

把 Wordpress 集成到 zen-cart 里方法

字体大小: [小](#) [中](#) [大](#)

作者: [闻庭生](#) | 分类: [zen cart 插件精解](#) | 浏览: 4 | 评论: [暂时没有评论](#)

如果你的 Zen-cart 需要一个 Blog 来发布一些你的最新动态, 可以试试 Wordpress, 并且用 [WOZ](#) 这个 Zen-cart 的 Module 来把它们关联起来。[WOZ](#) 的全名叫 [Wordpress On Zen-cart](#)。

下面就详细介绍一下 W0Z 的安装方法, 当然你可以在 [WOZ](#) 包里找到[英文的安装说明](#)。我所使用的版本为 zen-cart1.3.8 和 wordpress2.6, 至于其它版本能否集成, 没有测试过。

Wordpress on Zen-cart 总的来说有两种安装模式:

第一种是把 Wordpress 的文件全部解压到 Zen-cart 的根目录下, 另一种方法是把 Wordpress 单独放到 Zen-cart 下新建的一个目录里。

两种方法最后访问 Wordpress 的地址都是:

[Zen-Cart URL]/index.php?main_page=wordpress

第一步, 安装 Wordpress

把 Wordpress 解压到你的 Zen-cart 目录下, 然后把 Wordpress 的文件夹名改为 blog, 呵, 看起来感觉会爽一些。

这一步的安装和普通的 Wordpress 安装没啥区别。

第二步: 设置 Wordpress

在 Wordpress 的后台设置一下, setting -> General -> Blog address(URL)
把 Blog address (URL)改为[Zen-Cart URL], 即 http://your.zen-cart.com

把主题设为默认主题。

第三步: 上传 W0Z 的文件

建议用 ftp 直接覆盖:

把 W0Z 解压后的 ZC_ROOT 目录下的文件上传到服务器 Zen-cart 的安装目录下。
把 W0Z 解压后的 WP_ROOT 目录下的文件上传到服务器 WordPress 的安装目录下。

再把/ZC_ROOT/includes/templates/MY_TEMP/里的 CSS 文件上传到你的 Zen-cart 模板里的 CSS 文件夹里。这个是 Wordpress 的显示主题。

第四步：修改 Wordpress 的文件

在你的 Wordpress 安装目录下，找到[Path of WordPress]/wp-include/template-loader.php，把所有的“**exit;**”都替换成“**return;**”。如果你的 Wordpress 版本比较新，就不需要做这一步的工作了。

第五步：修改 Zen-cart 文件

打开 Zen-cart 安装目录下的[Path of Zen-Cart]/includes/extra_configures/wordpress-config.php，把

```
define ('ABSPATH', '/var/www/vhost/example.com/public_html/blog/');
```

替换为

```
define ('ABSPATH', '[Path of WordPress]');
```

比如说，在我本地我是改为这样的东西：

```
define ('ABSPATH', 'E:/mywebs/zen-cart/blog/');
```

记住，这个路径得是你服务器上的绝对路径。

第六步：布局设置

在 Zen-cart 后台管理面板的 **admin panel > tools > Layout Boxes Controller** 里，选择打开你需要的 Wordpress 侧栏框。所有和 Wordpress 有关的侧栏框都是以 WP_XXX 格式存在的。

第七步：检查

现在你可以通过

```
http://your.zen-cart.com/index.php?main_page=wordpress
```

这个网址来访问你的 Wordpress 了。

如果你能看到 Wordpress，则表明你已经把 Wordpress 集成到了 Zen-cart 里了。

完成了这七步，基本上就把 Wordpress 的显示页面整合到了 Zen-cart 里面。

版权修改

问题：如何清除 zencart 页脚上的 " 版权所有(c)2003 Zen Cart. Powered by Zen Cart

解决方案:"版权所有 (c) 2003 Zen Cart" 覆盖了模板的设计和图形。如果您使用原有的模板和图形，您必须保留版权信息。如果您使用自己的模板和图形，您可以修改版权信息。

如果您想将商店放在 Zen Cart 网站上展示，您必须在页脚上保留"Powered by Zen Cart"。保留"Powered by Zen Cart"对您和 Zen Cart 都有好处，因为相关的互惠链接能提高您在搜索引擎的排名。如果您不想在 Zen Cart 网站上展示，您可以去掉"Powered by Zen Cart"，但不可以修改它。

如果要定制这段文字，在 \includes\languages\schinese.php 文件中第 17 行。

本文转自最模板,原文地址：<http://www.zuimoban.com/php/zencart/317.html>

zen-cart 实现 MSN 在线聊天

作为一个外贸网店，怎么能少一个与客户沟通的通道，为网站添加一个 MSN 在线聊天按钮可以方便客户何乐而不为。但是，现在网上大部分的 MSN 聊天代码已经不能使用，出现信息无法发送的情况。

正确的方法如下

1. 登陆

<http://settings.messenger.live.com/applications/websettings.aspx>，申请 MSN 在线通，同时登入个人的 Windows Live ID。

2. 进入申请页面按提示设置，最后会生成一段 Html 文本。黏贴到需要显示的地方即可。

第一修改 zen-cart 二次开发教程

以下是 zen cart 首页程序的修改。根据各个文件修改不同的功能。希望这些能对你有所帮助。

首页界面：//include/templates/zccn/common/tpl_main_page.php

首页主样式表：//include/templates/zccn/css/schinese_stylesheet.css

首页左边栏目：

/includes/templates/template_default/common/tpl_box_default_left.php

商品分类内容修改：

/includes/modules/sideboxes/categories.php

/includes/templates/zccn/sideboxes/tpl_categories.php

推荐产品

/includes/modules/sideboxes/featured.php

/includes/templates/template_default/sideboxes/tpl_featured.php

新进产品

/includes/modules/sideboxes/whats_new.php

/includes/templates/template_default/sideboxes/tpl_whats_new.php

特价产品

/includes/modules/sideboxes/specials.php

/includes/templates/template_default/sideboxes/tpl_specials.php

数据库表

/includes/database_tables.php

首页中间商品修改:

/includes/templates/template_default/common/main_template_vars.php

/includes/templates/template_default/templates/tpl_index_default.php

内页中间商品修改:

/includes/templates/template_default/templates/tpl_index_categories.php

首页中间 August 新进商品

/includes/templates/template_default/templates/tpl_modules_whats_new.php

首页中间 August 特价商品

/includes/templates/template_default/templates/tpl_modules_specials_default.php

首页中间 推荐商品

/includes/templates/template_default/templates/tpl_modules_featured_products.php

首页右边栏目:

/includes/templates/template_default/common/tpl_box_default_right.php

商品搜索

/includes/templates/template_default/sideboxes/tpl_search.php

客服中心

/includes/templates/template_default/sideboxes/tpl_ezpages.php

畅销商品

/includes/templates/template_default/sideboxes/tpl_best_sellers.php

商品评论

/includes/templates/template_default/sideboxes/tpl_reviews_write.php

在线名单

/includes/templates/template_default/sideboxes/tpl_whos_online.php

第二修改

有的程序一看就明白,然而真要进行修改和扩展,还真不容易, ICommerce 就是这样的程序;然而有的程序,看起来很复杂,以至于一时间连某个页面都找不到在哪儿,这种程序架构比较复杂,初级网站编程人员很难看明白,但是当你真正的掌握到了它的程序结构后,你一定会为架构师的思维拍案叫绝,因为你会发现,若要开发一些新功能会变得如此的方便, zen-cart 就是如此.

zen-cart 是国外比较优秀的开源电子商务网点源码, zen-cart 程序源于 os-commerce, 借鉴了 os-commerce 的很多成果,而在扩展性方面, zen-cart 显得又比前辈做得更好.

可以看到, zen-cart 前台所有的页面地址是如下格式:

1. http://127.0.0.1/zencart/index.php?main_page=index&cPath=1_4
2. http://127.0.0.1/zencart/index.php?main_page=product_info&cPath=1_4&products_id=1
3. http://127.0.0.1/zencart/index.php?main_page=site_map

也许你会觉得很困惑,怎么所有的页面都是 index.php 啊?这么大的一个网店系统,难道一个 index.php 页面就可以搞定了吗?这时,你可能就需要耐心的去分析一下 index.php 页面了.

经过分析,你就会发现,其实 index.php 页面并不想你所想象的那么神奇,那么深不可测. 相对于其他文件, Index.php 更像是一个管理者, 一个司令员,

当它接到指令后，并不是自己来处理，而是把任务分配给他的下属。至于要完成什么任务，那么就要看接收到的指令 `main_page` 的内容了。

在 `index.php` 页面的开头，作者介绍了该页面的处理过程，如表格 2-1 所示。

```
* index.php represents the hub of the Zen Cart MVC system
*
* Overview of flow
* <ul>
* <li>Load application_top.php - see {@tutorial initsystem}</li>
* <li>Set main language directory based on $_SESSION["language"]</li>
* <li>Load all *header_php.php files from
includes/modules/pages/PAGE_NAME/</li>
* <li>Load html_header.php (this is a common template file)</li>
* <li>Load main_template_vars.php (this is a common template
file)</li>
* <li>Load on_load scripts (page based and site wide)</li>
* <li>Load tpl_main_page.php (this is a common template file)</li>
* <li>Load application_bottom.php</li>
* </ul>
*
* @package general
* @copyright Copyright 2003-2005 Zen Cart Development Team
* @copyright Portions Copyright 2003 osCommerce
* @license http://www.zen-cart.com/license/2_0.txt GNU Public License V2.0
* @version $Id: index.php 2942 2006-02-02 04:41:23Z drbyte $
```

表格 1. 加载 `application_top.php` 文件

`Application_top.php` 文件中将进行许多常量的定义，加载配置文件等工作

2. 设置语言目录

这里定义了一个变量`$language_page_directory`，代表当前的语言目录，该变量的值和`$_SESSION["language"]`值息息相关。

3. 加载“`includes/modules/pages/PAGE_NAME/`”目录中的所有`header_php.php`文件。

可以看到，在每个网页内容的文件夹中，都有`header_php.php`文件，加载页面时，首先会加载该文件。

4. 加载`html_header.php`文件

`html_header.php`文件包含了在`<head></head>`标记中出现的内容，默认放在`includes/templates/common`目录下。当然在特定的情况下，加载的可能不是这个文件，你可以再模板级或者页面级来重写这个文件，从而定义特殊的`html_header.php`文件。

通过金字塔视图，可以看到`html_header.php`文件的优先级别，越靠近金字塔顶端，优先级别越高，如果优先级别高的文件已经存在，则不会加载优先级别比它低的文件了。



图 5. 加载`main_template_vars.php`文件

`main_template_vars.php`文件根据`$_GET["main_page"]`的值实现了页面的跳转逻辑。该文件的加载过程和`html_header.php`文件的加载过程异曲同工，`html_header.php`加载的是网页头部`<head></head>`中的内容，而文件`main_template_vars.php`却决定了要加载的网页的主体内容。

通过观察`template_default`中`common`下面的`main_template_vars.php`中的代码，可以知道，若在存储页面文件

的” `includes/modules/pages/$_GET["main_page"]`” 中增加一个名为 `main_template_vars.php` 的文件，则将直接加载这个文件。否则，将加载当前模板中的” `tpl$_GET["main_page"]_default.php`” 文件。（注意，这里实际上没有真正加载文件，而只是将要加载的文件的路径 `$body_code` 进行设置）

如果您对这两个文件的加载过程还不胜了解，那么请试着做下面的实验。

在地址栏中访问

” `http://127.0.0.1/zencart/index.php?main_page=news_list`”，将会出现网页未找到的提示信息。这是可以理解的，因为 `news_list` 是我自己想象的，`zen-cart` 没有提供这个页面，我自己也没有安装类似的插件。

然后在目录” `/includes/modules/pages`” 中新建一个文件夹，命名为 `news_list`，再访问该链接，就不会提示找不到链接了，至此一个新的页面就建好了，接下来的工作就是要丰富该页面的内容了。这将在后面的章节中予以描述。

6. 加载 `on_load` 脚本

`on_load*.js` 文件中包含了要在 `<body>` 标记的 `onload` 属性中出现的内容，即在页面加载完成后要执行的脚本。可以在同一个页面中定义多个 `on_load*.js` 文件，这些内容将会同时被执行。

`on_load*.js` 文件可能是页面级和站点级的。页面级的 `on_load*.js` 文件只在当前页面中有效，而站点级 `on_load*.js` 文件将对站点中的每个页面有效。有关如何定义页面级和站点级的 `onload` 事件，将在后面的章节予以描述。

在这里，页面级和站点级的 `on_load*.js` 文件内容都将读入 `$za_onload_array[]` 数组中，然后组合到变量 `$zv_onload` 中，为后面做好准备。`$zv_onload` 最终将作为到 `<body>` 的 `onload` 属性值。

7. 加载 `tpl_main_page.php` 文件

这里定义将选择影响页面布局的模板，可以在具体的页面中选择，或者使用 `zen-cart` 默认的模板（一个标准的三列板式），模板中会对 `main_template_vars.php` 中定义的变量 `$body_code` 所指文件予以加载。

8. 加载 `application_bottom.php` 文件

`application_bottom.php` 文件中将进行一些清理操作。

第三修改

本文介绍了基于 Zen Cart 搭建外贸网站的流程以及在使用 Zen-Cart 网站中遇到的一些常见问题，是作者开发经验的总结，对希望快速了解和掌握 Zen Cart 的 PHPer 有所帮助。

1、安装配置 PHP+MySQL 环境建议安装 PHP 集成环境包(方便快捷).网上也有许多此类教程，在此就不赘叙了。如果是虚拟主机的话，需要服务器支持 Zen-Cart 的安装运行环境要求。

2、下载 Zen-Cart 源程序，解压并上传到网站空间的对应目录下，在浏览器的地址栏里面输入 `http://domainname/zc_install/index.php`，然后按照提示步骤进行安装...安装完成之后，转到前台会发现两条显眼的红色背景的 Warning，这是要求你删除或重命名安装目录的名字，以及让你修改配置文件的权限(FTP 改不了的话直接修改 `includes/init/includes/init_header.php` 文件：`define(WARN_CONFIG_WRITEABLE, true);` 里 `true` 改为 `false`)，以防被别有用心者利用，按照要求做，然后再刷新你的网页就 OK 了。

3、安装 Easy Popular 批量商品管理软件，然后批量导入数据。下载批量商品管理插件，解压之后，里面有一个若干个 txt 文件和一个 catalog 的目录，将 catalog 目录下的文件全部拷贝到你的程序安装目录下。刷新后台，在 Tools 栏目下便可以发现多了 Easy Popular 的菜单；点击进入，网页顶部会有相关 Configuration 提示要你安装，根据提示点击 [here](#)，这样就成功安装了 EP 插件。然后是向数据库导入数据，这里要求上传产品图片到程序安装目录下的 images 目录下，可能产品图片非常多，手工编辑记录数据很麻烦，本人写了一个专用的小工具可根据产品图片直接生成对应的 excel 报表，如有需要的，可联系本人。

4、模版修改，主要是三个大的方面：顶部的 Banner 和 Logo、导航、Flash/JS 图片展示、整体的色调修改。导航栏在 `includes/templates/模版文件目录/commom/tpl_header.php`。在这个文件下 id 为 `topLinks` 的 div 里面 `require` 了一个文件，里面是输出导航菜单的。可以将这个 `require` 改为我们需要的东西，也可以直接去 `includes/templates/template_default/templates/tpl_ezpages_bar_header.php` 里面修改，对应的 CSS 位于 `includes/templates/模版文件目录/css/styleSheet.css` 里面的 `#topLinks` 里面。对于三栏的模版，左右两侧栏目的宽度都可以在 Configuration 下的布局设置里可直接修改，但对于中间一栏的设置则需要修改 CSS 代码了，对应的 CSS 代码位于 `includes/templates/模版文件目录/css/styleSheet.css` 文件中的 `#mainWrapper` 中，改过这里之后，你便会发现顶部的 LOGO 和导航那部分的宽度与整体不协调了，我们继续修改本文件中的 `#logoWrap` 的宽度就可以了。默认模版左右侧栏的标题背景颜色是不相同的，若要把他们修改为一样的颜色该怎么修改呢？有两种办法，一种就是继续在上面的 CSS 文件里面直接修改背景图片，另外一种就是将模版文件下 images 目录下的带有 `Headingbg` 的图片换成一样的图片。

5、后台常用设置

- (1) 根据需要开启左右侧栏的内容：Tools→Layout Boxes Controller
- (2) 购物支付及免运费模块设置：Modules→Payment Modules 安装配置支付模块 Modules→Shipping Modules 免运费模块设置 (Free Shipping) Modules→Order Total Modules 可设置优惠券、税款等。
- (3) 国家地区：Locations→Countries (外贸英文网站可移除中国地区 Code (2) CHN)
- (4) 汇率设置：Localization→Currencies (使用 IPS 等支付方

式注意保留人民币 CNY)(5)开户关闭注册或结账的条款:在后台 Configuration-->条款选项里设置 6、局部细节上的调整: information 栏目里面的内容[在 Tool→Define Pages Editor 菜单下]填充、sponsors 栏目里面的广告图片[在 Tool→Banner Manager]更改、底部的版权信息(includes/languages/English.php)需要做对应的修改等等。7、产品图片放大效果:在后台 Tools 栏目下点击 Image Handler2 下拉菜单→点击 Install Image Handler 安装图像管理模块;然后在 Configuration 中设置图像参数, IH 放大(中)小图像设为 yes, 在商品信息 - 图像宽度/图像高度 分别设定要放大的图片大小。我们自己开发主要是修改下面两个文件:

includes/templates/dreamugg/templates/tpl_specials_default.php
includes/functions/html_output.php
8、是不是觉得自己网站浏览人数少的让人尴尬? 有办法
includes/modules/sideboxes/whos_online.php 里面自己修改吧, 其中 \$n_guests 是游客人数, \$n_members 是注册会员人数, 自己可以据此去修改。另外一个就是 Powered by 信息的修改, 修改方法如下: includes/languages/english.php 里面修改第一个 define() 中对应处为自己所要信息即可。9、记得安装 IPS 等补丁(先执行 SQL 语句,再上传 IPS 补丁程序)...10、如果产品的分类比较多的话, 点击父分类的连接时, 会有子分类的连接, 但是这个链接平平无奇, 有些客户会认为他不是链接这样我们的产品浏览次数以及被购买的机会就降低, 所以必须对这种链接做一些修改。/includes/modules/category_row.php 里面修改相关代码即可。11、做外贸网站, Google 推广是一个很关键的环节, 所以这里就衍生了一个问题: 如何将 GGAD 跟踪代码添加到网站中去? 添加位置很灵活, 这里的例子仅供参考。全站跟踪代码添加到: includes/templates/使用的模版/common/tpl_footer.php
GGAD 定义操作跟踪代码: <!-- Google Code for 注册/登陆行为 Conversion Page -->
>includes/languages/english/login.php
>includes/languages/english/create_account.php<!-- Google Code for 购买/销售 Conversion Page -->
>/includes/templates/template_default/templates/tpl_checkout_success_default.php
>/includes/templates/template_default/templates/tpl_checkout_confirmation_default.php(183,1)<!-- Google Code for 销售机会 Conversion Page -->
>includes/languages/english/shopping_cart.php<!-- Google Code for 关键网页视图跟踪 Conversion Page --><!-- Google Code for 其它跟踪 Conversion Page -->
注意: 添加的时候, 如代码是直接加在网站开头(head 之前的时候)会导致整个页面的错位。12、后台显示的时区是默认值, 这对于我们来说很不方便, 如何修改呢? 找到 admin/includes/header.php 文件, 搜索到 echo date(r, time()) . GMT 位置处, 不用明说了, 大家都知道这里就是输出时间的, 在前面加上一句话 date_default_timezone_set(PRC); 存档, 到后台刷新一下看效果, okay, 现在就是北京时间了。(但生成的订单时间还是以网站所在服务器时间为准)13、由于 Google 广告申请不下来, 所以需要更换域名, 更换域名的话则需要修改网站里面的配置信息, 如何修改呢? 先修改两个配置文件 includes/configure.php
admin/includes/configure.php 把里面的定义的域名变量改为需要的域名或者使用一种一劳永逸的办法, 使用 http://\$_SERVER[HTTP_HOST]https://\$_SERVER[HTTP_HOST]替换对应位置处。接着就是修改 Logo、Banner, 这个没有什么值得说的, 没有技巧可言;接着就是 Copyright 信息和 Powered by 信息的修改。14、如何设置搜索引擎优化? 在后台的 Configuration-->搜索引擎优化选项下打开 SEO 功能, SEO 模块需要检查的就三个地方: (1). 确认网站支持 mod_rewrite, 且主机允许你使用自己的.htaccess 文件;(2). .htaccess 文件名字正确 (前面有个点) (3). 将根目录下的文件 htaccess_sample 改名为 .htaccess, 并修改其中的 /shop/为您的 zen-cart 目录。如果你的域名指向的目录是 zen-cart 的根目录, 那么就设置

为: RewriteBase /如果是 <http://www.domain.com/zencart/> 这样的形式打开网站, 那么 RewriteBase /zencart/15、如何添加新的页面? Payment Policy 设置: 后台 ==>Tool==>EZ-Pages==>Payment Policy==>edit: (如果没有 Payment Policy 这项,则點選 new file 按钮)在 Page Title 填写 Payment Policy 其它单选按钮不用改动, 只需在 HTML 编辑框内填写支付条款内容即可, 其它都不用填写。详见: [topic1947.html](#)16、网关支付接口相关设置 ZenCart 程序已集成了 PayPal 等支付网关接口, 只要设置好收款支付账号等相关参数即可。另外要安装的 IPS 等网关支付接口,可根据官方提供的支付接口进行整合。

17、商品描述之 Attributes Controller 扩展(在属性控制选项内容里添加设置)在 Catalog 下拉菜单下的 Attributes Controller 的标题 Option Name 下填写选项名称和 ID 并插入; 然后在 Option Values 中分别各个选项的值。另外, 如果所有商品属性都已设置好的话, 可以下载 型号/属性 以 TAB 分割的文本文件

18、设置优惠券: admin->Modules->Order Total->Discount Coupon19、产品重量的修改: (g 改为 kg)/includes/languages/english.php/includes/languages/english/modules/shipping/zones.php 在这两个文件里查找替换

20、在选择支付方式中添加支付条款的语句:在 /includes/languages/english/checkout_payment.phpdefine(TEXT_SELECT_PAYMENT_METHOD 或者: define(TABLE_HEADING_PAYMENT_METHOD 相应的地方添加。 .ccinfo {display:none; /*隐藏内卡标志*/}21、开启关闭注册或结账的条款在后台 Configuration-->条款选项 里设置

22、添加友情链接的方法这里利用到了 Banner Manager 广告代码显示友情链接, 首先在后台布局设置中的“广告显示组-页脚位置 1”填写你的广告组名称,如 Links. 然后在 Tools 栏目下的 Banner Manager 菜单新建“New Banner”; 在 Banner Group 填写框中输入广告组名称 Links, 并输入 Banner Title, 在 HTML Text 文本框内输入你要显示的友情链接 Html 代码, 点击 insert 即可。

第四修改

去掉产品目录旁边的产品总数方法在管理后台里, 进入“Configuration” —> “My Store” —> “Show Category Counts”选项, 设置为关即可。目录后面会有一个烦人的小箭头->按 Zen-cart 的默认配置, 目录后面有->(12356)这样风格的统计数字显示。如何去掉呢? 在管理后台里, 进入“Configuration” —> “Layout Settings” —> “Categories Separator between the Category Name and Count”选项, 按你的需要设置即可。在 Zen-cart 里如何修改“Powered by Zen Cart”的内容?这个可以在后台 My Store->footer text body 里进行修改, 也可以到 FTP 里的/include/template/你的模板/common/tpl_footer.php 里进行修改。页眉下面有一大堆烦人的 Categories tags, 怎么去掉? 后台管理 —> Configuration —> Layout Settings —> Categories-Tabs Menu ON/OFF 把这个属性设为关闭即可! 怎么修改 Contact Us 的页面内容与风格? 如果是内容的话, 在后台管理的 Tools —> Define Pages Editor 里进行页面内容的修改。如果是风格的话, 在 CSS 文件里找到对应的 id 或是 class 名进行修改。如果是要修改显示内容的话, 在你的模板目录下的 templates/tpl_contact_us_default.php 里修改, 如果你重写了页面内容的话, 请在 common 文件夹下找到相应的文件进行修改。打开产品根目录后, 会显示子目录的排列图

标，这个东西怎么修改？在模板目录下的 `common/tpl_columnar_display.php` 文件里，修改显示的内容。浏览器的抬头上写着“Zen Cart!, The Art of ...”，怎么修改？打开文件 `includes/languages/english/meta_tags.php`，在里面找到下面这段 `// page title define('TITLE', 'Zen Cart!');// Site Tagline define('SITE_TAGLINE', 'The Art of E-commerce');`把这些文字替换掉就可以了。当然，如果你是中文的页面，就要到相应的文件夹下面去找。我想把左、中、右三栏之间留一点空隙怎么办？Zen-cart 默认把左侧栏、中间产品栏、右侧栏排列得很紧密，我想把三栏之间的两个空隙弄小一些，怎么办？只要按这个方法，你可以随意调节三者之间的空隙大小。在后台 **Configuration** → **Layout Setting** 中，仔细调节 **Column Width – Left Boxes****Column Width – Right Boxes** 和 **Column Width -LeftColumn Width – Right** 的值，上面的这类带有 **Boxes** 的，是设定显示给我们看的 **box** 的大小，而没有 **Boxes** 的这类值，则是显示背景的大小。只要你把下面的值设得比上面的值大，就有空隙出来了。刚安装好的 Zen-cart，如何修改首页的 **Congratulations! you have successfully installed...If you want to change “Congratulations! You have successfully installed your Zen Cart; E-Commerce Solution” with your own text** open the `includes/languages/ENGLISH/index.php` file and find the following code:`// This section deals with the “home” page at the top level with no options/products selected/*Replace this text with the headline you would like for your shop. For example: 'Welcome to My SHOP!*/define('HEADING_TITLE', 'Congratulations! You have successfully installed your Zen Cart™ E-Commerce Solution.');`elseif (`$category_depth == 'nested'`) {`// This section deals with displaying a subcategory/*Replace this line with the headline you would like for your shop. For example: 'Welcome to My SHOP!'/define('HEADING_TITLE', 'Congratulations! You have successfully installed your Zen Cart™ E-Commerce Solution.');`};Replace the text starting “Congratulations” with your own text. Make sure that the single quote marks are not left out.如何修改左、右边框的显示风格？修改文件 `common/tpl_box_default_left.php` 或 `common/tpl_box_default_right.php` 即可。我得到了一个空白页面，怎么调试呢？最简单的方法，把以下代码加入到你的 `index.php` 里：`<?php define('STRICT_ERROR_REPORTING', true);?>`这样，你访问页面产生的错误消息就会输出到页面了。当然，记得把这个关掉，如果你把错误 **fix** 掉了的话，免得给用户看到。我要单独修改 **Categories** 里的内容，怎么办？不单是 **Categories**，所有的 **sidebox** 的内容，都可以在 `/includes/templates/YourTemplate/sideboxes` 文件夹里进行修改。`tpl_categories` 文件，就是修改 **Categories** 里面的显示内容的，当然，你修改了里面的 **CSS** 标签，就修改了 **Categories** 的样式。你可以通过 `common/tpl_box_default_left.php` 和 `common/tpl_box_default_right.php` 这两个文件来对左栏和右栏的内容进行大的调整。一般来说，你只要修改 **sideboxes** 里面对应的文件就能搞定你所需要的东西了。Posted in **Zen-Cart 系列知识, 常用知识 | No Comments** ?zen-cart 全新做一个自己的模板应参考文档(转载)星期天，七月 12th, 2009 Zen Cart 的模板设计比较复杂，需要一定的时间来熟悉。一旦你了解了它的结构，就会慢慢习惯了。首先要阅读常见问答部分的：如何添加、制作新模板。Zen Cart 的设计没有什么特别，与以前设计 **HTML** 页面是一样的。只是整个页面分成了好几个部分，并加入了 **PHP** 代码。通常，页面分为页眉(**header**)，页脚(**footer**)，边框(**sideboxes**)。所以设计页面的时候，要记住 Zen Cart 是如何组织这些页面的。页面是通过 **CSS** 样式表来控制的。样式表控制表格单元的背景图案、字体的颜色和样式等等。所以，假如你需要修改边框标题栏的字体，那么查看样式表文件。Zen Cart 在页面添加图像有两种方式。可以使用图像目录的相对路径，或者在模板中用 `php` 变量定义图像。如果你使用 **https** 服务器，并且采用相对图像路径，那么 **https** 的图像目录下也要有同

样的图像，否则 https 服务器很可能会给出警告提示。Zen Cart 可以设置成任意的 html/flash 的界面，只是比通常的 html 页面的设计费时。你可以从修改缺省的模板开始，先修改 CSS 文件和三栏格式的界面。开始先采用不同的颜色，很快就可以设计出完全不同的风格。最后，在设计模板前要先计划好你网页的内容，事半功倍。下面是 zen cart 页面各部分相对应模板文件的一个列表：文件路径注释 index.php 主文件 includes/templates/[custom template folder]/common/html_header.php 页面的 head 部分 includes/templates/[custom template folder]/common/tpl_main_page.php 页面的 body 部分 includes/templates/[custom template folder]/common/tpl_header.php 所有页面的页眉 (column left) includes/templates/[custom template folder]/common/main_template_vars.php 决定页面的内容部分，缺省为 'tmp_index_default.php' 首页 - 缺省 includes/templates/[custom template folder]/templates/tmp_index_default.php 首页模板文件 首页 - 显示分类 includes/templates/[custom template folder]/templates/tpl_index_categories.php 首页上显示分类时的模板文件 includes/modules/[custom template folder]/pages/index/category_row.php 选择要显示的分类 includes/templates/[custom template folder]/templates/tpl_index_category_row.php 显示分类首页 - 显示指定分类 includes/templates/[custom template folder]/templates/tpl_index_product_list.php 首页上显示指定的分类时采用的模板文件 includes/modules/[custom template folder]/product_listing.php 将商品数据添加到数组 includes/templates/[custom template folder]/templates/tpl_modules_product_listing.php 显示商品数量和商品导航菜单 includes/templates/[custom template folder]/common/tpl_list_box_content.php 显示商品数组商品信息页面 includes/templates/[custom template folder]/templates/tpl_product_info_display.php 显示单件商品信息 购物车页面 includes/templates/[custom template folder]/templates/tpl_shopping_cart_default.php 购物车页面 (column right) includes/templates/[custom template folder]/common/tpl_footer.php 所有页面的页脚

正确地定制您的网站 - 文件替代机制

由 [Jack](#) » 2005-07-25 21:14

本文翻译自 NetworkDad 的文章 “Properly customizing Your Site”，适当修改。

原文地址：<http://www.zen-cart.com/forum/showthread.php?t=7263>

~~~~~

很多用户对 zen cart 做了自己的修改，非常好！但同时，一些用户修改了很多重要的文件，例如：

```
/includes/filenames.php  
/includes/languages/schinese.php  
/includes/languages/schinese/*.php
```

等等...



友情提示：如果你希望以后的升级更容易，不要修改这些文件！

Zen Cart 的文件替代机制的功能非常强，你不需要修改上面的这些文件。

如果你修改了这些文件，下次升级的时候，Zen Cart 会覆盖这些文件，你做的修改会丢失。

好了，如何避免这种情况呢？如何使用文件替代机制？

下面，我一步步说明如何使用文件替代机制，而不是去修改 Zen Cart 的核心文件！

下面的所有例子，都假设你：

1. 使用简体中文。其它语言相应修改就好。
2. 例子使用的模板名字是 custom

“custom” 目录名可以是任何名字... 只是在所有地方都要一致，这样文件替代机制才能正常工作。

你可以命名为：

- mytemplate
- custom
- billg
- helluvacart
- zenrocks
- thisseemstoolongatemplatename
- supercalifragilisticexpialidocious
- tpl

还有很多 !!!!

附录及声明：如果我漏了什么，或者说错了，请告知，我会尽快修改！不作任何担保。一个月后凭收据退款！😄

— 注，请先阅读 — 如何添加/建立新的模板？

=====

修改缺省语言文件 - /includes/languages/schinese.php

例子：你想修改标题，将 “Zen Cart!”，改为 “我的商店名称！”

在目录/includes/languages 下，建立一个与你的模板同名的目录，例如 /includes/languages/custom.

将 schinese.php 文件复制到该目录。

接着，你就可以修改文件 /includes/languages/custom/schinese.php:

```
define('TITLE', 'Zen Cart!');
```

改为:

```
define('TITLE', '我的商店名称!');
```

=====

修改语言文件 - /includes/languages/schinese/\*.php (所有该目录下的文件)

例子: 你想修改 account.php 中的文字 - 比如你要将 '帐号' 改为 '档案'

建立一个与你的模板同名的目录。假设模板名是 custom。那么目录就是:

```
/includes/languages/schinese/custom
```

将原来的文件 /includes/languages/schinese/account.php 复制到新目录:

```
/includes/languages/schinese/custom
```

接着，修改该新的 custom 目录下的文件。这样，该修改会 '替代' 原来

/includes/languages/schinese 目录中的相应文件，从而不会影响和修改核心文件。

=====

修改模板文件 - /includes/templates/template\_default/templates/tpl\_\*\_default.php

例如: 你想修改 tpl\_account\_default.php 文件中的一些 HTML

在你的模板目录中( /includes/templates/custom ) 你应该已经有一个模板目录了

( /includes/templates/custom/templates )。

将文件 tpl\_account\_default.php 复制到该目录中。然后你就可以自己修改了，它会自动 '替代' /includes/templates/template\_default/templates 目录中的相应文件。

你只要复制需要修改的文件到自己的模板目录，如果你未做修改，那么缺省情况下，商店会从 template\_default 目录调用缺省文件。

=====

## 修改边框模板

这和修改通用模板是一样的(见上)，除了边框的路径是：

/includes/templates/custom/sideboxes

=====

修改边框 - /includes/modules/sideboxes/\*.php

例子：你想修改边框 information.php 文件，加上另外一个链接。

在边框文件的目录中(/includes/modules/sideboxes)，建立另一个和模板 custom 同名的子目录，就是/includes/modules/sideboxes/custom

将文件 information.php 复制到该新目录中

接着你就可以修改新的 information.php，它会替代/includes/modules/sideboxes 目录中的缺省 information.php 文件

你可以核对该边框是否被替代，登录管理页面 -> 工具 -> 外观控制。在栏目 '方框文件名' 下，如果文件名显示为红色，就说明它被替代了。

=====

调用自定义文件名 - /includes/filenames.php

例子：你新建了一个页面 about\_us，需要调用该文件名。

不要在/includes/filenames.php 文件中添加一个新的定义，你应该在已有的目录 /includes/extra\_datafiles 中定义该文件名。

建立新文件 about\_us\_filenames.php，然后放在目录/includes/extra\_datafiles 中。

在该文件中，要有以下代码：

代码: [全选](#)

```
<?php
// About Us Filename Define
##define('FILENAME_ABOUT_US', 'about_us');
?>
```

该文件将自动被调用，就象该目录中的其它文件一样，因此系统知道你的定制文件。

尽量用同样的办法新建文件... 例如，调用文件 `blah`，将它命名为 `blah_filenames.php`

=====

调用自定义数据库表 - `/includes/database_tables.php`

例子：你为自己的 UPS 跟踪系统建了个数据库 `ups_track`，需要为代码定义数据表名。

不要在文件 `/includes/database_tables.php` 中定义，你要在已有的目录 `/includes/extra_datafiles` 中定义数据表。

建立新文件 `ups_track_database_tables.php`，置于目录 `/includes/extra_datafiles` 中。

在该文件中，要有以下代码：

代码： [全选](#)

```
<?php
// UPS Tracking Table
#define('TABLE_UPS_TRACK', 'ups_track');
?>
```

该文件将自动被调用，就象该目录中的其它文件一样，因此系统知道你的定制数据库。

=====

额外语言文件

例如：你新建了一个 Resource Center 方框，需要一个额外文件来调用额外语言定义。

在目录 `/includes/languages/schinese/extra_definitions` 中，你可以建立一个文件 `resource_center_box_defines.php`。在该文件中，你可以放置新的方框的所有语言定义。

该文件将自动被调用，保证你的自定义方框或页面的语言定义被调用。

=====

给已有文件增加额外 Javascript

例子：你想为你的 Bizrate 帐号调用一些 javascript，这样，订单完成后，就会从已有页面调用 `checkout_success.php`。



在目录/includes/modules/pages/checkout\_success 中，建一个新文件 jscript\_bizrate.js

在该文件中，放置 Bizrate 给你的 javascript 代码，例如：

代码：[全选](#)

```
<script language="javascript" ##type="text/javascript"><!--
  whatever the heck bizrate gave you
//--></script>
```

同样地.. 如果你使用 Google 的转换跟踪系统。建立另外一个文件，例如 jscript\_google.php

=====

添加和修改 CSS 文件

例子：你为自己的网站新建了一个 CSS 类

不要添加到已有的 CSS 文件/includes/templates/custom/css/styleSheet.css 中，要在同一目录下，另建一个文件 styleSheet\_YOURFILENAME.css，然后将你的 CSS 数据加在这里。

=====

最后，谈谈升级！

现在，你正确使用 Zen Cart 的替代机制了！恭喜，你可以放心了。你不用担心升级文件会覆盖你的定制文件，因为所有的定制文件都保存在你自己的目录里！

好吧，你升级了 Zen Cart。但如果你替代的文件有新的代码呢？所以你需要比较 custom 目录和新的核心文件。

例子：你修改了/includes/languages/custom/schinese.php 文件，升级后，/includes/languages/schinese.php 文件中有些新的定义。显然你要把这些新的定义加入到你的 chinese.php 文件中。

最容易的方法就是使用文件比较工具！

推荐使用 WinMerge，也是开源免费。

有了文件比较工具，你就可以比较你的/includes/languages/custom/schinese.php 文件， 和新的核心文件/includes/languages/schinese.php，然后很容易将新的修改加入到你的替代文件中。

# 常用 CSS 元素

## div ul dl dt ol 的简单解释及布局定位

post by waipiu / 2009-4-14 11:43 Tuesday xhtml 发表评论 几个 css 元素的简单解释 div ul dl dt oldiv, 这个很常见, 块级元素, div 尽量少用, 和 table 一样, 嵌套越少越好

ol 有序列表。

```
<ol>
<li>.....</li>
<li>.....</li>
<li>.....</li>
</ol>
```

表现为:

```
1.....
2.....
3.....
```

ul 无序列表, 表现为 li 前面是大圆点而不是 123

```
<ul>
<li>.....</li>
<li>.....</li>
</ul>
```

很多人容易忽略 dl dt dd 的用法

dl 内容块

dt 内容块的标题

dd 内容

可以这么写:

```
<dl>
<dt>标题</dt>

<dd>内容 1</dd>
<dd>内容 2</dd>

</dl>
```

dt 和 dd 中可以再加入 ol ul li 和 p

理解这些以后，在使用 div 布局的时候，会方便很多，w3c 提供了很多元素辅助布局

由于项目中编写文档结构、编写 CSS 的人员较多，并与程序员协同工作，就需要统一 class 与 id 的名称，前天花了一点时间，按照大多人的习惯，制定了下面的常用关键字：

容器：container/box

头部：header

主导航：mainNav

子导航：subNav

顶导航：topNav

网站标志：logo

大广告：banner

页面中部：mainBody

底部：footer

菜单：menu

菜单内容：menuContent

子菜单：subMenu

子菜单内容：subMenuContent

搜索：search

搜索关键字：keyword

搜索范围：range

标签文字：tagTitle

标签内容：tagContent

当前标签：tagCurrent/currentTag

标题：title

内容：content

列表：list

当前位置：currentPath

侧边栏：sidebar

图标：icon

注释：note

登录：login

注册：register

列定义：column\_1of3 (三列中的第一列)

column\_2of3 (三列中的第二列)

column\_3of3 (三列中的第三列)

## 代码精简

使用 DIV CSS 布局，页面代码精简，这一点相信对 XHTML 有所了解的都知道。代码精简所带来的直接好处有两点：一是提高 spider 爬行效率，能在最短的时间内爬完整个页

面，这样对收录质量有一定好处;二是由于能高效的爬行，就会受到 spider 喜欢，这样对收录数量有一定好处。

表格的嵌套问题搜索引擎一般不抓取三层以上的表格嵌套，这一点一直没有得到搜索引擎官方的证实。我的几项实验结果没有完全出来，但根据目前掌握的情况来看，spider 爬行 Table 布局的页面，遇到多层表格嵌套时，会跳过嵌套的内容或直接放弃整个页面。

使用 Table 布局，为了达到一定的视觉效果，不得不套用多个表格。如果嵌套的表格中是核心内容，spider 爬行时跳过了这一段没有抓取到页面的核心，这个页面就成了相似页面。网站中过多的相似页面会影响排名及域名信任度。

而 DIV CSS 布局基本上不会存在这样的问题，从技术角度来说，XHTML 在控制样式时也不需要过多的嵌套。搜索引擎优化及营销都是非常有利的。搜索引擎表示排名规则会倾向于符合 W3C 标准的网站或页面，但事实证明使用 XHTML 架构的网站排名状况一般都不错。这一点或许会有争议，但樂思蜀本人保持这样的观点，有异议者可以拿三组以上基本同等质量的网站对比观察。内容来自中国站长资讯网([www.chinahtml.com](http://www.chinahtml.com))

我想，这样的情况可能不是排名规则，最大的可能还是 spider 爬行网站时，出现以上差异导致收录质量的不同。

毕竟廖胜于无，建议建站或改版的朋友们，技术许可的情况下，还是选择 DIV CSS 布局为好。

CSS 布局常用的方法：float:nonelefttright

取值：

none: 默认值。对象不飘浮

left: 文本流向对象的右边

right: 文本流向对象的左边

它是怎样工作的，看个一行两列的例子

xhtml:

```
<div id="wrap">
<div id="column1">这里是第一列</div>
<div id="column2">这里是第二列</div>
<div class="clear"></div>
</div>
```

CSS:

```
#wrap{width:100;height:auto;}
#column1{float:left;width:40;}
#column2{float:right;width:60;}
.clear{clear:both;}
```

position:static|absolute|fixed|relative



取值:

**static:** 默认值。无特殊定位, 对象遵循 HTML 定位规则

**absolute:** 将对象从文档流中拖出, 使用 **left**, **right**, **top**, **bottom** 等属性相对于其最近的一个最有定位设置的父对象进行绝对定位。如果不存在这样的父对象, 则依据 **body** 对象。而其层叠通过 **z-index** 属性定义

**fixed:** 未支持。对象定位遵从绝对(**absolute**)方式。但是要遵守一些规范

**relative:** 对象不可层叠, 但将依据 **left**, **right**, **top**, **bottom** 等属性在正常文档流中偏移位置

它来实现一行两列的例子

xhtml:

```
<div id="wrap">
<div id="column1">这里是第一列</div>
<div id="column2">这里是第二列</div>
</div>
```

CSS:

```
#wrap{position:relative;width:770px;}
#column1{position:absolute;top:0;left:0;width:300px;}
#column2{position:absolute;top:0;right:0;width:470px;}
```

他们的区别在哪?

显然, **float** 是相对定位的, 会随着浏览器的大小和分辨率的变化而改变, 而 **position** 就不行了, 所以一般情况下还是 **float** 布局!

## CSS 常用布局实例

单行一列

```
body{ margin:0px;padding:0px;text-align:center;}
#content{ margin-left:auto;margin-right:auto;width:400px;}
```

两行一列

```
body{ margin:0px;padding:0px;text-align:center;}
#content-top{ margin-left:auto;margin-right:auto;width:400px;}
#content-end{ margin-left:auto;margin-right:auto;width:400px;}
```

三行一列

```
body{ margin:0px;padding:0px;text-align:center;}
#content-top{ margin-left:auto;margin-right:auto;width:400px;width:370px;}
#content-mid{ margin-left:auto;margin-right:auto;width:400px;}
#content-end{ margin-left:auto;margin-right:auto;width:400px;}
```

单行两列

```
#bodycenter{ width:700px;margin-right:auto;margin-left:auto;overflow:auto;}
#bodycenter#dv1{ float:left;width:280px;}
#bodycenter#dv2{ float:right;width:420px;}
```

两行两列

```
#header{ width:700px;margin-right:auto;margin-left:auto;overflow:auto;}
#bodycenter{ width:700px;margin-right:auto;margin-left:auto;overflow:auto;}
#bodycenter#dv1{ float:left;width:280px;}
#bodycenter#dv2{ float:right;width:420px;}
```

三行两列

```
#header{ width:700px;margin-right:auto;margin-left:auto;}
#bodycenter{ width:700px;margin-right:auto;margin-left:auto;}
#bodycenter#dv1{ float:left;width:280px;}
#bodycenter#dv2{ float:right;width:420px;}
#footer{ width:700px;margin-right:auto;margin-left:auto;overflow:auto;clear:both;}
```

单行三列

绝对定位

```
#left{ position:absolute;top:0px;left:0px;width:120px;}
#middle{ margin:0px 190px 0px 190px;}
#right{ position:absolute;top:0px;right:0px;width:120px;}
```

float 定位

xhtml:

```
<div id="wrap">
<div id="column">
<div id="column1">这里是第一列</div>
<div id="column2">这里是第二列</div>
<div class="clear"></div>
</div>
<div id="column3">这里是第三列</div>
<div class="clear"></div>
</div>
```

CSS:

```
#wrap{ width:100;height:auto;}
#column{ float:left;width:60;}
#column1{ float:left;width:30;}
#column2{ float:right;width:30;}
#column3{ float:right;width:40;}
.clear{ clear:both;}
```

float 定位二

xhtml

```
<div id="center" class="column">
<h1>This is the main content.</h1>
```

```

</div>
<div id="left"class="column">
<h2>Thisistheleftsidebar.</h2>
</div>
<div id="right"class="column">
<h2>Thisistherightsidebar.</h2>
</div>
CSS
body{
margin:0;
padding-left:200px;
padding-right:190px;
min-width:200px;
}
.column{
position:relative;
float:left;
}
#center{
width:100;
}
#left{
width:200px;
right:200px;
margin-left:-100;
}
#right{
width:190px;
margin-right:-100;
}

*html#left{
left:190px;
}

```

这样的指令：(position)，在 DreamWeaver 中文版中翻译为“定位”，常用的属性有两个：relative(相对)与 absolute(绝对)。有很多朋友对这条指令的用法还是不清楚，这里做一些细致的讲解。

`position:relative`; 表示相对定位，被定位了这个属性的标签在所属的范围内可以进行上下左右的移，这里的移动与 `padding` 或是 `margin` 所产生的位置变化是不一样的。`padding` 与 `margin` 是元素本身的一种边距与填充距离并不是真正的移动，而被定义为 `relative` 的元素是真正的移动，这所产生的移动距离是从 `margin` 的外围到父级标签内侧之间这一段。

`position:absolute`; 表示绝对定位，如果定义了这个属性的元素，其位置将依据浏览器左上角的 0 点开始计算，并且是浮动正常元素之上的。那么当你需要某个元素定位在浏览器内容区的某个地方就可以用到这个属性。

于是产生了一个问题：现在大家做的网页大部分是居中的，如果我需要这个元素跟着网页中的某个元素位置，不论屏幕的分辨率是多少它的位置始终是针对页内的某个元素的，靠单纯的 `absolute` 是不行的。

正确的解决方法是：在元素的父级元素定义为 `position:relative`;(这里可以是祖父级，也可以是 `position:absolute`;，多谢谢 old9 的提出)需要绝对定位的元素设为 `position:absolute`;

这样再设定 `top,right,bottom,left` 的值就可以了，这样其定位的参照标准就是父级的左上角 `padding` 的左上侧。

## 一.选择符模式

模式/含义/内容描述

\*

匹配任意元素。(通用选择器)

E

匹配任意元素 E (例如一个类型为 E 的元素)。(类型选择器)

E F

匹配元素 E 的任意后代元素 F。(后代选择器)

E > F

匹配元素 E 的任意子元素 F。(子选择器)



**E:first-child**

当元素 E 是它的父元素中的第一个子元素时，匹配元素 E。( :first-child 伪类)

**E:link E:visited**

如果 E 是一个目标还没有访问过(:link)或者已经访问过(:visited)的超链接的源锚点时匹配元素 E。(link 伪类)

**E:active E:hover E:focus**

在确定的用户动作中匹配 E。(动态伪类)

**E:lang(c)**

如果类型为 E 的元素使用了(人类)语言 c (文档语言确定语言是如何被确定的)，则匹配该元素。( :lang() 伪类)

**E F**

如果一个元素 E 直接在元素 F 之前，则匹配元素 F。(临近选择器)

**E[foo]**

匹配具有 "foo" 属性集(不考虑它的值)的任意元素 E。(属性选择器)

**E[foo="warning"]**

匹配其 "foo" 属性值严格等于 "warning" 的任意元素 E。(属性选择器)

**E[foo~="warning"]**

匹配其 "foo" 属性值为空格分隔的值列表，并且其中一个严格等于 "warning" 的任意元素 E。(属性选择器)

**E[lang|= "en"]**

匹配其 "lang" 属性具有以 "en" 开头(从左边)的值的列表的任意元素 E。(属性选择器)

**DIV.warning**

仅 HTML。用法同 DIV[class~="warning"]。(类选择器)

E#myid

匹配 ID 等于 “myid” 的任意元素 E。(ID 选择器)

我们用下面的例子来解释“父元素”、“子元素”、“父/子”及“相邻”这几个概念。

```
<div title="这是一个 div">
```

```
<h1>这是是 h1 的内容</h1>
```

```
<p>这是一个段落 p 的内容! <strong>这里是 strong 的内容</strong>这是一个段落 p 的内  
容! </p>
```

```
</div>
```

从以上代码中，我们可以找出这样的关系：

h1 和 p 同为 div 的“儿子”，两者分别同 div 形成“父/子”关系。

h1,p,strong 都是 div 的“子元素”。（三者都包含在 div 之内）

div 是 h1 和 p 的“父元素”。

strong 和 p 形成“父/子”关系，strong 的“父元素”是 p。

但 strong 和 div 并非“父/子”关系，而是“祖孙”关系，但 strong 依然是 div 的“子（孙）元素”。

div 是 h1 p strong 三者的“祖先”，三者是 div 的“子（孙）元素”。

h1 和 p 两者是相邻的。

继承上面的实例来具体演示一下 E F 的关系：假如，我们需要将 strong 内的内容二字变为绿色，我们可以有哪些方法呢？

```
div strong {color:green;}
```

```
p > strong {color:green;}
```

```
div > strong {color:green;}
```

临近选择器和通用选择器：通用选择器以星号“\*”表示，可以用于替代任何 tag。

实例：

```
h2 * { color:green }
```

## 二.选择符分类介绍

### 1.通配选择符

语法:

```
* { sRules }
```

说明:

通配选择符。选定文档目录树(DOM)中的所有类型的单一对象。

假如通配选择符不是单一选择符中的唯一组成,“\*”可以省略。

示例:

```
*[lang=fr] { font-size:14px; width:120px; }
```

```
*.div { text-decoration:none; }
```

### 2.类型选择符

语法:

```
E { sRules }
```

说明:

类型选择符。以文档语言对象(Element)类型作为选择符。

示例:

```
td { font-size:14px; width:120px; }
```

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

伪类可以看做是一种特殊的类选择符，是能被支持 CSS 的浏览器自动所识别的特殊选择符。它的最大的用处就是可以对链接在不同状态下定义不同的样式效果。

## 1. 语法

伪类的语法是在原有的语法里加上一个伪类（pseudo-class）：

```
selector:pseudo-class {property: value}
```

（选择符:伪类 {属性: 值}）

伪类和类不同，是 CSS 已经定义好的，不能象类选择符一样随意用别的名字，根据上面的语法可以解释为对象（选择符）在某个特殊状态下（伪类）的样式。

类选择符及其他选择符也同样可以和伪类混用：

```
selector.class:pseudo-class {property: value}
```

（选择符.类:伪类 {属性: 值}）

## 2. 锚的伪类

我们最常用的是 4 种 a（锚）元素的伪类，它表示动态链接在 4 种不同的状态：link、visited、active、hover（未访问的链接、已访问的链接、激活链接和鼠标停留在链接上）。我们把它们分别定义不同的效果：

```
a:link {color: #FF0000; text-decoration: none}
```

```
a:visited {color: #00FF00; text-decoration: none}
```

```
a:hover {color: #FF00FF; text-decoration: underline}
```

```
a:active {color: #0000FF; text-decoration: underline}
```

（上面这个例子中，这个链接未访问时的颜色是红色并无下划线，访问后是绿色并无下划线，激活链接时为蓝色并带有下划线，鼠标在链接上时为紫色并带有下划线）

注意：有时这个链接访问前鼠标指向链接时有效果，而链接访问后鼠标再次指向链接时却无效果了。这是因为你把 a:hover 放在了 a:visited 的前面，这样的话由于后面的优先级高，当访问链接后就忽略了 a:hover 的效果。所以根据叠层顺序，我们在定义这些链接样式时，一定要按照 a:link，a:visited，a:hover，a:active 的顺序书写。

## 3. 伪类和类选择符

将伪类和类组合起来用，就可以在同一个页面中做几组不同的链接效果了，例如，我们定义一组链接为红色，访问后为蓝色；另一组为绿色，访问后为黄色：

```
a.red:link {color: #FF0000}
```

```
a.red:visited {color: #0000FF}
```

```
a.blue:link {color: #00FF00}
```

```
a.blue:visited {color: #FF00FF}
```

现在应用在不同的链接上：

```
<a class="red" href="...">这是第一组链接</a>
```

```
<a class="blue" href="...">这是第二组链接</a>
```

#### 4. 其他伪类

此外 CSS2 还定义了首字和首行（`first-letter` 和 `first-line`）的伪类，可以对元素的首字或首行设定不同的样式。

下面看这个例子，我们在段落标记里定义文本首字尺寸为默认大小的 3 倍：

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

```
p:first-letter {font-size: 300%}
```

```
</style>
```

.....

```
<p>
```

这是一个段落，这个段落的首字被放大了。

```
</p>
```

我们再定义一个首行样式的例子：

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

```
div p:first-line {color: red}
```



</style>

.....

<div>

<p>

这是段落的第一行

这是段落的第二行

这是段落的第三行

</p>

</div>

（上例中段落的第一行为红色，第二、三行为默认颜色）

注意：首字和首行的伪类需要 IE5.5 以上的版本支持。

## 1. Block 和 inline 元素对比

所有的 HTML 元素都属于 block 和 inline 之一。

block 元素的特点是：

总是在新行上开始；

高度，行高以及顶和底边距都可控制；

宽度缺省是它的容器的 100%，除非设定一个宽度

<div>, <p>, <h1>, <form>, <ul> 和 <li>是块元素的例子。

相反地，inline 元素的特点是：

和其他元素都在一行上；



```
<body>
```

```
<div class="container">
```

然后 CSS 这样设计：

```
#container
```

```
{
```

```
min-width: 600px;
```

```
width:expression(document.body.clientWidth < 600? "600px": "auto" );
```

```
}
```

第一个 min-width 是正常的；但第 2 行的 width 使用了 Javascript，这只有 IE 才认得，这也会让你的 HTML 文档不太正规。它实际上通过 Javascript 的判断来实现最小宽度。

同样的办法也可以为 IE 实现最大宽度：

```
#container
```

```
{
```

```
min-width: 600px;
```

```
max-width: 1200px;
```

```
width:expression(document.body.clientWidth < 600? "600px" : document.body.clientWidth > 1200? "1200px" : "auto");
```

```
}
```

#### 4. IE 与宽度和高度的问题

IE 不认得 min-这个定义，但实际上它把正常的 width 和 height 当作有 min 的情况来使。这样问题就大了，如果只用宽度和高度，正常的浏览器里这两个值就不会变，如果只用 min-width 和 min-height 的话，IE 下面根本等于没有设置宽度和高度。

比如要设置背景图片，这个宽度是比较重要的。要解决这个问题，可以这样：

```
.box
```

```
{  
  
width: 80px;  
  
height: 35px;  
  
}  
  
body .box  
  
{  
  
width: auto;  
  
height: auto;  
  
min-width: 80px;  
  
min-height: 35px;  
  
}
```

所有的浏览器都可以使用第一个 **box** 设置，但 **IE** 不认得第 2 段设置，因为其中用到了子选择器命令。第 2 个设置更特殊些，所以它会覆盖掉第 1 个设置。

在以前的 web 开发栏目中，我介绍了不少处理 **HTML** 表格的方法，尽管现在使用表格来布局网页的方法已经不再时髦了，但是您依然可以使用表格来显示表列数据。

显示和样式化表格的方法有很多种，在这篇文章中，我将介绍使用 **CSS** 对表格边框进行样式化的方法。

## 链接

**CSS2** 表格模型是基于 **HTML 4.01** 表格模型的。表格包含了一个可选的锚标记和单元格以及数据行，表格模型包含以下的元素：表格、锚、数据行、数据行组、数据列、数据列组和单元格。这篇文章将集中讲解表格中各个元素的边框处理方法。

## 边框

根据不同的需求，您可以对表格和单元格应用不同的边框。您可以定义整个表格的边框也可以对单独的单元格分别进行定义。**CSS** 的边框属性可以指定边框的大小以及颜色和类型。以下的代码定义了宽度为 5 个像素的黑色实线边框：

```
TABLE { 5px solid black; }
```

除此以外，您还可以使用相同的语法为表格中单独的单元格分别指定边框属性。您可以使用以下的属性值来定义边框类型：

**l none:** 指定表格没有边框，所以边框宽度为 0。

**l dotted:** 由点线组成的表格边框。

**l dashed:** 由虚线组成的表格边框。

**l solid:** 由实线组成的表格边框。

**l Double:** 由双实线组成的表格边框。

**l Groove:** 槽线效果边框。

**l ridge:** 脊线效果边框，和槽线效果相反。

**l inset:** 内凹效果边框。

**l outset:** 外凸效果边框，和内凹效果相反。

每个边框类型都可以指定一种颜色，边框是绘制在背景颜色上的，列表 A 使用边框属性来样式化整个表格以及锚标记和单独的单元格。

```
<html>
```

```
<head><title>HTML Table</title></head>
```

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

```
TABLE {
```

```
background: blue;
```

```
border-collapse: separate;
```

```
border-spacing: 10pt;
```

```
border: 5px solid red; }
```

```
TD, TH {
```

```
background: white;
```

```
border: inset 5pt;
```

```
horizontal-align: right; }
```

```
CAPTION { border: ridge 5pt blue; }
```

```
</style><body>
```

```
<table summary="TechRepublic.com - Tables and CSS">
```

```
<caption>First Quarter Sales</caption>
```

```
<thead><tr>
```

```
<thabbr="salesperson" scope="col">Person</th>
```

```
<thabbr="sales" scope="col">Sales</th>
```

```
</tr></thead>
```

```
<tbody><tr>
```

```
<td>Mr. Smith</td>
```

```
<td>600.00</td>
```

```
</tr><tr>
```

```
<td>Mr. Jones</td>
```

```
<td>0000.00</td>
```

```
</tr><tr>
```

```
<td>Ms. Williams</td>
```

```
<td>0000.00</td>
```



```
</tr></tbody>
```

```
<tfoot><tr>
```

```
<td colspan="2">Let's sale, sale, sale!</td>
```

```
</tr></tfoot></table></body></html>
```

## 列表 A

这个例子展示了很多可供使用的表格边框的选项，您可以使用您熟悉的度量单位（像素、磅，行长单位等）。设定边框的尺寸，您可以使用十六进制数值或颜色名称来指定边框颜色。以下的例子演示了定义边框的方法。

```
border: 5px solid red;
```

在这一条语句中融合了宽度、样式和颜色属性的定义，但是您也可以对这些元素进行单独定义，如下所示：

```
border-width: 5px;
```

```
border-style: solid;
```

```
border-color: red;
```

除了将表格作为一个整体进行定义，您也可以将表格边框的四个部分分别进行定义，包括顶部、底部、左边和右边。列表 B 中的代码将刚才的例子中的表格分成四个部分单独定义。

## CSS 布局常用的方法

```
float:nonelefttright
```

取值：

**none:** 默认值。对象不飘浮

**left:** 文本流向对象的右边

right: 文本流向对象的左边

它是怎样工作的，看个一行两列的例子

xhtml 代码:

Example Source Code

```
<div id="wrap">

<div id="column1">这里是第一列</div>

<div id="column2">这里是第二列</div>

</div>
```

CSS 代码:

Example Source Code

```
#wrap{ width:100;height:auto;}

#column1 { float:left;width:40;}

#column2{ float:right;width:60;}

.clear{clear:both;}
```

position:static|absolute|fixed|relative

取值:

static: 默认值。无特殊定位，对象遵循 HTML 定位规则

absolute: 将对象从文档流中拖出，使用 left, right, top, bottom 等属性相对于其最近的一个最有定位设置的父对象进行绝对定位。如果不存在这样的父对象，则依据 body 对象。而其层叠通过 z-index 属性定义

fixed: 未支持。对象定位遵从绝对(absolute)方式。但是要遵守一些规范

relative: 对象不可层叠，但将依据 left, right, top, bottom 等属性在正常文档流中偏移位置

它来实现一行两列的例子

xhtml 代码:

Example Source Code

```
<div id="wrap">

<div id="column1">这里是第一列</div>

<div id="column2">这里是第二列</div>

</div>
```

CSS 代码:

Example Source Code [www.52css.com]

```
#wrap{position:relative;width:770px;}

#column1{position:absolute;top:0;left:0;width:300px;}

#column2{position:absolute;top:0;right:0;width:470px;}
```

他们的区别在哪?

显然, float 是相对定位的, 会随着浏览器的大小和分辨率的变化而改变, 而 position 就不行了, 所以一般情况下还是 float 布局!

在以前, 为 Web 页面创建一个打印机友好的版本意味着要设计一个布局和格式都经过修改的单独页面, 这样才能够在打印的时候获得令人满意的效果。现在, 通过使用结构化的 XHTML 和 CSS, 你可以实现同样的效果而只用花费少得多的精力。

从屏幕显示到打印效果

大多数的 Web 页面都是设计适用于在计算机屏幕上观看的。然而, 有的时候用户需要将某些页面打印出来, 也许就是为了保留一个长期的记录, 或者将其用作方便的离线参考资料。

现在的麻烦是，让 Web 页面在计算机彩色屏幕上看起来引人注目和五彩缤纷的很多特性，都无法在打印版的 Web 页面上表现出相同的效果——尤其当打印机是黑白的时候。在被降级为灰度打印的时候，彩色的组合会失去（原有的）对比效果；图形会看起来失真，而且耗费太长的打印时间；在 Web 页面上起着重要作用的导航按钮在打印页面上也毫无用处。

为了克服这些问题，Web 的创建者常常会为页面专门设计一个打印机友好的版本，这样访问者就有打印的欲望。打印机友好的版本通常都包括有和主要 Web 页面相同的内容，但是会省略掉大多数的图形、背景和导航元素。页面还会把彩色转换成某种形式，以便生成能够让人接受的灰度图像。

## CSS 的解决方案

使用结构化的 XHTML 标示和 CSS 格式将内容和表示分离开来的一个优势在于，通过更改 CSS 样式，你可以很轻易就把内容重新格式化。因此，创建一个打印机友好的页面就是把一个不同的 CSS 文件链接到相同的 XHTML 页面上。

你可以同时把屏幕样式表和打印样式表链接到同一个 XHTML 文件里，所以就没有必要单独创建一个打印机友好的页面，只需要一个打印机友好的样式表就行了。当你在链接代码里加入多媒体类型的文件时，这就是在告诉浏览器为了进行屏幕输出要遵循或者忽略哪些 CSS 的规则，而为了打印输出要使用哪些规则。

下面是链接到一对 CSS 文件的例子：

```
<link rel="stylesheet" type="text/css" media="screen" href="mysite-screen.css"
/>
<link rel="stylesheet" type="text/css" media="print" href="mysite-print.css" />
```

如果需要支持老版本的浏览器，那你就必须坚持使用 CSS1 的媒体描述符 screen 和 print。它们是相互排斥的，因此在为屏幕显示而生成页面的时候，浏览器会忽略掉打印样式表，反之亦然。所以，每个样式表都需要包含相同的样式选择器，但是有不同的规则声明，以便为不同的输出设备分别生成页面样式。