，两者都被缓存了。清空主题注册表的第1点

和第2点，同时也清空了.info的缓存。

- 预处理函数
- 默认的基本变量
- 特殊情况下的主题注册表
- 使用模板建议(suggestions)
- 核心模板和建议(suggestions)

预处理函数

预处理函数仅适用于模板形式的主题钩子.它的主要作用是设置模板文件((.tpl.php)中所用到的变量。在预处理器(Preprocessor)中，一般涉及不到普通的主题函数。

注意：

- 在提供模板建议（template suggestions）时也会用到预处理器。
- 在drupal5中，函数_phptemplate_variables提供了同样的功能，在drupal6中，为了以后版本的兼容性，最好不要用这个函数。

对于单个主题钩子，可以有多个预处理器。内核，模块，引擎，主题，每层都可以有一个预处理器，来逐步的构建显示在模板文件中的变量集。通过将大部分逻辑放到这些预处理器中，可使得模板文件更加简洁，易于使用。

下面是预期的预处理器。当它们同时存在时，按照下面的顺序运行：

1. template_preprocess
   -这个是由内核提供的，也是始终存在的。这里声称的变量在所有的模板钩子中都可以使用.
2. template_preprocess_hook
   -实现了主题钩子的内核或者模块提供该处理器。特定于某个钩子的变量，通常首先在这里生成。
3. moduleName_preprocess
   -不要将这个与前面的预处理器混淆了。对于那些最初没有实现钩子的模块，它允许影响变量集。它将在所有的钩子中运行。
4. moduleName_preprocess_hook
   -和第3个一样，但是特定于某个钩子。
5. engineName_engine_preprocess
   -主题引擎的预处理器。适用于所有的钩子。
6. engineName_engine_preprocess_hook
   -主题引擎的另一个预处理器，特定于单个钩子。
7. engineName_preprocess
   -这是第一个可以在主题内部使用的预处理器。命名方式为，主题所用引擎名称+预处理器名。适用于所有的钩子。
8. engineName_preprocess_hook
   -这个和第7个一样，但是特定于单个钩子。
9. themeName_preprocess
   -命名方式为：主题名+预处理器名。适用于所有的钩子。
10. themeName_preprocess_hook
    -与前者一样，但是特定于单个钩子。

这里有多种方式可修改变量集。在大多数情况下，只有前两个预处理器存在。第一个，添加了所有的默认基本变量，而第2个添加了特定于该主题钩子的变量。第3方模块，如果用到了第3和第4个预处理器的话，需要添加注释对其进行详细说明。这里就不对此展开讨论了。

尽管可以这样做，但是默认的PHPTemplate没有对变量集进行修改。(5 & 6)

从列表中的第7个开始，所有的预处理器都是放置在主题中的。这个预处理器列表最多是可以超过10个的，那就是使用子主题，子主题是基于第9和第10个预处理的前缀主题名的，但是这种情况在实际中很少用到。

注意：

- 一般推荐在基主题的预处理器中使用引擎名称(7 & 8)。这有利于代码在主题之间的迁移，同时有利于在Drupal.org上发布代码片断。
- 而只有在子主题(sub-themes)中才使用主题名称(9 & 10)。这将减少潜在的重名冲突，在PHP中是不允许重名的。
- 为了识别你主题的预处理器，与钩子相关联的模板必须位于主题内部。如果存在默认的模板的话，将其拷贝到你的主题下面，并清空注册表。如果你要将一个主题函数转化为一个模板，参看前面页面的“以模板方式注册钩子”(http://drupal.org/node/173880#convert-type)。

注意，这些函数中都没有返回值，所有的变量都是通过引用传递的，前面都有符号“&”,比如&$var。

由于这里适用的是引用方式，所以在前面设置的变量，在后面的预处理器中都会存在，所以你一定要小心，不要在这里出什么乱子。重置以前的变量是可以的，但重置以后，你总会疑神疑鬼，感情哪里会出漏子。

这个例子，来自于实现了钩子"foo"的模块：

```
<?php
function template_preprocess_foo(&$variables) {
  $variables['foo_list'] = array(
    'list item 1',
    'list item 2',
    'list item 3',
  );
}
?>
```

在主题的预处理器中添加变量集：

```
<?php
function drop_preprocess_foo(&$variables) {
  // Do not do this unless you mean to:
  $variables['foo_list'] = array('list item 4');
```

```
  // Instead do this:
  $variables['foo_list'][] = 'list item 4';
}
?>
```

在模板文件中使用的变量，就是$variables的键。所以，在上面的例子中，在模板文件中可用的变量就是$foo_list。

原文：http://drupal.org/node/223430

译者：葛红儒, http://zhupou.cn,

默认的基本变量

原文：http://drupal.org/theme-guide
译者：葛红儒，http://zhupou.cn，
下面是在所有的模板文件(http://drupal.org/node/190815)中都可以使用的基本变量。它们是通过预处理器函数(http://drupal.org/node/223430)，template_preprocess生成的。特定于模板文件的变量，其相关说明位于模板文件中。

$id
模板的编码。模板每被调用一次，它增1。
$zebra
"odd" 或者 "even"。两者随着模板的适用交替改变。
$directory
主题的相对路径，相对于安装路径。例如："sites/all/themes/myTheme"

$is_admin
布尔值。当访问者为站点管理员（user 1）时返回TRUE;
$is_front
布尔值。当当前页面为首页时，返回TRUE;
$logged_in
布尔值。访问者为站点会员，登陆并通过验证时，返回TRUE;
$db_is_active
布尔值。当数据库可用时，返回TRUE。这只在“维护模式下的主题化”(http://drupal.org/node/195435)中有用,此时站点可能会遇到数据库问题。
$user
当前访问者的用户对象。把数据放到这里可能不大安全。对于可疑字符串，一定要用check_plain。

特殊情况下的主题注册表

在继续阅读本节以前,你首先要熟悉主题注册表的目的(http://drupal.org/node/173880#theme-registry)。这里的指导，将覆盖如何手动的注册一个主题钩子，并解释如何才能手动操作。

大部分需要手动注册主题的情景，是与表单相关联的。表单元素可以主题化，但是它们的

主题化是以另一种方式进行的。对于基本元素，比如复选框，单选框，提交按钮，下拉菜单，等等。这些元素都是可以主题化的，对它们的覆写不需要手动的去注册与之相关联的钩子。对于一些定制表单，每个元素都以非常特别的方式来放置，这时就需要手动的去注册了。对于那些，已经设计好，主体化过的，并且注册过的表单，就不需要手动注册了。对于那些没有主题化的表单将使用默认方式显示它们(http://api.drupal.org/api/file/developer/topics/forms_api.html/6)。

表单注册例子：

在下面这个例子中，search.module注册了两个搜索表单，搜索框和搜索区块。每个表单都有一个唯一的Id与之相关联。即可充当注册编号，又可充当主题钩子。在这里，它是"search_theme_form" 和"search_block_form"。

```
<?php
function search_theme() {
  return array(
    'search_theme_form' => array(
      'arguments' => array('form' => NULL),
      'template' => 'search-theme-form',
    ),
    'search_block_form' => array(
      'arguments' => array('form' => NULL),
      'template' => 'search-block-form',
    ),
    ...
  );
}
?>
```

表单API将它的显示控制权交给了注册钩子的处理器。在这个例子中，它注册了默认参数('arguments')和模板类型（'template'）。只有当主题层(see image)下面的层次中已注册了主题钩子时，Drupal才能自动找到钩子。

注册一个未被注册的表单：

另一个搜索表单还没有被注册过。它的id为"search_form"，用在主搜索页面。表单的数据是在一个函数中构建的(http://api.drupal.org/api/function/search_form/6)，这和其它表单一样，但它需要表单API根据它的数据结构来处理它的显示。我们可通过覆写来对其进行扩展，你需要在你的主题中使用hook_theme对其进行注册。将下面的代码放到你主题的template.php文件中，将"drop"前缀改为你主题的名字。主题钩子就是表单的id：

```
<?php
function drop_theme() {
  return array(
    'search_form' => array(
      'arguments' => array('form' => NULL),
    ),
  );
}
?>
```

主题化过的表单都有一个参数"form"。由于这里没有声明模板(template),那么钩子将作为主题函数的形式存在，而不是一个模板。主题函数必须使用它的主题名作为前缀。这里不能使用phptemplate_*。所以，根据上面的信息，那么主题函数应该是这样的：

```
<?php
function drop_search_form($form) {
  $simple = '';
  foreach (element_children($form) as $element) {
    if ($element == 'advanced') {
      $advanced = drupal_render($form[$element]);
```

```
    }
    else {
      $simple .= drupal_render($form[$element]);
    }
  }
  return $advanced . $simple;
}
?>
```

这里所做的唯一修改就是高级搜索表单的位置。

- 子主题中覆写了一个表单，如果这个表单在基表单中已经注册过的话，那么就不需要手动的再为这个表单进行注册了。记住，底层注册过了的，上层都不需要再注册了，对于基主题来说，子主题就是它上面的一层。
- 在将来的版本中，这将会更加自动化。开发者知道主题者(themers)的负担。

手动操作

主题注册表的手动操作，使一个高级特性。如果你安装了devel模块的话，点击区块中的"Theme registry"(主题注册表)可阅读进一步的详细信息。也可参看http://api.drupal.org/api/function/theme_get_registry/6。

待续。。。

原文：http://drupal.org/node/223463
译者：葛红儒，http://zhupou.cn,

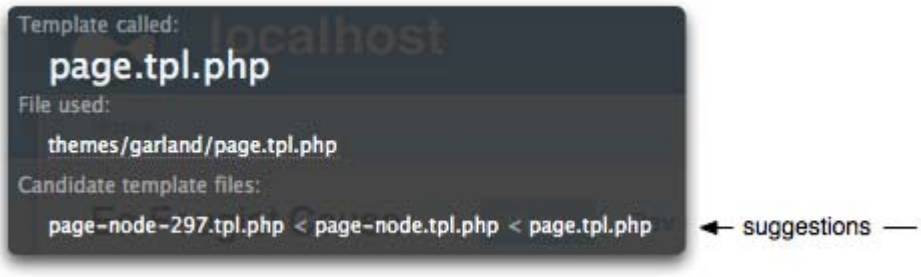
使用模板建议(template suggestions)

原文：http://drupal.org/node/223440

译者：葛红儒， http://zhupou.cn,

模板建议是基于已有.tpl.php文件的可选模板文件。当满足特定的条件，并且相应的文件存在时，就使用这些建议。每一层次，包括内核、模块、主题引擎、主题，都可以提供相应的建议。你可以把它们当作“命名提示”(naming hints),来告诉系统根据合适的环境选择合适的模板。这种想法很简单，但是这一特性却非常强大，它使得在模板层上也能进行定制。

Devel模块展示了“页面”模板的可用的模板建议。



内核的所有建议的列表可参看“内核模板和建议”。

这些命名建议是在预处理函数中设置的。内核已经提供了大量的建议。如果你需要进一步的扩展它，你需要在你的主题下面的template.php文件中为相应的钩子添加一个预处理器。本例中，我们为"page"主题钩子添加了扩展建议。这里的钩子可以为任何模板钩子。

前缀"drop"应改为你主题的名字。

```
<?php
function drop_preprocess_page(&$variables) {
  global $user;

  // Add a single suggestion.
  if (module_invoke('throttle', 'status') && isset($user->roles[1])) {
    $variables['template_file'] = 'page-busy';
  }

  // Add multiple suggestions.
  if (!empty($user->roles)) {
    foreach ($user->roles as $role) {
      $filter = '![^abcdefghijklmnopqrstuvwxyz0-9-_]+!s';
      $string_clean = preg_replace($filter, '-', drupal_strtolower($role));
      $variables['template_files'][] =  'page-'. $string_clean;
    }
  }

}
?>
```

有两种方式添加这些建议。

1. 1,键'template_file'接收一个单独的建议，并具有优先级。如果条件满足，并且文件存在的话，将会使用这个建议，忽略其它的建议。
2. 2,键'template_files' (复数)可以接受一个建议数组。它们按照先进后出的顺序进行处理。所以，首先应该向里面添加最一般的，然后逐步添加相对特殊一些的，这样就可以根据特殊性选择相应的建议了。

在上面的例子中，当达到了节流上限时，并且访问用户为匿名时，Drupal将尝试使用名为"page-busy.tpl.php"的文件。而其它代码，则告诉Drupal根据当前用户角色选择相应的模板，比如"page-authenticated-user.tpl.php"。如果找不到的话，就使用基模板"page.tpl.php"。这是非常简单的例子。你可以根据你可用的数据来自己设置上下文。

一些注意点：

- 当向'template_files'也就是数组中添加建议时。不要对其进行重置，因为变量是使用饮用传递的。如果重置的话，所有在内核和模块中进行的设置将全部丢失。

```
<?php
  // Do not do this:
  $variables['template_files'] = array('hook-suggestion');

  // Instead do this:
  $variables['template_files'][] = 'hook-suggestion';
?>
```

- 建议前面使用与之相连的钩子作为前缀。这样更加干净，并将文件聚到了一块。同时还减

小了Drupal将该模板注册为其它钩子的风险。
- 使用连字符来代替下划线。主模板都不使用下划线的。
- 建议只有和基模板放在一块时，才起作用。模板可以放在主题下面的恶任意子目录下，但是必须将其放到一块。
- 建议的修改添加删除，不需要清空主题注册表。只有基模板注册时才需要。而Drupal可以自动识别建议的。

核心的模板和建议

原文：http://drupal.org/node/190815
译者：葛红儒， http://zhupou.cn,

默认模板:
这些为内核提供的默认模板(.tpl.php)文件.模板文件内部包含了相应变量和模板用途的文档.这有对于所有模板都可用的默认变量集.
由PHPTemplate在5.x中处理的模板,也被删除了. PHPTemplate不再处理模板文件.
为了覆写这些模板,你只需要将其拷贝到你的主体目录下,并清空主题注册表.

Aggregator(聚合器)
"modules/aggregator/..."
- aggregator-feed-source.tpl.php
- aggregator-item.tpl.php
- aggregator-summary-item.tpl.php
- aggregator-summary-items.tpl.php
- aggregator-wrapper.tpl.php

Block (区块)
"modules/system/..."
- block.tpl.php

"modules/block/..."
- block-admin-display-form.tpl.php

Book (书)
"modules/book/..."
- book-all-books-block.tpl.php
- book-export-html.tpl.php
- book-navigation.tpl.php
- book-node-export-html.tpl.php

Comment (评论)
"modules/comment/..."
- comment-folded.tpl.php
- comment-wrapper.tpl.php
- comment.tpl.php

Forum (论坛)
"modules/forum/..."
- forum-icon.tpl.php
- forum-list.tpl.php
- forum-submitted.tpl.php
- forum-topic-list.tpl.php

- forum-topic-navigation.tpl.php
- forums.tpl.php

Node (节点)
"modules/node/..."
- modules/node/node.tpl.php

Poll (投票)
"modules/poll/..."
- poll-bar-block.tpl.php
- poll-bar.tpl.php
- poll-results-block.tpl.php
- poll-results.tpl.php
- poll-vote.tpl.php

Profile (外形)
"modules/profile/..."
- profile-block.tpl.php
- profile-listing.tpl.php
- profile-wrapper.tpl.php

Search (搜索)
"modules/search/..."
- search-block-form.tpl.php
- search-result.tpl.php
- search-results.tpl.php
- search-theme-form.tpl.php

User (用户)
"modules/user/..."
- user-picture.tpl.php
- user-profile-category.tpl.php
- user-profile-item.tpl.php
- user-profile.tpl.php

System (系统)
"modules/system/..."
- page.tpl.php
- maintenance-page.tpl.php
- box.tpl.php

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.

下面列出的为默认建议.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-wrapper-country.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。

原文：http://drupal.org/node/190815
译者：葛红儒， http://zhupou.cn,

样式表

在下面我们将讨论Drupal是如何通过.info文件来处理样式表的.而在子页面中,我们将讨论更高级的功能,比如通过API添加样式表的功能.完全基于CSS的主题,其样式表的相关信息也可以放在这里.

有几点需要注意。每一个核心组建或者模块都将提供一个默认输出。包括文本标记和相应的样式表。(更多解释，可参看文本标记的覆写)。 由于Drupal的易扩展性，对于主题设计者来说，处理浏览器端的一切，将会是一个极大的负担。这些默认的输出，将会根据主题设计者的判断，作出修改。与主题函数和模板的覆写一样，内核和模块提供的样式表也可以被覆写。不要直接修改。所有的修改都应该放在你主题目录里。

注意：

- 注意，修改样式表时，首先要确保仅用了CSS优化。它位于"Administer > Site configuration > Performance"。如果启用了的话，你将看不到你对CSS所做的修改，除非你清空聚合到一块的样式表。修改完以后，你可以在启用它。
- .info文件被缓存了。所以样式表的添加或者删除都不会显示出来，直到你将缓存清空。（不要将它与主题注册表混淆了。）为了清空缓存，你可以这样：
  1. 点击位于"Administer > Site configuration > Performance"的清空(Clear)按钮
  2. 如果启用了devel区块的话(devel模块生成的)，点击链接"Empty cache"。
  3. 简单的访问主题选择页面"Administer > Site building > Themes"。

添加样式表：

在默认情况下，如果你主题的.info文件中没有定义样式表的话，将会使用"style.css"文件。添加其它的样式表非常简单，只需要定义一个新的'stylesheets'键，媒体属性(media property)和样式表的名称。注意，定了了定制的样式表后，就不再加载默认的"style.css"了。如果你的主体需要使用它的话，你需要在info文件中明确声明它。

```
; Add a stylesheet for all media
stylesheets[all][] = theStyle.css

; Add a stylesheets for screen and projector media
stylesheets[screen, projector][] = theScreenProjectorStyle.css

; Add a stylesheet for print media
stylesheets[print][] = thePrintStyle.css
```

一些注意点：

- 注意[]的使用，在[media]和“= styleName.css”之间，使用了[]。
- 这里定义的顺序，就是样式表在页首出现的顺序。
- 样式表可以出现在子目录下面，比如stylesheets[all][] = stylesheets/styleName.css.这样易于管理样式表。

覆写内核和模块的样式表:

为了覆写内核或者模块的样式表,可以在主题的.info文件中对其进行重定义。以"system-menus.css"为例。它位于"modules/system/system-menus.css"。使用下面的代码，将会忽略这个样式表，取而代之的是主题下面的拷贝。

```
stylesheets[all][] = system-menus.css
```

当覆写模块的样式表时，应该将样式表文件放在你的主题下面，路经和文件名一定要匹配。这样，就使用主题提供的样式表来替换默认的样式表了。

如果添加的样式表，在主题内部不存在的话，将会忽略内核或者模块的样式表。这个特性是在Drupal6.0中设计的，在Drupal 6.3中已被纠正。

一些注意点：

- 覆写一个核心CSS文件，将会阻止加载默认的"style.css"文件。当你需要这些默认文件时，一定要明确的对其进行定义。
- 覆写的样式表与原始样式表在媒体(media)类型上必须匹配。
- 样式表内部的所有链接元素必须正确。你需要多检查一下'url()'属性或者'@import'规则，以确保里面的资源确实存在。
- 样式表在页面头部的出现顺序将被修改。修改以后，将会产生相应的级联效应。
- 一些内核和模块的样式表是根据条件自动加载的。使用.info文件覆写将会使得这些文件将被永远调用。
- 如果在主题中的修改不是很重要的话，可以考虑使用CSS选择器来覆写样式表，而不是覆写整个文件。

覆写基主题的样式表：

下面的内容适用于子主题。为了在子主题中禁用基主题的一个样式表，你可以在.info文件中重定义样式表。这与覆写模块或者内核样式表的方式相同。

基主题和子主题必须包含这一项：

```
stylesheets[all][] = masterStyle.css
```

如果文件存在于子主题中的话，就使用这个文件，而如果忽略该文件的话，将会阻止加载它。

原文：http://drupal.org/node/171209

译者：葛红儒， http://zhupou.cn,

通过API添加样式表

通过.info文件添加样式表,对于大多数主题来说,已经足够了.由于.info文件是静态的，所以不能动态的添加样式表。依据主体是如何处理样式表的，将它们放到一块也是可以的。当你有所疑虑的时候，使用.info文件就可以了。

有两个API函数可用来处理样式表，drupal_add_css 和drupal_get_css。下面是一个动态添加样式表的例子。
将前缀"drop"改为你的主题名。

```
<?php
function drop_preprocess_page(&$variables) {
  $front_style = path_to_theme() .'/front-page.css';
  $path_style = path_to_theme() .'/path-'. arg(0) .'.css';

  if (file_exists($front_style) && $variables['is_front']) {
    $include_style = $front_style;
  }
  elseif (file_exists($path_style)) {
    $include_style = $path_style;
  }

  if (isset($include_style)) {
    drupal_add_css($include_style, 'theme', 'all', FALSE);
    $variables['styles'] = drupal_get_css();
  }
}
?>
```

在上面的例子中，访问首页，将会加载样式表"front-page.css"，而访问其它页面，则会根据内部路径的不同加载其它的样式表。例如，对于页面，http://example.com/admin，将会使用"path-admin.css"。

一些注意点：

- 根据样式表加载的时间和地点，将会调用drupal_get_css来加载要添加的样式表。在模板预处理页面template_preprocess_page）首先对其进行初始回显。关于预处理器顺序的具体细节，可参看预处理器和变量。

- 在drupal_add_css存在一个参数，用来聚合添加的文件。当要添加的样式是非常动态的话，可以考虑像前面的例子那样，禁用该参数，这是由于向一个大的聚合文件中添加一个比较小的文件需要重新创建一个新的聚合CSS文件。生效后，加载页面的速度将会变慢，并且耗费更多的带宽。

原文：http://drupal.org/node/225868
译者：葛红儒， http://zhupou.cn

覆写内核样式表

原文：http://drupal.org/node/263967
译者：葛红儒， http://zhupou.cn,
默认情况下, Drupal安装后,各个部分是带有样式的. 尽管默认样式非常方便,但是当你使用一个主题时,你经常会根据你的设计来修改这些设置.
(Please fill this in as much as possible.)
(请往这里添加更多的相关内容,越多越好)

- 定制ul列表样式

定制ul list-style

难易程度: 中到高级.

例1:基本
在这个例子中,我们将创建一个无序列表(ul),它将在特定区域展示一个光盘,跟着是一列项目,而在网站的其它部分将会隐藏这些信息.接着你可以决定,在你的网站中,你要为哪些列表添加样式.

注意:
注意,你一定要在测试站点进行试验,而对于要用到的文件要有备份.
我们将使用一个div,它的名字为"content".你的div的名称可能有所不同.在div中,我们放置ul列表,我们将对其添加样式:

```
<div id="content">
<ul class="unordered_list_in_content">
<li>list item 1</li>
<li>list item 2</li>
<li>list item 3</li>
</ul>
</div>
```

在你的样式表中你将使用:

```
ul { list-style-type:none } /* This will turn off all list-style-types in your
theme */
#content ul.unordened_list_in_content { list-style-type: disc }
```

现在,你应该可以在你选择的区域看到带有光盘的列表,而不是在网站的其它部分.

例2:让我们添加一些有趣的列表样式
在一个主题中关闭所有的list-style,使用:
ul {list-style:none}

代替
```
ul {list-style-type:none}
```

你可以这样:
```
ul {list-style:disc inside}
```
代替
```
ul {list-style-type:disc}
```

这样你最终的样式表将会是:

```
ul {list-style:none}
#content ul.unordered_list_in_content { list-style: disc inside}
```

注意:如果在你的主题中,有多个样式的话,你需要确保没有list-styles来覆写你的CSS,否则它还是显示不出来.

注意: list-style-type已被禁用,更多细节参看http://www.w3.org/TR/html401/struct/lists.html.

提示:这也适用于节点内部的ul,比如一个page 或者story.

原文：**http://drupal.org/node/263979**
译者：葛红儒， http://zhupou.cn

支持RTL语言

添加对RTL(右到左)语言的支持,涉及到覆写横向的样式,可以通过级联和基于相对的样式表进行命名来实现.对RTL样式表的包含是自动完成的.基于站点语言的设置来决定是否包含RTL样式表.

例如,在核心主题Garland中, "style.css"是主样式表.而对于从右到左的语言,比如阿拉伯语或希伯来语,它将会包含"style-rtl.css".对两个样式表的加载次序是,首先加载主样式,然后加载RTL样式.这允许后者对前者的覆盖,从而在主样式中,不用担心RTL样式表中所需要考虑的特殊性.

这里有一个编码标准,用来管理规则.依赖于横向定位或者纬度的规则,应该带有注释 /*LTR */以指示该属性是特定于从左到右布局的.这包括floats, margins, padding,等等.内置文本应该自动浮动,而主题则通过"page.tpl.php"模板设置文档的语言.

例如 基样式:
```
ul.primary-links {
  margin-top: 1em;
  padding: 0 1em 0 0; /* LTR */
  float: left; /* LTR */
  position: relative;
}
```

相应的RTL样式:
```
ul.primary-links {
  padding: 0 0 0 1em;
  float: right;
}
```

由于和主CSS文件一起使用,这使得可以非常容易的指出在RTL样式中,哪些地方需要修改.
注意,如果你的主体覆写一个模块的样式,那么与之关联的RTL样式将被忽略,除非你明确的

对其进行声明.

原文：http://drupal.org/node/222782
译者：葛红儒， http://zhupou.cn，

JavaScript & jQuery

jQuery基本

Drupal 6.0到6.2包含的是jQuery 1.2.3。在Drupal 6.3中，已升级到jQuery 1.2.6，到目前为止，这是最新版本了。如果你的站点需要最新的jQuery版本的话，那么你可以使用jQuery升级模块。

默认JavaScript文件

与style.css类似，现在也存在一个能被自动加载到主题中的JavaScript文件，即为script.js。这个文件应该放在主题的根目录下面。

JavaScript主题化

对于JavaScript代码，现在有一个主题化机制。与自动加载的script.js一起，它使得主题开发者对于Drupal站点上的脚本事件拥有更多的控制权。通常，人们的JavaScript代码生成标记文本(markup)，并将其插入到页面中。然而，在脚本中，可能包含一些应编码进去的HTML，而这些硬编码是不允许被修改的。

模块在Drupal.theme.prototype的命名空间下提供了默认的主题函数。例如，如果我想创建自己的主题函数 powered(为了展示"powered by Drupal"图标)，它应该看起来是这样的

```
Drupal.theme.prototype.powered = function(color, height, width) {
  return '<img src="/misc/powered-'+ color +'-'+ height +'x'+ width +'.png" />";
}
```

我调用它的方式是

```
$('.footer').append(Drupal.theme('powered', 'black', 135, 42));
```

这将在页脚添加一个按钮。

如果我想在我的主题中覆写这个函数，可以使用我主题目录下面的文件来代替/misc，我可以这样：

```
Drupal.theme.prototype.powered = function(color, height, width) {
  return '<img src="/sites/all/themes/powered_images/powered-'+ color +'-'+ height
+'x'+ width +'.png" />";
}
```

这个利用了普通JavaScript的优点。前面的函数放在主题里面的一个JavaScript文件中，从而避免修改相关模块的JavaScript文件。

JavaScript主题函数的返回类型可以多种多样。既可以是简单的字符串，也可以是复杂的数据类型，比如对象，数组，和jQuery元素。参考原始的函数，看它返回什么，在你的定制主题函

数中就应该返回什么。

仍然需要的：jQuery介绍，如何添加聚合(http://drupal.org/node/119441的补丁)等等。

原文：http://drupal.org/node/171213,

译者：葛红儒，http://zhupou.cn,

区块,内容和它们的区域

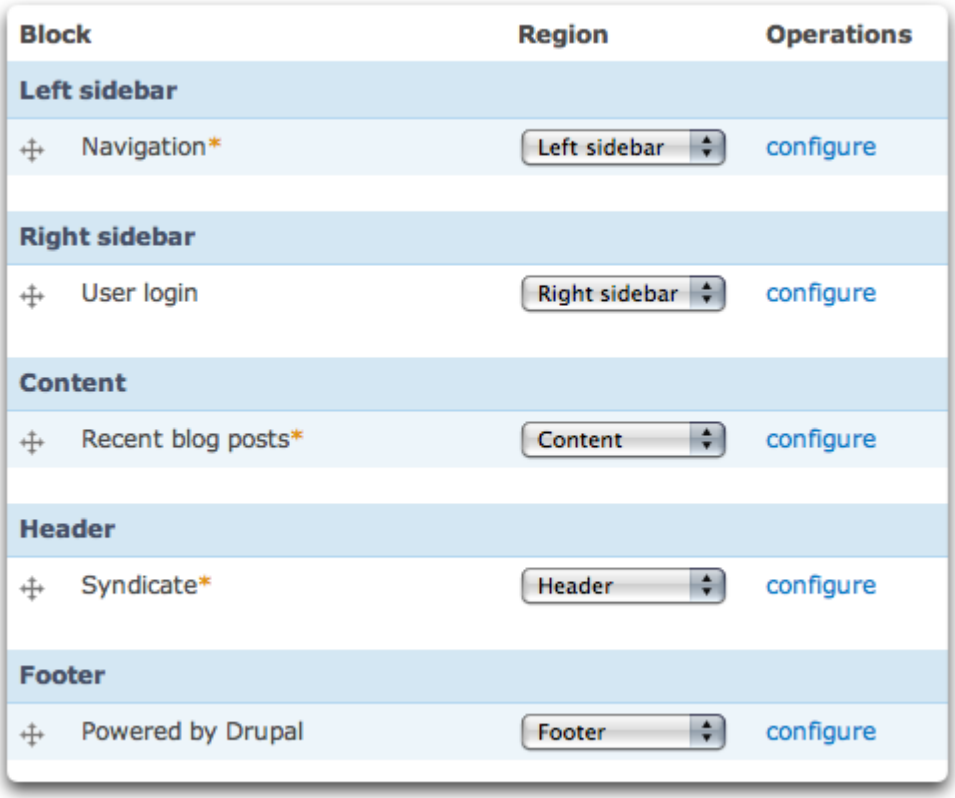
主题中可用的区块区域是在.info文件中定义的。定义方式为键'regions'+[],[]之间为内部名称，再加“=”，最后跟上外部名称，例如regions[theRegion] = The region label。如果没有定义的话，将会使用下面的默认值。

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

记住，内部名称将自动转化为"page.tpl.php"模板中的区域变量。在前面的例子中，[left]区域将通过$left变量输出所有指定给该区域的区块。PHP变量的命名存在一些限制，所以区域的内部名称也要遵守相同的约束。它只能包含字母数字字符，以及下划线，注意数字不能放 在开始位置。

方括号外面的外部名称，是用在区块管理面的，用来标记区域名，"Administer > Site building > Blocks"。

下面是Garland的区块管理页面 ：



一些注意点：

- *每个单独区块的显示都可以使用模板文件。*
- *添加一个定制区域，就会阻止使用默认的. 如果你想继续保留这些默认的区域，你需要手工的将其添加进去.*
- *区域的定义顺序将反映到区块配置页面. 例如Garland使用的为默认区域. 注意图中区域的顺序.*
- *. info文件中的内容被缓存了。所以对它的修改都不会显示出来。（不要将它与主题注册表混淆了。）为了清空缓存，你可以这样：*
  1. *点击位于"Administer > Site configuration > Performance"的清空(Clear)按钮*
  2. *如果启用了devel区块的话(devel模块生成的)，点击链接"Empty cache"。*
  3. *简单的访问主题选择页面"Administer > Site building > Themes"*

升级提示：

- *在Drupal 5以及更低版本中，是使用ThemeName_regions() 或EngineName_regions()声明区域的. 而在Drupal6中，这两个方法已不再被推荐使用.*
- *如果你要将你的主题从以前的版本升级到Drupal6，其中用到$sidebar_left和$sidebar_right的话，需要将其改名为$left and $right.*
- *在以前的版本中， $footer_message区域变量是将页脚区域(footer region)和页脚消息(footer message)混到了一起. (在"Administer > Site configuration > Site information"中设置). 而在Drupal6中，这两个变量是分开的，所以你需要单独创建一个$footer变量.*

原文：http://drupal.org/node/171224

译者：葛红儒， http://zhupou.cn

手工的将内容指派给特定区域

使用drupal_set_content可以将内容手工的添加到区域中. 例如，drupal_set_content('header', 'Welcome!')将文本'Welcome!'添加到页首区域. 下面是一个更

有用的例子,构建所有评论的总结并将其放到"right"区域.
将前缀"drop"改为你主题的名字.更多信息参看预处理器

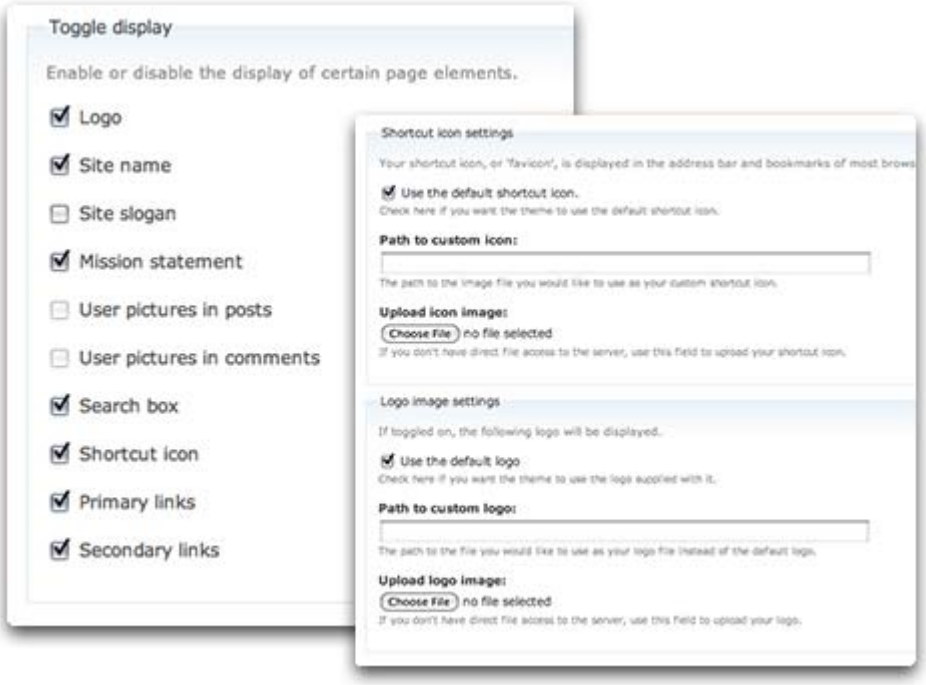
```
<?php
function drop_preprocess_comment(&$variables) {

  // Setup a few variables.
  $comment = $variables['comment'];
  $title = l(
    $comment->subject,
    comment_node_url(),
    array('fragment' => "comment-$comment->cid")
  );
  $new_marker = $comment->new ? t('new') : '';
  $by_line = t('by') .' '. theme('username', $comment);

  // Form the markup.
  $summary = '<div class="comment-sidebar">';
  $summary .= "<span class=\"title\">$title $new_marker</span>";
  $summary .= "<span class=\"credit\">$by_line</span>";
  $summary .= '</div>';

  // Set the comment into the right region.
  drupal_set_content('right', $summary);
}
?>
```

注意通过这个函数设置内容,发生在区块区域回显以前,它是这样调用的
template_preprocess_page > theme_blocks > drupal_get_content.

原文：**http://drupal.org/node/171224**
译者：葛红儒， http://zhupou.cn

定制主题设置

在主题配置页面,由主题输出的各种页面元素都可以被启用或者禁用.配置页面位于"Administer > Site building > Themes > themeName".例如,在该页面上,取消对"Site slogan"的选中,就可以禁用掉站点标语(slogan)了.



这些复选框,是根据.info文件中的特性(features)生成的.声明方式为'features'+“[]”+“=”+特性本身，例如features[] = the_feature。如果一个也没有定义的话，将会使用下面的默认值

```
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
```

为了禁用一些特性，只需要在.info文件中添加你需要的特性就可以了。定义需要的，就会将不需要的忽略掉。有些特性也会启用一些相关的表单字段。例如，'logo'将启用一个上传字段，用于上传图片。

一些注意点：

- .info文件中的内容被缓存了。所以对它的修改都不会显示出来。（不要将它与主题注册表混淆了。）为了清空缓存，你可以这样：
  - 点击位于"Administer > Site configuration > Performance"的清空(Clear)按钮
  - 如果启用了devel区块的话(devel模块生成的)，点击链接"Empty cache"。
  - 简单的访问主题选择页面"Administer > Site building > Themes"。

- 已不再支持hook_features()

- 高级主题设置
- 集成颜色模块

原文：http://drupal.org/node/221905

译者：葛红儒，http://zhupou.cn,

高级drupal主题设置

在Drupal的后台，每个主题都拥有一个配置页面，位于admin/build/themes/settings/themeName。在该页面中有一个表单，里面包含一些标准设置，比如“Logo image settings”和“Shortcut icon settings.”。

现在，在Drupal 6中，主题作者可以通过向该表单添加额外的设置来定制这个页面。在Drupal 5中，主题作者和主题用户首先需要安装主题设置API模块（5.x-2.1或者最新版本），这样才能使用下面介绍的方法。

为你的自定义主题设置添加表单小部件（widgets）

首先，在你的主体目录下面，创建一个theme-settings.php文件，并添加一个themeName_settings() 或者themeEngineName_settings()函数。最好是用themeEngineName_settings()表单，这样他人会更容易的基于你的主题创建子主题。在函数中，应该使用表单API来创建要添加的表单小部件。

例如：为了向Garland主题添加设置，需要在主题的theme-settings.php文件中添加garland_settings() 或者phptemplate_settings()函数。

如果一个用户以前保存过主题设置表单，那么保存过的值将通过$saved_settings参数传递给这个函数。对于添加到表单的小部件，其返回类型应该为表单API数组。

下面例子中的注释给出了更详细的解释：

```
<?php
// An example themes/garland/theme-settings.php file.

/**
* Implementation of THEMEHOOK_settings() function.
*
* @param $saved_settings
*   array An array of saved settings for this theme.
* @return
*   array A form array.
*/
function phptemplate_settings($saved_settings) {
  /*
   * The default values for the theme variables. Make sure $defaults exactly
   * matches the $defaults in the template.php file.
   */
  $defaults = array(
    'garland_happy' => 1,
    'garland_shoes' => 0,
  );

  // Merge the saved variables and their default values
  $settings = array_merge($defaults, $saved_settings);

  // Create the form widgets using Forms API
```

```
  $form['garland_happy'] = array(
    '#type' => 'checkbox',
    '#title' => t('Get happy'),
    '#default_value' => $settings['garland_happy'],
  );
  $form['garland_shoes'] = array(
    '#type' => 'checkbox',
    '#title' => t('Use ruby red slippers'),
    '#default_value' => $settings['garland_shoes'],
  );

  // Return the additional form widgets
  return $form;
}
?>
```

注意，主题作者可以使用高级表单API（auto-completion，可伸缩字段集）和JQuery javascript来创建复杂的动态的表单。

在你的主题文件中获取关于设置的值

为了在主题的template.php 或者.tpl.php文件中取回设置的值，可以简单的使用theme_get_setting('varname')。具体细节参看Drupal API：http://api.drupal.org/api/6/function/theme_get_setting。

例如：

```
<?php
$happy = theme_get_setting('garland_happy');
?>
```

原文：http://drupal.org/node/177868

译者:葛红儒，http://zhupou.cn,

初始化默认值

初始化默认值

由于我们不能保证用户是否会访问页面admin/build/themes/settings/themeName，所以我们需要为我们的自定义设置初始化默认值。

只有当我们提交过admin/build/themes/settings/themeName后，主题设置变量才被保存起来，所以在我们的template.php文件中，我们需要检查变量的设置情况。如果没被设置，我们需要将其设置为默认值。我们是这样做的，首先取出一个变量，查看它是不是为null，如果是的话，我们使用variable_set()来保存默认值，接着使用theme_get_setting('', TRUE)在Drupal内部强制刷新设置。

在你的template.php文件的顶部附近添加下面的代码：

```
<?php
/*
* Initialize theme settings
*/
if (is_null(theme_get_setting('garland_happy'))) {  // <-- change this line
```

```
  global $theme_key;

  /*
   * The default values for the theme variables. Make sure $defaults exactly
   * matches the $defaults in the theme-settings.php file.
   */
  $defaults = array(             // <-- change this array
    'garland_happy' => 1,
    'garland_shoes' => 0,
  );

  // Get default theme settings.
  $settings = theme_get_settings($theme_key);
  // Don't save the toggle_node_info_ variables.
  if (module_exists('node')) {
    foreach (node_get_types() as $type => $name) {
      unset($settings['toggle_node_info_' . $type]);
    }
  }
  // Save default theme settings.
  variable_set(
    str_replace('/', '_', 'theme_'. $theme_key .'_settings'),
    array_merge($defaults, $settings)
  );
  // Force refresh of Drupal internals.
  theme_get_setting('', TRUE);
}
?>
```

注意，在前面代码中第一行的变量“garland_happy”，应被替换为你自定义主题设置中的变量名，而$defaults数据也应该是从你的theme-settings.php文件中拷贝过来的。

为新版本主题添加额外的设置

当你的主题发布了1.0版本以后，对于2.0版本，你将想要添加一些额外的自定义设置。这一过程非常直接。这里你需要注意的是第3步中的代码初始化。

1. 在你的theme-settings.php文件中，向变量$defaults 和$form添加新的设置。
2. 在你的template.php文件中，在初始化主题设置代码中，向变量$defaults中添加设置。
3. 在你的template.php文件中，修改初始化代码，从你新添设置中任选一个，查看它是否存在。例如，如果你添加了多个设置，包括一个garland_slippers设置，你应该将初始化主题设置代码的第一行改为：
   ```
   if (is_null(theme_get_setting('garland_slippers'))) {
   ```
4.
5. 这样就能将新加的自定义设置的默认值，添加到已保存的自定义设置的值中了。

原文：http://drupal.org/node/177868
译者:葛红儒， http://zhupou.cn,

集成颜色(color)模块

Color.module允许管理对一个主题的颜色方案进行全面的修改.通过选择一个5色的调色板(从一个颜色集中选出或者手工给出),你可以修改整个主题的颜色.该模块可以修改样式表和重新显示的图片.然而,主题必须为此提供特定的钩子,而所创建的设计必须能够适应这一模块.本文档揭示了创建一个多颜色主题的基本知识.

设计

由于color.module的工作方式,并不是每个设计都能使用color.module.

我们在基本设计中放置一个透明的图片,在基本设计中,东西都有了,就缺背景颜色了.接着,我们将这个图片放到一个彩色背景上面,从而得到一个彩色的版本.最终,我们将这个合成的图片切成更小的图片然后将它们分别保存到图片文件中.

我们根据你的定义,还对样式表进行了处理,还修改了所有的颜色.该模块以调色板作为参考, 平滑的对颜色作出了修改.对于没有出现在调色板中的颜色,那么就为其选出一个最接近它的调色板颜色(无论它是一个链接,文本或者背景颜色).

所以,在photoshop的设计模式中,一定包含一个多层的文件,而在层次堆栈的最底部包含一个或者多个背景颜色层,而其它层则放在了背景颜色层的上面.当你保存基本图片时,你需要把所有的层合并到一起,使得背景颜色层融合到其中.以Garland的base.png文件为例(使用图片编辑器打开它以查看其透明性).这里有一个视频,讲述了如何使用Photoshop创建你自己的base.png文件.

在这一过程中生成的所有文件都将被写入到/files/css中,以取代默认图片.这意味着如果没有color.module的话,彩色主题仍然可以工作,不过使用的是默认颜色方案(color scheme).

原文: http://drupal.org/node/108459
译者: 葛红儒, http://zhupou.cn

集成color模块实践

让我们以Garland为例.最重要的文件位于themes/garland/color子目录中:

base.png

它包含了主题的基本设计,它是合成的,并将被切分成多个图片.

color.inc

这个文件包含了用来对主题着色的所有参数.参看下面.

preview.css

这个样式表是用来在颜色修改器上生成预览的.

preview.png

这个图片是用来在颜色修改器上生成预览的.

color/color.inc的使用,使得颜色挑选器出现在主题的设置中.它是一个规范的PHP文件,其中包含一个$info数组,其值如下:

Schemes(方案)

```
<?php
array('schemes' => array(
    '#0072b9,#027ac6,#2385c2,#5ab5ee,#494949' => t('Blue Lagoon (Default)'),
    '#464849,#2f416f,#2a2b2d,#5d6779,#494949' => t('Ash'),
    '#55c0e2,#000000,#085360,#007e94,#696969' => t('Aquamarine'),
    '#d5b048,#6c420e,#331900,#971702,#494949' => t('Belgian Chocolate'),
    '#3f3f3f,#336699,#6598cb,#6598cb,#000000' => t('Bluemarine'),
    '#d0cb9a,#917803,#efde01,#e6fb2d,#494949' => t('Citrus Blast'),
    '#0f005c,#434f8c,#4d91ff,#1a1575,#000000' => t('Cold Day'),
    '#c9c497,#0c7a00,#03961e,#7be000,#494949' => t('Greenbeam'),
    '#ffe23d,#a9290a,#fc6d1d,#a30f42,#494949' => t('Mediterrano'),
    '#788597,#3f728d,#a9adbc,#d4d4d4,#707070' => t('Mercury'),
    '#5b5fa9,#5b5faa,#0a2352,#9fa8d5,#494949' => t('Nocturnal'),
    '#7db323,#6a9915,#b5d52a,#7db323,#191a19' => t('Olivia'),
```

```
    '#12020b,#1b1a13,#f391c6,#f41063,#898080' => t('Pink Plastic'),
    '#b7a0ba,#c70000,#a1443a,#f21107,#515d52' => t('Shiny Tomato'),
    '#18583d,#1b5f42,#34775a,#52bf90,#2d2d2d' => t('Teal Top'),
  ));
?>
```

这个条目包含了一个简单的数组,里面为预定义的配色方案.每个条目必须有5个颜色,一个标题,其格式如上所示.

第一个方案是作为参考用的,必须与主题的默认图片和样式表紧密匹配.否则,最终的颜色可能就是不是用户想要的了.关于颜色是如何计算的,更多信息可参看'stylesheets'(“样式表”)一节.

要拷贝的图片

```
<?php
  array('copy' => array(
    'images/menu-collapsed.gif',
    'images/menu-expanded.gif',
    'images/menu-leaf.gif',
  ));
?>
```

这个数组包含了一列不能修改的图片.它们将和生成的图片和样式表放置在一起.

填充区和梯度

为了对图片着色,我们创建一个和基图片一样大小的目标图片,并且画出彩色区域和一个梯度.我们可以使用(x, y, width, height)声明区域的坐标,从而定义它们的位置,这样做非常灵活.

```
<?php
  array('gradient' => array(0, 37, 760, 121));
?>
```

You can specify one vertical two-color gradient.

```
<?php
  array('fill' => array(
    'base' => array(0, 0, 760, 568),
    'link' => array(107, 533, 41, 23),
  ));
?>
```

你可以为每个调色板颜色声明相应的区域.而在该区域中就会使用选择的颜色进行填充.可用颜色有'base'(基本), 'link'(链接), 'top'(顶部), 'bottom' (底部)和'text'(文本).

切图

接着,你要进行切图了,你需要在基图片上定义相应的区域.这里,你还需要使用(x, y, width,

height)作为坐标,以及在样式表中所用到图片的名字. The logo and screenshot slices are special and always take the same filename.标志和截图(logo和screenshot)有些特殊，它们采用一贯的文件名。截图需要被重新调整大小，调整为150x90像素。

```
<?php
  array('slices' => array(
    'images/body.png'                          => array(0, 37, 1, 280),
    'images/bg-bar.png'                        => array(202, 530, 76, 14),
    'images/bg-bar-white.png'                  => array(202, 506, 76, 14),
    'images/bg-tab.png'                        => array(107, 533, 41, 23),
    'images/bg-navigation.png'                 => array(0, 0, 7, 37),
    'images/bg-content-left.png'               => array(40, 117, 50, 352),
    'images/bg-content-right.png'              => array(510, 117, 50, 352),
    'images/bg-content.png'                    => array(299, 117, 7, 200),
    'images/bg-navigation-item.png'            => array(32, 37, 17, 12),
    'images/bg-navigation-item-hover.png'      => array(54, 37, 17, 12),
    'images/gradient-inner.png'                => array(646, 307, 112, 42),

    'logo.png'                                 => array(622, 51, 64, 73),
    'screenshot.png'                           => array(0, 37, 400, 240),
  ));
?>
```

文件

最后，你需要为你的主题声明文件所在的位置。你需要为预览准备一个图片和样式表，还需要一个基图片。

```
<?php
array(
  'preview_image' => 'color/preview.png',
  'preview_css' => 'color/preview.css',
  'base_image' => 'color/base.png',
);
?>
```

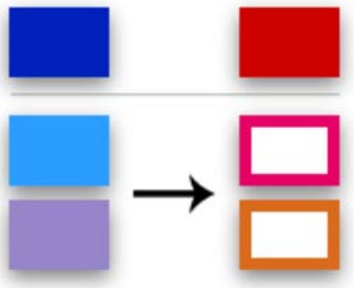
* 在Drupal6中，Color.module不再需要基图片了，这意味着不使用图片也可以使用该模块。

样式表

color.module将会读入一个主题的style.css文件，以及使用@import语句引入的其它样式，并创建一个新的style.css文件。它将使用一个调色板颜色作为参考，根据上下文，修改CSS中的颜色。

- **Links（链接）**：使用'link'颜色，根据规则它适用于一个元素。
- **Text（文本）**：使用'text'颜色，根据规则它适用于color: styles中。
- **Base（基本）**：'base'颜色，适用于其它所有地方。

然而，如果样式表中的一个颜色与参考颜色中的一个完全匹配的话，那么将会忽略上下文，而将会使用相匹配的那个颜色。



例如，假定你默认的参考颜色为深蓝色（dark blue），但是你把它修改为了红色（red）.你的默认样式表包含了浅蓝和灰紫色，都相对于这个参考颜色。

最终颜色(紫红色和棕色)与红色的差别，类似于原始颜色与蓝色的差别。用技术术语来说：色调，饱和度和亮度上的相对差别将被保存起来。

如果你发现color.module使用了错误的参考颜色，可以尝试将不同的部分放到独立的CSS规则中，每一个都使用自己的选择器{ ... }，这样就不会和上下文冲突了。

注意，如果你在修改颜色方案以后，编辑了你的样式表，你需要重新提交颜色方案，以重新生成颜色偏移后的版本。

如果希望样式表中的一些颜色不被修改，你需要在它们的CSS上面放置下面的标记：

```
/*************************************************************
* Color Module: Don't touch                                  *
*************************************************************/
```

你只能在你的style.css文件中使用这个标记。它在全局都适用，所以如果你在一个引入的样式表中使用它的话，@import语句下面的所有颜色都将被保留起来。

使颜色相匹配

生成的图片，与生成的样式表中的颜色一定要匹配，这一点非常重要。否则，边缘看起来就会

异常丑陋。

为了能够匹配起来，基图片中的像素在区域中必须使用一个简单的颜色，因为在这里必须与CSS定义的颜色相匹配。因为我们不知道CSS定义的颜色出现在基图片中的哪个位置，我们可以使用一个全局复合颜色，它在整个设计中都是相同的。Garland使用了白色。注意，在Garland基图片中，的确包含了比如灰色和黑色像素，但是只有出现在图片使用它作为背景颜色的区域(例如，页首)。除了白色以外，黑色或者灰色也是很好的候选。

```
<?php
  array('blend_target' => '#ffffff');
?>
```

刨根究底的人可以阅读一下color.module中的代码，特别是_color_shift()函数，这能帮助你理解这是如何实现的以及为什么这样。

PHPTemplate的修改

最终，你需要在你的主题中适用color.module的钩子。我们以一个PHPTemplate主题为例，但是这也适用其它的引擎。

在你主题的template.php文件中，添加下面的代码片断（Drupal 6.x）：

```
<?php
/**
* Override or insert PHPTemplate variables into the templates.
*/
function phptemplate_preprocess_page(&$vars) {
  // Hook into color.module
  if (module_exists('color')) {
    _color_page_alter($vars);
  }
}
?>
```

在Drupal 5.x中，你需要添加下面的代码：

```
<?php
/**
* Override or insert PHPTemplate variables into the templates.
*/
function _phptemplate_variables($hook, $vars) {
  if ($hook == 'page') {

    // Hook into color.module
    if (module_exists('color')) {
      _color_page_alter($vars);
```

```
    }
    return $vars;
  }
  return array();
}
?>
```

这将允许模块覆写你主题的logo，样式表和截图。如果你要在_phptemplate_variables中作其它的修改，你需要将它们合并到这段代码中。

原文：http://drupal.org/node/108459

译者：葛红儒， http://zhupou.cn,

维护(maintenance)页的主题化

当drupal站点处于离线模式下时,就会使用维护页.你可以在"Administer > Site configuration > Site maintenance"页面启用这一模式.这一模式也将关闭数据库连接.在默认情况下,不管你是否选用了其它主题,对于这个页面都会使用drupal核心主题Minnelli.为了让维护页使用你的drupal主题,你需要在"settings.php"文件中对其进行设置,你可以在"sites/default" 或者"sites/your.domain.com"目录下找到"settings.php"文件.

在该文件中,通过设置$conf变量,让其使用你drupal主题的内部名称,从而为维护页启用新的drupal主题:

```
<?php
$conf['maintenance_theme'] = 'themeName';
?>
```

接着,拷贝你的"page.tpl.php"并将其改名为"maintenance-page.tpl.php",或者拷贝"modules/system/maintenance-page.tpl.php"模板到你的主题目录下面,对maintenance-page.tpl.php文件进行编辑,使其与你站点的其它部分相匹配.然后,将站点设为离线模式,退出登录,来验证所作的修改.考虑到数据库连接失败,你可以尝试关闭你的数据库.对于所有用到数据库的函数,都将会首先使用db_is_active对其进行检查.在模板中也可以使用变量$db_is_active.

为了阻止出现数据库连接的警告信息,你可以再加一个模板文件"maintenance-page-offline.tpl.php",使用你的自定义消息来修改$content变量.该模板文件是基于maintenance-page.tpl.php的一个模板建议,所以你需要将它们放到一块.

在这种模式下,将会包含一个"maintenance.css"文件.它位于"modules/system/maintenance.css".你可以按照样式表一节中的讲解,来覆写这个文件.

注意:

- 你的drupal站点的初始安装和升级,都依赖于核心主题Minnelli 和Garland.不能因为这些模式对这两个主题进行修改.
- 在离线模式下,主题注册表没有被缓存起来.

原文：http://drupal.org/node/195435
译者：葛红儒， http://zhupou.cn

解决drupal主题中的问题

在你构建网站的所有努力中，对于你的用户来说，最重要的就是站点的外观了。为了使你的drupal主题对于每个浏览器，每个模块，用户所选的主题都兼容的话，你需要花费很大的功夫。

首先你需要熟悉基本的CSS概念（Cascading Style Sheets）。关于Css的资源可参看CSS Discuss 或者HTML dog 。在CSS Zen Garden有一篇非常好的概述文章，介绍了Css的强大。

一定要通过XHTML 1.0 Strict验证。

现在有很多品牌的浏览器，而每个品牌的浏览器又有多个版本，而每种浏览器都有特定厂商维护的，所以你很难掌握一个浏览器在特殊的情况下会干些什么。这就带来了不兼容问题。如果你把注意力放到了这些不兼容性上，就会把问题复杂化，也使得你很难获得他人的帮助。只有当你使用通过验证的XHTML和CSS，并且它在符合标准的浏览器上能够正常工作的情况下，你才可以开始调试。人们经常听到设计者抱怨“但是但是但是，如果我让它通过合法性验证以后，那么外观就会再次变得丑陋起来，我不想再干第2遍这样枯燥无味的工作了”。在这种情况下，设计者得到的漂亮的外观，是建立在错误（bug）之上的！而在错误之上再进行调试，是理应被禁止的。

而事实上：一个经过验证、干净的布局在90%的情况下都能正常工作。在剩下的10%中，9%的问题很容易就得到了修复，而只有1%的情况需要更多的努力。真的！

如果你想验证你的整个站点，你可以使用这些工具。

你的页面不可能在所有的地方看起来都一样

另一个需要重点注意的地方是，HTML和CSS的本质。它们意味着在不同的地方有着不同的外观。比如，我的移动手机不能为你显示基于javascript的6栏布局。而且它也不应该能够。Safari 和Konqueror决定不支持表单中的特定样式（出于安全性和桌面兼容性考虑）。大的屏幕将对你的字体重新调整大小，这将破坏你的固定布局。而旧式的显示器则使用的是更大一些的字体大小集。而网络较慢的用户常常看不到图片。或者甚至是CSS。

所以你要记住，你的样式对于浏览器来说仅仅是一个建议，建议它以特定方式显示。而它不是一个命令。

管理drupal主题不兼容性的工具

1. 我们推荐你使用一个遵从标准的浏览器作为起点，比如Firefox。Firefox允许你对你网页的某些部分进行高亮显示，右击“查看所选区域的源文件”，可以帮你理解你主题使用的CSS类。理解CSS类是如何作用于底层的xhtml的，是理解你主题的关键所在。
2. 使用标准的CSS命名规范。我们推荐为你的CSS类采用这些命名规范。
3. 为你的主题选择一个合法的DOCTYPE类型，并且包含一个DocType Declaration(DTD)。
4. 为了帮助分析你的HTML 和CSS，我们推荐你为Firefox安装Firebug插件。这个工具非常有用，它允许你查看你的HTML和CSS，并实时的对其进行修改，从而评估修改所带来的效果。另一个非常有用的Firefox插件是Web Developer toolbar。它包含了许多方便的功能。
5. FireFox扩展插件查看格式化源文件，为你展示格式化的，代码着色过的源文件，并为每个元素提供可选的CSS信息。
6. 首先你需要确保你的HTML或者xHTML通过合法性验证，然后才能开始修改你的CSS来修正你的bugs。Firefox 的Web Developer toolbar（web开发者工具栏）内置了一个web验证器。Opera也内置了合法性验证，只需要按下Ctrl+Alt+V就可以了。
7. 一个更高级的工具是Watchfire WebXACT tool，可用于检查代码，分析HTML页面，定位错误。
8. 如果你发现了一个模块输出了非法的XHTML，你需要为该模块提交一个问题（issue），里面附上截图，以说明问题所在。

9. 你可以使用Lynx viewer来查看搜索引擎是如何看待你的站点的。
10. 在IE下面，你站点网页中定位问题，可以参考下面的文章进行解决：Position Everything Internet Explorer Primer（IE中对所有东西的定位，初级读本）。
11. 在Quirks mode（奇特模式）中，你可以找到许多问题的解答。

原文：http://drupal.org/node/37156
译者：葛红儒， http://zhupou.cn,

解决drupal主题中的问题(1)

跨浏览器兼容性(FireFox, Internet Explorer, Opera, Safari)

很难在所有的浏览器下检查你的主题。这里有一些工具，能够帮助你在多个浏览器下检查你的主题。

1. Browser Shots是免费的，但是获取截图需要花费一些功夫。
2. BrowserCam是收费的，有24小时的试用期。

在Windows上，你可以使用Internet Explorer，还可以下载Firefox 或者Opera。在Linux上，你可以使用Konqueror，一个基于KHTML的浏览器，而Safari用在MacOS上，Opera, Firefox可在Linux上使用，而IE也可运行在WINE。在Mac OSX上，你可以使用Safari，下载Firefox 或者Opera。

颜色和图形问题

1. 如果你想尝试选择颜色的话，你可以使用颜色方案（Color schemes）。
2. 如果在你的日志中生成了未能找到favicon的错误，你可以使用这两个工具来创建一个favicon，favicon from pics 和 favicon generator（生成器）
3. 要检查对色盲用户的影响，参看Vischeck

选择一个基主题

如果你要选择一个主题作为起点的话，对于基于CSS的主题来说，Zen 或 Foundation是不错的选择。如果你想使用表格来管理布局的话，可以使用BlueMarine。

特定于模块的CSS

一些Drupal模块自带了一个默认的CSS文件。你应该使用一个工具，比如开发者的工具栏，来检查模块的CSS是否对你的元素起作用，以及所引起的问题。当你安装一个新的模块时，你可以到模块所在的文件夹中看一看，它是不是包含了一个CSS文件。

你主题中的实际调试问题

对主题的调试没有简单的方案可行。如果你遇到了问题，那么你最好选用简单的基主题，或者尽可能的选择一个接近你最终目标的能够工作的主题。学习你主题中的CSS类，对于理解在哪里修改CSS至关重要。通过IRC可以找到其他的主题开发者，你可以把你做过的写成文当，简明教程。把你的主题提交到Drupal.org，这样他人可以对你的工作进行检查并提出反馈意见。和PHP程序员交朋友，他们能够帮你理解底层的PHP主题模块是如何工作的，这对你的工作也很有帮助。考虑相互帮助，以提高自己的技能。

本页的作者为来自于CivicSpace Labs和Theodore Serbinski的Kieran Lal, 和Trae McCombs。如果你想贡献自己的力量，或者让Drupal的主题制作更简单的话，可以加入主题邮件列表，或者直接联系Kieran。

基本的主题帮助

主题目标

对你站点主题的改进,可以帮助你完成许多目标.特别是,对主题的改进可以帮你完成以下业务目标:

- 为你的站点带来流量
- 帮你增加销售量
- 为站点用户提供更多的信息
- 帮助站点用户相互协作.

本文的目的是,帮助你实现下列的主题开发目标:

- 小的设计修改不会带来浏览器兼容性问题
- 在你的主题上实现设计者的wireframes.
- 选择一个基主题以进行定制
- 基于一个已存在的主题构建一个新的主题

改进Drupal中的主题帮助

在最近的一个Drupal管理员用户体验的调查显示,制作主题是最难的一个Drupal管理任务.我们做了一系列的采访,更好的了解到了主题制作者在开发主题时,需要完成的目标和任务.

基本Drupal主题任务

无论你使用的是什么样的主题,下面的这些基本的主题任务,都是你必须掌握的.

- Find and open a Cascading Style Sheet(CSS) file for your theme
- 为你的主题查找和打开一个CSS文件
- Copy and paste CSS code
- 复制和粘贴CSS代码
- Learn the CSS attributes in your theme
- 在你的主题中学习CSS属性

我们推荐你使用Firefox,再装上两个扩展插件,开发者工具栏和查看格式化源文件.我们还建议你使用标准的CSS ID名,与Andy Clarke 和 Eric Meyer建议的一样.

- 修改颜色
- 使用另一主题的一部分.例如,将CivicSpace管理主题拷贝到你的站点主题里面.
- 如何修改一个已有主题的布局.比如,修改列数,页首和页脚的位置,主体的可变性与静止性.

困难的主题任务

我们给出了一些需要详细解释的困难的主题任务.一些任务既是基本的任务也是困难的任务,我们需要为其花费更多的墨水,以解释这些任务的难点.

- CSS布局开发
- 找出一个好的基主题作为起点

我们在项目模块(project module)添加了分类功能,这样就允许了对主题所进行的分类.

- 修改已有主题
- 学习CSS类和IDs

我们推荐你使用Firefox,再装上两个扩展插件,开发者工具栏和查看格式化源文件.我们还建议你使用Eric Meyer的标准CSS类名.

- 如何为一个类添加内边距(padding)或者外边距(margins)
- 从一个主题中删除文本和图片.比如,删除“submitted by”，这可能是也可能不是主题问题。
- 如何修改表单的默认输出，通常情况下都是非常困难的
- 当模块生成的XHTML与主题相冲突时，如何进行调整

应该向该模块提交一个问题。里面要包含一个展示冲突页面的图片，以及它在一个基本主题下是如何显示的。从模块中摘出相应的XHTML代码片断，并指出期望的XHTML应该是什么样的。

- 使用PHP变量
- 禁止对容器使用固定的宽度
- 对于一些复杂的或者灵活的设计，比如fullheight，需要大量的工作来实现max-width 和min-width。
- 当插入一个设计者想要的图形效果时，要保证在IE下面也能正常工作。
- IE和Firefox的兼容性问题
- 从头编写一个主题

原文：http://drupal.org/node/39451

译者：葛红儒， http://zhupou.cn,

drupal主题编码习惯

主题作者应该仔细的编写干净、结构良好的代码，这和其它项目上的程序员一样。这样做，可以使代码更容易阅读、理解和维护。虽然不同的组织有着不同的习惯，但是最好大家都遵从Drupal标准，这样有利于协作工作或者需求帮助。

- 缩进采用2个空格；而不是一个tab键
- HTML标签的开始和结束部分的缩进一定要匹配。
- PHP 和HTML的缩进要区分开来

不是这样：

```
  ...
  <?php if ($header): ?>
  <div id="header">
    <?php print $header; ?>
  </div>
  <?php endif; ?>
  ...
```

应该这样：

```
  ...
<?php if ($header): ?>
  <div id="header">
    <?php print $header; ?>
  </div>
<?php endif; ?>
  ...
```

这样，良好的缩进，使得一眼就能够找到匹配的开始和结束标签了。

- 最好在HTML中使用PHP，而不是在PHP中使用HTML。例如，

不应这样：

```
<?php
if (!$page) {
  print "<h2><a href=\"$node_url\" title=\"$title\">$title</a></h2>";
}

if ($submitted) {
  print "<span class=\"submitted\">$submitted</span>";
}
?>
```

应该这样：

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

<?php if ($submitted): ?>
  <span class="submitted"><?php print $submitted ?></span>
<?php endif; ?>
```

毕竟，PHP是一个嵌入到HTML中的脚本语言-----而不是其它方式(HTML嵌入到PHP中)。

原文：http://drupal.org/node/1965

译者：葛红儒， http://zhupou.cn,

主题截图指南

一个Drupal截图

4.5以后的版本中,每个主题都需要一个截图,以screenshot.png的形式放置在主题目录下面.在Drupal安装的主题列表页面(在Drupal 5.x或者最新版本中,位于Administer > Site building > Themes),将会用到这些截图.所有截图最好能保持一致.核心主题截图的制作指南如下(从一个空Drupal站点开始):

1. 以用户1的身份登录.
2. 为了得到一些菜单项,启用下面的模块: aggregator, blog, node, page, search, story和tracker.
3. 打开主题支持的特性(logo(标识), site name(站名), slogan(标语), search box(搜索框)).如果需要的话,可以添加一些一级和二级链接.我们建议使用"Link 1" "Link 2" "Link 3",你可以将它们链接到,比如"user/1".
4. 将站名设为Drupal,标语设为Community Plumbing
5. 创建一个story节点:

Donec felis eros, blandit non

Morbi id lacus. Etiam malesuada diam ut libero. Sed blandit, justo nec euismod laoreet, nunc nulla iaculis elit, vitae. Donec dolor. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Vivamus vestibulum felis nec libero. Duis lobortis. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Nunc venenatis pretium magna. Donec dictum ultrices massa. Donec vestibulum porttitor purus. Mauris nibh ligula, porta non, porttitor sed, fermentum id, dolor. Donec eu lectus et elit porttitor rutrum. Aenean justo. Phasellus augue tortor, mattis nonummy, aliquam euismod, cursus eget, ipsum. Sed ultricies bibendum ante. Maecenas rhoncus tincidunt eros.

6. 查看该节点,确保所有标签都能被看到.截图.
7. 对截图进行切图,大小为420x254,对生成的图片调整大小, 调整为150x90(原始尺寸的35%).在这里只展示有用的页面元素（菜单，标签，标题，链接）。不要包括浏览器的东西（工具栏，状态栏，滚动条等）。
8. 为了更加清晰一些，可以对缩略图使用一个标准的'sharpen'过滤器。
9. 保存为PNG格式，选择paletted colorspace，这样可以减小文件大小。

例子：

用于Drupal.org的截图

Drupal.org上项目的缩略图，（显示在http://drupal.org/project/Themes和独立的主题项目页面），也使用前面的指南，但略有不同：

- 你需要为站点填充更多的内容。例如，为story节点添加一个评论或者添加一些区块。
- 截图应该显示整个页面，当然仍然不能包含浏览器本身的东西（工具栏，状态栏，滚动条

等）。
- 使得原始截图的宽度为1000像素，这样缩略图的大小为原始尺寸的30%左右。
- 尽量减小图片的大小：保存为PNG或者JPEG，这样能压缩10-20%。这将减小Drupal.org上主题列表页面上图片的加载时间。
- 为了将图片添加到你的项目上，点击项目节点上的编辑标签，展开"Attached Images"部分。找到你本地的截图，将其添加上去。当你提交了你的修改以后，就会创建一个缩略图，显示在项目页面的右上角处。

如果你需要修改你上传的图片，点击你项目页面上的缩略图（这将链接到图片节点页面），然后点击编辑标签。接着，你可以上传一个新的图片，来代替旧的。

注意，这个放置缩略图的方法，和以前这里列出的方法完全不同。已经不再使用旧方法（链接到CVS版本）了。这个新的方法，不再需要站点管理员干预图片的添加和修改了。

原文：http://drupal.org/node/11637

译者：葛红儒，http://zhupou.cn,

将你的主题添加到Drupal.org

为了将你的主题添加到Drupal.org上,它必须是**GPL**的.不要在里面包含一些有版权的东西,特别是那些你不希望看到别人重用或者修改的东西.

主题的追踪方式和代码一样,都使用CVS资源库.你将需要申请一个CVS帐号.获得帐号以后,你就可以将你的主题检入到Drupal CVS资源库中了.创建一个工程,系统将会自动为你创建下载的页面.

添加主题以后,用户就可能经常的提出建议、文件bug，一般都希望你能够随着Drupal版本的升级不断的更新你的主题。

参看截图指南。

关于如何在Drupal.org上贡献代码和主题，以及维护一个工程的流程的更多信息，可参看开发者手册，点击这里。

原文：http://drupal.org/node/14208
译者：葛红儒， http://zhupou.cn,