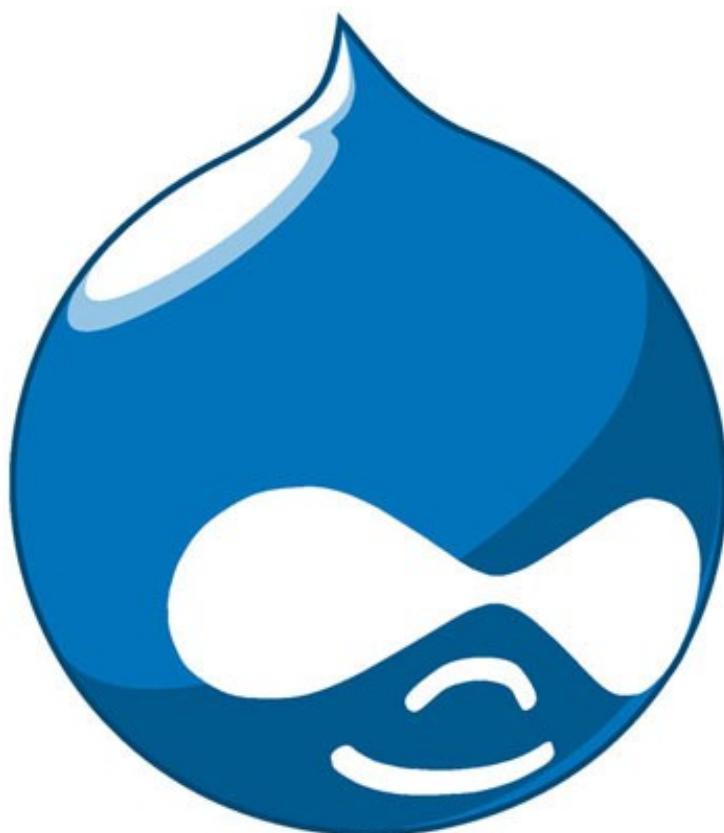


Уроки Drupal

-

или сказ о том как создать сайт



Оглавление

Drupal - Урок 1 - Блог, установка, настройка и русификация.....	4
Drupal - Урок 2 - Создание человеко понятных адресов и облака тегов(Модули Drupal Pathauto, Taxonomy, Tagadelic).....	15
Модуль PathAuto.....	15
Облако тегов.....	18
Drupal - Урок 3 - Прикручиваем FCKeditor, и учим его загружать изображения (Модули Drupal FCKeditor, IMCE, IMCE crop).....	24
Модуль FCKeditor.....	24
Модуль Similar By Terms.....	30
Drupal - Урок 4 - Разбираемся с модулем Views. Теория и практика.....	31
Первая задача.....	38
Вторая задача.....	45
Самый трудный пример, и самый последний (ну наконец то).....	48
Drupal - Урок 5 - Меню и то с чем его едят. Создание выпадающих меню, и развертывающихся без перезагрузки страницы.....	56
Drupal - Урок 6 - Модуль CCK. Создаем свой content type, и добавляем к нему поля.....	64
Drupal - Урок 7 - Таксономия.....	70
Drupal - Урок 8 - Еще немного про модуль Views. Работаем с Exposed filters и Arguments.....	76
Drupal - Урок 9 - Темизация.....	88
Drupal - Урок 10 - Руководство по созданию тем в Drupal 6.....	105
Часть первая. Анатомия темы Drupal 6.....	105
Часть вторая. Определение параметров и компонентов темы.....	108
Под-темы, их структура и наследование.....	108
Структура файла .info.....	110
Примеры файлов .info из тем ядра.....	113
Значения файла .info используемые по умолчанию.....	114
Назначаем регионам содержимое.....	115
Проверка занятости региона.....	118
Делаем настройки доступными на странице администрирования.....	120
Расширенные настройки темы.....	121
Добавляем форму-виджет для любых настроек темы.....	121
Получение значений опций в файлах вашей темы.....	122
Инициализация значений по умолчанию.....	123
Добавление настроек в новую версию вашей темы.....	124
Дизайн.....	125
Практика.....	125
Копируемые изображения.....	127
Заполнение областей и градиентов.....	127
Нарезка изображения.....	127
Файлы.....	128
Стили.....	128
Делаем что бы цвета совпадали.....	129
Очищаем кеш темы.....	131
Часть третья. Работа с CSS	131
Добавление таблицы стилей	132
Добавление стилей с помощью API	132
Создание и интерпретация CSS шаблона (style.css.php).....	134

Переопределяем таблицу стилей модулей или базовой темы.....	134
Переопределяем таблицы стилей ядра и поставляемых с ним модулей.....	135
Переопределение таблицы стилей базовой темы.....	135
Стандартные стили и классы ядра Drupal.....	136
Поддержка языков "с права налево" (RTL).....	137
Drupal - Урок 11 - Модуль Panels.....	138
Drupal - Урок 12 - Работа с изображениями. Модуль Image cache.....	141
Drupal - Урок 13 - введение в Ubercart, создаем простой онлайн-магазин.....	150
Drupal - Урок 14 - Создание опросов и страницы "Обратная связь" с помощью модуля Webform.....	157
Drupal - Урок 15 - Автоматизация. Модули Triggers и Workflow.....	167
Drupal - Урок 16 - Пишем свой модуль для Drupal.....	185
Приложение.....	194
Drupal 6 - переменные файла page.tpl.php.....	194
Drupal 6 - переменные файла node.tpl.php.....	195
Drupal 6 - переменные файла block.tpl.php.....	196
Drupal 6 - переменные файла box.tpl.php.....	197
Drupal 6 - переменные файла comment.tpl.php.....	197

Drupal - Урок 1 - Блог, установка, настройка и русификация.

By RemaGe - Posted on 26 Июнь 2009

Создание сайтов (веб-программирование) — одна из самых быстроразвивающихся веток программирования. Число сайтов с каждым днем увеличивается в геометрической прогрессии. Способы создания тоже немного изменились, для того что бы сделать персональную страничку, форум, или блог, уже не нужно месяцами корпеть над программным кодом. Можно взять готовое, настроить, и использовать в свое удовольствие. В этой статье, я расскажу вам как создать персональный блог наподобие этого. Потратив менее часа времени (это зависит от сообразительности), вы получите готовый продукт без строчки программирования. Этой статьёй я начну уроки посвящённые тому, как создать сайт на CMF **Drupal**.

Для создания блога нам потребуется:

1. Компьютер.
2. Руки.
3. Ноги (В туалет там сходить, или на кухню за чаем)
4. Голова.
5. Drupal (Заготовка нашего сайта)
6. Настроенный веб-сервер с PHP и MySQL (Настройку веб-сервера я пропущу, это вы легко найдете в интернете).

Для начала скачаем Drupal. Его можно найти как на официальном сайте - Drupal.org, так и на русскоязычном зеркале — Drupal.ru. Я предпочитаю второй, так как на нем всегда свежая версия перевода. Заходим на drupal.ru/downloads и скачиваем 2 файла



Главная

Скачать Drupal

Пришло: [admin](#) ат. 10/08/2008 - 19:44

Другие статьи по теме: [Drupal 6.x](#) [Drupal 5.x](#) [модули Drupal](#) [расширения](#) [скрипты Drupal](#) [ссылки](#) [теги](#)

Стабильные версии

- [Drupal 6.12](#) (последнее обновление 13.05.2009 1.05Mb);
- Предпоследняя версия перевода для версий Drupal начиная с 6.9 перевода разбитый по файлы модулей (последнее обновление с drupaler.ru 19.01.2009)
- Для версий Drupal 6.x: [расовый перевод интерфейса \(в формате PO, все в одном файле\)](#)
- Drupal 5.18 (последнее обновление 13.05.2009 766Кб)
- Для версий Drupal 5.x: [расовый перевод интерфейса \(в формате PO, все в одном файле\)](#)

Dev-версии

Внимание! Эти версии Drupalа находятся на этапе разработки. Использовать их можно или для ознакомления с функционалом, который идет нас в будущих версиях системы, или для помощи разработчикам. Использовать эти версии для рабочих проектов не стоит не рекомендуется.

- [Drupal 7.x-dev](#)

Пакеты в дистрибутивах

Мы настоятельно же рекомендуем из сборки от заинтересованных разработчиков или инсталляции с помощью опросок:

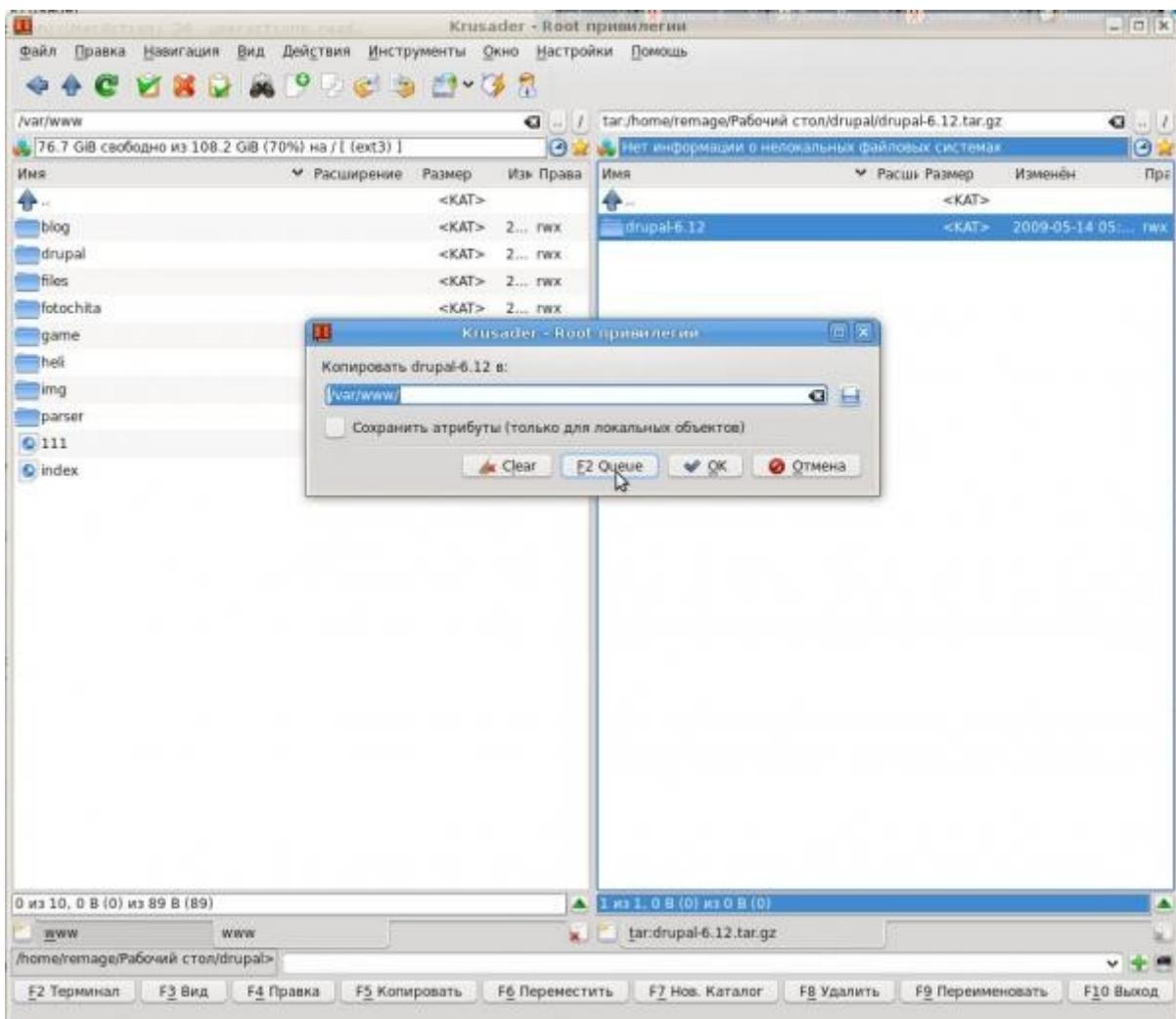
Debian GNU/Linux

В Debian Lenny [код](#)
В Debian Squeeze [код](#)
В Debian Sid ([unstable](#))
См. [документы в разных](#)

Gentoo GNU/Linux

Пакет входит в состав [portage.com/www-a](#)

Теперь нужно распаковать папку с Drupal в папку веб-сервера (для пользователей Linux она выглядит примерно как «/var/www/», должно получиться «/var/www/drupal-6.12», если возникнут проблемы, спрашивайте, постараюсь помочь).



Ну что же, половина работы сделана). Попробуем зайти на сайт) У меня получилась ссылка «localhost/drupal-6.12», у вас она может быть другой, если вы увидели не ошибку, а приветливую синюю мордашку, значит вы на верном пути.

The screenshot shows the 'Choose language' step of the Drupal 7.x installation process. On the left sidebar, under the 'Choose language' section, there is a green checkmark next to 'Choose language'. Below it, other steps are listed: 'Verify requirements', 'Set up database', 'Install site', 'Configure site', and 'Finished'. The main content area is titled 'Choose language' and contains two options: 'Install Drupal in English' (selected) and 'Learn how to install Drupal in other languages'.

Нажимаем на «Install Drupal in English», у меня появились 3 ошибки

The screenshot shows the 'Requirements problem' step of the Drupal 7.x installation process. On the left sidebar, under the 'Verify requirements' section, there is a green checkmark next to 'Choose language'. Below it, other steps are listed: 'Set up database', 'Install site', 'Configure site', and 'Finished'. The main content area is titled 'Requirements problem' and states: 'The following errors must be resolved before you can continue the installation process:'. It lists three errors:

- The Drupal installer requires that you create a settings file as part of the installation process.
 - Copy the `./sites/default/default.settings.php` file to `./sites/default/settings.php`.
 - Change file permissions so that it is writable by the web server. If you are unsure how to grant file permissions, please consult the [on-line handbook](#).
- The directory `sites/default/files` does not exist. An automated attempt to create this directory failed, possibly due to a permissions problem. To proceed with the installation, either create the directory and modify its permissions manually, or ensure that the installer has the permissions to create it automatically. For more information, please see `INSTALL.txt` or the [on-line handbook](#).

Please check the error messages and [try again](#).

1. Drupal просит скопировать файл /sites/default/default.settings.php в ту же папку, но при этом изменить его имя на settings.php
2. У созданного файла нужно изменить права доступа на 777 (Все могут читать, исполнять и изменять).
3. В папке sites/default/ нужно создать папку files, и изменить права на 777.

После исправления ошибок обновляем страницу, если все сделано правильно то мы получим

Database configuration

Basic options

To set up your Drupal database, enter the following information.

Database name: The name of the mysql database your Drupal data will be stored in. It must exist on your server before Drupal can be installed.

Database username:

Database password:

[Advanced options](#)

Save and continue

Тут нужно ввести название базы данных, пользователя, и пароль. Заходим в phpmyadmin (который у вас конечно же настроен), и создам новую базу

The screenshot shows the phpMyAdmin interface for the MySQL localhost server. In the top navigation bar, the 'Базы данных' (Databases) tab is selected. Below it, the 'MySQL localhost' section is active. A form is displayed for creating a new database, with the name 'drupal6' entered in the input field. A red oval highlights this input field. To the right of the input field is a dropdown menu set to 'Сравнение' (Compare) and a blue button labeled 'Создать' (Create), which is also highlighted with a red oval. At the bottom of the form, there is a note about matching the connection to MySQL.

Я назвал базу drupal6, вбиваем название, пользователя MySQL(обычно это root), и пароль MySQL (если вы установили пароль), кликаем кнопку «Save and continue», если все сделано правильно, получаем страницу «Configure site», и очередную ошибку. Drupal сообщает что все работы производимые с файлом /sites/default/settings.php окончены, и нужно убрать у него право на запись.

Site information

Site name: *
localhost

Site e-mail address: *

The From address in automated e-mails sent during registration and new password requests, and other notifications. (Use an address ending in your site's domain to help prevent this e-mail being flagged as spam.)

Administrator account

The administrator account has complete access to the site; it will automatically be granted all permissions and can perform any administrative activity. This will be the only account that can perform certain activities, so keep its credentials safe.

Username: *

Spaces are allowed; punctuation is not allowed except for periods, hyphens, and underscores.

E-mail address: *

All e-mails from the system will be sent to this address. The e-mail address is not made public and will only be used if you wish to receive a new password or wish to receive certain news or notifications by e-mail.

Password: *

Confirm password: *

Server settings

Default time zone:
Thursday, June 25, 2009 - 23:30 +1000 ▾
By default, dates in this site will be displayed in the chosen time zone.

Clean URLs:
 Disabled
 Enabled
This option makes Drupal emit "clean" URLs (i.e. without ?q= in the URL).
Your server has been successfully tested to support this feature.

Update notifications:
 Check for updates automatically
With this option enabled, Drupal will notify you when new releases are available. This will significantly enhance your site's security and is **highly recommended**. This requires your site to periodically send anonymous information on its installed components to drupal.org. For more information please see the [update notification information](#).

Вбиваем настройки:

1. Site name: - Имя сайта, то что будет отображаться в заголовке
2. Site e-mail address: - E-mail сайта, вбиваем свой, за неимением что нибудь

наподобие `qwe@qwe.er`

3. Username: - Логин администратора, например Admin (потешим свое самолюбие)
4. E-mail address: - E-mail администратора, по умолчанию, тот же что и у сайта.
5. Password: - пароль администратора, нужно написать что нибудь сложное, например 123, или 111.
6. Confirm password: - у Drupal плохая память, с первого раза пароль запомнит едва, повторяем.
7. Default time zone: - местное время.
8. Clean URLs: - «чистые ссылки», если включить, то ссылки будут вида «ваш-сайт/admin/settings/clean-urls» вместо «ваш-сайт/?q=admin/settings/clean-urls», у меня включено)
9. Update notifications: - если поставить галочку, Drupal сам будет проверять не появилось ли к нему обновление, или к одному из используемых модулей\тем оформления. Крайне удобно.

Жмем «Save and continue». Поздравляю, Drupal успешно установлен, заходим на наш сайт, и получаем приветственное сообщение «Welcome to your new Drupal website!». Ну что же, теперь давайте все переведём.

Заходим в меню «Administer», и видим Админку, именно здесь происходит вся настройка сайта. Сбоку мы видим блок «Site building», заходим во вкладку «Modules», и видим список модулей которые идут в базовой конфигурации, на данный момент нам нужно активировать модуль «Blog», именно он создаст функционал блога, и модуль «Locale», с помощью него мы русифицируем сайт. Прокручиваем вниз, жмем кнопку «Save configuration». Модули установлены, блоги начнут работать сразу, русский язык требует ещё немного работы. Идём в меню «Administer», блок «Site configuration», вкладка Languages, нажимаем кнопку «Add language»,

Languages [List](#) [Add language](#) [Configure](#)

This page provides an overview of your site's enabled languages. If multiple languages are available and enabled, the text on your site interface may be translated. Registered users may select their preferred language on the *My account* page, and site authors may indicate a specific language when creating posts. The site's default language is used for anonymous visitors and for users who have not selected a preferred language.

For each language available on the site, use the *edit* link to configure language details, including name, an optional language-specific path or domain, and whether the language is natively presented either left-to-right or right-to-left. These languages also appear in the *Language* selection when creating a post of a content type with multilingual support.

Use the [add language page](#) to enable additional languages (and automatically import files from a translation package, if available), locate strings for manual translation, or the [import page](#) to add translations from individual .po files. A number of contributed translation packages containing .po files are available on the [Drupal.org translations page](#).

Enabled	Code	English name	Native name	Direction	Default
<input checked="" type="checkbox"/>	en	English	English	Left to right	<input checked="" type="radio"/>

В появившемся выпадающем меню выбираем «Русский» (неожиданно, неправда ли?), сохраняем, должно получиться так

Languages [List](#) [Add language](#) | [Configure](#)

The language Russian has been created and can now be used. More information is available on the help screen.

This page provides an overview of your site's enabled languages. If multiple languages are available and enabled, the text on your site interface may be translated. Registered users may select their preferred language on the *My account* page, and site authors may indicate a specific language when creating posts. The site's default language is used for anonymous visitors and for users who have not selected a preferred language.

For each language available on the site, use the *edit* link to configure language details, including name, an optional language-specific path or domain, and whether the language is natively presented either left-to-right or right-to-left. These languages also appear in the *Language* selection when creating a post of a content type with multilingual support.

Use the [add language page](#) to enable additional languages (and automatically import files from a translation package, if available), the [translate interface page](#) to locate strings for manual translation, or the [import page](#) to add translations from individual .po files. A number of contributed translation packages containing .po files are available on the [Drupal.org translations page](#).

Enabled	Code	English name	Native name	Direction	Default	Weight	Operations
<input type="checkbox"/>	en	English	English	Left to right	<input checked="" type="radio"/>	0	edit
<input checked="" type="checkbox"/>	ru	Russian	Русский	Left to right	<input type="radio"/>	0	edit delete

[Save configuration](#)

нажимаем ссылочку «import page».

Translate interface Overview Search Import Export

This page imports the translated strings contained in an individual Gettext Portable Object (.po) file. Normally distributed as part of a translation package (each translation package may contain several .po files), a .po file may need to be imported after off-line editing in a Gettext translation editor. Importing an individual .po file may be a lengthy process.

Note that the .po files within a translation package are imported automatically (if available) when new modules or themes are enabled, or as new languages are added. Since this page only allows the import of one .po file at a time, it may be simpler to download and extract a translation package into your Drupal installation directory and add the language (which automatically imports all .po files within the package). Translation packages are available for download on the [Drupal translation page](#).

Import translation

Language file:
 
A Gettext Portable Object (.po) file.

Import into:
 Russian 
Choose the language you want to add strings into. If you choose a language which is not yet set up, it will be added.

Text group:
 Built-in interface
Imported translations will be added to this text group.

Mode:
 Strings in the uploaded file replace existing ones, new ones are added
 Existing strings are kept, only new strings are added

Language file: - выбираем скачанный нами ранее файл русификации (у меня он называется drupal-6.6-ru.po), и нажимаем «import».

Хуууух, Almost there.

Теперь еще раз заходим во вкладку Languages, и выбираем русский язык по умолчанию.

Languages [List](#) Add language Configure

This page provides an overview of your site's enabled languages. If multiple languages are available and enabled, the text on your registered users may select their preferred language on the [My account](#) page, and site authors may indicate a specific language which default language is used for anonymous visitors and for users who have not selected a preferred language.

For each language available on the site, use the edit link to configure language details, including name, an optional language-specific direction (the language is natively presented either left-to-right or right-to-left). These languages also appear in the Language selection when using multilingual support.

Use the [add language](#) page to enable additional languages (and automatically import files from a translation package, if available), locate strings for manual translation, or the [import page](#) to add translations from individual .po files. A number of contributed translation modules are available on the [Drupal.org translations page](#).

Enabled	Code	English name	Native name	Direction	Default	Weight
<input checked="" type="checkbox"/>	en	English	English	Left to right	<input type="radio"/>	0
<input checked="" type="checkbox"/>	ru	Russian	Русский	Left to right	<input checked="" type="radio"/>	0

Сохраняем... Вуаля, сайт изучил русский язык. В меня «Создать материал-> Запись блога», вы можете создавать записи в блоге.

На этом урок закончен.

В следующей статье, мы поговорим о том как:

1. Установить модуль pathauto, который для статьи с названием «В пасти безумия», сделает красивую ссылку вида «/content/v-pasti-bezumiya».
2. Выучим страшные слова «таксономия» и «тагаделик», с помощью которых призовем вельзевула.
3. Прикрутим к сайту облако тегов.
4. И, если хватит места, установим модуль Fck editor, который позволит создавать записи в блоге с помощью удобного меню в стиле Open Office, и одним махом заливать картинки на сервер, изменять их размер, и вставлять в статьи).

Drupal - Урок 2 - Создание человеко понятных адресов и облака тегов(Модули Drupal Pathauto, Taxonomy, Tagadelic).

By RemaGe - Posted on 27 Июнь 2009

Продолжаем наши уроки о том, как создать сайт. Сегодня мы разберем такие модули Drupal как Pathauto (его работа приятна взгляду, и говорят помогает в SEO), Taxonomy (его должен знать на пять любой уважающий себя Drupal-разработчик) и Tagadelic (он создает «Облако тегов»).

Модуль PathAuto

В стандартной поставке **Drupal** есть модуль **Path**, он предназначен для создания алиасов, то есть если у вас есть страница с адресом <http://ваши сайт/node/1/>, то с помощью этого модуля вы можете создать к ней алиас, и страница так же будет доступна по адресу <http://ваши сайт/content/klevaya-stranica>, но вбивать алиас к каждой странице накладно, для автоматизации этого процесса предназначен модуль **PathAuto**. Для начала давайте его скачаем) он доступен по [ссылке](#)

- **Path Redirect** when installed Pathauto will provide a new "Update Action" in case your URLs change. This is the recommended update action and is considered the best practice for SEO and usability.
- **Global Redirect** ensures that your content is only visible at the one, best URL possible (i.e. requests for node/2 are forwarded to *alias-for-node-2*).
- **Pathologic** (untested, but may help with updating outdated links in existing content)
- **Transliteration** useful if your patterns contain characters beyond ASCII 96 and you want them transliterated into ASCII 96 (used for Pathauto 6.x-2.x, in 6.x-1.x and earlier transliteration was handled differently...)
- **Sub-path URL Aliasing** provides a way to use one alias for multiple URLs as long as the so that "node/1" is aliased to "about-my-site" then you can use "about-my-site/edit" to access "node/1/edit". Neat!

Co-Maintainers

Pathauto is co-maintained by [mikeryan](#) and [Frederik 'Freso' S. Olesen](#).

Releases

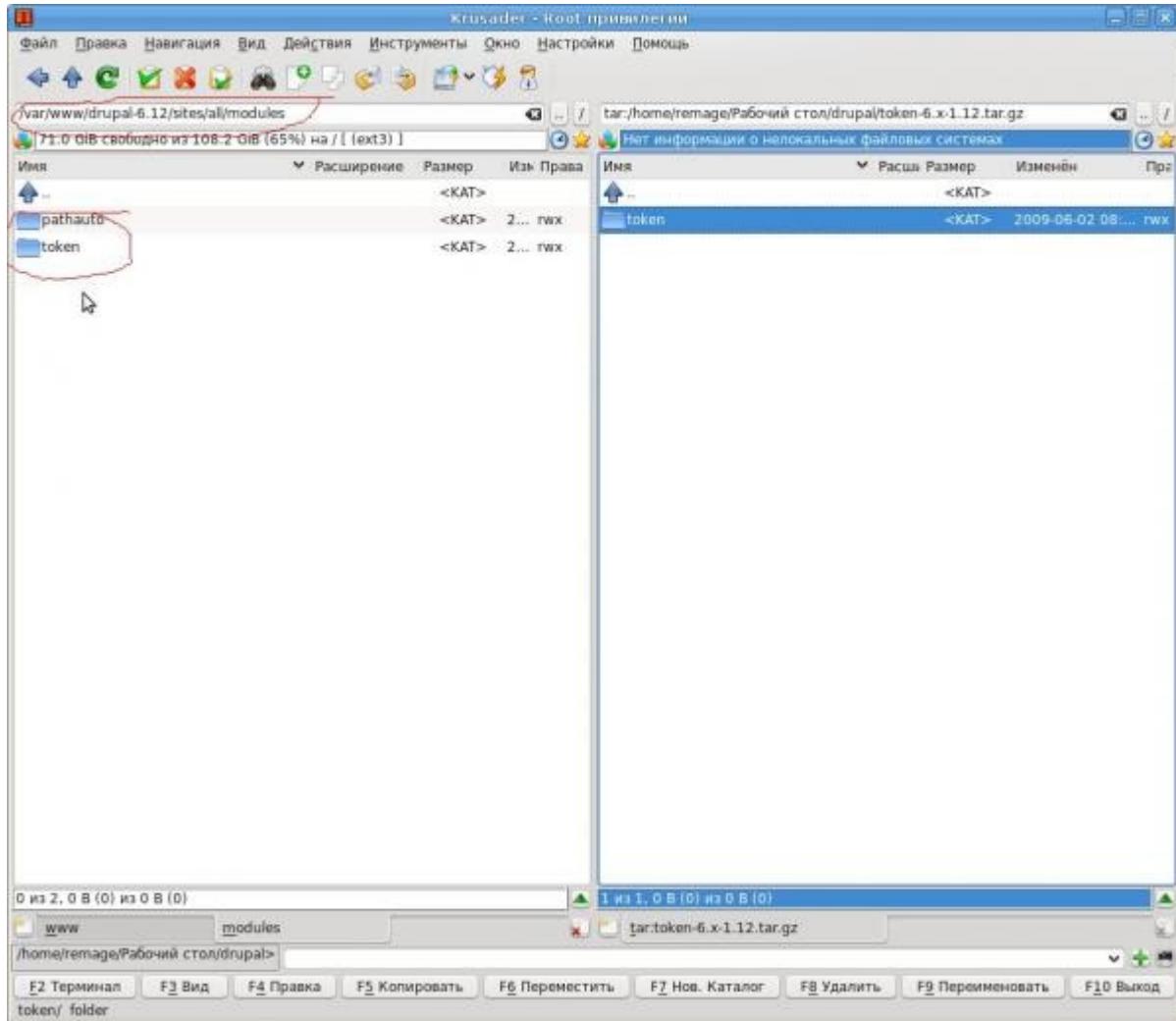
Official releases	Date	Size	Links	Status
6.x-1.1	2008-Jun-26	101.05 KB	Download · Release notes	Recommended for 6.x ✓
5.x-2.3	2008-Jun-26	96.65 KB	Download · Release notes	Recommended for 5.x ✓

Development snapshots

Development snapshots	Date	Size	Links	Status
6.x-1.x-dev	2009-May-03	128.51 KB	Download · Release notes	Development snapshot ✘
5.x-2.x-dev	2009-Jun-11	98.52 KB	Download · Release notes	Development snapshot ✘

• [View all releases](#) 

Поскольку у нас 6 версия Drupal, то и Pathauto мы скачиваем для 6 версии. Разархивируем модуль, и помещаем его в папку sites/all/modules, которая находится в каталоге вашего сайта (если там нет папки modules, то просто создайте ее). Так же, для работы PathAuto требуется модуль [Token](#), скачайте его по ссылке, и закиньте в ту же папку



Теперь на нашем сайте, с правами администратора, заходим в меню «Управление → Конструкция сайта → Модули», и активируем модули Path, PathAuto и Token. Сохраняемся.

<input checked="" type="checkbox"/>	Path	6.12	Позволяет переименовывать ссылки. Требуется для: Pathauto (отключён)
<input type="checkbox"/>	PHP filter	6.12	Позволяет внедрение кода PHP на страницах.
<input type="checkbox"/>	Ping	6.12	Оповещает другие сайты при обновлении вашего сайта.
<input type="checkbox"/>	Poll	6.12	Позволяет проводить на сайте опрос по различным темам, с возможностью выбора нескольких одновременно.
<input type="checkbox"/>	Profile	6.12	Поддерживает настраиваемые анкетные данные (профили) пользователей.
<input type="checkbox"/>	Search	6.12	Позволяет проводить на сайте поиск по ключевым словам.
<input type="checkbox"/>	Statistics	6.12	Журнал статистики доступа к вашему сайту.
<input type="checkbox"/>	Syslog	6.12	Регистрирует и записывает события в системный журнал.
<input checked="" type="checkbox"/>	Taxonomy	6.12	Позволяет упорядочить материалы по категориям. Требуется для: Forum (отключён)
<input type="checkbox"/>	Throttle	6.12	Поддерживает механизм авто-регулятора для контроля нагрузки на сайт.
<input type="checkbox"/>	Tracker	6.12	Позволяет пользователям следить за изменениями на сайте. Зависит от: Comment (отключён)
<input type="checkbox"/>	Trigger	6.12	Включает действия, которые будут запущены при возникновении обычных событий системы, на нового материала.
<input checked="" type="checkbox"/>	Update status	6.12	Проверяет статус доступных обновлений для Drupal и ваших установленных модулей и тем.
<input type="checkbox"/>	Upload	6.12	Позволяет пользователям закачивать и прикреплять файлы к материалам сайта.

▷ Ядро - обязательное

▷ Другой

Включено	Имя	Версия	Описание
<input checked="" type="checkbox"/>	Pathauto	6.x-1.1	Provides a mechanism for modules to automatically generate aliases for the content they manage. Зависит от: Path (отключён), Token (отключён)
<input checked="" type="checkbox"/>	Token	6.x-1.12	Provides a shared API for replacement of textual placeholders with actual data. Требуется для: Pathauto (отключён), TokenSTARTER (отключён), Token actions (отключён)

В принципе уже должно работать, но не все браузеры понимают русские символы, поэтому предлагаю перестраховаться, и настроить транслитерацию. Для этого находим в папке модуля **PathAuto** файл «i18n-ascii.example.txt», и изменяем его имя на «i18n-ascii.txt»

Идем в меню «Управление», ставим отображение по модулям.

Управление По задачам **По модулям**

Эта страница показывает все возможные административные настройки для каждого модуля.

Block Управляет блоками, которые расположены вокруг областей с основным содержимым.	Blog Позволяет создавать легко и регулярно обновляемые пользователем страницы или блоги.
<ul style="list-style-type: none"> ◦ Настройка прав доступа ◦ Блоки ◦ Справка 	<ul style="list-style-type: none"> ◦ Настройка прав доступа ◦ Справка

И ищем модуль **Path**, заходим во вкладку «Синонимы», там заходим в «Синонимы «Настройки адреса материала», и во вкладке «Основные настройки» устанавливаем галочку «Transliterate prior to creating alias». Сохраняемся. Модуль настроен -)

Облако тегов

Для создания облака тегов, нам потребуются 2 модуля, **Taxonomy**, который входит в стандартную поставку, и **Tagadelic**, который можно скачать [тут](#).

Модуль **Taxonomy** один из ключевых модулей **Drupal**, он отвечает за категории, и позволяет создавать неограниченное количество схем классификации.

Другими словами, если мы хотим создать на сайте раздел новости, мы сначала сделаем классификацию(словарь) «Новости», и при создании статей будем приписывать их к этому словарю, а при выводе пользователю, будем отображать записи принадлежащие к словарю «Новости». По описанию все выглядит немного путанно, но на деле все просто и прозрачно. Скачаем модуль **Tagadelic**, и активизируем привычным способом (Модуль **Taxonomy** по умолчанию активирован, если это не так, исправьте).

Теперь идем в меню «Управление → Таксономия» и нажимаем «Добавить словарь»,

Название словаря: Теги (или что-нибудь что понравится вам больше)

Типы материалов: Запись в блоге (Ведь мы создаем блог -))

В меню «Настройки» устанавливаем галочки «Теги» (Термины будут создаваться пользователями при создании материала, путём набора слов, разделённых запятыми) и «Обязательно» (По крайней мере, один термин должен быть выбран во время создания материала).

Таксономия Список Добавить словарь

Настройте, как ваш словарь будет представлен администраторам и пользователям, и какие типы материалов будут возможность пользователям создавать термины при добавлении материалов, вводя список слов, разделенных запятой выбираясь из выпадающего списка и создаваться они будут лишь пользователями с правами «управлять» тэксономией

Идентификация

Название словаря: Теги
Имя словаря, например, «Частки».

Описание:

Описание словаря. Может использоваться модулями.

Справочный текст:

Инструкция, показываемая пользователю при выборе терминов, например «щедите список слов, разделенных запятой».

Типы материалов

Типы материалов:

- Page
- Story
- Запись в блоге

Выберите типы материалов, которые будут классифицированы с помощью данного словаря.

Настройки

Теги
Термины будут создаваться пользователями при создании материала, путем набора слов, разделенных запятой.

Множественный выбор
Позволяет назначать материалам сразу несколько терминов из этого словаря (для тегов это всегда так).

Обязательно
Приайди же, один термин должен быть выбран во время создания материала.

Weight:

Теперь нам нужно решить где на сайте будет отображаться наше облако. Идем в меню «Управление → Конструкция сайта → Блоки». Здесь можно управлять блоками вашего сайта (подробнее о блоках можно почитать [тут](#)). Ищем блок «Tags in Теги» (Если вы конечно назвали словарик «Теги»), и значение области устанавливаем в «Левая колонка» (например), так же блок можно перетащить в любую область мышкой. Сохраняем, и нажимаем ссылку «Настроить», напротив названия нашего блока.

Блок	Область	Операции
Левая колонка		
+ Вход в систему	Левая колонка	настроить
+ Навигация	Левая колонка	настроить
+ Tags in Теги*	Левая колонка	настроить

Здесь мы можем задать заголовок блока (по умолчанию будет «Tags in Теги»), я поставил «Облако тегов», количество выводимых тегов, оставил по умолчанию 12, так же можно определить пользователей которые будут видеть этот блок, и страницы на которых он будет отображаться, эти настройки я не трогал.
Сохраняем.

Блок "Tags in Теги"

Настройки, специфичные для блока

Заголовок блока:

Перезаписать заголовок блока по умолчанию. Используйте <споле>, чтобы не показывать заголовок или оставьте это умолчанию.

Количество тегов:

Количество отображаемых в данном блоке тегов.

Настройки видимости для пользователя

Настройки видимости:

- Пользователи не могут управлять видимостью блока.
- Показать этот блок по умолчанию, но позволить пользователям скрывать его.
- Скрыть этот блок по умолчанию, но позволить пользователям показывать его.

Позволить пользователям переключать видимость этого блока в своих пользовательских установках.

Настройки видимости для ролей

Показать блок определенным ролям:

- anonymous user
- authenticated user

Показывать блок только выбранным ролям. Если ни одна роль не выбрана, блок будет виден всем.

Установки видимости для страницы

Показывать блок на определенных страницах:

- Показывать на всех страницах кроме перечисленных.
- Показывать только на перечисленных страницах.
- Показать, если данный PHP-код возвращает TRUE (PHP-режим, только для экспертов).

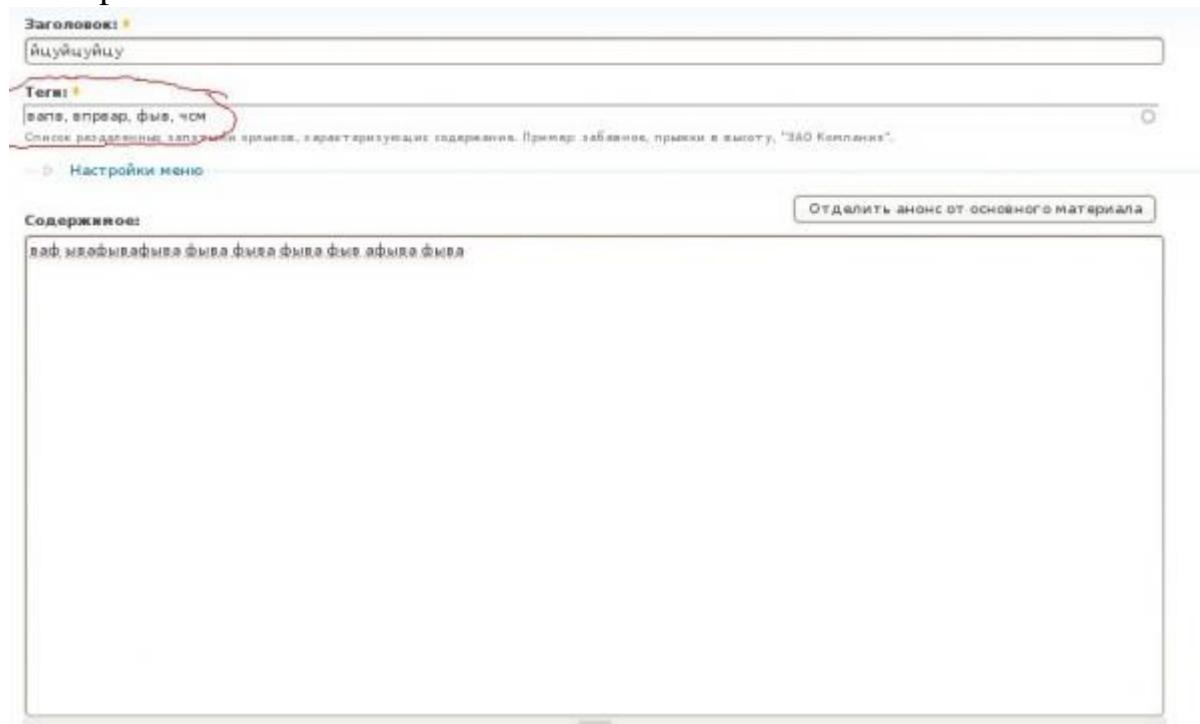
«Облако тегов» - настроено, осталось проверить. Создаем новую запись в блоге («Создать материал → Запись в блоге»), и вбиваем:

Заголовок: заголовок блога.

Теги: теги по которым его можно будет найти.

Содержимое: текст блога.

И сохраняемся.



Ну и где облако, спросит наивный читатель? По правде говоря я и сам не знаю почему не работает, разберусь попозже...(((

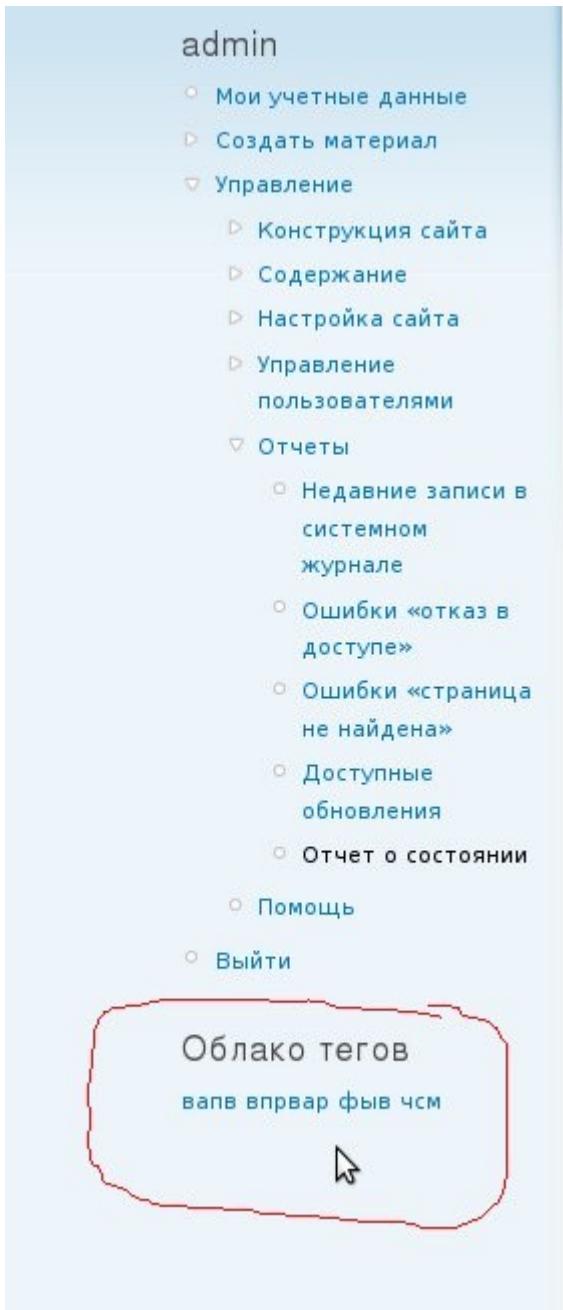
Шутка, новые теги появляются в блоге только после запуска кроны, идем в «Управление → Отчеты → Отчет о состоянии», и запускаем крон.

Отчет о состоянии

Здесь вы найдете краткий обзор параметров вашего сайта, а также проблем, которые были обнаружены. Данную информацию полезно добавлять в ваши вопросы на drupal.org или на других форумах поддержки. Обратите внимание, чтобы пользоваться сайтом drupal.org нужно владеть техническим английским. Обращайтесь на drupal.ru если вы русскоязычны.

Drupal	6.12
✓ GD библиотека	2.0 or higher
✓ PHP	5.2.6~Ubuntu4.1
✓ PHP register globals	Отключено
✓ База данных MySQL	5.0.75
✓ Библиотека Unicode	PHP расширение Mbstring
✓ Веб-сервер	Apache/2.2.11 (Ubuntu) PHP/5.2.6~Ubuntu4.1 with Suhosin-Patch
✓ Выполнение регулярных процедур (задачи cron)	Последний запуск 1 день 18 часа назад
Вы можете запустить выполнение регулярных процедур (cron) вручную .	
✓ Доступ к update.php	Защищен
✓ Обновления базы данных	Обновлен
✓ Ограничение памяти PHP	64М
✓ Сообщение об обновлениях	Включено
✓ Состояние обновления ядра Drupal	Обновлен
✓ Статус обновления для модулей и тем.	Обновлен
✓ Файл настроек	Защищен
✓ Файловая система	Запись разрешена (публичный метод загрузки)

Если все было сделано правильно, то появится долгожданное «облако».



Вывод тегов можно настроить в меню «Управление (по модулям) → Tagadelic → Облако тегов».

На этом урок закончен.

В следующей статье, мы поговорим о том как:

1. Установить модуль Fck editor, который позволит создавать записи в блоге с помощью удобного меню в стиле Open Office, и одним махом заливать картинки на сервер, изменять их размер, и вставлять в статьи).
 2. Сделать чтобы в блогах внизу выводился блок «Похожие статьи».
- А еще позже поговорим о модулях CCK, Views, Calendar, и о том как сделать сайт с поддержкой многоязычности.

Drupal - Урок 3 - Прикручиваем FCKeditor, и учим его загружать изображения (Модули Drupal FCKeditor, IMCE, IMCE crop).

By RemaGe - Posted on 28 Июнь 2009

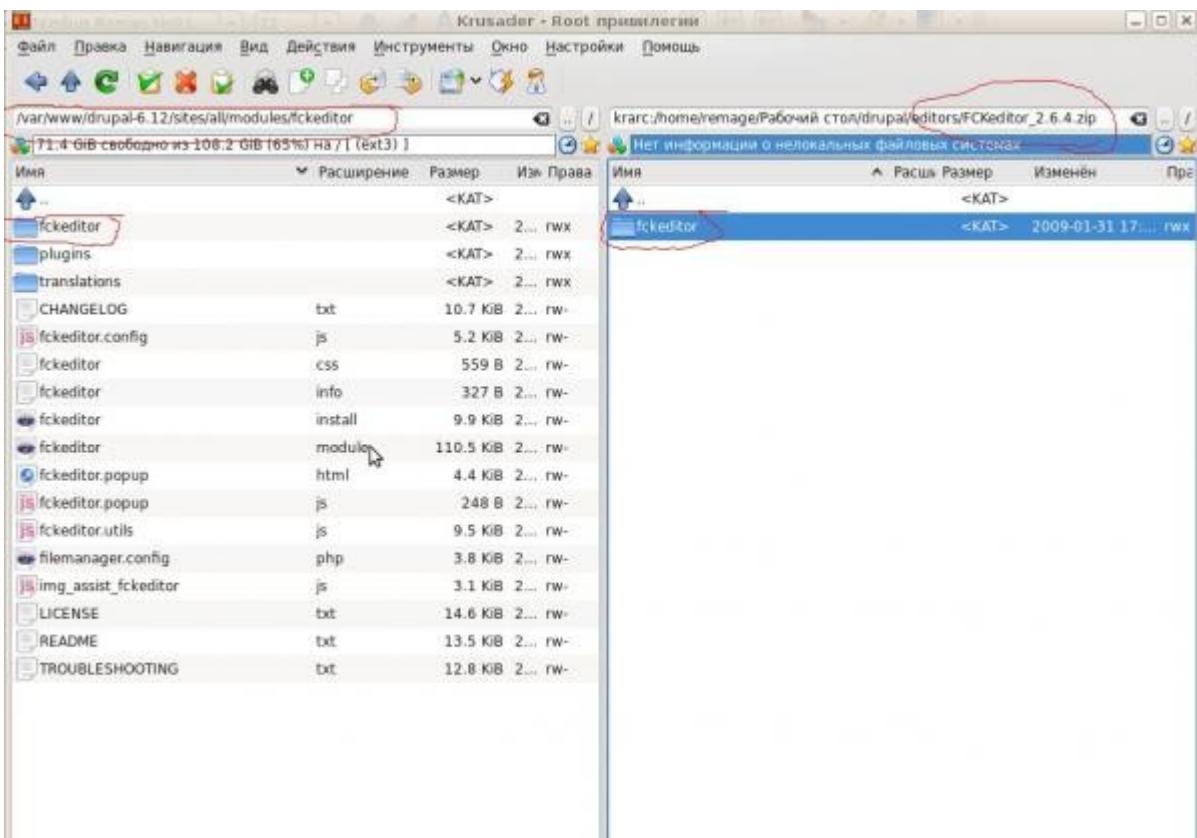
Продолжаем уроки Drupal, о том как создать сайт. Сегодня мы прикрутим к сайту FCKeditor, который превращает создание статей в чистое удовольствие, и с помощью модуля IMCE научим его заливать на сервер изображения.

Модуль FCKeditor

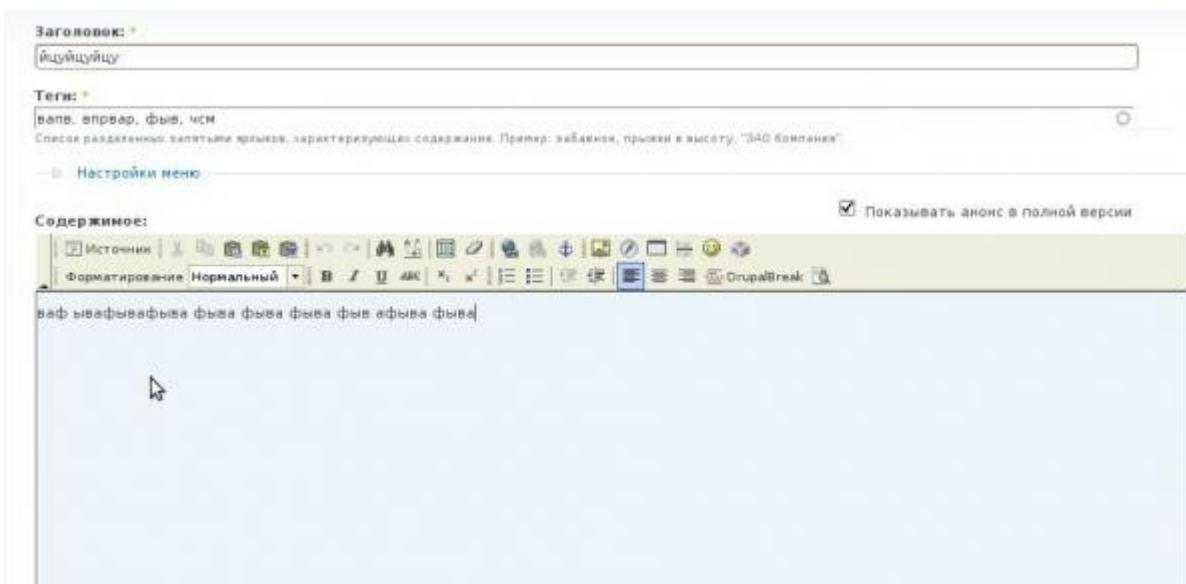
Изначально ввод текста в Drupal довольно таки неуклюж. Это решается сторонними модулями, которые в большом количестве находятся на Drupal.org. Одним из самых популярных является **FCKeditor**, это вызвано колоссальным удобством, и большими возможностями по настройке. Сам **FCKeditor** скачиваем, на данный момент доступна версия 2.6.4, а [здесь](#) скачиваем «лыжи», модуль для **Drupal** который встроит в него **FCKeditor**. Закидываем модуль скачанный с Drupal.org в папку /sites/all/modules нашего сайта, и распакуем **FCKeditor** в папку только что скопированного модуля.

[Модуль FCKeditor](#)

[CKeditor](#)



Включаем модуль привычным способом в меню «Управление → Конструкция сайта → Модули». Сразу после установки модуль готов к работе.



Теперь поговорим о проблеме возникающей у многих пользователей, - загрузка изображений на сервер средствами **FCKeditor**. В принципе этот функционал есть и в самом **FCKeditor**, но намного проще это реализовать с помощью модуля [IMCE](#), и для увеличения функционала так же скачаем [IMCE mkdir](#), и [IMCE Crop](#), первый будет создавать на сервере папки, а второй обрезать изображения до нужного размера. Устанавливаем модули.

Для того что бы эта связка начала работать, нужно настроить профиль, и дать доступ одной из ролей, но администратора в разрешениях нету, что же делать? Мы создадим еще одну роль, и настроим для нее доступ, а администратор унаследует права от нее (не нашел как это сделать без создания дополнительной роли, если кто то знает — просвятите -)).

Идем в меню «Управление → Управление пользователями → Роли», и добавляем новую роль - «модератор».

Имя	Операции
анонимный пользователь	неизменяемо изменить права
зарегистрированный пользователь	неизменяемо изменить права
модератор	Добавить роль

Далее переходим в меню «Управление → Управление пользователями → Разрешения», и и даем модератору право на доступ к FCKeditor (access fckeditor).

Право доступа	анонимный пользователь	зарегистрированный пользователь	модератор
модуль block			
управлять блоками	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
использовать PHP для управления видимостью блоков	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
модуль blog			
создавать записи в блоге	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
удалять любую запись в блогах	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
удалять свои записи в блоге	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
редактировать любую запись в блогах	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
редактировать свои записи в блоге	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
модуль comment			
иметь доступ к комментариям	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
управлять комментариями	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
размещать комментарии	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
размещать комментарии без проверки	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
модуль fckeditor			
access fckeditor	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>

Сохраняем, заходим «Управление → настройка сайта → FCKeditor». Настроим профиль Advanced, кликаем ссылку «изменить» напротив профиля Advanced.

Basic setup:

Устанавливаем галочку «Модератор»

Editor appearance:

Toolbar – устанавливаем "Drupal Full" (у модератора будут все возможности)

Язык – выбираем Ru

И по желанию можно изменить Skin, мне нравится Office2003

CSS:

Editor CSS – устанавливаем FCKeditor default (это не обязательно, но могут возникнуть проблемы при использовании сторонних тем для сайта)

File browser settings:

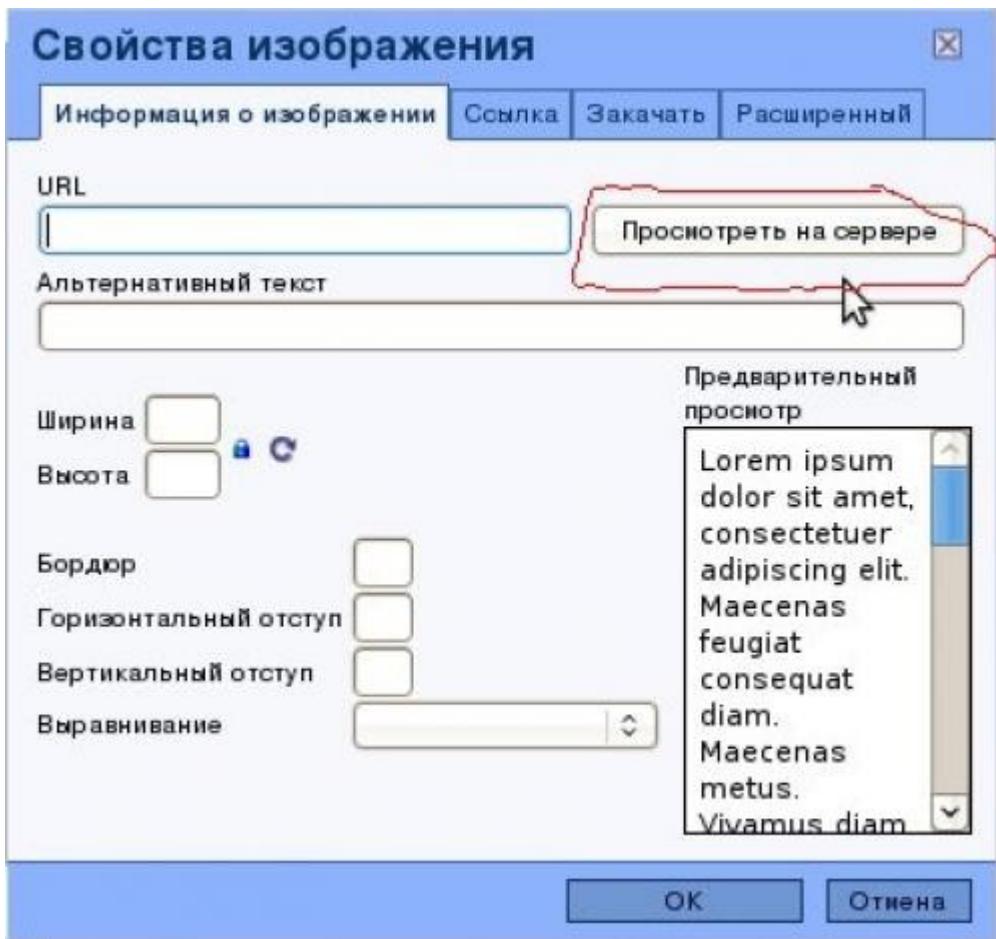
File browser type – ставим IMCE

остальное можно не трогать, сохраняем.

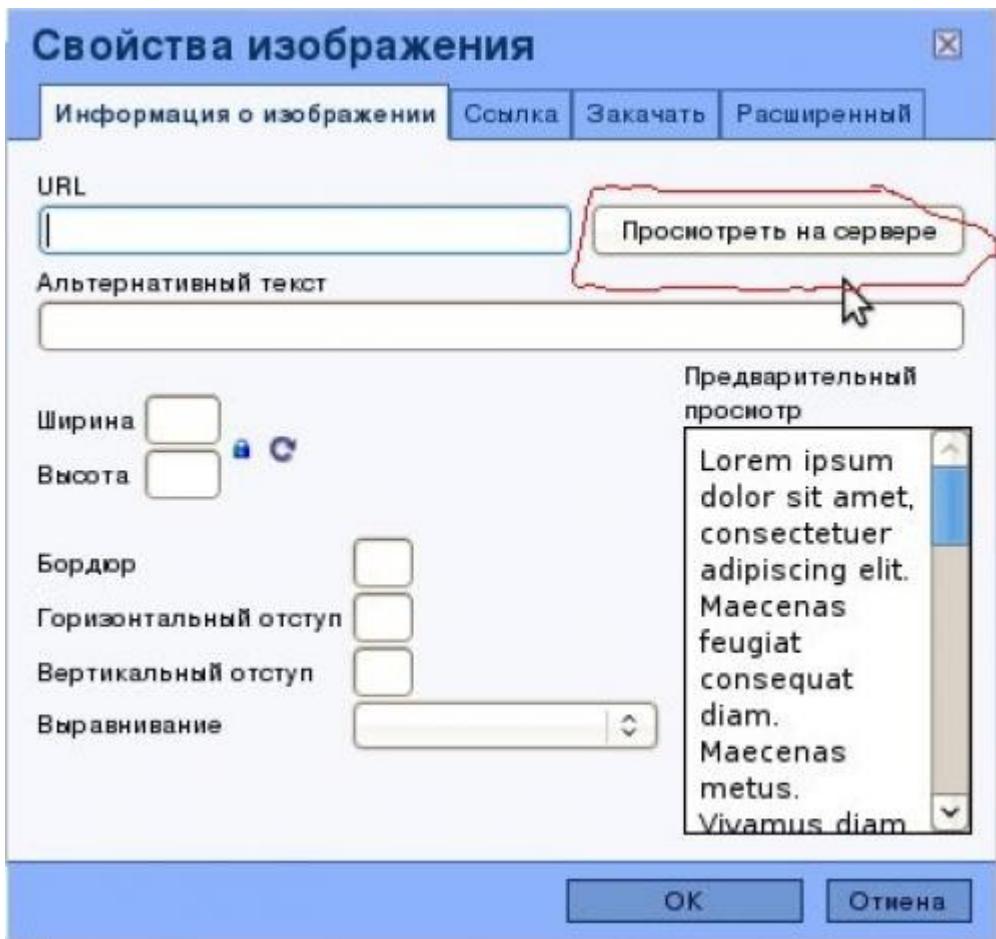
Если вы все сделали правильно, то меню FCKeditor должно принять вид



а в меню добавления изображения должна появиться кнопка «посмотреть на сервере»

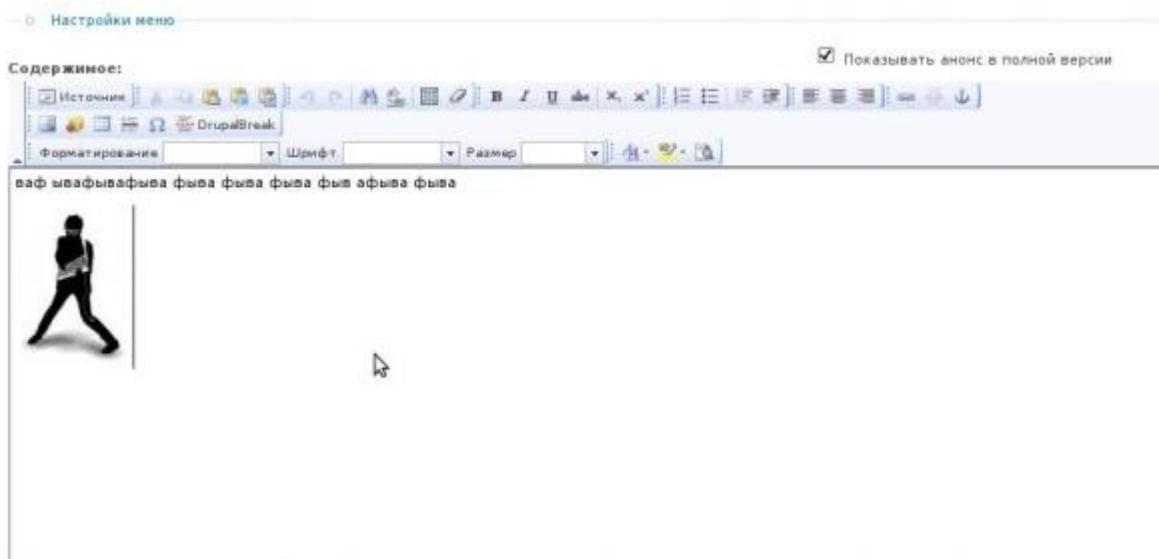


Теперь давайте попробуем что нибудь закачать -) Нажимаем кнопку добавления рисунка «рис» в появившемся меню жмем кнопку «посмотреть на сервере»



Видим папки находящиеся на сервере, за загрузку изображений отвечает ссылка загрузить, жмем «обзор», выбираем нужный нам рисунок, и нажимаем «загрузить», через мгновение файл появляется на сервере, теперь его можно скаллировать (подогнать под нужные нам размеры) с помощью ссылки «Resize», и обрезать с помощью ссылки «Crop», для вставки его в наш документ, нужно нажать ссылку «Send to FCKEditor».

Готово!



Чтобы рисунки отобразились, так же нужно установить Формат ввода в «Full HTML».

Из соображений безопасности формат ввода «Full HTML» можно давать лишь администратору, или модератору, а обычным пользователям «Filtered HTML», в который нужно добавить необходимые теги, как это сделать мы обсудим позже.

Модуль Similar By Terms.

Модуль Similar By Terms служит для создания блока "Похожие статьи", это полезно для SEO, а так же удобно для пользователей, скачать его можно [тут](#), установка довольно таки проста, и во многом похожа на установку модуля Tagadelic, который мы обсуждали во втором уроке. Будем считать что это дамашнее задание -)

На этом урок закончен.

Не совсем уверен о чем мы будем разговаривать в следующей статье, но скорее всего мы прикрутим блок «календарь», который будет выводить наши записи по датам.а внимание -)

Drupal - Урок 4 - Разбираемся с модулем Views. Теория и практика.

By RemaGe - Posted on 02 Июль 2009

Одним из самых полезных **модулей Drupal** для **создания сайта**, является модуль **Views**, кроме того, он так же является одним из самых сложных, и наверное самым большим (архив Drupal 6-12 весит около мегабайта, архив Views 6.x-2.6 весит полтора мегабайта).

Суть модуля проста, - он выводит данные в нужном нам виде, вывод можно настраивать по многим параметрам, и всячески преобразовать. Для начала модуль нужно [скачать](#), и установить (это мы обсуждали в прошлых уроках).

Активировать нужно 2 модуля (они идут в 1 архиве **Views**) **Views** и **Views UI**, первый выводит списки из базы так как вы задали, второй — это визуальная оболочка, с помощью которой создаются Views. После активации в меню «Administer → Site building» (Управление → Конструкция сайта) появится вкладка Views. Зайдем.

The screenshot shows the 'Views' administration page. At the top, there are tabs for 'Views' (selected), 'Create' (with a 'Add' button), 'Import', and 'Tools'. Below the tabs, there's a message about the advanced help module. A search bar and filter buttons ('Storage', 'Type', 'Tag', 'Displays') are present. A sorting section ('Sort by: Имя', 'Order: Up', 'Apply') follows. The main area lists several views:

- По умолчанию Материнский view: archive (default)**
Title: Monthly archive
Path: archive
Блок, Страница
Status: Enable
- По умолчанию Комментарий view: comments_recent (default)**
Title: Последние комментарии
Path: comments/recent
Блок, Страница
Status: Enable
- По умолчанию Материнский view: frontpage (default)**
Path: frontpage
Блок, Страница
Status: Enable
- По умолчанию Материнский view: glossary (default)**
Path: glossary
Страница
Status: Enable
- По умолчанию Материнский view: taxonomy_term (default)**
Path: taxonomy/term/%
Блок, Страница
Status: Enable
- По умолчанию Материнский view: tracker (default)**
Title: Последние публикации
Path: tracker
Страница
Status: Enable

Здесь мы видим виды которые уже созданы по умолчанию, их может быть и больше в зависимости от установленных модулей. Эти виды неактивны, их можно включить с помощью ссылки «Enable», и использовать в своих целях. Но, !!МЫ!! как истинные «джедаи» попробуем сделать свой (пусть он и будет маленький, неказистый и неработающий). Кликаем по кнопке «Add».

Появляется первое меню создания вида.

View name: *

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

View description:

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

View tag:

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

View type:

Материал
Nodes are a Drupal site's primary content.

File
Files maintained by Drupal and various modules.

Locale source
A source string for translation, in English or the default site language.

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

Term
Taxonomy terms are attached to nodes.

Комментарий
Comments are responses to node content.

Пользователь
Users who have created accounts on your site.

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

Next >

Тут 2 обязательных поля: заголовок (название вида., вводить английские буквы в нижнем регистре), и тип вида (View type — это то откуда мы будем брать информацию для вывода). Напишем в заголовке first (ведь это наш первенец, не так ли?), а тип вида поставим «Материал» (Node), будем выводить материалы. Нажимаем «Next», и попадаем на страницу где мы будем задавать параметры вывода, первоначально она выглядит ужасно, но это только в первый раз.

If you install the advanced help module from http://drupal.org/project/advanced_help, views will provide more and better help. [Hide this message.](#)

View first, displaying items of type Материал.

Экспортировать Clone

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

Страница Add display Analyze

View settings
Tag: None

Basic settings
Имя: Defaults
Заголовок: None
Style: Unformatted
Row style: Fields
Use AJAX: Нет
Use pager: Нет
Items to display: 10
More link: Нет
Distinct: Нет
Access: Unrestricted
Caching: None
Exposed form in block: Нет
Заголовок: None
Подвал: None
Empty text: None
Theme: Information

Relationships
None defined

Arguments
None defined

Fields
Материал: Заголовок: Заголовок

Sort criteria
None defined

Фильтры
None defined

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

Сохранить Отмена Удалить

Live preview

Display: Arguments:

Defaults Предпросмотр

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

тут можно создать несколько display (видов вывода, наиболее используемые это блок, и страница). Один display уже есть — default, если вы хотите создать несколько схожих дисплеев, то можно задать основные настройки в default, создаваемые display наследуют эти настройки. Если вы хотите что бы настройки у дисплея отличались от настроек Default, их нужно переназначить, для этого служит кнопка Override.

Block: How should this view be styled

Status: using default values.

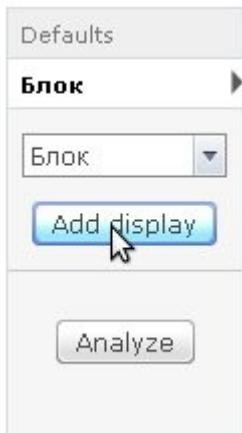
Override

Calendar
 Grid
 HTML List
 Table
 Unformatted

If the style you choose has settings, be sure to click the settings button that will appear next to it in the View summary.
You may also adjust the settings for the currently selected style by clicking on the icon.

Update default display Cancel

Давайте добавим блок.

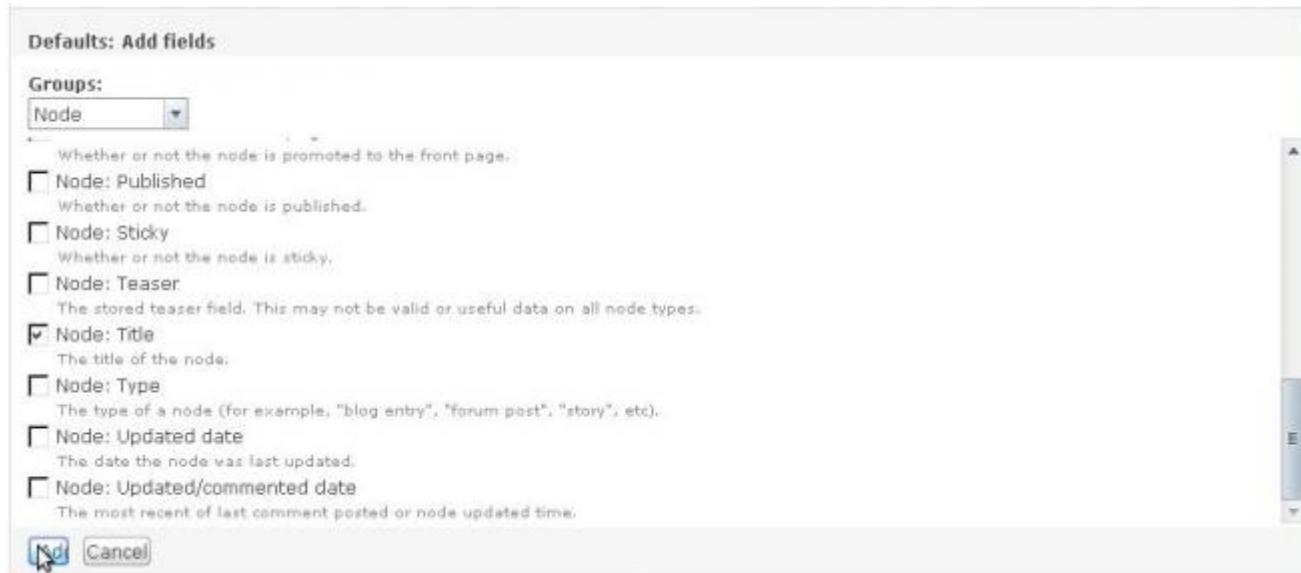


Рассмотрим блок Fields, он является одним из основных, так как в нем мы выбираем какие поля (Fields) будет отображать в нашем виде.



На блоке вы видите 2 кнопки, первая служит для добавления поля, вторая для выставления полей в порядке нужном вам.

При нажатии на кнопку добавления поля, внизу отобразятся поля которые можно добавить, и выпадающий список сортировки. Если мы хотим добавить в вывод заголовки материалом, нам нужно выбрать Node: title, и нажать Add.



Появится меню Configure field, в котором мы можем задать Имя поля, с которым он будет выводиться, и проставить галочки:

Exclude from display: загружать поле, но не отображать.

Rewrite the output of this field: заменить вывод этого поля на свойтекст.

Output this field as a link: Выводимый текст можно сделать ссылкой.

Trim this field to a maximum length: можно указать максимальный размер текста, остальное обрежется.

Link this field to its node: запись будет ссылкой, в зависимости от поля, адрес ссылки меняется.

Defaults: Configure field Node: Title

Label:

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

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

Rewrite the output of this field
If checked, you can alter the output of this field by specifying a string of text with replacement tokens that can use any existing field output.

Output this field as a link
If checked, this field will be made into a link. The destination must be given below.

Trim this field to a maximum length
If checked, this field will be trimmed to a maximum length in characters.

Link this field to its node
This will override any other link you have set.

Update **Cancel** **Remove**

Остановимся поподробнее на блоке «**Basic settings**», здесь собраны основные свойства нашего вида. При нажатии какого либо свойства внизу появляется панель редактирования.

View first, displaying items of type Материал.

Блок: *Display the material as Block*

Basic settings

- Имя: Блок
- Заголовок: None
- Style: **Unformatted**
- Raw style: *Fields*
- Use AJAX: *Hi* Change the style plugin.
- Use pager: *Нет*
- Items to display: *10*
- More link: *Нет*
- Distinct: *Нет*
- Access: *Unrestricted*
- Caching: *None*
- Exposed form in block: *Нет*
- Заголовок: None
- Логика: None
- Empty text: None
- Theme: Information

Block settings

- Admin: None
- Caching: Do not cache

Блок: How should this view be styled?

Status: using default values.

Grid
 HTML List
 Unformatted
 Таблица

If the style you choose has settings, be sure to click the settings button that will appear next to it in the View summary.
You may also adjust the settings for the currently selected style by clicking on the icon.

Override

Update default display **Отмена**

Сохранить **Отмена** **Удалить**

Обсудим основные свойства:

Name: имя дисплея, можно не менять, но когда дисплеев много, помогает не запутаться.

Title: Заголовок дисплея (в нашем случае заголовок блока, по умолчанию).

Style: Стиль вывода. Очень важное свойство, определяет как будет выводиться ваш вид. Всего 4 варианта (если не добавлять модули):

Grid — данные выводятся «решеткой», можно задать количество столбцов, выравнивание (по горизонтали/вертикали) и нужно ли группировать по какому либо полю.

фывфывфыв фывыв ыв	ывавы
йцуйцуйцу	ывапыва
1111111	1231ads
1231ads	

на рисунке выводятся заголовки страниц, **Style: Greed**, **Grouping field: None**, **Number of columns: 4**, **Alignment:Horizontal**.

HTML list: обычный HTML список, из свойств **Grouping field** (аналогичен тому что в Greed), и **List type** (Unordered- обычный, Ordered — нумерованный)

1. фывфывфыв фывыв ыв
2. ывавы
3. йцуйцуйцу
4. ывапыва
5. 1111111
6. 1231ads
7. 1231ads



те же самые заголовки, но **Style: HTML list**, **List type: Ordered**.

Unformatted — данные выводятся без форматирования, есть возможность их сгруппировать.

фывфывфыв фывыв ыв
ывавы
йцуйцуйцу
ывапыва
1111111
1231ads
1231ads



Style: Unformatted

Table: поля (field) выводятся в столбцах таблицы, можно задать сортировку по любому полю.

1231ads	06/30/2009 - 21:37
1231ads	06/30/2009 - 21:37
1111111	06/30/2009 - 21:36
ывапыва	06/28/2009 - 21:41
йцуйцуйцу	06/27/2009 - 18:22
ывавы	06/27/2009 - 17:44
фывфывфыв	06/27/2009
Фывывыв	- 17:36

выведены 2 поля: заголовок материала, и дата публикации, сортируется по дате публикации. **Style: Table, Default sort order: Descending** (нисходящая).

Use AJAX: использовать ли AJAX для вывода вида. Если, например, используется постраничный вывод, то переход между страницами будет происходить без перезагрузки страницы.

Use pager: Выводить ли данные постранично, страницы можно показывать 2 способами: **Full** (1 2 3 4 следующая > последняя ») и **Mini** (1 of 4 >>), свойство **Pager element** вам понадобится при использовании нескольких pager в одном виде.

Items per page: Сколько записей выводить на 1 странице.

Distinct: если включить, то будут отображаться только уникальные записи.

Access: задает доступ к виду, Role — только пользователям с определенной ролью, Permission (Право доступа) — только пользователям с определенным правом доступа (User management → Permission).

Caching: будет ли кешироваться блок.

Header, Footer: здесь можно вставить текст который будет до нашего блока, и после, возможна вставка HTML.

Empty text: этот текст выведется в том случае, если вид не вернет записей.

Theme: можно задать стилю какой темы будет соответствовать наш блок.

Блок Sort criteria: по любому полю можно задать сортировку по возрастанию

или убыванию.

Блок Filters служит для задания фильтров вывода. Например можно вывести только те записи, у которых id меньше 5, или Node: Type равен «Blog entry».

Модуль **Views** очень обширен, мы рассмотрели основы, остальное я надеюсь вы разберете сами. Теперь давайте от теории перейдем к практике, выполните несколько заданий (ниже я приведу решение, но желательно что бы вы справились сами):

1. Создайте страницу в которой будут выводиться последние 5 комментариев (автор, текст обрезаный до 20 символов, и дата публикации).
2. Создайте RSS-поток который будет выводить данные из предыдущего дисплея(возможно тут возникнут некоторые проблемы, но я в вас верю).
3. С помощью модуля [calendar](#), создайте блок с календарем, что бы по клику на даты выводились записи из блога за эту дату.

Первая задача.

1. Создаем вид с **View type: Comment**, назовем его **last_comments**.

Views List **Add** Import Tools

View name: *

last_comments

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

View description:

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

View tag:

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

View type:

Node

Nodes are a Drupal site's primary content.

Comment

Comments are responses to node content.

File

Files maintained by Drupal and various modules.

Node revision

Node revisions are a history of changes to nodes.

User

Users who have created accounts on your site.

The view type is the primary table for which information is being retrieved. The view type controls what arguments, fields, sort criteria, and filters can be used. The view type cannot be changed.

Next

2. В только что созданном виде создаем дисплей типа page



3. Добавим 3 поля: Автор комментария (Comment: Author), текст (Comment: Body), и дату создания (Comment: Post date).

BLOCK Display the view as a BLOCK.

Basic settings

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

Relationships

None defined

Arguments

None defined

Fields

Comment: Author **Author**

Comment: Body Body
Comment: Post date Post date

Block settings

Admin: None
Caching: Do not cache

Block: Configure field **Comment: Author**

Status: using default values.

Label:

Author

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

Exclude from display

Check this box to not display this field, but still load it in the view. Use this option to not show a grouping for theming.

Rewrite the output of this field

If checked, you can alter the output of this field by specifying a string of text with replacement tokens that

Output this field as a link

If checked, this field will be made into a link. The destination must be given below.

Trim this field to a maximum length

If checked, this field will be trimmed to a maximum length in characters.

Link this field to its user or an author's homepage

[Update default display](#) [Cancel](#) [Remove](#)

4. настроим поле Автор. Удалим Label, и зделаем имя ссылкой на домашнюю страницу.

Use pager: No
Items to display: 10
More link: No
Distinct: No
Access: Unrestricted
Caching: None
Exposed form in block: No
Header: None
Footer: None
Empty text: None
Theme: Information

Fields	+	↑↓
Comment: Author		
Comment: Body Body		
Comment: Post date Post date		

Block settings

Admin: None
Caching: Do not cache

Block: Configure field **Comment: Author**

Status: using default values.

Label:

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

Exclude from display

Check this box to not display this field, but still load it in the view. Use this option to not show a grouping theming.

Rewrite the output of this field

If checked, you can alter the output of this field by specifying a string of text with replacement tokens th

Output this field as a link

If checked, this field will be made into a link. The destination must be given below.

Trim this field to a maximum length

If checked, this field be trimmed to a maximum length in characters.

Link this field to its user or an author's homepage

5. Настроим Body, удалим Label, и поставим обрезку до 20 символов.

Block: Configure field *Comment: Body*

Status: using default values.

Label:

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

Exclude from display

Check this box to not display this field, but still load it in the view. Use this option to not show a grouping field in theming.

Rewrite the output of this field

If checked, you can alter the output of this field by specifying a string of text with replacement tokens that can us

Output this field as a link

If checked, this field will be made into a link. The destination must be given below.

Trim this field to a maximum length

If checked, this field be trimmed to a maximum length in characters.

Maximum length:

20

The maximum number of characters his field can be.

Trim only on a word boundary

If checked, this field be trimmed only on a word boundary. This is guaranteed to be the maximum characters stat boundaries this could trim a field to nothing.

Add an ellipsis

If checked, a "..." will be added if a field was trimmed.

Strip HTML tags

If checked, all HTML tags will be stripped. 

Field can contain HTML

If checked, HTML corrector will be run to ensure tags are properly closed after trimming.

[Update default display](#)

[Cancel](#)

[Remove](#)

6.У даты удалим Label.

7.Сохраним дату (1), и сохраним весь вид(2)

Block: Configure field Comment: Post date

Status: using overridden values.

Label:

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

Exclude from display
Check this box to not display this field, but still load it in the view. Use this option to not show a group in theming.

Rewrite the output of this field
If checked, you can alter the output of this field by specifying a string of text with replacement tokens th

Output this field as a link
If checked, this field will be made into a link. The destination must be given below.

Trim this field to a maximum length
If checked, this field will be trimmed to a maximum length in characters.

Date format:

 1

2

8. Зададим количество комментариев Items to display: 5.

9. Поставим Style: Table, сортировку по дате на понижение.

Name: Defaults
 Title: None
Style: Table
 Use AJAX: No
 Use pager: No
 Items to display: 5
 More link: No
 Distinct: No
 Access: Unrestricted
 Caching: None
 Exposed form in block: No
 Header: None
 Footer: None
 Empty text: None
 Theme: Information

None defined

None defined

Fields

Comment: Author Author
 Comment: Body Body
 Comment: Post date Post date

Defaults: Style options

Place fields into columns; you may combine multiple fields into the same column. If you do, the separator in the column specified will be used to separate the fields. Check the sortable box to make that column click sortable, and check the default sort radio to determine which column will be sorted by default, if any. You may control column order and field labels in the fields section.

Field	Column	Separator	Sortable	Default sort
Author	Author ▾		<input checked="" type="checkbox"/>	<input checked="" type="radio"/>
Body	Body ▾		<input type="checkbox"/>	<input type="radio"/>
Post date	Post date ▾		<input checked="" type="checkbox"/>	<input checked="" type="radio"/>
None				

Grouping field:
 You may optionally specify a field by which to group the records. Leave blank to not group.

Override normal sorting if click sorting is used
 Enable Drupal style "sticky" table headers (Javascript)
(Sticky header effects will not be active for preview below, only on live output.)

Default sort order:
 If a default sort order is selected, what order should it use by default?

10. Последнее что нужно сделать, это задать путь (я вбил «comm»), и сохранить,

Page settings
Path: comm
 Menu: No menu

Page: The menu path or URL of this view

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

если все сделано правильно, то по ссылке ваш_сайт/comm будет доступна наша страница

Author	Body	Post date
admin	Всем пока	07/02/2009 - 00:27
admin	Всем привет	07/02/2009 - 00:27
admin	sdfghsdfh	07/01/2009 - 21:11
qwe	An example	06/28/2009 - 23:50

→

 Drupal

Вторая задача.

1.дисплей у нас уже настроен, нужно только прикрутить к нему RSS. Заходим в на вид last_comments. И создаем новый дисплей Feed.



2.Изменяем Row style:на Comment.

Style: RSS Feed

Row style: Comment

Items to display: 5

Distinct: No

Access: Unrestricted

Caching: None

Exposed form in block: No

Header: None

Footer: None

Empty text: None

Theme: Information

Feed settings

Path: None
Attach to: None

Feed: How should each row in this view be

Comment

3. Вбиваем путь (**Path**) по которому будет располагаться наш RSS. Я поставил «rsscomm»

Feed settings

Path: None

Attach to: None

Feed: The menu path or URL of this view

This view will be displayed by visiting this path on your site. It is recommended that the path be something like "path/%/%/feed" putting one % in the path for each argument you have defined in the view.

4. Присоединяем RSS-поток к дисплею Page.

Feed settings

Path: rsscomm

Attach to: None

Feed: Attach to

Defaults
 Page

The Feed icon will be available only to the selected displays.

5. Сохраняемся.

Access: unrestricted
Caching: None
Exposed form in block: No
Header: None
Footer: None
Empty text: None
Theme: Information

Feed settings
Path: rsscomm
Attach to: Page

The style selected does not utilize fields.

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

Save Cancel Delete

6.Заходим на страницу нашего вида, если все было сделано правильно, внизу появится значек RSS.

Author	Body	Post date
admin	Всем пока	07/02/2009 - 00:27
admin	Всем привет	07/02/2009 - 00:27
admin	sdfghsdfh	07/01/2009 - 21:11
qwe	An example	06/28/2009 - 23:50

Drupal

При клике на который появится сам RSS-поток, в Opere он выглядит так:

Подписаться на эту ленту, используя Opera Mail Now!

http://localhost/drupal-6.12/rsscomm

Всем пока

Thursday July 02, 00:27:40 GMT+1000 2009
admin

Всем пока

Всем привет

Thursday July 02, 00:27:29 GMT+1000 2009
admin

Всем привет

sdfghsdfh

Wednesday July 01, 21:11:00 GMT+1000 2009
admin

sdfghsdfh

Opera Mail Now!

Эта страница была создана Opera из http://localhost/drupal-6.12/rsscomm

Самый трудный пример, и самый последний (ну наконец то)

1. Первоначально нам нужно [скачать](#) дополнительный модуль и установить его, который в свою очередь требует модули [Date Api](#) и [Date timezone](#), скачиваем и устанавливаем их.

Enabled	Name	Version	Description
<input checked="" type="checkbox"/>	Calendar	6.x-2.1	Views plugin to display views containing dates as Calendars. Depends on: Views (enabled), Date API (enabled), Date Timezone (enabled) Required by: Calendar iCal (disabled), Calendar Popup (disabled)
<input type="checkbox"/>	Calendar iCal	6.x-2.1	Adds ical functionality to Calendar views. Depends on: Views (enabled), Date API (enabled), Calendar (enabled), Date Timezone (enabled)
<input type="checkbox"/>	Calendar Popup	6.x-2.1	Replaces the links to calendar items with a javascript popup that gracefully regresses if javascript is disabled. Depends on: Calendar (enabled), Views (enabled), Date API (enabled), Date Timezone (enabled)
<input type="checkbox"/>	Date	6.x-2.2	Defines CCK date/time fields and widgets. Depends on: Content (missing), Date API (enabled), Date Timezone (enabled) Required by: Date Tools (disabled)
<input checked="" type="checkbox"/>	Date API	6.x-2.2	A Date API that can be used by other modules. Required by: Calendar (enabled), Calendar iCal (disabled), Date (disabled), Date Locale (disabled) (disabled), Date Repeat API (disabled), Date Timezone (enabled), Date Tools (disabled), Calendar (disabled)
<input type="checkbox"/>	Date Locale	6.x-2.2	Allows the site admin to configure multiple formats for date/time display to tailor dates for a specific locale. Depends on: Date API (enabled), Locale (disabled)
<input type="checkbox"/>	Date PHP4	6.x-2.2	Emulate PHP 5.2 date functions in PHP 4.x, PHP 5.0, and PHP 5.1. Required when using the Date module. Depends on: Date API (enabled)
<input type="checkbox"/>	Date Popup	6.x-2.2	Enables jquery popup calendars and time entry widgets for selecting dates and times. Depends on: Date API (enabled), Date Timezone (enabled)
<input type="checkbox"/>	Date Repeat API	6.x-2.2	A Date Repeat API to calculate repeating dates and times from iCal rules. Depends on: Date API (enabled)
<input checked="" type="checkbox"/>	Date Timezone	6.x-2.2	Needed when using Date API. Overrides site and user timezone handling to set timezone names correctly. Depends on: Date API (enabled) Required by: Calendar (enabled), Date (disabled), Date Popup (disabled), Calendar iCal (disabled), Calendar Popup (disabled)
<input type="checkbox"/>	Date Tools	6.x-2.2	Tools to import and auto-create dates and calendars. Depends on: Content (missing), Date (disabled), Date API (enabled), Date Timezone (enabled)

2. Создаем новый вид, цепляем его к Node.

View name: *

This is the unique name of the view. It must contain only alphanumeric characters and underscore template names for this view. If overriding a module provided view, the name must not be changed.

View description:

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

View tag:

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

View type: Node

Nodes are a Drupal site's primary content.

 Comment

Comments are responses to node content.

 File

Files maintained by Drupal and various modules.

 Node revision

Node revisions are a history of changes to nodes.

 User

Users who have created accounts on your site.

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

[Next](#)

3. Создаем дисплей Block.



4. Свойство Style: выставляем равным Calendar (Этот стиль появился у нас после установки модуля Calendar).

Basic settings

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

Block settings

Admin: None
Caching: Do not cache

Block: How should this view be styled

Status: using default values.

- Calendar
- Grid
- HTML List
- Table
- Unformatted

If the style you choose has settings, be sure to

You may also adjust the **settings** for the current

[Update](#) [Default display](#) [Cancel](#)

5.Добавляем аргумент Date: Date (node).

Basic settings

Name: Block
 Title: None
Style: Calendar
 Use AJAX: No
 Use pager: No
 Items to display: 10
 More link: No
 Distinct: No
 Access: Unrestricted
 Caching: None
 Exposed form in block: No
 Header: None
 Footer: None
 Empty text: None
 Theme: Information

Relationships

None defined

Arguments

None defined

Fields

None defined

Block settings

Admin: None
 Caching: Do not cache

Block: Add arguments

Groups:

Date

Date: Date (node)
 Filter any Views node date field by a date argument, using any common ISO date/period format (i.e MM-DD--P3M, P90D, etc).

Date: Date (users)
 Filter any Views users date field by a date argument, using any common ISO date/period format (i.e MM-DD--P3M, P90D, etc).

Add **Cancel**

6. На странице редактирования аргумента устанавливаем галочки **Provide default argument, Current date, Month, Node: Post date.**

- Summary, sorted ascending
- Summary, sorted descending
- Provide default argument

Provide default argument options

Wildcard title:

Bce

The title to use for the wildcard

Default argument type:

- Fixed entry
- Node ID from URL
- PHP Code
- User ID from URL
- User ID from logged in user
- Current date
- Current node's creation time
- Current node's update time

Validator options

Validator:

<Basic validation> ▾

Action to take if argument does not validate:

Hide view / Page not found (404) ▾

Granularity:

- Year
- Month
- Day
- Hour
- Week

Select the type of date value to be used in defaults, summaries, and navigation. For example, a granularity of 'current month' summarize by month in summary views, and link to the next and previous month when using date

Date year range:

-3:+3

Set the allowable minimum and maximum year range for this argument, either a -X:+X offset from the current year, or a specific year range, like '2005:2010'. When the argument is set to a date outside the range, the page will be

Date field(s):

- Comment: Post date
- Node: Last comment time
- Node: Post date
- Node: Updated date
- Node revision: Created date
- User: Created date
- User: Last access
- User: Last login

Select one or more date fields to filter with this argument. Do not select both the 'From date' and 'To date' for needed.

Method:

7.Добавляем поля **Node: Post date**, и **Node: Title**, у первого выставляем **Date format** в Custom, **Label** удаляем. У второго ставим галочку **Exclude from**

display.

8. В блоке **Filters** добавляем параметр **Node: Type**, и выставляем параметр **Is one of — Blog Entry**

Defaults: Configure filter Node: Type

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

Operator:

Is one of
 Is not one of

Node type:

Blog entry
 Page
 Story

Buttons:
Update Cancel Remove

9. Все сохраняем, выводим блок, должно получиться так



Не очень красиво, но работает, внешний вид — это тема для другого урока. Но пока ничего не выводит. Почему? Потому что нужно создать вид **Page** на котором будут выводиться выбранные материалы. Собираем волю в кулак, осталось совсем чуть чуть.

1. В этом же виде создаем дисплей **Page**.

2. В **Page settings** ставим **Path: callendar/%** (% - это то место куда наш блок будет подставлять дату, что бы Page знал данные за какую дату отображать)

Page settings

Path: calendar/%
Menu: No menu

Page: The menu path or URL of this view

http://localhost/drupal-6.12/ calendar/%

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

Update **Cancel**

3. Переопределяем (кнопочка **Override**) у Page аргумент **Date: Date (node)** **Node: Post date**, меняем Month на Day (что бы на страничке выводился только один день).

Page: Configure Argument Date: Date (node)

Status: using overridden values.

Title:
The title to use when this argument is present. It will override the title of the view and titles from previous arguments. You can use percent substitution here to replace with argument titles. Use "%1" for the first argument, "%2" for the second, etc.

Breadcrumb:
The Breadcrumb title to use when this argument is present. If no breadcrumb is setted here, default Title values will be used; see "Title" for percent substitutions.

Action to take if argument is not present:

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

Provide default argument options

Default argument type:

- Fixed entry
- Node ID from URL
- PHP Code
- User ID from URL
- User ID from logged in user
- Current date
- Current node's creation time
- Current node's update time

Wildcard:
If this value is received as an argument, the argument will be ignored; i.e., "all values".

Wildcard title:
The title to use for the wildcard in substitutions elsewhere.

Validator options

Validators: <Basic validation> ▾

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

Granularity:

- Year
- Month
- Day
- Hour
- Week

Select the type of date value to be used in defaults, summaries, and navigation. For example, a granularity of 'month' will set the default date to the current month, summarize by month in summary views, and link to the next and previous month when using date navigation.

4. В блоке **Fields** переопределяем **Node: Title**, убираем галочку **Exclude from display**, и ставим **Link this field to its node**, что бы заголовки отображались, и были ссылкой на материал.

5. Сохраняем. Пробуем. Теперь при клике по соответствующей дате будут выведены записи блога созданные в тот день. Вы можете изменить параметр **Style** у дисплея **Page**, пробуйте и у вас получится. А как темизировать виды, я раскажу вам как нибудь потом)

На этом урок по созданию сайта на Drupal закончен. Но не законченны уроки -)

Drupal - Урок 5 - Меню и то с чем его едят. Создание выпадающих меню, и развертывающихся без перезагрузки страницы.

By RemaGe - Posted on 04 Июль 2009

Прошлый урок был непрост. Поэтому сегодня мы слегка пофилоним) Давайте разберемся с такой важной частью сайта, как меню. Создание меню настолько же универсально в **Drupal**, как и все остальное. Урок проводится на чистом сайте Drupal-6.13, на английском языке из дополнительных модулей только **Pathauto**.

Управление меню сконцентрировано в «Administer > Site building > Menus».

The screenshot shows the Drupal administration interface under 'Site building > Menus'. The 'List menus' tab is active. Below it, there's a brief description of what menus are: 'Menus are a collection of links (menu items) used to navigate a website. The menus currently available on your site are displayed below. Select a menu from this list to manage its menu items.' To the right of this text is a link '[more help...]' with a small arrow icon. The main content area is divided into three sections: 'Navigation', 'Primary links', and 'Secondary links'. Each section has a brief description and a small icon. A red box highlights the 'List menus' tab and the first three menu items ('Navigation', 'Primary links', 'Secondary links').

В самом верху мы видим 3 управляющие кнопки: List menus (Вывести список меню системы), Add menus (Создать новое меню), Settings (Опции, вернемся к ним чуть позже).

Изначально в Drupal создано 3 меню: Navigation, Primary links, Secondary links. Первое из них основное, оно включено по умолчанию (на самом деле работают все 3, но пункты\подпункты есть только в первом) и отображается так (Если вы вошли на сайт под профилем администратора, у простых смертных меню изначально поменьше, но сути это не меняет):

rem

- [My account](#)
- ▷ [Create content](#)
- ▷ [Administer](#)
- [Log out](#)

Кроме стандартных меню, вы можете создать сколь угодно много своих, при создании к нему сразу же создается блок, и его можно поместить в любой регион (справочка: окно сайта в Drupal делится на регионы, а в них уже вставляются блоки). Внимательный читатель заметит - «Для чего же созданы меню Primary links и Secondary links, если они изначально пустые, и я могу создать сколько захочу таких же?». Как я понимаю, основная причин в том, что почти любой дизайн Drupal, включает 2 региона созданных специально для меню, у них свой внешний вид, и они более подходят под дизайн стилистически, давайте попробуем добавить в меню Primary links и Secondary links по пункту, и посмотреть как же эти меню будут выглядеть в стандартной теме Drupal.

Кликаем по ссылке Primary links.

Primary links

List items Add item Edit menu

To rearrange menu items, grab a drag-and-drop handle under the *Menu i* handle by clicking and holding the mouse while hovering over a handle icon configuration button at the bottom of the page.

There are no menu items yet.

Сверху так же 3 кнопки: List items (список пунктов меню), Add items (Добавить пункт), Edit menu (Редактировать меню). Нажимаем Add items.

Primary links

[List items](#)[Add item](#)[Edit menu](#)

Menu settings

Path: *

The path this menu item links to. This can be an internal Drupal path such as `node/add` or an external URL.

Menu link title: *

The link text corresponding to this item that should appear in the menu.

Description:

The description displayed when hovering over a menu item.

 Enabled

Menu items that are not enabled will not be listed in any menu.

 Expanded

If selected and this menu item has children, the menu will always appear expanded.

Parent item:

<Primary links> ▾

The maximum depth for an item and all its children is fixed at 9. Some menu items may not be visible.

**Weight:**

0 ▾

Optional. In the menu, the heavier items will sink and the lighter items will be positioned nearer to the top.

[Save](#)

И видим окно добавления меню:

Path — адрес страницы на которое ведет меню, сейчас для нас не важно, но поле обязательно, по этому введите (например) — «`node`», и меню будет ссылаться на материалы (это главная страница по умолчанию).

Menu link title — как будет выглядеть этот пункт в меню, пусть будет «О авторе».

Enabled — включен ли пункт меню.

Expanded — если у пункта есть подпункты, то он будет по умолчанию развернут.

Parent item — к какому меню прицепим пункт..

Weight — вес пункта, нужен для задания расположения пунктов относительно друг друга. Легкие идут в начале, тяжелые в конце.

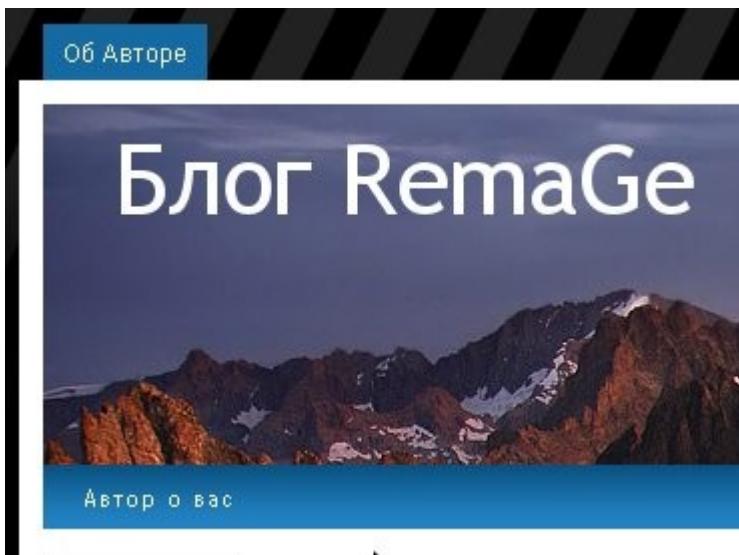
Сохраним, и проделаем тоже самое с Secondary links, только в Menu link title забьем «Автор о вас». У меня получилось так:

The screenshot shows a list of menu items. One item is selected, indicated by a blue background. The menu item has the following details:

Enabled	Expanded	Operations
<input checked="" type="checkbox"/>	<input type="checkbox"/>	edit delete

The link title is "Автор о вас". The menu item is part of a list where other items like "lit menu" and "Secondary links" are also visible.

В теме которая установлена на этом блоге, аналогичные пункты меню выглядят так:



По желанию в эти регионы можно вставить любое меню, это можно настроить в «Administer > Site building > Menus > Settings».

Menus

List menus

Add menu

Settings

The menu module allows on-the-fly creation of menu links in the menu. You can add new menu links to any menu by clicking the "Add menu link" button.

Default menu for content:

Primary links ▾

Choose the menu to be the default in the menu options in the content type settings.

Source for the primary links:

Primary links ▾

Select what should be displayed as the primary links.

Source for the secondary links:

Secondary links ▾

Select what should be displayed as the secondary links. You can choose from the list of menus. The active primary menu link will be displayed as secondary links.

[Save configuration](#)

[Reset to defaults](#)

Default menu for content — для любой добавленной записи можно создать ссылку в меню, этот параметр определяет какое меню там будет по умолчанию.

Source for the primary links — задаем какое меню будет в первом регионе.

Source for the secondary links — задаем какое меню во втором регионе.

Для примера давайте вставим в Source for the primary links меню Navigation.



Попробуем создать свое меню.

Создайте меню, я назвал его "Первый пошел".

Создаем контент типа «Page» (Create content > Page)

Title — введите «Пункт 1», «Body» можно не вводить.

Во вкладке «Menu settings» вбейте «Menu link title:» - «Пункт 1», это то как будет отображаться наше меню. «Parent item» - выберете «Первый пошел», и сохраните. Теперь перейдите в «Administer > Site building > Block», найдите блок «Первый пошел», и вставьте его в лодин из регионов, я перетащил в «Left sidebar».

Block	Region	Operations
Left sidebar		
+ Navigation	Left sidebar	configure
+ User login	Left sidebar	configure
+ Первый пошел	Left sidebar	configure
Drag to re-order		
Right sidebar		
No blocks in this region		
Content		
No blocks in this region		
Header		
No blocks in this region		
Footer		
+ Powered by Drupal	Footer	configure
Disabled		
+ Recent comments	<none>	configure
+ Syndicate	<none>	configure
+ Primary links	<none>	configure
+ Secondary links	<none>	configure
+ Who's new	<none>	configure
+ Who's online	<none>	configure

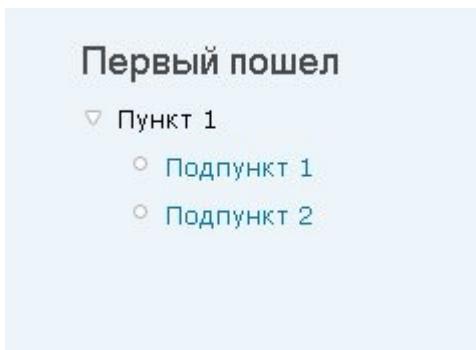
Сохраните. В левом регионе отобразился наш блок.

- ▷ [Reports](#)
- [Help](#)
- [Log out](#)

- Первый пошел**
- [Пункт 1](#)

Теперь добавьте еще 2 страницы, но «Parent item» поменяйте на «Пункт 1». У

меня вышло так:

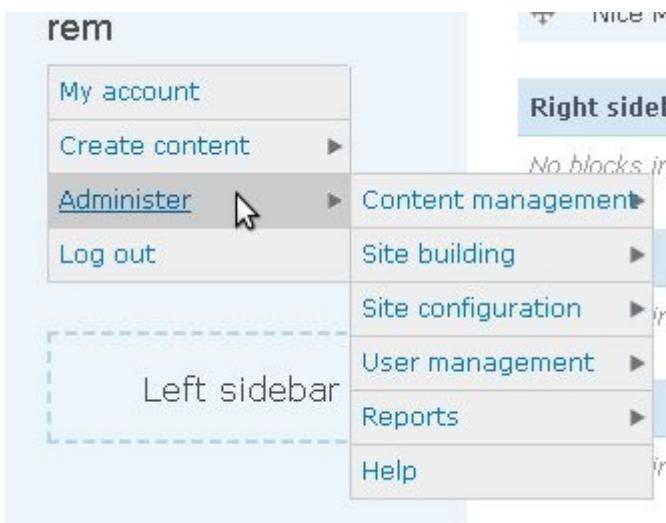


Если вы хотите добавить пункт меню на уже существующий материал, то это можно сделать в «Administer > Site building > Menus», просто войдите в меню в котором хотите добавить пункт, и нажмите «Add item». В поле «Path» укажите путь к вашему материалу.

Кроме того, в Drupal множество модулей улучшающих внешний вид меню, рассмотрим некоторые из них.

Nice menus

Функционально, и симпатично, создает выпадающие, при наведении на них мыши, меню. Можно сделать их выпадающими как горизонтально, так и вертикально.



DHTML Menu

Все меню раскрываются без перезагрузки страницы. Крайне удобно, несколько раз использовал.

YUI Menu

Выпадающее многоуровневое меню.

Большое количество модулей вы сможете найти на сайте drupal.org, неполный список можно найти [здесь](#).

На этом урок закончен.

Оставайтесь на линии, уроки о том как создать сайт на Drupal еще не окончены -)

Drupal - Урок 6 - Модуль ССК. Создаем свой content type, и добавляем к нему поля.

By RemaGe - Posted on 14 Июль 2009

Одним из модулей-столпов на которых зиждется **Drupal**, наравне с **Views**, является модуль **ССК (Content Construction Kit)**. Он позволяет создавать произвольные типы документов, а так же добавлять к ним новые поля. Скачать модуль можно [тут](#).

В пакет модулей ССК входят следующие модули:

Content — Основной модуль, без него ничего работать не будет. Позволяет создавать новые типы документов (Content types).

Content Copy — позволяет импортировать/экспортировать определение полей.

Content Permissions — позволяет устанавливать права доступа на уровне полей.

Fieldgroup — с помощью него можно группировать поля.

Node Reference — добавляет к документу ссылку на другой документ.

Number — добавляет численное поле.

Option Widgets — добавляет дополнительные типы выбора для текстовых и числовых полей.

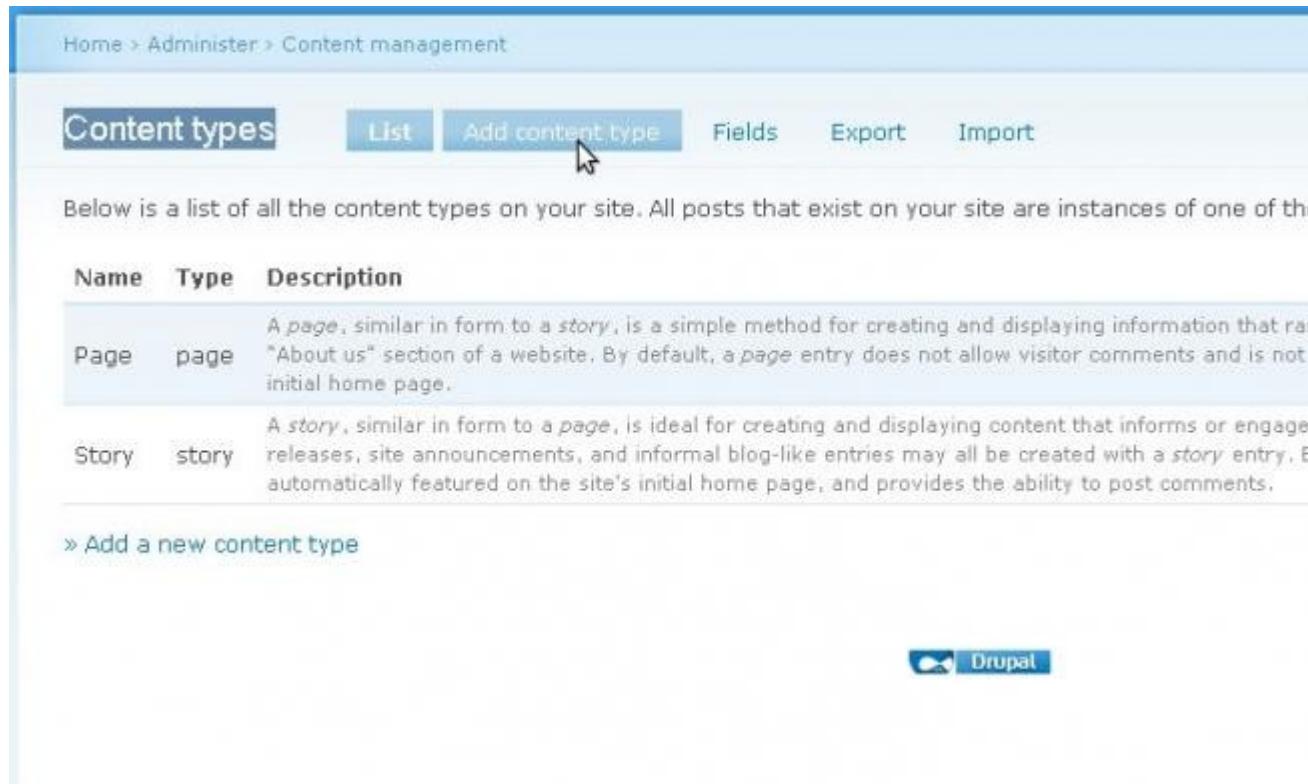
Text — добавляет текстовые поля.

User Reference — позволяет добавлять поле с ссылкой на пользователя.

Зайдем в меню «Administer > Site building > Modules» и включим их все.

Enabled	Name	Version	Description
<input checked="" type="checkbox"/>	Content	6.x-2.4	Allows administrators to define new content types. Required by: Content Copy (disabled), Content Permissions (disabled), Fieldgroup (disabled), Node Reference (disabled), Option Widgets (disabled), Text (disabled), User Reference (disabled)
<input checked="" type="checkbox"/>	Content Copy	6.x-2.4	Enables ability to import/export field definitions. Depends on: Content (disabled)
<input checked="" type="checkbox"/>	Content Permissions	6.x-2.4	Set field-level permissions for CCK fields. Depends on: Content (disabled)
<input checked="" type="checkbox"/>	Fieldgroup	6.x-2.4	Create display groups for CCK fields. Depends on: Content (disabled)
<input checked="" type="checkbox"/>	Node Reference	6.x-2.4	Defines a field type for referencing one node from another. Depends on: Content (disabled), Text (disabled), Option Widgets (disabled)
<input checked="" type="checkbox"/>	Number	6.x-2.4	Defines numeric field types. Depends on: Content (disabled)
<input checked="" type="checkbox"/>	Option Widgets	6.x-2.4	Defines selection, check box and radio button widgets for text and numeric fields. Depends on: Content (disabled) Required by: Node Reference (disabled), User Reference (disabled)
<input checked="" type="checkbox"/>	Text	6.x-2.4	Defines simple text field types. Depends on: Content (disabled) Required by: Node Reference (disabled), User Reference (disabled)
<input checked="" type="checkbox"/>	User Reference	6.x-2.4	Defines a field type for referencing a user from a node. Depends on: Content (disabled), Text (disabled), Option Widgets (disabled)

В меню «Administer > Content management > Content types» добавим новый тип документа.



Home > Administer > Content management

Content types [List](#) [Add content type](#) [Fields](#) [Export](#) [Import](#)

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

Name	Type	Description
Page	page	A page, similar in form to a story, is a simple method for creating and displaying information that requires no interaction with visitors. It is ideal for creating static pages like "About us" section of a website. By default, a page entry does not allow visitor comments and is not automatically featured on the site's initial home page.
Story	story	A story, similar in form to a page, is ideal for creating and displaying content that informs or engages visitors. Press releases, site announcements, and informal blog-like entries may all be created with a story entry. Stories are automatically featured on the site's initial home page, and provides the ability to post comments.

[» Add a new content type](#)

 Drupal

Необходимо заполнить 2 поля: **Name** (Имя для пользователя) и **Type** (имя для машины).

Content types List Add content type Fields Export Import

To create a new content type, enter the human-readable name, the machine-readable name, and all c users of your site will be able to create posts that are instances of this content type.

Identification

Name: *

The human-readable name of this content type. This text will be displayed as part of the list on the create content page and contain only letters, numbers, and spaces. This name must be unique.

Type: *

The machine-readable name of this content type. This text will be used for constructing the URL of the create content page and contain only lowercase letters, numbers, and underscores. Underscores will be converted into hyphens when constructing the L

Description:

A brief description of this content type. This text will be displayed as part of the list on the create content page.

— ▷ Submission form settings
— ▷ Workflow settings
— ▷ Comment settings

Save content type

Сохраняем новый тип данных, сейчас он практически ничем не отличается от типа Story, давайте его разнообразим. Жмем «Edit».

Первый first

edit manage fields delete

Следом нажимаем «Manage fields».

Первый Edit Manage fields  Display fields

Identification

Name: *

Первый

The human-readable name of this content type. This text will be displayed as part of the title and contain only letters, numbers, and **spaces**. This name must be unique.

Type: *

first

The machine-readable name of this content type. This text will be used for constructing URLs and form fields. It can contain lowercase letters, numbers, and underscores. Underscores will be converted into hyphens.

Description:

После включения модуля ССК тут добавилась вкладка Add



Add

+ New field

Label

field

- Select a field type -

Type of data to store

Select a widget

+ New group

Label

group

Group name (a-z, 0-9, _)

Save

где можно добавить новое поле, и новую группу полей. Для добавления нового поля нужно ввести:

Label — заголовок поля который увидят пользователи.

Name — машинное имя.

Type of data to store — тип поля (у нас их 6, но на drupal.org можно скачать множество дополнительных модулей).

Form element to edit the data — Элемент формы для редактирования поля. Если по-русски, то тут мы выбираем как будем вводить данные в это поле если например у поля типа «Text» выбрать «Text field», то будет однострочное поле для ввода текста, а если выбрать «Text area», то получим многострочное поле. У разных типов полей разные варианты элементов формы.

Добавим поле с заголовком «Text», именем «text», типом «Text», элементом формы «text field», и нажмем «Save».

Первый settings

These settings apply only to the `text` field as it appears in the `Первый` content type.

Size of textfield: *

60

Help text:

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

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

Default value

text:

PHP code

Global settings

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

Required

Number of values:

1

Maximum number of values users can enter for this field.

'Unlimited' will provide an 'Add more' button so the users can add as many values as they like.

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

Text processing:

Plain text

Filtered text (user selects input format)

Maximum length:

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

Allowed values

[Save field settings](#)

Здесь можно настроить только что добавленное поле.

Size of textfield — размер поля.

Default value — значение по умолчанию.

Required — если галочка установлена, то при добавлении материала данное поле будет обязательно для заполнения.

Number of values — количество значений которое может ввести пользователь.

Я оставил все значения без изменений, и сохранил. Только что созданное поле отобразилось в списке, его можно перетащить мышкой, тем самым выбрав его месторасположение.

Label	Name	Type	Operations
+ Title	Node module form.		
text	field_text	Text	Configure Remove
Menu settings	Menu module form.		
Drag to re-order			
+ Body	Node module form.		

Add

Для того что бы изменить то, как поле будет отображаться, нужно зайти во вкладку «Display fields».

Field	Label	Teaser	Exclude	Full node	Exclude
text	Above	Default	<input type="checkbox"/>	Default	<input type="checkbox"/>
<input type="button" value="Save"/>					

Здесь выводится список полей. Мы можем изменить:

Label — заголовок. **Above** — отображается выше содержимого поля. **Inline** — отображается на одной линии с содержимым. **<Hidden>** - не показывать заголовок.

Teaser — как отобразить поле в превью. **Default** — по умолчанию. **Plain text** — обычный текст. **Trimmed** — обрезанный. **<Hidden>** - не показывать поле в превью.

Exclude — исключить поле из превью.

Full node — значения аналогичны Teaser, но распространяются не на превью, а на полный документ.

Exclude — исключить поле из документа.

Таким же образом в тип документа можно добавить и другие поля. Для CCK доступно более 300 дополнительных модулей, и если вам нужно добавить какое либо поле не входящее в стандартную поставку, ищите модуль на drupal.org.

На этом урок закончен, чуть позже мы вернемся к модулю CCK, и рассмотрим работу с ним более подробно. А так же, я думаю, попробуем скрестить в одном уроке CCK и Views.

Drupal - Урок 7 - Таксономия.

By RemaGe - Posted on 29 Июль 2009

Новичков обычно пугает слово таксономия, в русском языке оно имеет более приятный аналог — категории. Из этого мы можем сделать вывод, что основное предназначение таксономии, это разбиение данных на категории.

Управление таксономией сосредоточено в «Administer > Content management > Taxonomy». Давайте представим, что нам нужно создать раздел новостей, притом новости будут делиться на «Пресс-релизы» и «Публикации», а во вкладке «Новости», будут доступны материалы из обеих вкладок. Для начала нам нужно создать словарь (vocabulary), для этого нажмем «Add vocabulary».

Name	Type	Operations
No vocabularies available.		

Для добавления обязательно ввести название словаря - «Vocabulary name:», назовём наш «Новости».

Во вкладке Content types нужно поставить галочки на типах материалов которые можно будет категоризировать с помощью нашего словаря. Установим галочку на Оба доступных типа. Во вкладке «Settings» можно указать что выбор категории обязателен (Required) для материалов, так же можно кликнуть по галочке «Tags», включение которой означает что пользователь сам сможет добавлять теги, галочка «Multiple select» разрешает указывать несколько тегов для каждого материала (при включенном «Tags» всегда «Multiple select»). «Weight:»

- определяет местоположение словаря в списке. Тут нам не нужно изменять ничего, сохраняем.

Identification

Vocabulary name: *

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

Description:

Description of the vocabulary; can be used by modules.

The ID for **excluding or including** this element is: edit-description - the path is

Help text:

Instructions to present to the user when selecting terms, e.g., "Enter a comma :"

Content types

Content types:

Page
 Story

Select content types to categorize using this vocabulary.

Settings

Tags
Terms are created by users when submitting posts by typing a comma separate

Multiple select
Allows posts to have more than one term from this vocabulary (always true for

Required
At least one term in this vocabulary must be selected when submitting a post.

Weight:

0 ▾

Vocabularies are displayed in ascending order by weight.



Словарь создан и появился в списке, теперь нам нужно добавить несколько

терминов, переходим по ссылке «Add term».

The screenshot shows the Drupal Taxonomy module interface. At the top, there are tabs for 'Taxonomy', 'list' (which is selected), and 'Add vocabulary'. A message says 'Created new vocabulary Новости.' Below this, a descriptive text explains the taxonomy module's purpose: categorizing content using tags and administrator-defined terms. It mentions creating a 'Vocabulary' to hold terms, and notes that terms can be tied to specific content types like 'Page, Story'. A note at the bottom reminds users to save changes by clicking the 'Save' button. A table below lists the vocabulary 'Новости' with its type 'Page, Story' and various operations: 'edit vocabulary', 'list terms', and 'add terms' (with a cursor icon).

Name	Type	Operations
Новости	Page, Story	edit vocabulary list terms add terms

Для начала добавим головной термин "новости" (По идеи он не обязателен, но с ним выбор категории будет выглядеть красивее). Просто введем название и сохраним.

The screenshot shows the 'Identification' tab of the term edit form. It has two sections: 'Term name:' with the value 'Новости' and 'Description:' which is currently empty. Below the description area, there is a note about the ID for excluding or including the element. At the bottom, there is an 'Advanced options' section and a 'Save' button with a cursor icon.

Следом за ним добавим вкладки «Пресс-релизы» и «Публикации», при этом нужно не забыть указать в «Advanced options», что «Родителем» («Parents:»), у них является термин «Новости»

Identification

Term name: *
Пресс-релизы
The name of this term.

Description:

A description of the term. To be displayed on taxonomy/term page
The ID for excluding or including this element is: edit-description

Advanced options

Parents:

<root>
Новости
Parent terms

Related terms:

<none>
Новости

Synonyms:

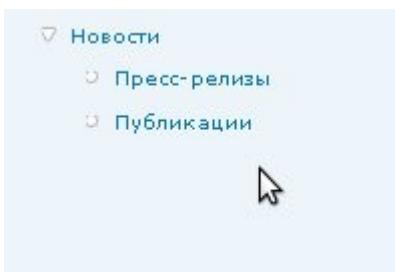
Synonyms of this term, one synonym per line.
The ID for excluding or including this element is: edit-synonyms -

Weight: *
0
Terms are displayed in ascending order by weight.



Теперь попробуем создать материал типа Page. Сразу после заголовка, у нас появился необязательный выпадающий список «Новости», создадим пару материалов с тегами «Пресс-релизы» и «Публикации».

Чтож, теперь нам нужно добавить пункты меню, по которым будут выводиться только материалы с нужным термином, как создавать меню мы уже обсуждали. Ссылки на необходимые термины можно взять в «Administer > Content management > Taxonomy», на странице «list terms», просто кликните по нужному термину, и скопируйте ссылку вверху, у меня термину «Пресс-релизы» соответствует ссылка «taxonomy/term/2». Для головной вкладки новости, нужно прописать ссылку "taxonomy/term/1/all". Готовое меню выглядит так:



В составлении путей для страниц таксономии есть несколько хитростей. Если мы хотим показать все материалы принадлежащие к определенному термину, то путь будет **taxonomy/term/n**, где n — это номер термина, для того что бы

показать материалы нескольких терминов, нужно добавить номера терминов через пробел, или через «+», например путь для меню «Новости», мы могли бы записать как **taxonomy/term/2+3**, если бы нам нужно было вывести материалы входящие сразу в несколько терминов (в нашем примере это невозможно, так как не была проставлена галочка «Tags», или «Multiple select»), то плюс нужно было бы заменить на «,» (**taxonomy/term/2,3**). Путь «**taxonomy/term/1/all**» означает что нужно вывести все материалы, которые входят в дочерние термины для термина с номером 1 (в нашем случае это «Новости»). Так же этот путь можно было записать как "**taxonomy/term/1/1". taxonomy/term/n1/n2** где n1 — это головной термин, а n2 — это до какой глубины мы будем выводить термины.

Drupal - Урок 8 - Еще немного про модуль Views. Работаем с Exposed filters и Arguments.

By RemaGe - Posted on 08 Август 2009

Не так давно разрабатывал сайт для одной гос конторы. Не скажу что в нем было что то особо трудное, но использование **Views** зашкаливало. **Views** поистине монструозен. Поэтому сегодня мы вернемся к нему, но урок будет более практический, чем теоретический. Мы рассмотрим использование такой полезной штуки как **Exposed Filter**, и слегка пощупаем **Arguments**, они с небольшой помощью модуля **Views bonus** сделают возможным скачку материалов в формате **DOC** (вообще их несколько на выбор).

Приступим -) У нас свежайшая подборка, **Drupal 6.13, CCK 6.x-2.5, Date 6.x-2.3, Views 6.x-2.6, Views Bonus Pack 6.x-1.0-beta4**. Если какого то модуля у вас нет, идем на Drupal.org и скачиваем его. Установку модулей мы уже проходили, нам нужны следующие:

CCK

Content

Option Widgets

Date/Time

Date

Date API

Date Timezone

Views

Bonus: Views Export

Views

Views UI

Включаем их.

Теперь создайте словарь терминов, сделайте его обязательным для типа данных page, и закиньте туда пару терминов. Так же создайте несколько материалов типа page. Идем в «**Administer > Site building > Views**» и создаем новый **Views** с «**View type: Node**». Добавляем в новый вид 3 поля (должен же вид что то показывать), **Node: Title, Node: Post date, Node: Teaser**. У поля **Node: Title** устанавливаем галочку «**Link this field to its node**».

Fields

+ ↑ ↓

- Node: Title
- Node: Post date
- Node: Teaser

Теперь давайте добавим парочку **Exposed Filter**. Разница у **Exposed** с обычными только в том, что пользователь может менять **Exposed**, тем самым сортируя материалы. Сначала давайте добавим фильтр **Date: Date (node)**.

Defaults: Add filters

Groups:

Date

Date: Date (node)
Filter any Views node date field.

Date: Date (users)
Filter any Views users date field.



В настройках укажем что сортировка будет происходить по **Node: Post date**

Defaults: Configure extra settings for filter Date: Date (node)

Date form type:

- Select
 Text

Choose the form element to use for date selection.

Granularity:

- Year
 Month
 Day
 Hour
 Minute
 Second

Select a granularity for the date filter. For instance, selecting 'day' will create a

Date year range:

-3:+3

Set the allowable minimum and maximum year range for this argument, either as a range of years, or as a single year, like '2005:2010'. When the argument is set to a date ou

Date field(s):

- Comment: Post date
 Node: Last comment time
 Node: Post date
 Node: Updated date
 Node revision: Created date
 User: Created date
 User: Last access
 User: Last login

Select date field(s) to filter with this argument.

Method:

- OR
 AND

Method of handling multiple date fields in the same query. Return items that matches in all selected date fields (date = field_1 AND field_2).

[Update](#)

[Cancel](#)

В следующей вкладке поставим что нужно найти все даты между 2 указанными значениями (**Operator: is between**), **Date default: now- 5 year**, **To date default: now+ 5 year** (2 этих значения указывают, что сначала отобразятся материалы у которых дата публикации лежит в десятилетнем промежутке, центр которого — сегодня). Теперь нажимаем кнопку **Expose**,

Defaults: Configure filter Date: Date (node)

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

Operator:

Absolute value

From date:

To date:

Relative value

Relative values will be used if no date is set above. Use 'now' to default to the current date at runtime or add modifiers like 'now +1 day'. The To date default value is used when the operator is set to 'between' or 'not between'. You can use any values PHP's date_create() can understand, like between '12AM today' and '12AM tomorrow'.

If the filter is exposed, these values will be used to set the initial value of the exposed filter. Leave both date and default values blank to start with no value in the exposed filter.

Date default:

To date default:

ПОЯВИТСЯ НЕСКОЛЬКО НОВЫХ ПОЛЕЙ ДОСТУПНЫХ ДЛЯ ИЗМЕНЕНИЯ.

Defaults: Configure filter Date: Date (node)

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

Operator:

Absolute value

From date:

To date:

Relative value

Relative values will be used if no date is set above. Use 'now' to default to the current date at runtime or add modifiers like 'now +1 day'. The To date default value is used when the operator is set to 'between' or 'not between'. You can use any values PHP's date_create() can understand, like between '12AM today' and '12AM tomorrow'.

If the filter is exposed, these values will be used to set the initial value of the exposed filter. Leave both date and default values blank to start with no value in the exposed filter.

Date default:

To date default:

Unlock operator

When checked, the operator will be exposed to the user

Filter identifier:

This will appear in the URL after the ? to identify this filter. Cannot be blank.

Label:

Optional

This exposed filter is optional and will have added options to allow it not to be set.

Remember

Remember the last setting the user gave this filter.

Из них, нас интересует «Label», заменим на «Дата публикации». Так же

интересна галочка **Unlock operator**, если кликнем по ней, и сохраним, кроме даты, пользователь сможет изменить и поле **Operator**, но нам это не нужно (во всяком случае пока). Сохраняем фильтр.

Так же добавим фильтр **Node: Title**, в настройках поставим **Contains** (будет выводить все материалы, в заголовках которых есть введенный текст), уберем галочку **Case sensitive**, чтобы не было разницы в Регистре букв, и нажмем **Expose. Label** поставим «Поиск в заголовке».

Defaults: Configure filter **Node: Title**

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

Operator: Is equal to Is not equal to Contains Contains any word Contains all words Starts with Ends with Does not contain

Value:

Unlock operator
When checked, the operator will be exposed to the user

Filter identifier:
This will appear in the URL after the ? to identify this filter. Cannot be blank.

Optional
This exposed filter is optional and will have added options to allow it not to be set.

Remember
Remember the last setting the user gave this filter.

Label:

Case sensitive
Case sensitive filters may be faster. MySQL might ignore case sensitivity.

Buttons:

И последний фильтр - **Taxonomy: Term**. В настройках укажем созданный нами словарь (я поэтично, и с большой фантазией, назвал его «qwe»). **Selection type: Dropdown**, то есть будет выпадающий список, если выбрать **Autocomplete**, то термины нужно будет вводить самому, и они будут авто дополняться. **Show hierarchy in dropdown** — если словарик имеет иерархию, то это отобразится в выпадающем меню.

Defaults: Configure extra settings for filter **Taxonomy: Term**

Vocabulary: qwe
Select which vocabulary to show terms for in the regular options.

Selection type: **Dropdown** **Autocomplete** **Show hierarchy in dropdown**

Buttons:

Defaults: Configure filter Taxonomy: Term

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

[Hide](#)

Operator:

- Is one of
- Is all of
- Is none of
- Is empty (NULL)
- Is not empty (NULL)

Select terms from vocabulary qwe:



Reduce duplicates

This filter can cause items that have more than one of the selected options to appear as duplicate results. If this filter causes duplicate results to occur, this checkbox can reduce those duplicates; however, the more terms it has to search for, the less performant the query will be, so use this with caution.

Unlock operator

When checked, the operator will be exposed to the user

Filter identifier:

tid

This will appear in the URL after the ? to identify this filter. Cannot be blank.

Label:

Выбрать раздел

Optional

This exposed filter is optional and will have added options to allow it not to be set.

Force single

Force this exposed filter to accept only one option.

Remember

Remember the last setting the user gave this filter.

Limit list to selected items

If checked, the only items presented to the user will be the ones selected here.

[Update](#) [Cancel](#) [Remove](#)

Тут я изменил только **Label**, но при желании можете поиграть с настройками.
Сохраняем фильтр, и вид в целом.

Теперь давайте добавим дисплей **Page**.

Defaults ▶

Page

Add display

Analyze

Зададим какой-нибудь **Path**, еще раз сохраним весь вид, и перейдем по вбитому пути.

Page settings

Path: [None](#)

Menu: [No menu](#)

Page: The menu path or URL of this view

<http://localhost/drupal6/> [to_be_or_not_to_be](#)

This view will be displayed by visiting this path on your site. For example, "node/%/feed".

[Update](#) [Cancel](#)

У меня получилось так:

[Edit] [Export] [Clone]

Дата публикации

From date:
2009 ▾ Aug 8 ▾
To date:
2014 ▾ Aug 8 ▾

Поиск в заголовке

Выбрать раздел

<Any> ▾ [Apply](#)

Бросок кобры
08/08/2009 - 20:41
G.I. Joe: The Rise of Cobra
экшн, США, 2009, 135 мин.
Режиссёр: Стивен Содмерс
В ролях: Дэн尼斯 Кузид, Ченнинг Татум, Марлон Уэйнс, Сиенна Миллер, Джозеф Гарден-Льюинт, Кристофер Экклс
Джонатан Прайс, Арнольд Вослу

Кислород
08/08/2009 - 20:42
Кислород
драма, Россия, 2009, 75 мин.
Режиссёр: Иван Вырыпаев
В ролях: Каролина Грушка, Алексей Филимонов
Был обычный летний день. Санек гулял по бульвару и с интересом рассматривал всех окружающих. Кто-то танцевал на скейтбордах, кто-то целовался, в общем, все как обычно, но... Санек повернул голову и увидел ЕЕ — свободную, которая стала для него — "Чистым КИСЛОРОДОМ"! И с этой минуты все, что было в его жизни раньше, потеряло значение.

по моему неплохо)

С первой задачей разобрались, займемся второй. Зайдем в созданный нами вид, и добавим дисплей типа **Feed. Style** поменяем на **DOC file**.

Feed Display the view as a feed, such as an RSS feed

Basic settings

Name: Feed
Title: None
Style: DOC file
Items to display: 10
Distinct: No
Access: Unrestricted
Caching: None
Exposed form in block: No
Header: None
Footer: None
Empty text: None
Theme: Information

Feed settings

Path: None
Attach to: None

Изменим **Path** на «**doc/%**» (% - это то место куда будет подставляться аргумент)

Feed settings

Path: None
Attach to: None

Feed: The menu path or URL of this view

http://localhost/drupal6/ doc/%

This view will be displayed by visiting this path on your site. It is recommended that the path be something like "path/%" putting one % in the path for each argument you have defined in the view.

Update **Cancel**

теперь нужно датьDrupalу нужный аргумент, для наших елей подойдет **Node: Nid**, то есть, идентификационный номер материала.

Feed Display the view as a feed, such as an RSS feed.

Basic settings

Name: Feed
Title: None
Style: DOC file
Items to display: 10
Distinct: No
Access: Unrestricted
Caching: None
Exposed form in block: No
Header: None
Footer: None
Empty text: None
Theme: Information

Relationships
None defined

Arguments
None defined

Fields
Node: Title
Node: Post date
Node: Teaser

Feed settings

Path: **doc/%**
Attach to: None

Feed: Add arguments

Groups:

Node ▾

In the form of WW (01 - 53).
 Node: Created year
In the form of YYYY.
 Node: Created year + month
In the form of YYYYMM.
 Node: Nid
The node ID of the node.
 Node: Title
The title of the node.
 Node: Type
The type of a node (for example, "blog entry", "forum post", "story", etc.).
 Node: Updated date
In the form of CCYYMMDD.
 Node: Updated day
In the form of DD (01 - 31).

Add **Cancel**

Нажмем **Override** (мы ведь не хотим добавить аргумент во все дисплеи?), и сохраним.

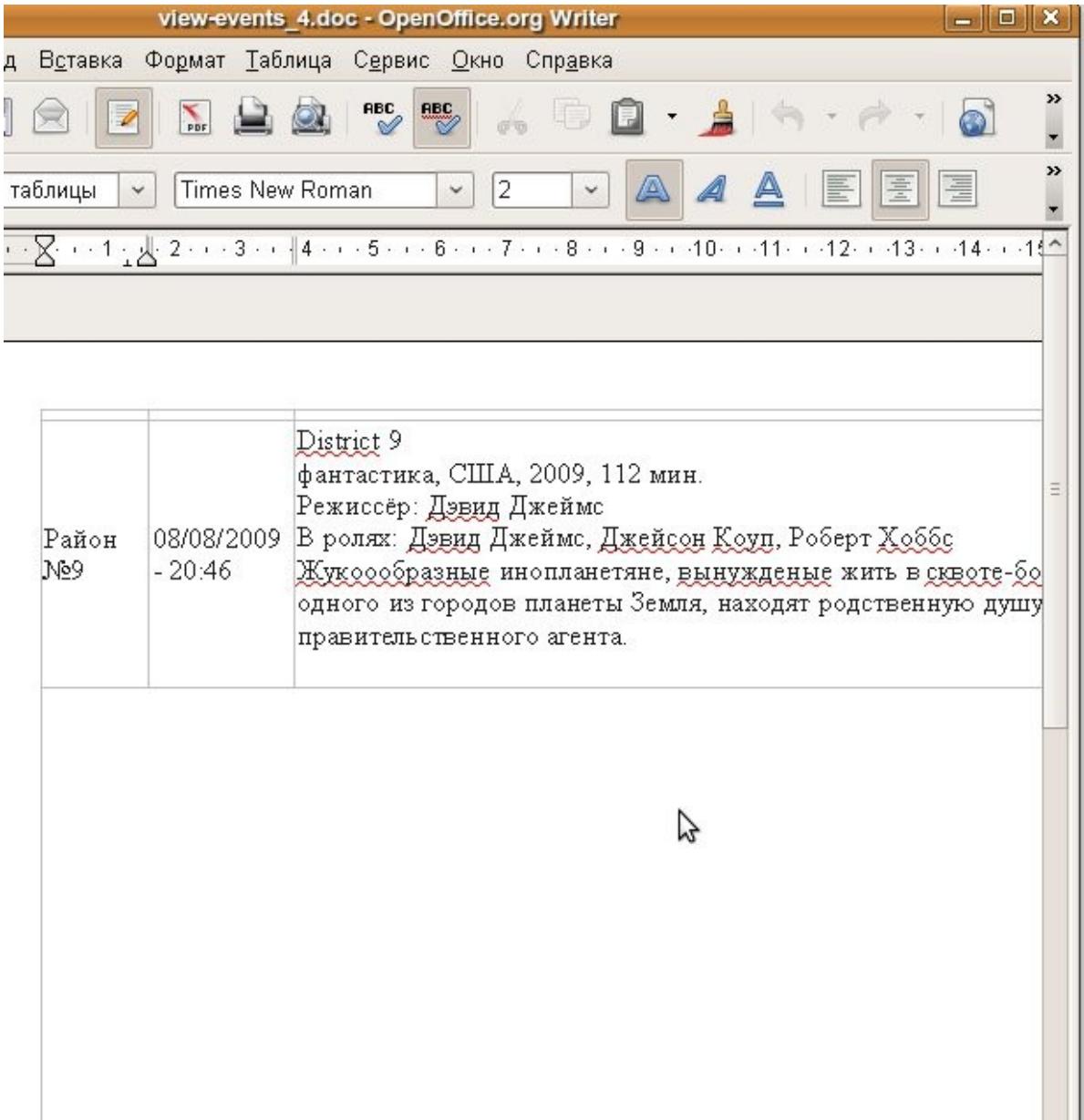
Уже должно работать, зайдем по адресу **ваш_сайт/doc/id_существующего_материала**, должна начать закачка doc файла) а внутри у нас...

Fatal error: Call to undefined method

`views_bonus_plugin_style_export_doc::sanitize_columns() in
/var/www/drupal6/sites/all/modules/views/theme/theme.inc on line 290`

Неожиданно не правда ли? Честно говоря не припоминаю что создавал такую ноду... Если порыскать по Drupal.org, то окажется что не я один создал материал подобного вида. Если у вас не появилась эта ошибка, значит вам повезло, если же появилась, то скачиваем [этот файл](#). Внутри папка «export», которой нужно заменить папку с аналогичным названием в модуле **Views bonus** (Кстати, в DEV версии модуля такой ошибки уже нет, так что скоро поправят и в этой).

Пробуем еще раз зайти по адресу
ваш_сайт/doc/id_существующего_материала (главное **Nid** поставьте правильный, я по ошибке вбил **Nid** несуществующего материала, и пол часа искал почему ничего не отображается). В итоге получим желаемое:



The screenshot shows a Microsoft Word document titled "view-events_4.doc" in OpenOffice.org Writer. The document contains a single table with two columns. The first column has two rows: "Район №9" and "08/08/2009 - 20:46". The second column contains movie details: "District 9", "фантастика, США, 2009, 112 мин.", "Режиссёр: Дэвид Джеймс", "В ролях: Дэвид Джеймс, Джейсон Коуп, Роберт Хоббс", and a long descriptive sentence about the plot.

Район №9	District 9 фантастика, США, 2009, 112 мин. Режиссёр: Дэвид Джеймс В ролях: Дэвид Джеймс, Джейсон Коуп, Роберт Хоббс Жукообразные инопланетяне, вынужденные жить в сквоте-бо одного из городов планеты Земля, находят родственную душу правительственного агента.
08/08/2009 - 20:46	

Ммм... ничего не забыли?) Что бы не вставлять ссылку каждый раз вручную, давайте чуть чуть подправим тему (Это не совсем правильный подход, и чуть позже будет урок по темизации Drupal, но на первый раз сойдет). Зайдите в папку с вашей темой (у меня стандартная тема Garland, она находится в папке **ваш_сайт/Themes/Garland**), тут нас интересует файлик **node.tpl.php** (он отвечает за отображение материалов). Внутренности его выглядят так:

```
<?php
// $Id: node.tpl.php,v 1.5 2007/10/11 09:51:29 goba Exp $
?>
<div id="node-<?php print $node->nid; ?>" class="node<?php if ($sticky) { print ' sticky'; } ?><?php if (!$status) { print ' node-unpublished'; } ?>"><?php print
$picture ?>

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

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

Как видите это обычный HTML файл, со вставками PHP, Добавим нашу ссылку

```
<?php
// $Id: node.tpl.php,v 1.5 2007/10/11 09:51:29 goba Exp $
?>
<div id="node-<?php print $node->nid; ?>" class="node<?php if ($sticky) { print ' sticky'; } ?><?php if (!$status) { print ' node-unpublished'; } ?>"><?php print
$picture ?>

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

```

<h2><a title=<?php print $title ?> href=<?php print $node_url ?>><?php print $title ?></a></h2>
<?php endif; ?>

<?php if ($submitted): ?>
<span class="submitted"><?php print $submitted; ?></span>
<?php endif; ?>
<div class="content clear-block"><?php print $content ?></div>
<div class="clear-block">
<div class="meta"><?php if ($taxonomy): ?>
<div class="terms"><?php print $terms ?></div>
<?php endif;?></div>
<?php if ($page): ?>
<a href=<?php print "http://localhost/drupal6/doc/".$nid ?>>скачать в Doc</a>
<?php endif; ?>

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

```

Если быть точным, то я добавил

```

<?php if ($page): ?>
<a href=<?php print "http://localhost/drupal6/doc/".$nid ?>>скачать в Doc</a>
<?php endif; ?>

```

Сначала идет проверка «**if (\$page)**», то есть, если страница отображается полностью, то печатаем ссылку, далее идет сама ссылка «**<a href="<?php print "http://localhost/drupal6/doc/".\$nid ?>>скачать в Doc**», здесь **\$nid** — эта идентификационный номер, у вас это будет выглядеть как «**<a href="<?php print "ВАШ_САЙТ/doc/".\$nid ?>>скачать в Doc**

На этом урок закончен. Старался расписать все понятнее, если что не серчайте)

Drupal - Урок 9 - Темизация

By RemaGe - Posted on 16 Август 2009

Из коробки Drupal располагает пятью темами. Естественно в интернете можно найти множество замечательных тем для Drupal, но человеческой душе всегда хочется чего то особенного. Либо тема совсем не подходит, либо хочется изменить какую то мелкую деталь. У вас есть свой сайт на Drupal, или вы как раз в процессе его создания? Давайте поговорим о том, как можно его приукрасить. Для того что бы создавать свои темы, или изменять существующие, понадобятся знания PHP, HTML и CSS (чем большие изменения вы хотите внести, чем большие навыки вам потребуются), к сожалению это не входит в тему урока, и вам нужно либо уже иметь знания по этим языкам, или найти уроки по ним где нибудь еще, хотя если вы не лишены смекалки, то кое что понять, и изменить, сможете и после прочтения этого урока.

По умолчанию в Drupal используется движок phptemplate (его вы можете найти в папке /themes/engines, хотя он вам вряд ли понадобится), его мы и помучаем -) Для примеров будем использовать стандартную тему bluemarine, так как по моему мнению она наиболее проста, и подходит для обучения (ну и потому что я делал на ее основе несколько дизайнов -)). Давайте скопируем папку **/themes/bluemarine**, в папку **/sites/all/themes** (если у вас нет папки themes, то создайте ее), и переименуем во что то типа my_theme.

Внутри мы найдем файл **bluemarine.info** (переименуйте его в **my_theme.info**) и 5 файлов с расширением .tpl.php, они то нам и нужны. Каждый из них задает отображение какой то части сайта. Для работы темы необходимы всего лишь 2 файла *.info, и page.tpl.php.

Давайте завершим создание нашей темы. Заходим в файл my_theme.info, он выглядит так:

```
; $Id: bluemarine.info,v 1.4 2007/06/08 05:50:57 dries Exp $
name = Bluemarine
description = Table-based multi-column theme with a marine and ash color scheme.
version = VERSION
core = 6.x
engine = phptemplate
; Information added by drupal.org packaging script on 2009-07-01
project = "drupal"
datestamp = "1246481719"
```

и замените содержимое на:

```
name = my_theme
```

description = Наша тема!!!!

version = VERSION

core = 6.x

engine = phptemplate

Как видите мы убрали лишнее, и изменили название (name), и описание (description). Теперь зайдем в «Administer > Site building > Themes», выберем нашу тему, и сохраним. Теперь можно спокойно экспериментировать -)

Screenshot	Name	Version	Enabled	Default
	Наша тема!!!!	6.13	<input checked="" type="checkbox"/>	<input checked="" type="radio"/>
	Bluemarine	6.13	<input type="checkbox"/>	<input type="radio"/>
	Chameleon	6.13	<input type="checkbox"/>	<input type="radio"/>
	Garland	6.13	<input type="checkbox"/>	<input type="radio"/>
	Marvin	6.13	<input type="checkbox"/>	<input type="radio"/>
	Minnelli	6.13	<input type="checkbox"/>	<input type="radio"/>
	Pushbutton	6.13	<input type="checkbox"/>	<input type="radio"/>
<p>Save configuration Reset to defaults</p>				

Немного теории:

Page — собственно вся страница, то как она будет выглядеть задается в файле page.tpl.php.

Region — страница включает в себя регионы, если вы добавляли\убирали блоки, то вы делали это как раз в регионах, по умолчанию: Left sidebar, Right sidebar,

Content, Header, Footer. В каждый из них можно вставлять блоки. Регионы можно добавлять\убирать, об этом мы поговорим позже.

Block — О блоках вы уже определенно слышали, они выводят какую-либо информацию в нужный нам регион. Их оформление задается в файле block.tpl.php.

Node — это содержание страницы, и задается файлом Node.tpl.php.

Box — выводит элементы страницы в указанном шаблоне. Задается файлом Box.tpl.php.

Давайте разберемся с файлом Page.tpl.php, и с тем как он выводит страницу. Можно разделить весь файл на 3 части: Шапка, вместилище данных, и погреб, почему такие нелепые названия? Потому что большинство боится лезть в темизацию, а то что смешно, - уже не страшно. Разберем шапку:
первая часть

```
<?php
// $Id: page.tpl.php,v 1.28.2.1 2009/04/30 00:13:31 goba Exp $
?>      <?php print $head ?>
<title></title>
<?php print $styles ?>
<?php print $scripts ?>
<script type="text/javascript">
<?php /* Needed to avoid Flash of Unstyle Content in IE */ ?> </script>
```

Тут все просто и не интересно, задается язык, прицепляются файлы CSS и javascript... Переходим ко второму куску:

```
<table cellspacing="0" cellpadding="0" border="0" id="header">
  <tbody>
    <tr>
      <td id="logo"><?php if ($logo) { ?><a title=<?php print t('Home') ?>
href=<?php print $front_page ?>"><img alt=<?php print t('Home') ?>" src=<?php
print $logo ?>" /></a><?php } ?>      <?php if ($site_name) { ?>
        <h1 class="site-name"><a title=<?php print t('Home') ?>" href=<?php print
$front_page ?>"><?php print $site_name ?></a></h1>
        <?php } ?>      <?php if ($site_slogan) { ?>
        <div class="site-slogan"><?php print $site_slogan ?></div>
        <?php } ?></td>
      <td id="menu"><?php if (isset($secondary_links)) { ?><?php print
theme('links', $secondary_links, array('class' => 'links', 'id' => 'subnavlist')) ?><?
php } ?>      <?php if (isset($primary_links)) { ?><?php print theme('links',
```

```

$primary_links, array('class' => 'links', 'id' => 'navlist')) ?><?php } ?>      <?php
print $search_box ?></td>
</tr>
<tr>
  <td colspan="2">
    <div><?php print $header ?></div>
  </td>
</tr>
</tbody>
</table>

```

тут тоже нет ничего сложного:

if (\$logo) Если задан логотип, печатаем логотип.

if (\$site_name) Если задано имя сайта, то и его напечатаем.

if (\$site_slogan) Если задан слоган сайта, он тоже заслужил быть напечатанным.

Немного знания английского, и чуть чуть смекалки, и все встает на свои места.

if (isset(\$secondary_links)) Если существует меню secondary_links, печатаем.

if (isset(\$primary_links)) То же самое касательно primary_links.

<?php print \$search_box ?> Выводим форму поиска.

<?php print \$header ?> А вот это совсем интересно, выводим регион Header, и все что мы в него засунули.

С шапкой вроде разобрались, перейдем к вместилищу данных, или как его называют англоязычныеaborигены — Content (по правде говоря я его и сам так часто называю):

```

<table cellspacing="0" cellpadding="0" border="0" id="content">
  <tbody>
    <tr>
      <?php if ($left) { ?>
        <td id="sidebar-left"><?php print $left ?>      </td>
      <?php } ?>
        <td valign="top"><?php if ($mission) { ?>
          <div id="mission"><?php print $mission ?></div>
        <?php } ?>
          <div id="main"><?php print $breadcrumb ?>
            <h1 class="title"><?php print $title ?></h1>
            <div class="tabs"><?php print $tabs ?></div>
            <?php if ($show_messages) { print $messages; } ?>      <?php print $help ?
          >      <?php print $content; ?>      <?php print $feed_icons; ?></div>
        </td>
      <?php if ($right) { ?>

```

```
<td id="sidebar-right"><?php print $right ?>    </td>
<?php } ?>
</tr>
</tbody>
</table>
```

Вместилище данных можно поделить на 3 части, левую колонку, правую колонку, и то что посередине.

if (\$left) Если в левом регионе что то есть (блоки), то
 <?php print \$left ?> Выводим левый регион, это и есть вся левая колонка.

Переходим к центру

if (\$mission) Если админ ввел «Миссию» сайта, печатаем ее.

<?php print \$breadcrumb ?> Выводим «хлебные крошки» (они выглядят примерно так «Home » Administer » Site building», и находятся вверху страницы)
<?php print \$title ?> Заголовок страницы.

<?php print \$tabs ?> Вкладки (если конечно есть)

if (\$show_messages) { print \$messages; } Выводятся системные сообщения.

<?php print \$help ?> Помощь. В основном ее можно найти в админке.

<?php print \$content; ?> Регион content.

<?php print \$feed_icons; ?> Иконка RSS.

if (\$right) Если в правом регионе что то есть (блоки), то

<?php print \$right ?> Выводим правый регион.

Вот и все вместилище данных -) Переходим к погребу (его так же называют футером, или подвалом):

```
<div id="footer">
  <?php print $footer_message ?>
  <?php print $footer ?>
</div>
<?php print $closure ?>
</body>
</html>
```

<?php print \$footer_message ?> выводим системные сообщения внизу страницы.

<?php print \$footer ?> Выводим регион footer.

<?php print \$closure ?> Страница закончена.

Вот и весь page.tpl.php, как видите ничего сложного, другие темы движка phptemplate могут быть больше и тяжелее, но созданы по образу и подобию этой.

Вы можете вносить необходимые изменения к коду, и они после обновления страницы непременно отобразятся. Например, удалим строчку <?php print \$breadcrumb ?>, и зайдем в админку

The screenshot shows the 'Content management' section of the Drupal administration interface. At the top, there are two tabs: 'By task' (selected) and 'By module'. Below the tabs, a message says 'Welcome to the administration section. Here you may control how your site's content is managed.' Under the heading 'Hide descriptions', there are several sections: 'Content management' (selected), 'Comments', 'Content', 'Content types', 'Post settings', and 'RSS publishing'. Each section has a brief description and a link to its configuration page.

- Content management**
Manage your site's content.
- Comments**
List and edit site comments and the comment moderation queue.
- Content**
View, edit, and delete your site's content.
- Content types**
Manage posts by content type, including default status, front page promotion, etc.
- Post settings**
Control posting behavior, such as teaser length, requiring pre-posting, and the number of posts on the front page.
- RSS publishing**
Configure the number of items per feed and whether feeds show teasers/full-text.

Хлебные крошки пропали, а теперь вернем их на место, и допишем за ними:
<?php print \$breadcrumb ?> Хлебные крошки

The screenshot shows the Drupal administration interface. In the top left, there's a sidebar with links: 'Home' (highlighted with a red box), 'Хлебные крошки', and 'Administer'. Below that is a navigation bar with 'By task' and 'By module' (also highlighted with a red box). A main content area starts with a welcome message: 'Welcome to the administration section. Here you may control ho...'. Below this is a 'Hide descriptions' link. The 'Content management' section is currently active, indicated by a blue header. It contains links for 'Comments', 'Content' (with a mouse cursor hovering over it), and 'Content types'. A small note at the bottom says 'Многие посты имеют контентные типы, включая стандартные'.

Простор для действий огромный, хотя конечно все основные данные о отображении данных и их расположении хранятся в файле style.css.
В файле page.tpl.php можно использовать еще и другие переменные, для того что узнать какие переменные есть в наличии, можете добавить код

```
<?php  
print '<pre>';  
print htmlspecialchars(print_r(get_defined_vars(), TRUE), ENT_QUOTES);  
print '</pre>';  
?>
```

например после строчки

```
<?php print $footer ?>
```

Переменных будет много, не пугайтесь, список переменных можно найти [тут](#).

Не бойтесь что нибудь изменить, и у вас все получится -)

Перейдем к файлу **Node.tpl.php**, на этом блоге изменения происходили в основном в этом файле.

```
<?php  
// $Id: node.tpl.php,v 1.7 2007/08/07 08:39:36 goba Exp $  
?  
<div class="node<?php if ($sticky) { print " sticky"; } ?><?php if (!$status) { print " node-unpublished"; } ?>">  
  <?php if ($picture) {  
    print $picture;
```

```

    }?>
    <?php if ($page == 0) { ?><h2 class="title"><a href=<?php print $node_url?
>"><?php print $title?></a></h2><?php }; ?>
        <span class="submitted"><?php print $submitted?></span>
        <div class="taxonomy"><?php print $terms?></div>
        <div class="content"><?php print $content?></div>
        <?php if ($links) { ?><div class="links">&raquo; <?php print $links?></div><?
php }; ?>
    </div>

```

В этой теме он довольно таки простой. Из интересных переменных **\$page**.

if (\$page == 0) Если \$page равно 1 (истина), то мы в режиме полного отображения материала, если же 0, то мы в тизере, эта строчка выводит ссылку на полный материал, если мы в тизере. У переменной \$page так же есть противоположная переменная **\$teaser**, которая истинна только в тизере.

Недавно я добавил к своим страницам на блоге кнопки добавления статьи в различные сервисы. Как это можно реализовать тут? Код который нужно добавить выглядит так

```
<script src="http://odnaknopka.ru/ok3.js" type="text/javascript"></script>
```

Он одинаков для всех страниц, и у вас будет выглядеть так же. По моему оптимальное место для добавления, это перед выводом **\$links**, и нужно не забыть что они не должны выводиться в тизере. Код который нужно добавить, у меня выглядит так:

```
<?php if ($page) { ?><script src="http://odnaknopka.ru/ok3.js"
type="text/javascript"></script><?php }; ?>
```

то есть, если мы не в тизере, то выводим кнопки.

вид измененного файла

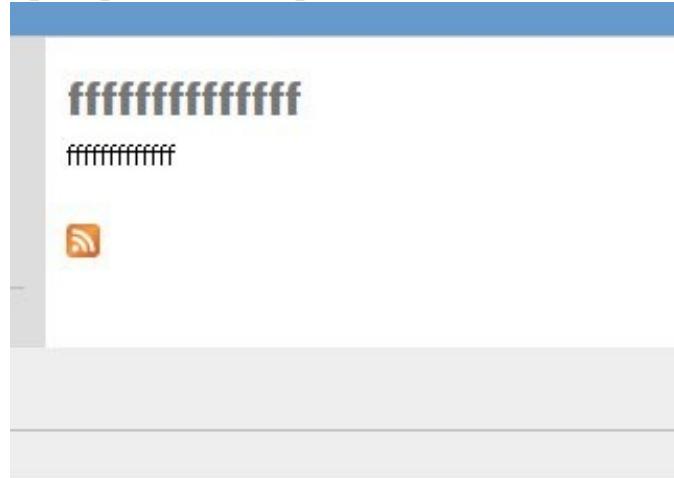
```

<?php
// $Id: node.tpl.php,v 1.7 2007/08/07 08:39:36 goba Exp $
?>
<div class="node<?php if ($sticky) { print " sticky"; } ?><?php if (!$status) { print "
node-unpublished"; } ?>">
    <?php if ($picture) {
        print $picture;
    }?>
    <?php if ($page == 0) { ?><h2 class="title"><a href=<?php print $node_url?
>"><?php print $title?></a></h2><?php }; ?>
        <span class="submitted"><?php print $submitted?></span>

```

```
<div class="taxonomy"><?php print $terms?></div>
<div class="content"><?php print $content?></div>
<?php if ($page) { ?><script src="http://odnaknopka.ru/ok3.js"
type="text/javascript"></script><?php } ; ?>
<?php if ($links) { ?><div class="links">&raquo; <?php print $links?></div><?
php } ; ?>
</div>
```

проверяем. В тизере нет:



а на полной страничке есть.



Таким же образом можно выводить картинки, произвольный текст, и все что вам захочется. Только, если будете выводить русский текст, помните, что сохранять файл нужно в UTF-8, иначе на выходе вместо букв получите кракозябры.

Block.tpl.php

```
<?php
// $Id: block.tpl.php,v 1.3 2007/08/07 08:39:36 goba Exp $
?>
```

```
<div class="block block-<?php print $block->module; ?>" id="block-<?php print  
$block->module; ?>-<?php print $block->delta; ?>">  
  <h2 class="title"><?php print $block->subject; ?></h2>  
  <div class="content"><?php print $block->content; ?></div>  
</div>
```

Давайте сделаем его ярко-красным, и с округлыми углами? Смотреться будет ужасно, но надо же с чего то начинать -) Как человек ориентированный на программирование, я не особо люблю возиться с дизайном, поэтому мы немного сфилоним. Идем на сайт <http://www.roundedcornr.com>, нас интересует вот эта менюшка:



Первое поле не трогаем, это радиус углов, и такой пойдет. Во втором пишем ddddःःःः, это цвет фона, и он у нас серый, в третьем пишем ff3030, это цвет самого блока — красный. Нажимаем кнопку Create RoundedCornr, и попадаем на вторую страничку.

RoundedCornr

What it looks like:

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

HTML code:

Replace the "Lorem ipsum..." text with your own content.

```
<div class="roundedcornr_box_419100">
  <div class="roundedcornr_top_419100"><div></div></div>
  <div class="roundedcornr_content_419100">
    Lorem ipsum dolor sit amet, consectetur
    adipisicing elit, sed do eiusmod tempor incididunt
    ut labore et dolore magna aliqua. Ut enim ad minim
    veniam, quis nostrud exercitation ullamco laboris
    nisi ut aliquip ex ea commodo consequat. Duis aute
    irure dolor in reprehenderit in voluptate velit esse
    cillum dolore eu fugiat nulla pariatur. Excepteur
    sint occaecat cupidatat non proident, sunt in culpa
    qui officia deserunt mollit anim id est laborum.
  </div>
  <div class="roundedcornr_bottom_419100"><div></div></div>
</div>
```

CSS code:

Add this code to your CSS stylesheet.

```
.roundedcornr_box_419100 {
  background: #ff3030;
}
.roundedcornr_top_419100 div {
  background: url(roundedcornr_419100_tl.png) no-repeat top left;
}
.roundedcornr_top_419100 {
  background: url(roundedcornr_419100_tr.png) no-repeat top right;
}
.roundedcornr_bottom_419100 div {
  background: url(roundedcornr_419100_bl.png) no-repeat bottom left;
}
.roundedcornr_bottom_419100 {
  background: url(roundedcornr_419100_br.png) no-repeat bottom right;
}
```

Download image files:

If you put the image files in a directory different from where the HTML page resides, remember to update the filepaths in the CSS code above.

- [roundedcornr_419100_tl.png](#)
- [roundedcornr_419100_tr.png](#)
- [roundedcornr_419100_bl.png](#)
- [roundedcornr_419100_br.png](#)

Здесь нас интересуют 4 вещи: как будет выглядеть наш блок, HTML код, который нужно будет вставить в файл block.tpl.php, CSS код, который нужно

будет вставить в файл style.css, и 4 картинки с углами, их мы закинем в папку /sites/default/files.

Начнем с CSS, копируем код, и вставляем в самый конец файла style.css, при этом нужно чуть чуть подправить названия файлов с углами, в оригинале

background: url(roundedcornr_419100_tl.png) no-repeat top left;

у меня

background:

url(http://localhost/drupal6/sites/default/files/roundedcornr_419100_tl.png) no-repeat top left;

то есть я просто добавил к названию **ВАШ_САЙТ/sites/default/files/**, что бы браузер знал где искать уголки, это нужно сделать в 4 местах.

Весь CSS код

```
.roundedcornr_box_419100 {  
    background: #ff3030;  
}  
.roundedcornr_top_419100 div {  
    background:  
url(http://localhost/drupal6/sites/default/files/roundedcornr\_419100\_tl.png) no-repeat  
top left;  
}  
.roundedcornr_top_419100 {  
    background:  
url(http://localhost/drupal6/sites/default/files/roundedcornr\_419100\_tr.png) no-repeat  
top right;  
}  
.roundedcornr_bottom_419100 div {  
    background:  
url(http://localhost/drupal6/sites/default/files/roundedcornr\_419100\_bl.png) no-repeat  
bottom left;  
}  
.roundedcornr_bottom_419100 {  
    background:  
url(http://localhost/drupal6/sites/default/files/roundedcornr\_419100\_br.png) no-repeat  
bottom right;  
}  
.roundedcornr_top_419100 div, .roundedcornr_top_419100,  
.roundedcornr_bottom_419100 div, .roundedcornr_bottom_419100 {
```

```
width: 100%;  
height: 30px;  
font-size: 1px;  
}  
.roundedcornr_content_419100 { margin: 0 30px; }
```

у вас он будет выглядеть чуть иначе.

Далее, копируем HTML код и вставляем в файл **block.tpl.php**, при этом старое содержимое файла вставляем вместо текста

Lorem ipsum dolor sit amet, consectetur
adipisicing elit, sed do eiusmod tempor incididunt
ut labore et dolore magna aliqua. Ut enim ad minim
veniam, quis nostrud exercitation ullamco laboris
nisi ut aliquip ex ea commodo consequat. Duis aute
irure dolor in reprehenderit in voluptate velit esse
cillum dolore eu fugiat nulla pariatur. Excepteur
sint occaecat cupidatat non proident, sunt in culpa
qui officia deserunt mollit anim id est laborum.

должно получиться примерно так

```
<?php  
// $Id: block.tpl.php,v 1.3 2007/08/07 08:39:36 goba Exp $  
?>  
<div class="block block-<?php print $block->module; ?>" id="block-<?php print  
$block->module; ?>-<?php print $block->delta; ?>">  
<div class="roundedcornr_box_419100">  
  <div class="roundedcornr_top_419100"><div></div></div>  
  <div class="roundedcornr_content_419100">  
    <h2 class="title"><?php print $block->subject; ?></h2>  
    <div class="content"><?php print $block->content; ?></div>  
  </div>  
  <div class="roundedcornr_bottom_419100"><div></div></div>  
</div>  
</div>
```

Мой код брать не следует, так как у вас так же будут отличаться названия стилей. Прочто сравните

Сохраняем 4 файлика с углами в папку **/sites/default/files** и обновляем страницу. Вуаля:



Вот мы и получили красные блоки, единственное что меня не устраивает, что блоки стали красными и в подвале. В Drupal хорошо то, что можно темизировать любой элемент по его id. Если посмотреть код страницы, то мы увидим что id блока navigation — **block-user-1**. Как происходит построение этого id? В файле block.tpl.php можно это увидеть.

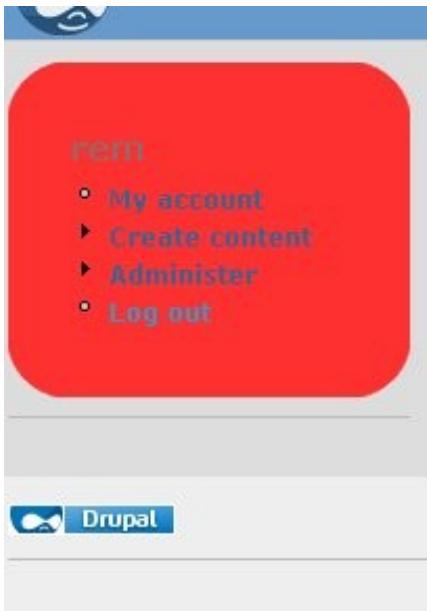
id="block-<?php print \$block->module; ?>-<?php print \$block->delta; ?>">

Сначала ставится block- к нему добавляется название модуля, а затем числовой идентификатор, названный delta.

Давайте сделаем копию файла block.tpl.php, и назовем его **block-user-1.tpl.php**, а затем вставим в block.tpl.php первоначальный код

```
<?php
// $Id: block.tpl.php,v 1.3 2007/08/07 08:39:36 goba Exp $
?>
<div class="block block-<?php print $block->module; ?>" id="block-<?php print
$block->module; ?>-<?php print $block->delta; ?>">
  <h2 class="title"><?php print $block->subject; ?></h2>
  <div class="content"><?php print $block->content; ?></div>
</div>
```

и обновим страницу



мы только что переопределили вывод блока `block.tpl.php`, так же можно поступить с любой страницей, нодой, и тд. Например, если мы хотим переопределить страницу с нодой под номером 4, то создаем файл **page-node-4.tpl.php**, а файл **node-story.tpl.php** переопределит вывод всех нод типа — story. В принципе тут — все. Используя похожий подход, вы сможете поменять многое в теме Drupal, или при желании создать свою, если конечно вы наделены дизайнерскими талантами (в отличии от меня <8-D).
Напоследок давайте создадим свой регион. Как я уже говорил по умолчанию их 5, но бывает ситуация, что свои блоки хочется засунуть еще куда нибудь, например форму поиска вставить рядом с логотипом (хотя есть переменная `$search_box`, для примера подойдет и форма поиска). Попробуем?
Все новые регионы нужно прописывать в файле `*.info`, вашей темы, при этом, если вы пропишете хоть 1 свой регион, то придется указать и 5 дефолтных.
Действуем -)

```
name = my_theme  
description = Наша тема!!!!  
version = VERSION  
core = 6.x  
engine = phptemplate  
  
name = my_theme  
description = Наша тема!!!!  
version = VERSION  
core = 6.x  
engine = phptemplate  
regions[left] = Left sidebar
```

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

Так выглядит наш файл после внесения изменений. Как видите, изначально мы вписали 5 стандартных регионов, а потом добавили свой - search (вы можете назвать его по своему), теперь давайте добавим его вывод в файл page.tpl.php, строка которую нужно добавить выглядит так

<?php print \$search; ?>

я вставил ее сразу после вывода логотипа

```
<?php if ($logo) { ?><a href="<?php print $front_page ?>" title="<?php print t('Home') ?>">"/></a><?php } ?>    <?php print $search; ?>    <?php if ($site_name) { ?></code></p>
<h1 class="site-name"><a href="<?php print $front_page ?>" title="<?php print t('Home') ?>"><?php print $site_name ?></a></h1>
<?php } ?>    <?php if ($site_slogan) { ?>
<div class="site-slogan"><?php print $site_slogan ?></div>
<?php } ?>
```

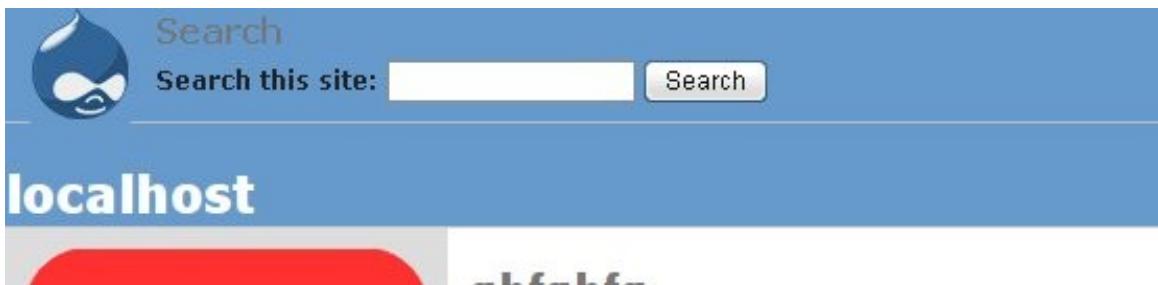
Естественно вашу фантазию никто не сдерживает, можете выводить регионы где захотите)

Сохраняем, и идем в «Administer » Site building » Blocks», если все сделано правильно, то должен появиться наш регион.

The screenshot shows the 'Blocks' configuration page in the Drupal admin interface. There are five regions listed:

- Header**: No blocks in this region.
- Footer**: Contains a block titled "Powered by Drupal" with a "configure" button.
- search**: No blocks in this region.
- Disabled**: Contains three blocks:
 - Primary links: Set to "<none>" with a "configure" button.
 - Recent comments: Set to "<none>" with a "configure" button.
 - Secondary links: Set to "<none>" with a "configure" button.

Вставляем туда блок поиска, и получаем:



вот такой вот ужас... Но я надеюсь общий смысл понятен.

На этом урок закончен.

Естественно это только базис темизации в Drupal. В следующих уроках мы непременно обсудим темизацию всего остального через файл template.php, создадим свой модуль, и наверно еще поработаем с Views (может и их темизируем). Если у кого то есть интересная тема для урока, пишите в комментах)

ЗЫ Пожалуйста, удалите тему которую мы сегодня на ваяли, она ужасна, я боюсь, потомки мне этого не простят.

Drupal - Урок 10 - Руководство по созданию тем в Drupal 6

Это руководство рассказывает о темизации Drupal 6 и выше.

Информацию по созданию тем для ранних версий Drupal, можно найти в «руководство по созданию тем для Drupal 5 и более ранних версий».

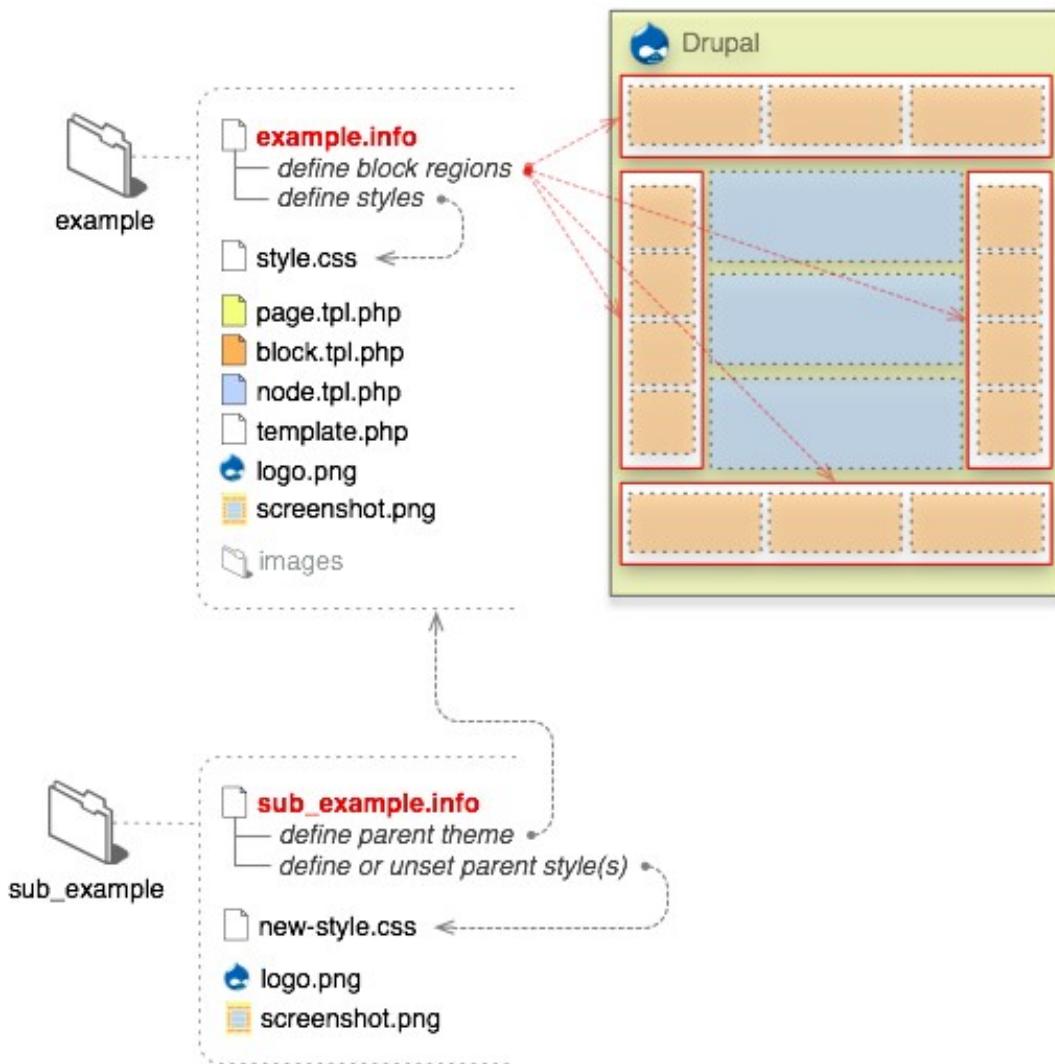
В Drupal 6 значительно изменилось создание тем. Список этих изменений, и информацию по переводу тем на Drupal 6 смотрите в «Обновление ваших тем». <http://drupal.org/update/theme> Это руководство так же отметит отличие от предыдущих версий, там где это будет необходимо.

Часть первая. Анатомия темы Drupal 6

By RemaGe - Posted on 01 Сентябрь 2009

Тема — это несколько файлов определяющих как будет выглядеть ваш сайт. Вы так же можете создать «**под-тему**» или другой вариант готовой темы.

Необходимым является только файл **.info**, но большинство тем и под-тем также используют иные файлы. Приведённая ниже диаграмма показывает какие файлы можно найти в обычной теме и под-теме.



.info (обязателен)

Файл .info - это все, что необходимо Drupal, для того что бы увидеть вашу тему. В нем определяются мета данные, стили, java-скрипты, регионы и т.д. Остальное по желанию.

Внутреннее имя темы так же определяется здесь. Например, если файл называется «drop.info», тогда Drupal определит имя темы как «drop». Drupal 5 и ранние версии для этого используют имя папки.

Файлы info для тем — новинка, появившаяся в Drupal 6. В пятой версии info файлы были только у модулей.

Файлы шаблонов (.tpl.php)

В этих шаблонах располагается xHTML код, и переменные PHP. Иногда они могут выводить другую информацию, например xml rss. Каждый .tpl.php файл определяет вывод какой то части информации. Они опциональны, и если какого

то файла нет, будет использован стандартный вывод. Воздержитесь от создания сложной логики в этих файлах. В большинстве случаев это должны быть xHTML-теги, и PHP-переменные. Некоторые из этих шаблонов находятся в папках ядра и модулей. Скопировав эти файлы в папку темы, вы заставите Drupal использовать вашу версию.

Примечание: Информация о теме кешируется. При добавлении или удалении шаблонов или функций, нужно сбросить кеш.

template.php

Вся программная логика, и обработка выходных данных находится в файле template.php. Он не обязательен, но его можно использовать для сохранения чистоты файлов .tpl.php, а так же для определения переменных до того как они ссылаются с разметкой файлов .tpl.php. Так же тут находятся функции пользователя, переопределения функций темы, и любые другие функции ответственные за вывод информации. Файл должен начинаться с "<?php", на закрывающий тег не нужен, рекомендуется его пропускать.

Под-темы

С первого взгляда, под-темы не отличаются от обычных тем. Основным отличием является то, что они наследуют ресурсы тем-родителей. Для создания под-темы, нужно определить «базовую тему» в файле .info. После этого тема унаследует ресурсы темы родителя. Возможен множественный уровень наследования, т.е. под-тема может указать в качестве «базовой темы» другую под-тему. Пределы наследования неопределенны.

Drupal 5 более старые версии требуют что бы под-темы находились в папке базовой темы. Для 6 версии это не актуально.

Остальное

Логотип и скриншот не обязательны для функционирования темы, но рекомендованы, особенно если вы внесли вашу тему в репозиторий Drupal. Скриншот будет показан на странице администрирования тем, и в свойствах пользовательского аккаунта (конечно, если выданы соответствующие права). Больше информации можно найти в [руководстве по добавлению скриншотов](#). Для обеспечения административной настройки пользовательского интерфейса или каких-либо «фич» наподобие логотипа, поиска, миссии сайта и т.д., может быть использован файл "theme-settings.php". Это продвинутый функционал. Более подробную информацию можно найти в [руководстве по расширенным настройкам](#).

Для работы модуля color, необходима папка "color" с файлом "color.inc", а также различные файлы поддержки.

Если вы хотите создать тему на основе темы ядра, создайте под-тему, или скопируйте и переименуйте тему. Изменение тем Garland или Minnelli крайне не рекомендовано, так как они используются для установки и обновления.

Все темы не являющиеся темами ядра, или модификации тем ядра должны находиться в папке "sites/all/themes". Если вы планируете запустить несколько сайтов из одной сборки Drupal, вы можете сделать тему доступной определенному сайту. Почитайте о этом [тут](#).

Часть вторая. Определение параметров и компонентов темы

By RemaGe - Posted on 07 Сентябрь 2009

В этом уроке объясняется как задать структуру темы, названия регионов и их количество, стили, и javascript, с помощью редактирования файла .info. В файле .info так же задается какие настройки темы будут доступны из интерфейса администратора.

Так как файл .info кешируется, перед тем как увидеть какие-либо изменения, - очистите кеш.

О файле .info

В Drupal 6, каждая тема должна иметь файл .info в своей папке. Без него тема не будет видна из Drupal. Файл .info должен иметь расширение «.info».

Машинное имя берется из названия темы. Например, если файл называется "drop.info", тогда Drupal определит имя как "drop". Имя должно начинаться с буквы, и не содержать пробелов, и знаков препинания. Разрешен знак подчеркивания, но не тире. Разрешены цифры, но они не должны стоять первыми.

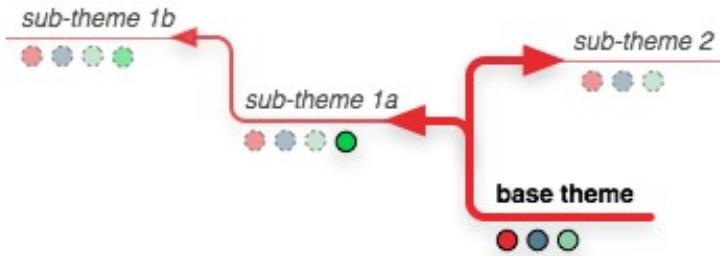
Примечание:

Внимание! Модули с тем же названием что и темы, могут помешать сайту функционировать. Это может привести к дублированию имен функций, что запрещено в PHP. Все установленные компоненты должны иметь уникальные имена. Для создаваемых тем рекомендуется использовать уникальный префикс, например у сайта example.com может быть тема ex_themetame.

Под-темы, их структура и наследование

Под- темы не отличаются от обычных тем ничем, кроме одного: они наследуют ресурсы родительской темы. Лимита на длину цепочки подключений под-тем к родителям нет. Под-тема может быть родителем другой под-темы, разветвляйте и организовывайте это как вам удобно. Из этого вытекает большой потенциал

под-тем.



Представьте себе: создаете основную тему-каркас, а детали и улучшения прикручиваете в под-теме. Можно создать ответвление в котором будет тестироваться другой дизайн. Используете мульти-сайтинг, но хотите что бы все сайты были в одном стиле? Легко. С помощью под-тем можно объединить многие дизайнерские ресурсы. Изменения специфичные для одного сайта, могут быть внесены в его под-тему, но если придется поменять что то везде разом, просто подправьте тему-родителя. При правильном планировании, возможности безграничны.

Для закрепления базовой темы, нужно включить в файл .info команду base theme, например что бы задать для текущей темы базовую тему "themeName", включите в файл следующую строку:

base theme = themeName

Что наследуется:

- Все стили включенные в базовую тему, хотя это можно контролировать.
- Все java-скрипты включенные в базовую тему.
- Все шаблоны (.tpl.php).
- Все что определено в файле "template.php". Это включает в себя переопределения функций, препроцессоры, и все остальное. Каждая под-тема может включать включать и свой template.php, вместе с родительским.
- Скриншот темы родителя, пока файл .info использует те же настройки, что и родительский.

Что не наследуется:

- logo.png установленный для данной темы. Это не распространяется на загруженные логотипы, так как они используются всегда.
- Некоторые настройки установленные в .info. Например регионы. Если вы не используете регионы создаваемые по умолчанию, убедитесь что регионы в "page.tpl.php" под-темы, совпадают с тем, что установлено в

файле .info. Базовая и под-темы могут использовать свои настройки регионов.

- Все что заданно в "theme-settings.php".
- Все что задано в папке color для поддержки модуля color.

Заметьте, что под-тема может быть помещена как в папке базовой темы, так и не в ней. До 6 версии, под-темы должны были находиться в папке базовой темы.

Структура файла .info

Синтаксис очень похож на структура файлов INI. Файл .info — это статичный текстовый файл, предназначенный для настройки темы. Каждая строка в нем, это ключ и значение, разделенные знаком равенства (например: key = value). Текст после точки с запятой, это комментарии. Некоторые ключи используют специальный синтаксис; они используют квадратные скобки для построения массивов. Если вы не знакомы с массивами, их использование должно быть понятным по примерам.

Drupal понимает ключи приведенные ниже. Значения по умолчанию будут использоваться для необязательных ключей, не записанных в файле.

- **name** **обязательно**
- **description** **рекомендовано**
- **screenshot**
- **version** **нежелательно**
- **core** **обязательно**
- **engine** **обязательно в большинстве случаев**
- **base theme**
- **regions**
- **features**
- **stylesheets**
- **scripts**
- **php**

name (обязательно)

Имя темы предназначенное для людей, теперь может быть задано отдельно от машинного, это накладывает меньше ограничений на используемые символы.

name = A fantasy name

description (рекомендовано)

Короткое описание темы. Оно отображается на странице выбора темы "Administer > Site building > themes".

description = Tableless multi-column theme designed for blogs.

screenshot

Необязательный ключ — screenshot, говорит Drupal где искать миниатюрное отображение темы, используемое на странице выбора темы (admin/build/themes). Если параметр пропущен, Drupal использует файл "screenshot.png" расположенный в папке темы.

Используйте этот параметр только в случае, если ваш скриншот называется не «screenshot.png», или если вы хотите поместить скриншот вне папки темы (например screenshot = images/screenshot.png)

screenshot = screenshot.png

version (нежелательно)

Ключ version будет добавлен автоматически drupal.org, когда вы выложите тему. Так что вы можете пропустить этот параметр в темах бесплатно выложенных для сообщества. Однако, если вы собираетесь выкладывать тему на drupal.org, вы можете задать своей теме любое значение version.

version = 1.0

core (обязательно)

Начиная с 6 версии, все файлы .info, должны указывать с какой версией ядра они совместимы. Указанное значение сравнивается с DRUPAL_CORE_COMPATIBILITY. Если они не совпадают, тема отключается.

core = 6.x

Архивирующий скрипт Drupal.org выставляет это значение автоматически, в зависимости от совместимости каждого релиза ядра Drupal. Так что у тем загружаемых с Drupal.org это значение всегда правильно.

engine (обязательно в большинстве случаев)

Движок используемый темой. Если ничего не указано, считается что тема использует движок по умолчанию. Большинство тем использует движок "phptemplate".

PHPTemplate определяет функции темы, и ее поведение. Пропустите этот параметр, если не понимаете для чего он.

engine = phptemplate

base theme

Под-темы указывают базовую тему. Это позволяет под-теме использовать ресурсы базовой темы. По-темы могут указывать другие по-темы в качестве базовых тем, создавая множественное наследование. Используйте машинное имя темы, для указания базовой темы. Это использовано в теме Minnelli, под-теме темы Garland.

`base theme = garland`

Больше информации доступно в разделе со структурой под-тем.

regions

Регионы можно задать в теме с помощью ключа 'regions', за которым должно идти машинное имя региона в квадратных скобках, и после знака равенства имя показываемое людям, т.е.

Если не определен ни один регион, будут использованы значения по умолчанию. Вы можете переписать значения под ваши нужды.

`regions[left] = Left sidebar`

`regions[right] = Right sidebar`

`regions[content] = Content`

`regions[header] = Header`

`regions[footer] = Footer`

features

Разные элементы страницы могут быть выключены или включены, для изменения на странице настройки темы. Ключ "features" задает какие из чекбоксов будут отображены на странице настройки темы. Можно выключить элементы не используемые данной темой. Что бы убрать какой-либо чекбокс, пропустите строчку с его определением. Однако, если не будет определено ни одного, будут выведены все чекбоксы.

Приведенный ниже пример, показывает выводом чего можно управлять.

`primary_links` и `secondary_links` не будут показаны администратору, так как их вывод за комментирован.

`features[] = logo`

`features[] = name`

`features[] = slogan`

`features[] = mission`

`features[] = node_user_picture`

`features[] = comment_user_picture`

`features[] = search`

```
features[] = favicon  
; These last two disabled by redefining the  
; above defaults with only the needed features.  
; features[] = primary_links  
; features[] = secondary_links
```

stylesheets

По традиции, тема использует файл style.css, но есть возможность добавить стили с помощью вызова drupal_add_css() в файле template.php. Начиная с 6 версии, стили так же можно добавить в файле .info.

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

scripts

По традиции, есть возможность добавить javascript с помощью вызова drupal_add_js() в файле template.php. Начиная с 6 версии, файл script.js включается автоматически (если существует в папке темы). Так же, javascript можно добавить в файле .info.

```
scripts[] = myscript.js
```

php

Определяет минимальную версию php, которую поддерживает тема. Значение по умолчанию равно константе DRUPAL_MINIMUM_PHP, которая задает минимальную версию PHP для ядра. Это может быть переопределено при необходимости. Должно быть добавлено для большинства тем.

```
php = 4.3.3
```

Примеры файлов .info из тем ядра



Garland:

```
; $Id: garland.info,v 1.5 2007/07/01 23:27:32 goba Exp $
```

```
name = Garland
description = Tableless, recolorable, multi-column, fluid width theme (default).
version = VERSION
core = 6.x
engine = phptemplate
stylesheets[all][] = style.css
stylesheets[print][] = print.css
```

; Information added by drupal.org packaging script on 2008-02-13
version = "6.0"
project = "drupal"
datestamp = "1202913006"

Minnelli под-тема темы Garland.:

```
; $Id: minnelli.info,v 1.7 2007/12/04 20:58:44 goba Exp $
name = Minnelli
description = Tableless, recolorable, multi-column, fixed width theme.
version = VERSION
core = 6.x
base theme = garland
stylesheets[all][] = minnelli.css
```

; Information added by drupal.org packaging script on 2008-02-13
version = "6.0"
project = "drupal"
datestamp = "1202913006"

Заметьте, что все после "; Information added by drupal.org packaging script on 2008-02-13" и ниже, это информация добавленная drupal.org. Вам не нужно добавлять это вручную.

Значения файла .info используемые по умолчанию

Когда значения не определены, темы использует приведенные ниже.

Примечание: эти значения переопределяются все сразу. То есть, если вы задали регион regions[sub_header] = Sub-header, то он затрет все регионы создаваемые по умолчанию. Для их возвращения, их нужно будет переопределить. Это так же работает и со стилями.

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 (минимальная поддержка)

DRUPAL_MINIMUM_PHP — это константа, она задает минимально необходимую версию PHP для ядра Drupal

php = DRUPAL_MINIMUM_PHP

Назначаем регионам содержимое

Регион — это область в теме, в которую можно вставлять блоги и другое содержимое. Регионы доступные в теме, задаются в файле .info. Регионы задаются с помощью ключа 'regions', за которым должно идти машинное имя

региона в квадратных скобках, и после знака равенства имя показываемое людям, т.е.

Если не определен ни один регион, будут использованы значения по умолчанию.

regions[left] = Left sidebar

regions[right] = Right sidebar

regions[content] = Content

regions[header] = Header

regions[footer] = Footer

Запомните, что внутренние имена регионов, автоматом конвертируются в имена переменных в шаблоне "page.tpl.php". В приведенном выше примере, регион [left] будет выводить все назначенные ему блоки, через переменную \$left. Есть некоторые ограничения, на имена задаваемые переменным PHP, убедитесь что внутренние/машинные имена соответствуют этим ограничениям. Основные ограничения состоят в том, что имя должно содержать только цифры, буквы и знак нижнего подчеркивания, первым символом должна быть буква.

Человеко-читаемые имена задаваемые в скобках, используются для обозначения региона на странице "Administer > Site building > Blocks".

Вот страница Blocks для темы Garland:

Block	Region	Operations
Left sidebar		
⊕ Navigation*	Left sidebar	configure
Right sidebar		
⊕ User login	Right sidebar	configure
Content		
⊕ Recent blog posts*	Content	configure
Header		
⊕ Syndicate*	Header	configure
Footer		
⊕ Powered by Drupal	Footer	configure

Примечания:

- Для вывода каждого блока можно создать свой шаблон (.tpl.php)
- Добавление пользовательского региона, затрет регионы «по умолчанию». Если вы хотите их сохранить, добавьте их вручную.
- Порядок в котором определены регионы сохранится на странице "Administer > Site building > Blocks". Например, Garland использует регионы по умолчанию, подметьте порядок вывода регионов на картинке.
- Тело страницы выводится в файле page.tpl.php (или его производных) через переменную \$content. Если в теме задан регион "content", то все блоки назначенные ему, будут выводиться через эту же переменную.

Файл .info кешируется, поэтому Drupal не заметит его изменения. Для очистки кеша, можно сделать следующее:

1. Кликните по "clear all cached data" на странице "Administer > Site configuration > Performance".
2. Если включен блок Devel (идет с модулем Devel), нажмите "Empty cache".
3. Зайдите на страницу "Administer > Site building > Themes".

Примечания по обновлению:

- В Drupal 5 и более ранних версиях, регионы задаются с помощью themeName_regions() или EngineName_regions(). Это не актуально для Drupal 6.
- Если бы обновляете вашу тему до версии Drupal 6, и в ней есть регионы \$sidebar_left и \$sidebar_right, переименуйте их в \$left и \$right.
- В версиях до 6, переменная \$footer_message объединяла регион footer с footer message (задается в "Administer > Site configuration > Site information"). Убедитесь что отдельная переменная \$footer создана (если ваша тема использует ее), начиная с 6 версии эти 2 элемента уже не объединяются.

Присвоение содержимого регионам вручную

Содержимое может быть присвоено регионам вручную, с помощью drupal_set_content(). Например, drupal_set_content('header', 'Приветствую!') выведет 'Приветствую!', в регионе header.

Приведем более полезный пример, и попробуем вывести сумму всех комментариев в регионе right. Измените префикс "drop" на название вашей темы. Больше информации на странице

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

```

// Задаем пару переменных
$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);

// формируем разметку
$summary = '<div class="comment-sidebar">';
$summary .= '<span class="title">' . "$title $new_marker</span>';
$summary .= '<span class="credit">' . "$by_line</span>';
$summary .= '</div>';

// Вставляем все в регион
drupal_set_content('right', $summary);
}
?>

```

Данная функция должна вызываться до того как получены блоки региона, это вызывается с помощью template_preprocess_page > theme_blocks > drupal_get_content.

Проверка занятости региона

Проверив содержание соответствующей переменной, можно узнать- пуст ли регион, и если нет, то что содержит.

```

<?php
if($left) {
    // do something
}
?>

```

Однако, переменные регионов не определены в шаблонах на уровне блока, ноды и вида.

Что бы это поправить, я адаптировал часть кода block.module, и создал функцию которая может быть помещена в template.php.

У функции один параметр (имя региона), она возвращает 1, если регион пуст,

или 0 в обратном случае. Функция учитывает текущий путь, и настройки видимости блока.

```
<?php
function region_empty($test_region) {
/* Проверяем регион на заполнение, если заполнен, возвращаем 1
 *
*/
$test_empty = 1;

$result = db_query_range('SELECT n.pages, n.visibility FROM {blocks} n WHERE
n.region="%s" AND n.theme="%s", $test_region, $GLOBALS["theme"], 0, 10);
if (count($result) > 0) {
  while ($node = db_fetch_object($result))
  {

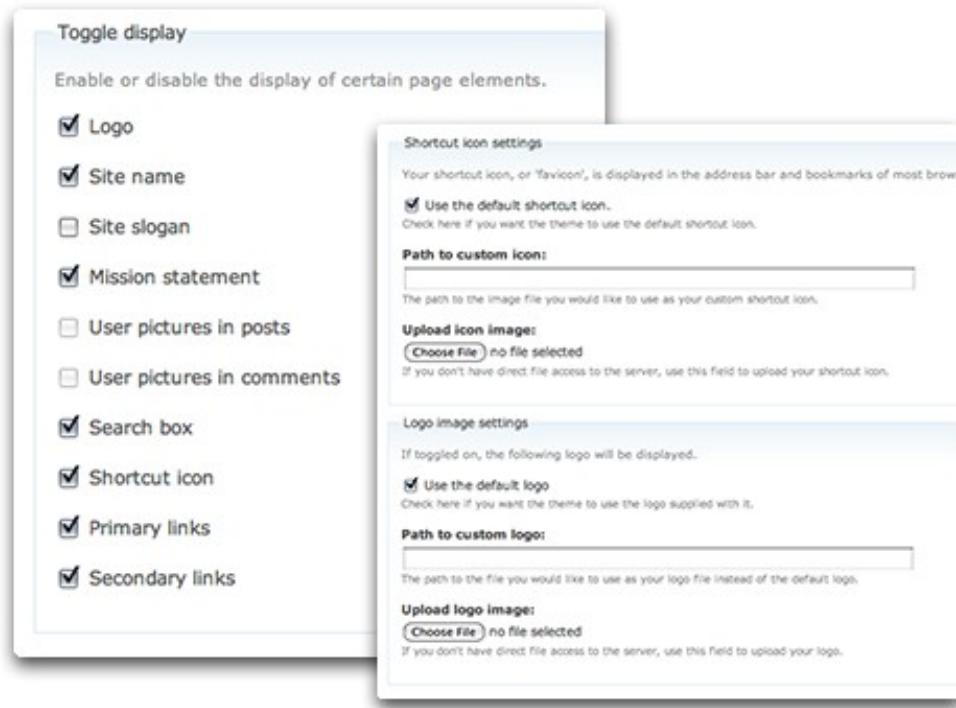
    if ($node->visibility < 2) {
      $path = drupal_get_path_alias($_GET['q']);

      $page_match = drupal_match_path($path, $node->pages);
      if ($path != $_GET['q']) {
        $page_match = $page_match || drupal_match_path($_GET['q'], $node->pages);
      }
      // Когда $block->visibility равно 0, блок показывается
      // на всех страницах, кроме заданных в $block->pages. Если 1, то
      // показывается только на страницах заданных в $block->pages.
      $page_match = !($node->visibility xor $page_match);
    } else {
      $page_match = drupal_eval($block->pages);
    }

    if ($page_match)
      $test_empty = 0;
  }
}
return $test_empty;
}
?>
```

Делаем настройки доступными на странице администрирования.

Разные элементы страницы выводимые темой, могут выключаться или включаться на странице настройки темы расположенной в "Administer > Site building > Themes > themeName". Например, если снять галочку "Site slogan", то вывод слогана будет подавлен.



Эти флаги отображаются, если они включены в файле .info. Они задаются ключом 'features' после которого идут пустые скобки, т.е. features[] = the_feature. Если ничего не задано, разрешены следующие чекбоксы

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

Что бы выключить ненужные функции, добавьте нужные в файл .info. Задание нужных функций, выключит остальные. Некоторые функции так же включают связанные формы. Например, 'logo' сделает доступной форму загрузки логотипа.

Файл .info кешируется, поэтому Drupal не заметит его изменения. Для очистки кеша, можно сделать следующее:

1. Кликните по "clear all cached data" на странице "Administer > Site configuration > Performance".
2. Если включен блок Devel (идет с модулем Devel), нажмите "Empty cache".
3. Зайдите на страницу "Administer > Site building > Themes".
4. hook_features() больше не поддерживается.

Расширенные настройки темы

Каждая тема имеет страницу настройки, она находится в admin/build/themes/settings/themeName. И на этой странице есть форма со стандартными опциями типа “Logo image settings” и “Shortcut icon settings.”

В Drupal 6 автор темы может разнообразить эту страницу добавлением дополнительных настроек. В Drupal 5 нужно установить модуль Theme Settings API прежде чем использовать метод описанный ниже.

Добавляем форму-виджет для любых настроек темы.

Первоначально нужно создать файл theme-settings.php в директории вашей формы, и добавить функцию themeName_settings() или themeEngineName_settings(). themeEngineName_settings() предпочтительнее, потому что позволяет другим пользователям проще создавать темы основанные на вашей. Для создания дополнительных форм, нужно использовать Forms API.

Например, для добавления настроек в тему Garland, нужно создать файл theme-settings.php, и добавить туда функцию garland_settings().

Если пользователь ранее сохранял форму настроек темы, сохраненные значения будут доступны в функции через параметр \$saved_settings. Виджеты добавляемые к форме должны быть возвращены как массив Forms API.

Комментарии в примере объясняют детали:

```
<?php  
// пример файла themes/garland/theme-settings.php  
  
/**  
 * Реализация функции THEMEHOOK_settings()  
 *  
 * @param $saved_settings  
 * массив сохранных настроек для этой темы.  
 * @return
```

```

* массив формы.
*/
function phptemplate_settings($saved_settings) {
/*
 * Значения по умолчанию для переменных темы. Убедитесь что $defaults
 * совпадает с $defaults в файле template.php
 */
$defaults = array(
  'garland_happy' => 1,
  'garland_shoes' => 0,
);

// Объединим сохраненные переменные, и их значения по умолчанию
$settings = array_merge($defaults, $saved_settings);

// создаем форму-виджет с помощью 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'],
);

// Возвращаем дополнительную форму-виджет
return $form;
}
?>

```

Автор темы может создавать комплексные, динамические формы с помощью Forms API (авто-дополнение, сворачивающиеся списки) и JQuery javascript.

Получение значений опций в файлах вашей темы.

Для получения значений в template.php или шаблонах .tpl.php нужно использовать функцию

theme_get_setting('varname'). Ищите детали в [Drupal API](#):

Например:

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

Инициализация значений по умолчанию

Поскольку мы не можем быть уверены что пользователь зайдет на страницу admin/build/themes/settings/themeName, мы должны убедиться что значения по умолчанию для наших настроек будут инициализированы.

Переменные настроек темы не будут установлены до тех пор, пока мы не передадим форму admin/build/themes/settings/themeName в первый раз, поэтому в файле template.php нужно проверить заданы ли переменные. Если не заданы, нужно присвоить им значения по умолчанию. Мы поймем это проверив не пуста ли одна из переменных. Если пуста, мы зададим значения по умолчанию с помощью variable_set(), и обновим настройки Drupal с помощью theme_get_setting("", TRUE).

Добавьте этот код в начале вашего template.php:

```
<?php  
/*  
 * Инициализируем настройки темы.  
 */  
if (is_null(theme_get_setting('garland_happy')))) { // <-- поменяйте эту строку  
    global $theme_key;  
  
    /*  
     * Значения по умолчанию для переменных темы. Убедитесь что $defaults  
     * совпадает с $defaults в файле template.php  
     */  
    $defaults = array(          // <-- измените этот массив  
        'garland_happy' => 1,  
        'garland_shoes' => 0,  
    );  
  
    // Получаем значения настройки темы по умолчанию.  
    $settings = theme_get_settings($theme_key);  
    // Не сохраняем toggle_node_info_variables.  
    if (module_exists('node')) {  
        foreach (node_get_types() as $type => $name) {  
            unset($settings['toggle_node_info_' . $type]);  
        }  
    }
```

```

}

// Сохраняем значения настройки темы по умолчанию.
variable_set(
  str_replace('/', '_', 'theme_'. $theme_key .' _settings'),
  array_merge($defaults, $settings)
);
// Обновляем настройки Drupal силой
theme_get_setting(" , TRUE);
}
?>

```

Переменная с именем “garland_happy” в первой строке, должна быть заменена на вашу переменную из файла theme-settings.php.

Добавление настроек в новую версию вашей темы.

После выпуска 1.0 версии вашей темы, вы, возможно, решите добавить некоторые дополнительные настройки в версию 2.0. Обратите внимание на код инициализации в 3 шаге:

1. В файле theme-settings.php добавьте новые настройки в переменные \$defaults и \$form.
2. В файле template.php добавьте настройки в переменную \$defaults в коде инициализации настроек темы.
3. В файле template.php обновите код инициализации что бы проверить наличие одной из новых настроек темы. Например, если вы добавили несколько настроек, и в их числе настройку garland_slippers, нужно изменить первую строчку кода инициализации настроек темы на:

```
if (is_null(theme_get_setting('garland_slippers'))) {
```

Это будет служить гарантией, что значения по умолчанию для новых настроек, сохранены вместе со значениями старых настроек.

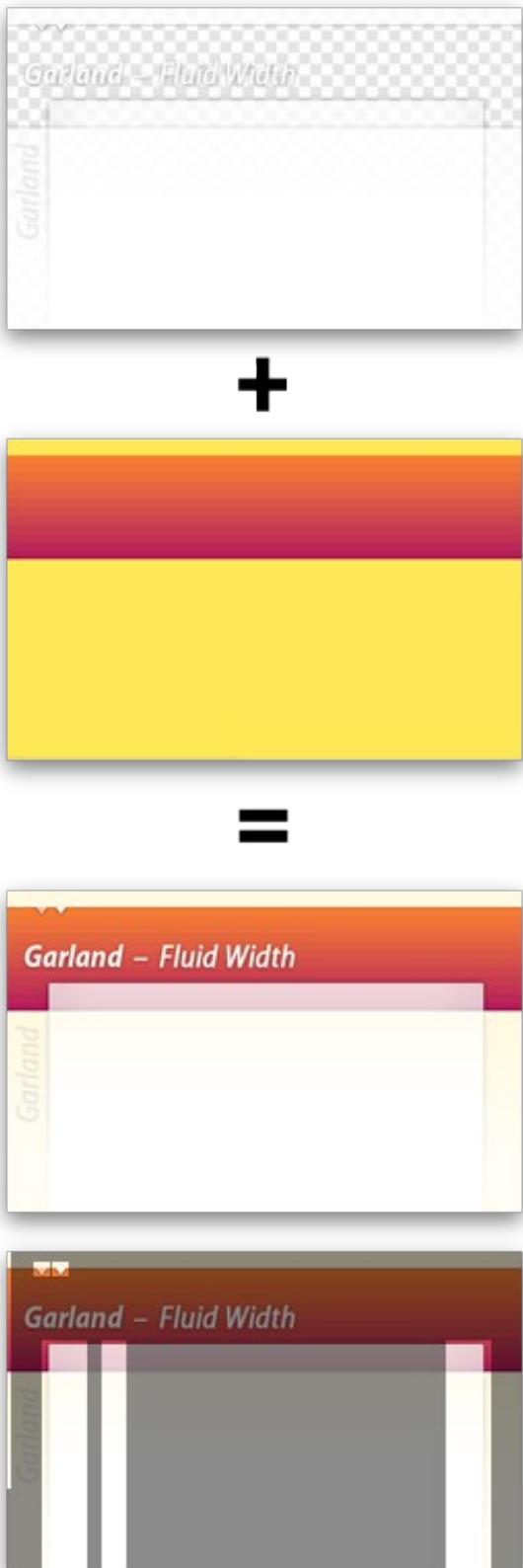
Интегрируем модуль color

Модуль Color позволяет менять администратору цветовую схему. Вы сможете изменить цвета выбранной схемы.

Модуль может менять стили, и перерисовывать изображения. Однако, тема должна предоставить конкретные хуки, а дизайн разрабатываться с учетом использования этого модуля.

Ниже объясняется базис создания темы с изменяемым цветом.

Дизайн



Важно понять как работает модуль color, не каждый дизайн может быть разукрашен.

Мы берем прозрачное изображение дизайна (основу), которое включает все, кроме бэкграунда. Затем ставим его поверх цветного бэкграунда, что бы получить разукрашенную версию. Наконец разрезаем цельное изображение на мелкие кусочки, и сохраняем в разные файлы.

Так же нужно изменить все цвета определенные в стилях. Модуль может плавно менять цвета, используя палитру. Цвета которых не в палитре, могут быть заменены на максимально подходящие (вне зависимости от того, чей цвет меняется: ссылка, текст, цвет бэкграунда).

Итак, макет фотошоп должен содержать файл со слоями, включающий один или несколько цветных слоев снизу, и со всем остальным наверху. Когда вы сохраняете основное изображение, нужно объединить все слои, при этом сделав цветные слои невидимыми. Посмотрите на [base.png](#) темы Garland. Так же есть [видео](#) показывающее как создать свой base.png используя фотошоп.

Все созданные файлы записываются в /files/css и используются вместо стандартных изображений. Это означает, что раскрашиваемая тема может работать «из коробки» без модуля color, в цветовой схеме по умолчанию.

Практика

Будем использовать тему Garland как пример. Наиболее важные файлы находятся в папке themes/garland/color:

base.png

Включает в себя основной дизайн темы.

color.inc

Файл включает в себя основные параметры для раскрашивания темы.

preview.css

Стиль используемый для генерации превью изменений цвета.

preview.png

Изображение используется для генерации превью изменений цвета.

Наличие color/color.inc делает доступным выбор цветов в настройках темы. Это обычный PHP-файл, который включает в себя массив \$info со следующими значениями:

Схемы

```
<?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'),
));
?>
```

Этот массив содержит заранее заданные цветовые схемы Каждая запись состоит из 5 цветов (основной цвет, цвет ссылок, верх шапки, низ шапки, и цвет текста) отформатированных как показано выше, а так же названия.

Первая схема используется как эталон, и должна максимально соответствовать цветам использованным в теме по умолчанию, рисункам, и стилям. Иначе, в финале можно получить не те цвета, которые хотел пользователь. Больше

информации о том как рассчитываются цвета, можно найти в секции «Стили».

Копируемые изображения

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

Массив включает в себя список изображений, которые не должны быть заменены. Они будут скопированы туда где сгенерированные изображения и стиль.

Заполнение областей и градиентов.

Для раскрашивания изображения, мы создаем изображение такого же размера, и рисуем цветные области и градиент. Для гибкости, расположение этих областей задается координатами (x, у, ширина, высота):

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

Вы можете задать вертикальный двух-цветный градиент.

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

Можно задать регион для каждого из цветов палитры. Регионы будут залиты выбранным цветом. Доступные цвета 'base', 'link', 'top', 'bottom' и 'text'.

Нарезка изображения

Далее вам нужно задать области базового изображения, для нарезки. Еще раз, вы указываете координаты (x, у, ширина, высота), вместе с названием изображения, использованным в стиле. Логотип и скриншот всегда называются одинаково.

Скриншот будет урезан до 150x90 пикселей.

```
<?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),
));
?>

```

Файлы

В конце вам нужно указать расположение файлов вашей темы. Вам нужно изображение+ стиль для превью, и базовое изображение:

```

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

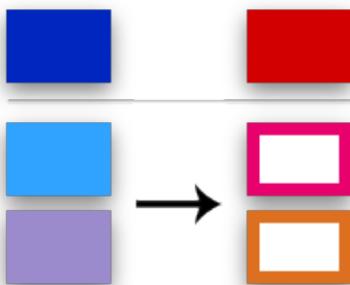
```

С 6 версии, модуль Color больше не требует наличия base_image, то есть, можно использовать модуль без изображений.

Стили

Модуль color считает файл style.css так же, как и любой другой импортированный с помощью @import, и создаст новый файл style.css. Он изменит цвета в CSS используя одну из выбранных цветовых палитр как эталонных, .в зависимости от контекста.

Однако, если цвет в стиле совпадает с одним из эталонных цветов, он будет проигнорирован.



Например, предположим что синий цвет — эталонный по умолчанию, но вы изменили его на красный. Ваш стиль включает голубой и фиолетовый цвета, оба относятся к цвету по умолчанию.

Результирующие цвета (лиловый и коричневый) отличны от красного, поскольку оригинальные цвета произошли от синего.

С технической точки зрения: сохраняются

относительные отличия в оттенке, насыщенности, и яркости.

Если модуль color использует неправильный эталонный цвет, постарайтесь раскидать разные части в разные CSS, каждый в свою секцию { ... }, что бы относительно контекста не возникало путаницы.

Если вы правили стиль темы после изменения цветовой схемы, нужно обновить цветовую схему для восстановления смешанных версий.

Если хотите что бы определенные цвета в стиле не были изменены, их нужно поставить после этого маркера:

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

Вы можете использовать этот маркер в файле style.css только 1 раз. Он действует глобально, то есть, если вы используете его внутри импортированного стиля, все цвета после @import так же останутся нетронутыми.

Делаем что бы цвета совпадали

Очень важно, что бы сгенерированные рисунки совпадали с сдвинутыми цветами в сгенерированном стиле. Иначе могут показаться уродливые края.

Что бы это работало, в тех местах где пиксели базового изображения пересекаются с цветами заданными в CSS, они должны быть простого цвета. Поскольку мы не знаем где в базовом изображении появятся цвета заданные в CSS, мы используем глобальный цвет для смешивания, который должен быть одинаков по всему дизайну. В Garland использован белый цвет. Заметьте что базовое изображение Garland, включает серые и черные пиксели, где в бэкграунде использованы только изображения (например заголовок). Другие цвета тоже отлично подходит.

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

Мазохисты могут заглянуть внутрь модуля color, особенно интересна функция _color_shift() (прим. переводчика — Судя по всему автор-старый мазохист), если хотите узнать что и почему происходит.

Изменения в PHPTemplate

И наконец, вам нужно вставить хук модуля color в вашу тему. В виде примера мы используем движок PHPTemplate, но это должно подойти и к другим движкам.

В файле template.php вашей темы добавьте этот снippet(Для Drupal 6.x):

```
<?php
/**
 * Переопределяем или вставляем переменные PHPTemplate в шаблоны.
 */
function phptemplate_preprocess_page(&$vars) {
    // Хук в color.module
    if (module_exists('color')) {
        _color_page_alter($vars);
    }
}
?>
```

В Drupal 5 вам понадобится этот снippet:

```
<?php
/**
 * Переопределяем или вставляем переменные PHPTemplate в шаблоны.
 */
function _phptemplate_variables($hook, $vars) {
    if ($hook == 'page') {

        // Хук в color.module
        if (module_exists('color')) {
            _color_page_alter($vars);
        }
        return $vars;
    }
    return array();
}
?>
```

Это позволит модулю заменить логотип, стиль, и скриншот вашей темы. Если вы внесете в _phptemplate_variables другие изменения, это так же нужно отразить

это в снippetе.

Очищаем кеш темы

Файл .info кешируется, поэтому Drupal не заметит его изменения. Для очистки кеша, можно сделать следующее:

1. Кликните по "clear all cached data" на странице "Administer > Site configuration > Performance".
2. Если включен блок Devel (идет с модулем Devel), нажмите "Empty cache".
3. Зайдите на страницу "Administer > Site building > Themes".

ЗЫ. Вторая часть перевода статьи с друпал орг. Если найдете ошибки, пишите в комментах, статья длинная, мог что то пропустить. В следующей части будет рассказано о работе с CSS при создании темы. Поэтому =====V

Часть третья. Работа с CSS

By RemaGe - Posted on 12 Сентябрь 2009

Большинство современных сайтов используют внешние таблицы стилей, для управления отображением страниц. В традиционной статичной HTML-странице, указатель на таблицу стилей должен быть помещен в страницу вручную с помощью HTML-кода (обычно в шапке страницы)

Например:

```
<link rel="stylesheet" type="text/css" href="/mytheme.css" />
```

Этот код говорит браузеру где искать таблицу стилей (mytheme.css) , которая контролирует отображение страницы.

Страница сгенерированная **Drupal**, для браузера, может выглядеть также: указатели на таблицы стилей помещенные в шапку. Отличие скрыто от глаз, оно в том, что указатели на таблицы стилей добавляются автоматически. Некоторые стили могут добавляться самой темой, другие прикрепляются разными **модулями Drupal** (для обеспечения стиля «по умолчанию», для выводимой модулем информации).

Вы можете добавить таблицы стилей в тему, а так же переопределить стандартные темы ядра, и поставляемых с ним модулей. При использовании под-темы можно перезаписать стили базовой темы.

Добавление таблицы стилей

Тут объясняется как добавить таблицу стилей используя **файл .info**.

Программное добавление таблицы стилей будет рассмотрено ниже.

Примечание:

При работе с CSS, убедитесь что оптимизация CSS выключена. Это находится на странице "Administer > Site configuration > Performance". Пока оптимизация стилей включена, изменения не будут видны. Вы можете включить оптимизацию вновь, когда закончите.

Файл .info кешируется, поэтому Drupal не заметит его изменения. Для очистки кеша, можно сделать следующее:

- Кликните по "clear all cached data" на странице "Administer > Site configuration > Performance".
- Если включен блок Devel (идет с модулем Devel), нажмите "Empty cache".
- Зайдите на страницу "Administer > Site building > Themes".

По умолчанию, если в файле .info не указанна ни одна тема, будет использоваться "style.css". Добавить новый стиль просто, для этого служит ключ 'stylesheets'. Запомните, что добавление какого-либо стиля предотвратит загрузку "style.css". Поэтому, если ваша тема использует «style.css», не забудьте добавить и его.

; Добавляем общую таблицу стилей
stylesheets[all][] = theStyle.css

stylesheets[screen, projector][] = theScreenProjectorStyle.css

; Добавляем стиль для печати
stylesheets[print][] = thePrintStyle.css

Примечания:

Обратите внимание на пустые скобки между [media] и = styleName.css.

Порядок в котором стили перечислены в шапке страницы, повторяет порядок заданный тут.

Таблицы стилей могут быть помещены в под-папках, то есть: stylesheets[all][] = stylesheets/styleName.css. Это удобно для упорядочивания.

Добавление стилей с помощью API

Возможности добавления стилей через файл .info должно быть достаточно для большинства тем. Учитывая статичность файла .info, темы не могут добавляться

динамически. В зависимости от того, как тема использует таблицы стилей, это может вообще не иметь значения. Если сомневаетесь – используйте файл .info. Есть API- функции для работы с таблицами стилей, **drupal_add_css** и **drupal_get_css**. Приведем пример динамического добавления стилей:

Измените префикс «drop» на название вашей темы

```
<?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();
  }
}
?>
```

Этот код подключит таблицу стилей "front-page.css" на главной страницу, или множество других стилей название которых основано на имени страницы. Например, на странице <http://example.com/admin> будет подключен стиль "path-admin.css".

Примечания:

В зависимости от того, где и когда добавляются стили, может понадобиться drupal_get_css, для того, что бы включить дополнительные стили. Изначально они задаются в template preprocess_page.

В функции drupal_add_css есть параметр отвечающий за агрегацию добавленного файла. Это нужно отключить (как в приведенном примере), когда подключение таблиц очень динамично, так как агрегирование создаст новый общий файл CSS. Это может замедлить создание страницы, и увеличить трафик.

Куда добавлять код

Этот код можно добавить в файл template.php в директорию вашей темы. Там

уже может существовать функция "phptemplate_preprocess_page". Можно просто включить ее в тело вашей функции XXX preprocess_page.

Так же, возможно придется заменить "variables" на "vars", что бы заставить ее работать.

Создание и интерпретация CSS шаблона (style.css.php)

Хотели создать динамичный CSS? Вот пример как задать шаблону базовый CSS, в который вы можете вставить переменные.

В этом примере мы будем динамически задавать размер заголовка страницы. Создайте шаблон CSS (style.css.php):

```
@CHARSET "UTF-8";  
#content h1.title{ font-size: <?php print $font_size; ?>px; }
```

Код для интерпретации шаблона и получения результирующей строки:

```
$variables = array(  
  'font_size' => 20,  
);  
  
extract($variables, EXTR_SKIP);  
ob_start();  
include('style.css.php');  
$css = ob_get_contents();  
ob_end_clean();
```

результирующая переменная \$css может быть использована для подключения или вставки стиля разными путями:

```
print '<style type="text/css">'.$css.'</style>';
```

Переопределяем таблицу стилей модулей или базовой темы

Ядро и поставляемые с ним модули по умолчанию определяют вид выводимых ими данных. Это включает как саму разметку, так и связанные с ней таблицы стилей. Благодаря расширяемой натуре Drupal, обработка всего что выводится в браузер, могло бы стать тяжким бременем для темизаторов. Но эти «значения по умолчанию» могут быть изменены по усмотрению темизатора. Таблицы стилей выводимые ядром и модулями могут быть переопределены так же, как и функции темизации и шаблоны. Не надо напрямую их изменять. Все изменения должны находиться в вашей теме.

Переопределяем таблицы стилей ядра и поставляемых с ним модулей

Для замещения таблицы стилей ядра или поставляемых с ним модулей, стиль должен быть

переопределен в файле .info. Возьмем для примера "system-menus.css". Он находится в "modules/system/system-menus.css". Для игнорирования системой оригинального файла, и загрузки вашего, добавьте следующую строку:

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

Когда переопределяется стиль модуля, актуальный файл CSS должен быть представлен внутри вашей темы и указывать на верное местоположение с совпадающим именем. Это гарантирует то, что оригинал будет замещен таблицей стилей вашей темы.

Примечания:

Переопределение CSS файлов ядра, предотвратит загрузку "style.css". Не забывайте определять «значения по умолчанию», если они используются.

Переопределения темы должны иметь те же медиа типы (media type) что и оригинальный стиль.

Любые присоединенные в тиле элементы, должны быть верны. Дважды проверьте что все свойства 'url()' или правила '@import' в файле, ссылаются на правильные ресурсы.

Порядок вызова стилей перечисленных в шапке страницы будет меняться. Это может изменить некоторые каскадные правила.

Некоторые стили ядра и модулей загружаются условно. Их переопределение в файле .info сделает их используемыми всегда.

Если нужны незначительные изменения, для переопределения стиля можно использовать CSS селекторы, вместо переопределения файла целиком.

Не забывайте очищать кеш после каких либо изменений! Зайдите в Administer > Site configuration > Performance. Прокрутите вниз, и кликните "Clear cached data."

Переопределение таблицы стилей базовой темы.

Что бы не загружать стиль базовой темы, вы можете перепределить таблицу стилей внутри файла .info. Точно так же, как и переопределение стиля модуля или ядра.

Значения в базовой и под-теме должны быть одинаковы:

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

Если файл существует внутри под-темы, он будет использован, если нет, то и файл базовой темы не будет загружен.

Стандартные стили и классы ядра Drupal

В Drupal принят модульный подход для CSS классов стандартных элементов страницы. Число классов одинаково везде. Этот список напоминает для чего какой класс, и где он появляется. Полный список классов можно найти в Zen starter kit.

Причение:

В загружаемых темах эти классы могут быть изменены, или добавлены новые.

Элементы страницы

.menu

Этот класс получают все меню, например меню navigation.

.block

Все блоки, <http://drupal.org/node/104319> — статья о стилизации блоков.

.links

Список ссылок, включая Primary и Secondary links в шапке страницы, а так же ссылки ноды и терминов таксономии.

Элементы нод

.node

Div обертывающий ноду, включая ее заголовок.

.node-title

Заголовок ноды.

.content

Тело ноды. Это относится и дополнительным модулям наподобие CCK или uploaded files.

.links

Относится ко всем UL являющимся списком ссылок, включая Primary и Secondary links в шапке страницы, а так же ссылки ноды, и термины таксономии. Однако ссылки ноды получают класс .links от вмещающего их DIV.

.terms

Термины таксономии, так же получающие .links и .inline.

.inline

Системный класс для пунктов UL размещенных по горизонтальной линии.

.feed-icon

Иконка RSS, обычно внизу области контента.

Поддержка языков "с права на лево" (RTL)

Включение поддержки RTL языков связанно с переопределением побочных стилей, и парного именования файлов с таблицами стилей.

Включение стилей RTL происходит автоматически. Включение зависит от языковых параметров установленных для сайта.

Например, в теме ядра Garland, "style.css" — основной стиль. Он так же включает "style-rtl.css" для языков RTL наподобие арабского. При включении 2 стилей, сначала грузится основной стиль, а потом RTL стиль. Это позволяет сделать каскадными все наборы правил, не беспокоясь о селекторах использованных в стилях RTL.

Существуют стандарты кодирования, что бы держать стили в порядке.

Например комментарий /* LTR */ ставится после строк связанных с LTR- языками. Это относится к floats, margins, padding и т.д.

Пример основного стиля:

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

Соответствующий RTL стиль:

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

При работе с основным файлом CSS, это позволяет минимизировать необходимые изменения.

ЗЫ. Третья часть перевода статьи с друпал орг. Если найдете ошибки, пишите в комментах, статья длинная, мог что то пропустить. В следующей части будет рассказано о работе с файлом template.php, и о том как изменить вывод всего что только можно в Drupal, хотя может быть я временно прерву эту серию переводов, и накатаю статью о SEO оптимизации сайта на Drupal.

Drupal - Урок 11 - Модуль Panels.

By RemaGe - Posted on 08 Октябрь 2009

Создание сайтов давно перестало быть простым делом. Подумать только, для написания мало-мальски нормального сайта нужно знание HTML, CSS, PHP, SQL, java script, и это только самые популярные в своих нишах языки. Радует то, что веб-разработчики в целом, и разработчики Drupal в частности, не сидят на месте, и пытаются всячески упростить жизнь обычных смертных. Сегодня мы поговорим о модуле Panels. Если модуль Views в основном служил для не напряжного составления SQL-запросов, и помогал наиболее комфортно представить их в нужном виде, то модуль Panels помогает расположить на странице (или в блоке) отдельные куски информации, без знаний CSS или HTML (Кстати, если не ошибаюсь у этих модулей один разработчик).

Для начала скачаем сам модуль. Не так давно вышел первый стабильный релиз, поэтому спешим скачать) Для своей работы Panels требуют ctools.

Итак, на операционном столе: Drupal 6.14, Ctools 6.1, Panels 6.3. Нужно установить модули: Chaos tools, Page manager, Mini panels, Panel nodes, Panels. Переходим на страницу «Your_sites/admin/build/panels», и наблюдаем такую картину:

Panel page — Добавить панель-страницу, то есть панель занимает целую страницу, при этом имеет свой путь и ссылку в меню. Так же в них можно передавать аргументы.

Panel node — Примерно то же самое что и Panel page, только слегка ограниченные. Панель встраивается в ноду.

Mini panel — Панель в виде блока.

Как я уже говорил, панели помогают располагать различную информацию на странице. Вставить туда можно многое, думаю лучше это будет показать на примере.

Давайте создадим Panel page и поиграемся с настройками.

Заходим на страницу «Home > Administer > Site building > Panels» и кликаем «Panel page».

И видим страницу с основными настройками:

Administrative title — имя страницы отображаемое для администратора. Я поставил «page».

Machine name — машинное имя, после сохранения, не изменяется. Я поставил «page».

Administrative description — описание панели, если делаете сайт навеселе, стоит заполнить. Не заполнял.

Path — путь по которому будет доступна эта страница. Поставил «page». Make this your site home page — чекбокс, если выбран, то данная панель заменит главную страницу.

Optional features — дополнительные настройки панели, я отметил все:

Access control — выбираем кто может просматривать страницу.

Visible menu item — добавляем ссылку в меню.

Selection rules — можно задать разные варианты панели, для разных пользователей.

Contexts — если в двух словах, то contexts позволяет использовать в панеле дополнительные объекты. Например, если вы добавите contexts «Node add form», то сможете вставить в панель форму добавления нод, притом как полностью, так и отдельные ее куски.

Нажимаем «Continue», и попадаем на страницу «Access control» (если вы конечно отметили этот чекбокс на первой странице).

Можно определять доступность страницы 3 методами:

PHP Code — вбиваем свою проверку на PHP, если она вернет true, пользователь увидит страницу.

User: permission — если у пользователя есть разрешение на что-то (например на создание нод типа story), то ему будет доступна и эта страница.

User: role — можно выбрать роли которым будет доступна эта страница.

All criteria must pass - Пользователь получит доступ к странице, если удовлетворены все условия.

Only one criteria must pass - Пользователь получит доступ к странице, если удовлетворено хотя бы 1 условие.

Переходим далее.

На этой странице можно задать элемент меню для страницы, я выбрал Normal menu entry, Title: page, Menu: Navigation, то есть в меню Navigation, появится ссылка «Page», на страницу с панелью.

Далее идут настройки вариантов панели, то есть, все вышеперечисленное относилось к панели в целом, все что идет далее, будет заполняться для каждого варианта.

Страница «Selection rules» в точности такая же как и «Access control», только критерии показа задаются ни ко всей странице, а к вариантам.

на странице Contexts можно добавить разные объекты сайта для отображения. Я добавил « Node add form»

и Node (перед этим создав страницу, и запомнив ее id)

Страница «Choose layout», здесь нужно выбрать на сколько регионов будем

разбивать страницу:

Крайне важное свойство, от которого зависит расположение элементов, я выбрал 2 колоночный вариант, и попал на последнюю страницу:

На которой и происходит наполнение панели информацией, жмем на шестеренку, должно отобразиться 2 пункта:

Add content — добавление содержимого,

Region style: Default — выбор стиля панели.

жмем Add content, и выбираем что же вставить в панель:

Все описывать не буду, т.к. Это долго и непродуктивно, да и в зависимости от ситуации, эта страничка может измениться. Например, в нашем случае, добавились 2 вкладки: Form, после добавления Contexts «Node add form», и Node, после добавления Contexts «Node». Да и лучше будет если вы сами попробуете разные варианты.

Посмотрите 2 последних пункта:

New custom content - позволяет добавить произвольное содержимое, то есть любой нужный вам текст.

Existing node — позволяет вставить существующую ноду.

Я добавил в первую вкладку тело существующей ноды, и блок «Who's online», а во вторую ноду целиком, и форму добавления ноды.

а так выглядит готовая страница (не забудьте сохранить результат):

Если вы справились с Panel page, остальное не должно вызвать у вас проблем.

Естественно, учитывая одного создателя, модули Views и Panels, отлично работают друг с другом (если честно, в основном я использовал Panels для того что бы быстро раскидать блоки Views), но об этом, и многом другом, мы поговорим в следующий раз.

Спасибо за внимание.

Drupal - Урок 12 - Работа с изображениями. Модуль Image cache.

By RemaGe - Posted on 20 Октябрь 2009

Интернет и графику в наше время уже не разъединить. Кому интересен голый текст? Drupal, конечно же, обладает средствами для работы с графикой.

Давайте обсудим это поподробнее.

Для начала нам понадобится сам Drupal, на сегодняшний день актуальна версия — 6.14, в придачу к нему установим [cck](#), [imagefield](#) и [filefield](#). Нам необходимы входящие в эти пакеты модули: Content, ImageField, FileField, Option Widgets. Устанавливаем все. Теперь мы можем прикреплять к нашим типам содержимого рисунки. Давайте попробуем видоизменить тип содержимого story. Заходим в «Administer > Content management > Content types». Находим Story, и нажимаем «manage fields», напротив него. Добавим поле с рисунком:

Label: image

Field name: image

Type of data to store: file

Form element to edit the data: Image



Сохраняем, появляется окно настройки поля, можно оставить все по умолчанию, хотя настройки довольно таки интересны:

Permitted upload file extensions — разрешенные типы файлов, по умолчанию png gif jpg jpeg.

Maximum/Minimum resolution for Images — максимальное и минимальные разрешенные размеры рисунков (в пикселях), по умолчанию 0, то есть можно загружать рисунки любых размеров, забивается в формате 640x480.

File path — путь, куда будут загружаться рисунки, по умолчанию «files/».

File size restrictions — максимальный размер загружаемых файлов.

Maximum upload size per file — максимальный размер загружаемого файла.

Maximum upload size per node — максимальный размер загружаемых файлов на ноду.

ALT text settings — добавлять ли к рисункам свойство ALT.

Title text settings — добавлять ли к рисункам свойство Title.

Default image — Если не загружено изображение, использовать ли какое либо по умолчанию?

Required — поле обязательно?

Number of values — Сколько изображений можно прикрепить к ноде?

image

Story settings

These settings apply only to the Image field as it appears in the Story content type.

Help text:

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

Allowed HTML tags: <a> <big> <code> <i> <ins> <pre> <q> <small>

Permitted upload file extensions :

png gif jpg jpeg

Extensions a user can upload to this field. Separate extensions with a space and do not include the leading dot. i.e.

Maximum resolution for Images:

0

The maximum allowed image size expressed as WIDTHxHEIGHT (e.g. 640x480). Set to 0 for no restriction. If a height is specified, resizing images on upload will cause the loss of EXIF data in the image.

Minimum resolution for Images:

0

The minimum allowed image size expressed as WIDTHxHEIGHT (e.g. 640x480). Set to 0 for no restriction. If a height is specified, resizing images on upload will cause the loss of EXIF data in the image.

—> **Path settings**

—> **File size restrictions**

—> **ALT text settings**

—> **Title text settings**

—> **Default image**

Global settings

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

Required

Number of values:

1

Maximum number of values users can enter for this field.

Unlimited will provide an Add more button so the users can add as many values as they like.

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

List field:

Я оставил все по умолчанию. После сохранения, при создании Story, к ним можно будет прикреплять рисунки.

The screenshot shows a modal dialog box on a Drupal website. At the top left of the dialog is the number '111'. Below it is the date and time 'Tue, 10/20/2009 - 22:12 — rem'. The main content of the dialog is a code snippet:

```
Parents: page.tpl.php  
Function called:  
phptemplate_menu_local_tasks()  
Candidate function names:  
garland_menu_local_tasks < phptemplate_menu_local_tasks <  
theme_menu_local_tasks  
Duration: 0.22 ms
```

Below the dialog, there is a link 'Add new comment' and a small cursor icon pointing towards it.

При помощи тематизации, вы легко сможете разместить картинку так, как вам нужно, но если вы хотите подогнать картинку под какие то только вам ведомые стандарты? Обрезать и скалировать каждое изображение фотошопом, жуть как неудобно... И тут на помощь приходит модуль [imagecache](#), как и большинство полезных модулей Drupal, сам он работать отказывается, и требует [imageapi](#). Устанавливаем ImageAPI, ImageCache, ImageCache UI, и поддержку одной из библиотек, в зависимости от того, какая установлена у вас на сервере, в моем случае это ImageAPI GD2. Сохраняем.

Кстати, для работы Image cache требуется включить Clean URL.

Теперь мы можем создавать пресеты (preset), или если вам будет проще, я бы назвал их «фильтрами». Заходим на страницу «Administer > Site building > ImageCache». И нажимаем «Add new preset».



Manage ImageCache presets.

Preset Name

Вводим имя фильтра



Save Preset

Сохраняем и попадаем на страницу выбора действий (Action) с изображением. К каждой картинке можно применить сразу несколько действий.

Edit preset: first

Preset Namespace:

first

The namespace is used in URL's for images to tell imagecache how to process an image.

Actions

▽ New Actions

- Add Crop - Crop an image to the rectangle specified by the given offset.
- Add Deprecated Scale - Precursor to Scale and Crop. Has inside and outside options.
- Add Desaturate - Convert an image to grey scale.
- Add Resize - Resize an image to an exact set of dimensions, ignoring aspect ratio.
- Add Rotate - Rotate an image.
- Add Scale - Resize an image maintaining the original aspect-ratio (or not).
- Add Scale And Crop - Resize an image while maintaining aspect ratio and crop.
- Add Sharpen - Sharpen an image using unsharp masking.



Save Preset

Preview:

Add Crop — обрезать картинку.

Add Deprecated Scale — Предшественник «Add Scale And Crop», исчезнет в версии ImageCache 2.1.

Add Desaturate — Сделать изображение черно-белым.

Add Resize — изменить размер изображения.

Add Rotate — повернуть изображение.

Add Scale — изменить размер изображения, с сохранением пропорций.

Add Scale And Crop — Изменить размер изображения до нужного, с сохранением пропорций, а потом обрезать.

Add Sharpen — изменить четкость изображения.

Я задал скалирование (Scale) до 200 пикселей по ширине, и сделал изображение черно-белым. Давайте добавим еще один фильтр, например, со скалированием до 400 пикселей по ширине, и поворотом (Add Rotate) изображения на 90 градусов.

Вот что у меня получилось.

Первый фильтр, я назвал его First

Edit preset: first

The directory `sites/default/files/imagecache/first` has been created.

Preset Namespace:

first

The namespace is used in URL's for images to tell imagecache how to process an image, names.

Actions

Action	Settings
⊕ Scale	width 200
⊕ Desaturate	
→ New Actions	



[Save Preset](#)

Preview:

http://localhost/drupal/sites/default/files/imagecache/first/imagecache_sample.jpg



второй фильтр, назвал Sec

The directory `sites/default/files/imagecache/sec` has been created.

Preset Namespace:

sec

The namespace is used in URL's for images to tell imagecache how to process an image, names.

Actions

Action	Settings
⊕ Scale	width 400
⊕ Rotate	90°
→ New Actions	

[Save Preset](#)

Preview:

http://localhost/drupal/sites/default/files/imagecache/sec/imagecache_sample.jpg



Теперь давайте сделаем что бы этот фильтр накладывался на загружаемые нами

изображения.

Зайдем в «Administer > Content management > Content types», нажмем «manage fields», напротив типа Story, и там «Display fields».

Field	Label	Teaser	Exclude	Full node	Exclude
Image	Image	Image	<input type="checkbox"/>	Image	<input type="checkbox"/>

Здесь можно задать как будет отображаться наше изображение.

Выберем **Label — Hidden**, что бы убрать противную надпись «image», у нод, **Teaser — First image**, то есть, на изображение в тизере будет накладываться первый фильтр, и **Full node — Sec image**, на изображение в полной ноде будет накладываться второй фильтр.

Проверяем работу... В тизере исходное изображение уменьшено, и в черно-белых тонах.

А в полной ноде, перевернуто и уменьшено.

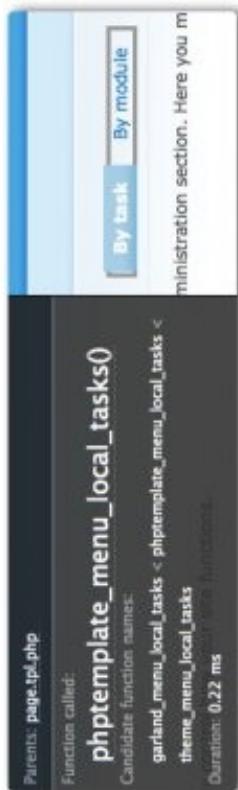
111

[View](#)

[Edit](#)

Tue, 10/20/2009 - 22:12 — [rem](#)

2222



[Add new comment](#)

Как вы уже наверно поняли, Image cache в связке с CCK, и Views, дает неограниченные возможности при умелом использовании, главное не бойтесь пробовать что то новое -)

Drupal - Урок 13 - введение в Ubercart, создаем простой онлайн-магазин.

By RemaGe - Posted on 26 Октябрь 2009

Все мы знаем, что кризис только усилил позиции и доходы онлайн-торговли. Не возьмусь объяснить почему, но люди стали охотнее вкладывать в это деньги. Вас посещала идея создать свой **онлайн-магазин**? Но не знаете как это реализовать, и не хочется отдавать кровно заработанные чужим дядям? Давайте поговорим о создании **онлайн-магазина на Drupal**.

Самые популярные пакеты модулей для реализации шопа, - это **Ubercart**, и **e-Commerce**, сегодня мы будем работать с первым из них. Для начала нам понадобится сам [drupal](#), и модули [ubercart](#): Cart, Conditional Actions, Order, Product, Store, и Catalog из Ubercart - core (optional). Так же, для работы Ubercart необходим модуль [token](#). Для работы с картинками установим модули ImageAPI, ImageAPI GD2, ImageCache, из пакетов [imagecache](#) и [imageapi](#).

Enabled	Name	Version	Description
<input checked="" type="checkbox"/>	Cart	6.x-2.0	REQUIRED, Contains: Order, Product, Store. Depends on: Configuration, Test Gateway (disabled), Shipping quote (disabled), Weight quote (disabled). Required by: Cart (disabled), Shipping quote (disabled), Weight quote (disabled).
<input checked="" type="checkbox"/>	Conditional Actions	6.x-2.0	REQUIRED, Creates: Order, Product, Store. Required by: Cart (disabled), Payment method (disabled), UPS.
<input checked="" type="checkbox"/>	Order	6.x-2.0	REQUIRED, Records: Order, Product, Store. Depends on: Configuration, Test Gateway (disabled), Role (disabled), Card (disabled), Quotes (disabled), Service (disabled). Required by: Cart (disabled), Shipping quote (disabled), Weight quote (disabled).
<input checked="" type="checkbox"/>	Product	6.x-2.0	REQUIRED, Adds: Order, Product, Store. Depends on: Configuration, Test Gateway (disabled), Role (disabled), Card (disabled), Quotes (disabled), Service (disabled), UPS (disabled).
<input checked="" type="checkbox"/>	Store	6.x-2.0	REQUIRED, Sets up: Order, Product, Store. Required by: Configuration, Test Gateway (disabled), CyberSource (disabled), Shipping quote (disabled), Weight quote (disabled), UPS (disabled).

Устанавливаем все. Магазин готов -)

В Ubercart автоматизировано очень многое, например, он автоматом создал словарь таксономии, который будет служить для нас каталогом продукции. Заходим в «Administer > Content management > Taxonomy > List terms», нажимаем «Add term».

Add term to Catalog

[List](#)[Add term](#)

Identification

Term name: *

The name of this term.

Image:[Обзор...](#)**Description:**
.....

A description of the term. Displayed to customers at the top of catalog pages.

Advanced options

[Save](#)

Вбейте пару терминов, я добавил термины «Футболки», и «Шапки».

Теперь перейдем в «Create content > Product» и создадим пару продуктов. Тут все довольно таки стандартно, только добавились поля специфичные для магазина, такие как размер, вес и т.д.

Костяк магазина уже готов.

Футболка рас

Mon, 10/26/2009 - 14:26 — rem

\$5.00

\$5.00

[Add to cart](#)[Add new comment](#)[Футболки](#)

Но чего то не хватает... Не хватает фотографии товара. Устанавливаем [cck](#), а именно модули: Content, Image field, file field и модуль Thickbox, для красивого отображения.

Переходим в «Administer > Store administration», и нажимаем «Click here».

The screenshot shows the left sidebar of the Drupal admin interface with the 'Store administration' menu selected. On the right, under the 'Reports' tab, there is a table titled 'Status messages:'.

Title	Description
Catalog vocabulary	Vocabulary Catalog has been
Images	Click here to automatically co The Image file field has no (This action is not required a

И, о чудо, к нашим товарам, добавилось поле для картинки. Для работы Thickbox зайдем в «Administer > Site configuration > Thickbox», и выставим Imagecache preset: product full.

Thickbox

Image field options (CCK)

Image field gallery:

- Per page gallery
- Per post gallery
- Per field gallery
- No gallery

Should the gallery be images within a single field, a single post (default) or all images?

Imagecache preset:

product_full



Select which Imagecache preset to use for viewing images in the thickbox.

Login settings

- Enable for login links

Automatically activate Thickbox for links to user/login.

Сохраняем, и пробуем.

Задаем в редактирование созданного продукта. И видим поле Image.
Добавляем пару картинок.

Image:



Alternate Text:

Футболка с крылышками

This text will be used by screen readers, search engines, or when the image cannot be loaded.

Title:

Футболка с крылышками

The title is used as a tool tip when the user hovers the mouse over the image.

[Remove](#)



Alternate Text:

This text will be used by screen readers, search engines, or when the image cannot be loaded.

Title:

The title is used as a tool tip when the user hovers the mouse over the image.

[Remove](#)



Alternate Text:

This text will be used by screen readers, search engines, or when the image cannot be loaded.

Title:

The title is used as a tool tip when the user hovers the mouse over the image.

[Remove](#)

[Обзор...](#) [Upload](#)



Maximum Filesize: 2 MB

Allowed Extensions: gif jpg png

Готовый результат

Футболка рас
Mon, 10/26/2009 - 14:26 — rem

\$5.00

[Add to cart](#)



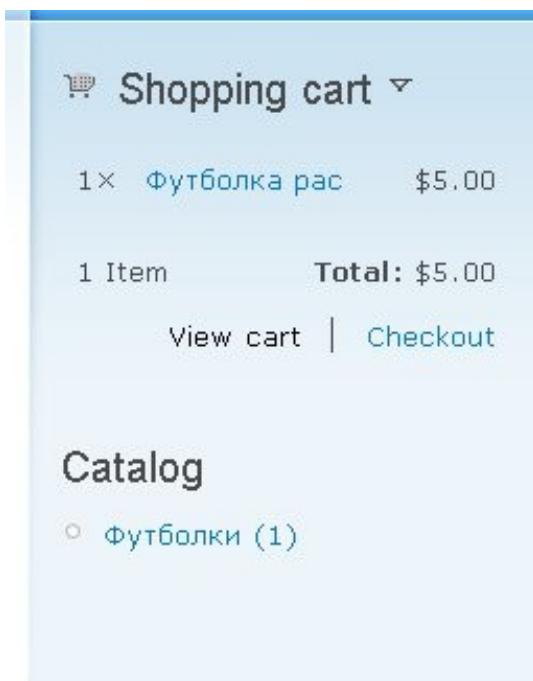
\$5.00

Add new comment

Футболки

Последний штрих.

Заходим в «Administer > Site building > Blocks», и выводим на обозрение блоки Shopping cart (корзина) и Catalog (каталог товаров).



Shopping cart ▾

1× Футболка рас \$5.00

1 Item Total: \$5.00

[View cart](#) | [Checkout](#)

Catalog

Футболки (1)

Магазин готов. Единственное что неплохо бы добавить, это оплату по вебмани, но у меня сейчас нет кошелька, и я не могу вам это показать (, поэтому оставим это и расширенные настройки магазина, на следующий урок -)

Спасибо за внимание.

Drupal - Урок 14 - Создание опросов и страницы "Обратная связь" с помощью модуля Webform.

By RemaGe - Posted on 22 Ноябрь 2009

Очень часто на Drupal.ru возникают вопросы, о том как создать форму контактов (или что-либо подобное). В поставке Drupal идет отличный модуль Contact, который реализует эту форму, а так же общение между пользователями сайта. Этот модуль используется на моем блоге (пока), а так же, если не ошибаюсь, на самом Drupal.ru, но сегодня мы поговорим о модуле . Он позволяет создавать опросы, страницы контактов, и многое другое. На одном из сайтов посвящённых Drupal даже продают за \$100 сборку простого магазина с использованием webform.

Для начала его нужно установить, думаю с этим у вас проблем не возникнет. После установки у нас появляются 2 новых страницы на вкладке Administer: **Webforms** (Content management) — на которой отображаются все созданные Webforms, и **Webform** (Site configuration) — страница с основными настройками.

Content types

Manage posts by content type, including default status, front page promotion, etc.

Post settings

Control posting behavior, such as teaser length, requiring previews before posting, and the number of posts on the front page.

RSS publishing

Configure the number of items per feed and whether feeds should be titles/teasers/full-text.

Taxonomy

Manage tagging, categorization, and classification of your content.

Webforms

View and edit all the available webforms on your site.

User management

Manage your site's users, groups and access to site features.

Access rules

List and create rules to disallow usernames, e-mail addresses, and IP addresses.

Permissions

Determine access to features by selecting permissions for roles.



Roles

List, edit, or add user roles.

User settings

Configure default behavior of users, including registration requirements, e-mails, and user pictures.

Users

List, add, and edit users.

Reports

View reports from system logs and other status information.

Recent log entries

View events that have recently been logged.

Top 'access denied' errors

View 'access denied' errors (403s).

Top 'page not found' errors

View 'page not found' errors (404s).

Available updates

Get a status report about available updates for your installed modules and themes.

Status report

Get a status report about your site's operation and any detected problems.

well as rename and reor

Modules

Enable or disable add-or

Themes

Change which theme yo

Site configuration

Adjust basic site configurati

Actions

Manage the actions defin

Administration theme

Settings for how your ac

Clean URLs

Enable or disable clean l

Date and time

Settings for how Drupal default timezone.

Error reporting

Control how Drupal deal error reporting.

File system

Tell Drupal where to stor

Image toolkit

Choose which Image too

Input formats

Configure how content is Also allows enabling of r

Logging and alerts

Settings for logging and system events to differen email, ...etc.

Performance

Enable or disable page c bandwidth optimization c

Site information

Change basic site inform address, mission, front p

Site maintenance

Take the site off-line for

Webform

Create configuration of w

На первой пока пусто, зайдем на вторую, посмотрим что тут можно изменить.

Available components — выбираем какие компоненты будут доступны при создании форм.

Markup — добавляет произвольный HTML (или даже PHP при желании).

Date — Выбор даты.

Time — Выбор времени.

Textarea - поле для ввода текста.

Pagebreak - элемент разбивает форму на страницы.

Grid — сетка из радио кнопок.

Hidden — скрытое поле, в нем можно хранить нужную нам информацию.

Fieldset — позволяет группировать поля.

Textfield - строка для ввода текста.

Email - поле для ввода почтового адреса.

File - позволяет пользователям прикреплять к форме файлы.

Select — позволяет добавлять чекбоксы, радио-кнопки.

Default e-mail values:

From address — адрес с которого придет письмо с результатами.

From name — имя отправителя.

Default subject — Тема письма.

Advanced options:

Allow cookies for tracking submissions — разрешить использовать cookies, для предотвращения повторного заполнения формы пользователем.

Default export format — формат экспорта по умолчанию.

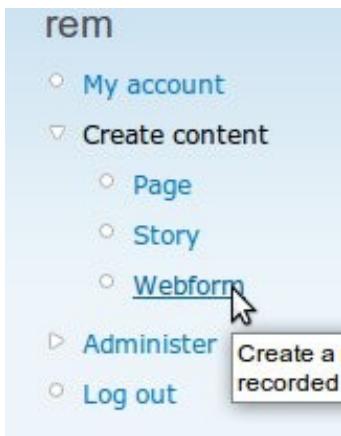
Default export delimiter — разделитель записей при экспорте по умолчанию.

Submission access control — как контролировать доступ: по ролям, или не контролировать.

Webforms debug — нужно ли вести лог.

Работать с webform можно и без этих настроек. Давайте попробуем создать свой Webform.

Кликните на «**Create content** → **Webform**», в меню navigation.



Страница добавления webform во многом схожа со страницей добавления ноды (которой webform по сути и является).

A screenshot of the 'Create Webform' configuration page. The page title is 'Create Webform'. Below it is a list of configuration sections, each preceded by a blue triangle icon: 'Menu settings', 'Webform Settings', 'Webform access control', 'Webform mail settings', 'Webform advanced settings', 'Revision information', 'Comment settings', 'Authoring information', and 'Publishing options'. At the bottom of the page are two buttons: 'Save' and 'Preview'.

Menu settings — при создании webform, можно сразу создать пункт меню на нее ссылающийся.

Webform Settings

Title — заголовок (обязательный параметр).

Description — описание.

Confirmation message or redirect URL — если введете сообщение, то оно будет показано пользователю после заполнения webform, если ссылку, то по ней пользователь будет перекинут.

Webform access control — какие роли имеют доступ к форме.

Webform mail settings:

E-mail to address — почтовый адресс, на который будут направляться результаты заполнения webform, можно оставить пустым, а можно ввести несколько через запятую.

E-mail from name, E-mail from address, E-mail subject — по умолчанию равны значениям вбитым на странице webform, но для каждого отдельного опроса (webform), можно вбить свои, или указать, чтобы их значения были равны какому-нибудь полю опроса.

Webform advanced settings:

Show complete form in teaser — показывать ли полную форму в тизере.

Limit the number of submissions a user may send within a specified time period — позволяет указать сколько раз пользователь может заполнить опросник за определенный промежуток времени. По умолчанию — без ограничения.

На рисунке задано, что пользователь может заполнять форму 3 раза в час.

Limit the number of submissions a user may send within a specified time period

Unlimited

Limit to submission(s) ▾

Submit button text — произвольный текст для кнопки submit.

Additional Validation — с помощью PHP, можно задать дополнительные параметры проверки.

Additional Processing — после проверки, с помощью PHP, можно произвести дополнительную обработку данных.

Остальные настройки повторяют настройки у стандартных нод.

Создадим свой опрос, вбив в title произвольный текст, и сохранив. Я назвал его «123».

123 View **Edit** Results

Configuration **Form components**

○ The new webform 123 has been created. Add new fields to your webform with the form below.
○ This webform is currently unpublished. After finishing your changes to the webform, use the Publish button below.

This page displays all the components currently configured for this webform node. You may add any number of components to the form, even multiple of the same component type. To add a new component, fill in a name and select a type from the fields at the bottom of the table. Submit the form to create the new component or update an existing one. You can change the values of any component by clicking on its name.

Click on any existing component's name to edit its settings.

Name	Type	Value	Mandatory	E-mail	Operations
No Components, add a component below.					
<input type="text"/> New component name	<input type="text" value="textfield"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="button" value="Add"/>	

После сохранения видим 5 кнопок:

View — показать готовый опрос.

Edit - редактировать.

Results — результаты.

Мы находимся в разделе **Edit**, он делится на 2:

Configuration — первая страница, с которой мы попали на эту.

Form components — тут мы находимся. На этой странице происходит добавление полей к опросу.

Добавим несколько полей. Я добавил поля: file, Markup, Pagebreak, textfield. У каждого из этих полей есть свои настройки, их я описывать не буду, так как их много, и они различны.

Configuration

Form components

This page displays all the components currently configured for this webform node. To add a new component, fill in a name and select a type from the fields at the bottom.

Click on any existing component's name to edit its settings.

Name	Type
⊕ file	file
⊕ markup	markup
⊕ pagebreak	pagebreak
⊕ text	textfield
⊕ New component name	textfield ▾

После сохранения эта форма станет доступна на странице webforms. Так же для нее можно создать пункт меню, или вывести ее на главную, поставив соответствующую галочку на странице настроек.

Теперь давайте попробуем создать форму обратной связи.

Создаем новый опрос, в title вбиваем «Обратная связь» и сохраняем.

Я добавил 3 поля к этому webform:

Имя — типа textfield.

E-mail — типа email.

Сообщение — типа textarea.

Name	Type	Value	Mandatory
⊕ Имя	textfield	-	<input checked="" type="checkbox"/>
⊕ E-mail	email	%user@email	<input checked="" type="checkbox"/>
⊕ Сообщение	textarea	-	<input checked="" type="checkbox"/>
⊕ New component name	textfield ▾		<input type="checkbox"/>



Сохраняем, и получаем готовую форму обратной связи. Единственное что было

бы неплохо добавить, так это капчу, для не зарегистрированных пользователей. Для этого устанавливаем captcha, и заходим на страницу «**Administer → User management → CAPTCHA**», там нужно установить галочку **Add CAPTCHA administration links to forms**.

Add CAPTCHA administration links to forms
This option makes it easy to manage CAPTCHA settings on forms. When enabled, you can manage CAPTCHA settings directly from the form administration pages, on all forms, except on administrative pages.

После этого, капча станет доступна на странице опроса.

Обратная связь [View](#) [Edit](#) [Res](#)

Sun, 11/22/2009 - 21:49 — rem

Имя: *

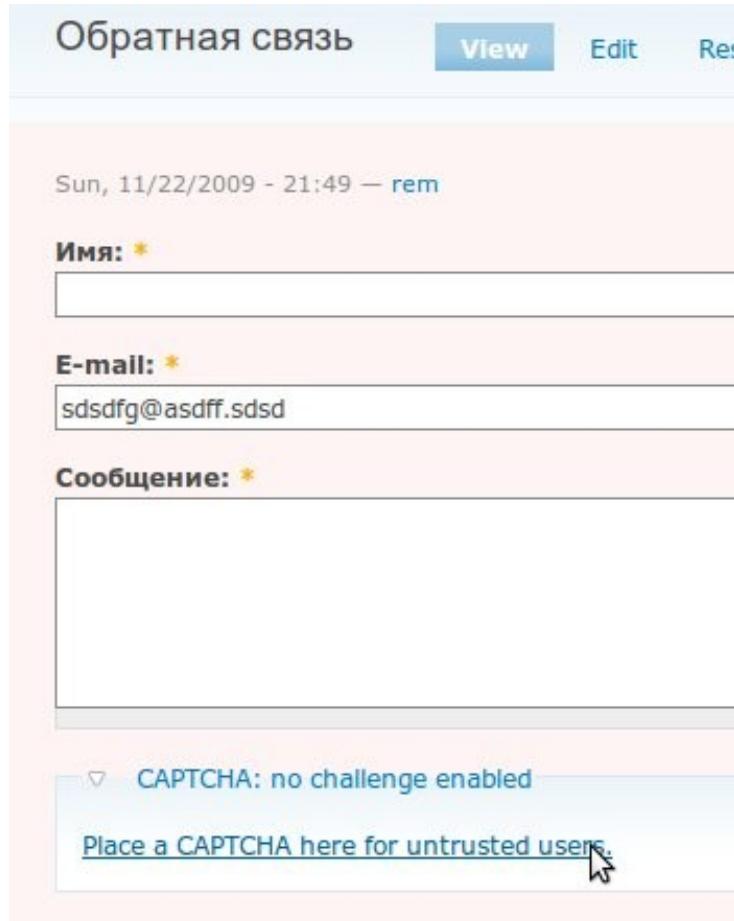
E-mail: *

sdsdfg@asdff.sdsd

Сообщение: *

▼ CAPTCHA: no challenge enabled

[Place a CAPTCHA here for untrusted users](#)

A screenshot of a Drupal contact form titled "Обратная связь". The form has fields for "Имя:" (Name), "E-mail:", and "Сообщение:" (Message). Below the message field, there is a section titled "CAPTCHA: no challenge enabled" with a link "Place a CAPTCHA here for untrusted users". A cursor arrow points to this link.

Кликаем по ссылке «Place a CAPTCHA here for untrusted users»

CAPTCHA point administration

Form ID:

The Drupal form_id of the form to add the CAPTCHA to.

Challenge type:▼

The CAPTCHA type to use for this form



в «Challenge type», нам нужно указать какой тип капчи будет выводиться пользователю. После выбора типа капчи, и сохранения, незарегистрированные пользователи (если разрешить им доступ к webform) увидят

Sun, 11/22/2009 - 21:49 — rem

Имя: ***E-mail: *****Сообщение: *****CAPTCHA**

This question is for testing whether you are a human visitor and to prevent automated spam :

Math question: *

$$4 + 0 = \boxed{}$$

Solve this simple math problem and enter the result. E.g. for 1+3, enter 4.

Спасибо за внимание -)

Если возникли какие-то вопросы, можете задать их на [форуме](#).

Drupal - Урок 15 - Автоматизация. Модули Triggers и Workflow.

By RemaGe - Posted on 20 Декабрь 2009

Автоматизация — это слово беспокоит людей с первой НТР произошедшей в 17 веке, что может быть лучше, чем поставить производство на поток? Отладить работу, и пожинать лавры?

Сегодня мы поговорим о том, что может предложить **Drupal**, для автоматизации некоторых процессов. Для начала давайте установим модуль **Trigger**, он идет с **Drupal**.



после установки на странице «**Administer**» появится 2 новых ссылки: **Triggers** и **Actions**. Если объяснить по простому, тригеры (Triggers), - это какие то системные события, например создание материала, или комментария, а действия (Actions) — это то, как мы можем на них отреагировать. Такая вот, причинно-следственная связь -)

Давайте создадим свое действие.

Заходим на страницу «**Administer** → **Actions**», и видим список действий созданных по умолчанию, а так же форму для создания своих действий.

Actions available to Drupal:

Action type	Description
comment	Unpublish comment
node	Publish post
node	Unpublish post
node	Make post sticky
node	Make post unsticky
node	Promote post to front page
node	Remove post from front page
node	Save post
user	Block current user
user	Ban IP address of current user

Make a new advanced action available

Давайте попробуем создать действие, например, мы хотим, что бы при определенном тригере, пользователя перекидывало на страницу www.yandex.ru (Смилостивись, и дай мне еще немного ТИЦа -)), для этого создаем действие «Redirect to URL».

Make a new advanced action available

В настройках вбиваем название (желательно его изменить, так как таких действий может быть несколько), и адрес:

Configure an advanced action

An advanced action offers additional configuration options which may be filled out below precise action taking place. This description will be displayed in modules such as the trigger descriptive as possible (for example, "Send e-mail to Moderation Team" rather than simply "Send e-mail").

Description:

Редирект к Яндексу

A unique description for this advanced action. This description will be displayed in the interface of modules such as the trigger.

URL: *

<http://yandex.ru>

The URL to which the user should be redirected. This can be an internal URL like node/1234 or an external URL like http://www.yandex.ru.



Действие добавилось в список, в отличие от добавленных по умолчанию, его можно изменить, или удалить:

node	Remove post from front page		
node	Save post		
system	Редирект к Яндексу	configure	delete
user	Block current user		

Теперь давайте закрепим наше действие, за каким-либо событием. Заходим на страницу «**Administer → triggers**»

Triggers

Comments

Content

Cron

Taxonomy

Users

Triggers are system events, such as when new content is added or when a user logs in. Trig unpublishing content or e-mailing an administrator. The [Actions settings page](#) contains a list actions.

Below you can assign actions to run when certain content-related triggers happen. For exam updated.

Trigger: When either saving a new post or updating an existing post

Trigger: After saving a new post

Trigger: After saving an updated post

Trigger: After deleting a post

Trigger: When content is viewed by an authenticated user

События разбиты по типам. Изначально мы попадаем на события связанные с материалами, давайте повесим наше действие, на событие «After saving a new post», то есть, после добавления материала, пользователь будет перекинут на яндекс.

Trigger: After saving a new post

Редирект к Яндексу

Прикрепляем...

Trigger: After saving a new post

Name	Operation
Редирект к Яндексу	unassign

Choose an action

и добавляем новый материал. Если все сделано правильно, вас перекинет на яндекс. Так же можно добавить любое другое действие, например, что бы на каждый коммент, вы получали сообщение. Конечно, доступно немного действий, но есть множество модулей, которые работают с модулем Trigger, и добавляют свои действия в систему. Например, модуль Token, который очень часто встречается в моих уроках (штука крайне полезная).

Other

Enabled	Name	Version	Description
<input type="checkbox"/>	Token	6.x-1.12	Provides a shared API for replacement of tokens with actual data. Required by: TokenSTARTER (disabled), Token actions
<input checked="" type="checkbox"/>	Token actions	6.x-1.12	Provides enhanced versions of core Drupal Token module. Depends on: Token (disabled)
<input type="checkbox"/>	TokenSTARTER	6.x-1.12	Provides additional tokens and a base on which other tokens. Depends on: Token (disabled)

Вместе с ним идет модуль **Token actions**, который добавляет 3 события:
 Послать пользователю сообщение,
 Послать e-mail,
 Редирект на страницу.

Но при этом, можно использовать токены. Например послать сообщение, включающее в себя текст свежедобавленного комментария, автора, время, и кучу другой информации.

Теперь давайте установим модуль [Workflow, и Workflow access..](#)

Он тесно связан с модулем **Trigger**, и позволяет добавить к типам материалов различные статусы, разграничить, какие роли могут производить перевод из статуса в статус, и вызывать при каждом переходе какое то действие.

После установки, появляется сообщение

The content access permissions need to be rebuilt. Please visit [this page](#).

переходим по ссылке, и жмем кнопку «**rebuild permissions**» (Друпал большой — ему видней).

Теперь заходим на страницу «**Administer → Workflow**».

Для того что бы понять как это все работает (с первого раза сам слегка затупил -)), давайте решим какую-либо задачу.

Например, нужно сделать сайт, и дать зарегистрированным пользователям возможность создавать статьи, которым будет присваиваться статус «Не проверено», после создания пользователю выводится сообщение о удачном добавлении, при этом статья не публикуется, и не выводится на главную. Пользователь с правами может дать статье статус «Проверено», или «Лучшее», в первом случае статья публикуется, во втором публикуется и выводится на главную.

Приступим.

Создаем свой **Workflow**

The screenshot shows the Drupal administrative interface under 'Site building'. In the top navigation bar, 'Home > Administer > Site building' is visible. Below this, the 'Workflow' section is displayed. There are two buttons: 'List' (which is greyed out) and 'Add workflow' (which is blue and has a cursor arrow pointing to it). A message below the buttons states: 'No workflows have been added. Would you like to [add a workflow?](#)'.

Пусть он называется «User_article_workflow»

Workflow [List](#) [Add workflow](#)

To get started, provide a name for your workflow. This will be used to identify the workflow in other parts of the site.

Workflow Name:

[Add Workflow](#)



Теперь нужно создать статусы, добавим «Не проверенно», «Проверено» и «Лучшее».

Home > Administer > Site building > Workflow

Add a new state to workflow User_article_workflow

The workflow *User_article_workflow* was created. You should now add states to it.

Enter the name for a state in your workflow. For example, if you were doing a review process, you might have states like "Not reviewed", "Reviewed" and "Approved".

State name: *

Enter the name for a state in your workflow. For example, if you were doing a review process, you might have states like "Not reviewed", "Reviewed" and "Approved".

Weight: 

In listings, the heavier states will sink and the lighter states will be positioned above them.

[Save](#)



Workflow

[List](#)[Add workflow](#)

The workflow state *Лучшее* was created.

Workflow

Operations

[User_article_workflow](#)[Add state](#) [Actions](#) [Edit](#) [Delete](#)

State

Operations

(creation)

Не проверенно

[Edit](#) [Delete](#)

Проверено

[Edit](#) [Delete](#)

Лучшее

[Edit](#) [Delete](#)

Each content type may have a separate workflow. The form for changing workflow state can be displayed when editing content.

Content Type

Workflow

Display Workflow

Page

<None>

 Post
 Comment

Story

<None>

 Post
 Comment[Save workflow mapping](#)

Статус (creation) добавляется по умолчанию.

Сохраняем, и идем в « **Administer → Content management → Content types** », добавляем свой тип материалов

Content types

List

Add content type

Below is a list of all the content types on your site. All posts that exist on yo

Name	Type	Description
Page	page	A page, similar in form to a story, is a simple method for "us" section of a website. By default, a page entry does not appear in the site's initial home page, and provides the
Story	story	A story, similar in form to a page, is ideal for creating announcements, and informal blog-like entries may be featured on the site's initial home page, and provides the

Name: Статья пользователя

Type: user_article

Workflow settings - Default options: все галочки сняты.

Content types

List

Add content type

To create a new content type, enter the human-readable name, the your site will be able to create posts that are instances of this conter

Identification

Name: *

Статья пользователя

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

Type: *

user_article

The machine-readable name of this content type. This text will be used for co numbers, and underscores. Underscores will be converted into hyphens when i

Description:

A brief description of this content type. This text will be displayed as part of t

Submission form settings

Workflow settings

Default options:

- Published
- Promoted to front page
- Sticky at top of lists
- Create new revision

Users with the *administer nodes* permission will be able to override these opt

Comment settings

[Save content type](#)

Сохраним, и возвращаемся на страницу «**Administer → Workflow**». Там уже

появился наш тип, ассоциируем его с Workflow «User_article_workflow».

Статья пользователя

User_article_workflow

Post
 Comment

Save workflow mapping

Осталось настроить права, и действия.

Но для начала добавим пользователя "модератор" (он не обязателен, но так нагляднее). Идем в «**Administer** → **User management** → **Roles**».

Name	Operations
anonymous user	locked edit permissions
authenticated user	locked edit permissions
Модератор	Add role

Возвращаемся на страницу «**Administer** → **Workflow**». Кликаем по ссылке Edit, напротив Workflow «User_article_workflow»,

Workflow	Operations
List	Add workflow
<p>The workflow mapping was saved.</p>	

Workflow	Operations
User_article_workflow	Add state Actions Edit Delete
State	Operations

и видим таблицу, в которой нужно указать, какие роли, могут переводить из одного состояния в другое.

From / To →	Не проверенно	Проверенно	Лучшее
(creation)	<input checked="" type="checkbox"/> author <input type="checkbox"/> anonymous user <input checked="" type="checkbox"/> authenticated user <input type="checkbox"/> Модератор	<input type="checkbox"/> author <input type="checkbox"/> anonymous user <input type="checkbox"/> authenticated user <input type="checkbox"/> Модератор	<input type="checkbox"/> author <input type="checkbox"/> anonymous user <input type="checkbox"/> authenticated user <input type="checkbox"/> Модератор
Не проверенно		<input type="checkbox"/> author <input type="checkbox"/> anonymous user <input type="checkbox"/> authenticated user <input checked="" type="checkbox"/> Модератор	<input type="checkbox"/> author <input type="checkbox"/> anonymous user <input type="checkbox"/> authenticated user <input checked="" type="checkbox"/> Модератор
Проверенно	<input type="checkbox"/> author <input type="checkbox"/> anonymous user <input type="checkbox"/> authenticated user <input checked="" type="checkbox"/> Модератор		<input type="checkbox"/> author <input type="checkbox"/> anonymous user <input type="checkbox"/> authenticated user <input checked="" type="checkbox"/> Модератор
Лучшее	<input type="checkbox"/> author <input type="checkbox"/> anonymous user <input type="checkbox"/> authenticated user <input checked="" type="checkbox"/> Модератор	<input type="checkbox"/> author <input type="checkbox"/> anonymous user <input type="checkbox"/> authenticated user <input checked="" type="checkbox"/> Модератор	

Из состояния «(creation)» в непроверенное может перевести авторизованный пользователь и автор (какое то состояние у статьи должно быть, а так как выбор один, при создании статьи авторизованным пользователем, она автоматом примет статус «Не проверено» у Модератора будет выбор), остальное может модератор.

Workflow tab permissions

Галочка Модератору.

Далее идут таблицы доступа.«Administer → Workflow»

Access control

Не проверенно

Roles who can view posts in this state:	Roles who can edit posts in this state:	Roles who can delete posts in this state:
<input type="checkbox"/> anonymous user	<input type="checkbox"/> anonymous user	<input type="checkbox"/> anonymous user
<input type="checkbox"/> authenticated user	<input type="checkbox"/> authenticated user	<input type="checkbox"/> authenticated user
<input checked="" type="checkbox"/> Модератор	<input checked="" type="checkbox"/> Модератор	<input checked="" type="checkbox"/> Модератор
<input checked="" type="checkbox"/> author	<input checked="" type="checkbox"/> author	<input type="checkbox"/> author

Проверенно

Roles who can view posts in this state:	Roles who can edit posts in this state:	Roles who can delete posts in this state:
<input checked="" type="checkbox"/> anonymous user	<input type="checkbox"/> anonymous user	<input type="checkbox"/> anonymous user
<input checked="" type="checkbox"/> authenticated user	<input type="checkbox"/> authenticated user	<input type="checkbox"/> authenticated user
<input checked="" type="checkbox"/> Модератор	<input checked="" type="checkbox"/> Модератор	<input checked="" type="checkbox"/> Модератор
<input checked="" type="checkbox"/> author	<input type="checkbox"/> author	<input type="checkbox"/> author

Лучшее

Roles who can view posts in this state:	Roles who can edit posts in this state:	Roles who can delete posts in this state:
<input checked="" type="checkbox"/> anonymous user	<input type="checkbox"/> anonymous user	<input type="checkbox"/> anonymous user
<input checked="" type="checkbox"/> authenticated user	<input type="checkbox"/> authenticated user	<input type="checkbox"/> authenticated user
<input checked="" type="checkbox"/> Модератор	<input checked="" type="checkbox"/> Модератор	<input checked="" type="checkbox"/> Модератор
<input checked="" type="checkbox"/> author	<input type="checkbox"/> author	<input type="checkbox"/> author



Save

Состояние «Непроверенно»

Видят: Модератор, Автор

Редактируют: Модератор, Автор

Могут удалить: Модератор

Состояние «Проверено»

Видят: Все

Редактируют: Модератор

Могут удалить: Модератор

Состояние «Лучшее»

Видят: Все

Редактируют: Модератор

Могут удалить: Модератор

Сохраняем. Теперь нужно присвоить действия, но для начала нужно создать действие, которое покажет пользователю сообщение, после создания статьи.

Переходим на «**Administer → Actions**» и добавляем действие «**Display a message to the user**».

Configure an advanced action

An advanced action offers additional configuration options which is recommended, in order to better identify the precise action taken by the trigger module when assigning actions to system events, so it is like "Moderation Team" rather than simply "Send e-mail").

Description:

Сообщение пользователю добавившему статью

A unique description for this advanced action. This description will be displayed in the moderation queue module.

Message: *

Статья сохранена, и ожидает проверки модератором.



The message to be displayed to the current user. You may include the following variables: %teaser, %body. Not all variables will be available in all contexts.

Save

Description: Сообщение пользователю добавившему статью

Message: Статья сохранена, и ожидает проверки модератором.

Возвращаемся на «**Administer** → **workflow**». Жмем ссылку **Actions**

User_article_workflow Add state Actions Edit Delete

И добавляем нужные события (событие «Save post» добавляется автоматом).

Trigger: When *user_article* moves from *User_article_workflow: (creation)* to *User_article_workflow: Не проверено*

Name	Operation
Сообщение пользователю добавившему статью	unassign
Choose an action	<input type="button" value="Assign"/>

Trigger: When *user_article* moves from *User_article_workflow: Не проверено* to *User_article_workflow: Лучшее*

Name	Operation
Promote post to front page	unassign
Publish post	unassign
Save post	unassign
Choose an action	<input type="button" value="Assign"/>

Trigger: When *user_article* moves from *User_article_workflow: Проверено* to *User_article_workflow: Не проверено*

Name	Operation
Unpublish post	unassign
Remove post from front page	unassign
Save post	unassign
Choose an action	<input type="button" value="Assign"/>

Trigger: When *user_article* moves from *User_article_workflow: Проверено* to *User_article_workflow: Лучшее*

Name	Operation
Promote post to front page	unassign
Save post	unassign
Choose an action	<input type="button" value="Assign"/>

Trigger: When <i>user_article</i> moves from <i>User_article_workflow: Лучшее</i> to <i>User_article_workflow: Не проверенно</i>	
Name	Operation
Unpublish post	unassign
Remove post from front page	unassign
Save post	unassign
<input type="button" value="Choose an action"/>	<input type="button" value="Assign"/>

Trigger: When <i>user_article</i> moves from <i>User_article_workflow: Лучшее</i> to <i>User_article_workflow: Проверенно</i>	
Name	Operation
Remove post from front page	unassign
Save post	unassign
<input type="button" value="Choose an action"/>	<input type="button" value="Assign"/>

Trigger: When <i>user_article</i> moves from <i>User_article_workflow: Не проверенно</i> to <i>User_article_workflow: Проверенно</i>	
Name	Operation
Publish post	unassign
Save post	unassign
<input type="button" value="Choose an action"/>	<input type="button" value="Assign"/>

Теперь идем в «**Administer → User management → Permissions**» и разрешаем авторизованным пользователям создавать и редактировать свои статьи типа «*user_article*»

node module		
access content	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
administer content types	<input type="checkbox"/>	<input type="checkbox"/>
administer nodes	<input type="checkbox"/>	<input type="checkbox"/>
create page content	<input type="checkbox"/>	<input type="checkbox"/>
create story content	<input type="checkbox"/>	<input type="checkbox"/>
create user_article content	<input type="checkbox"/>	<input checked="" type="checkbox"/>
delete any page content	<input type="checkbox"/>	<input type="checkbox"/>
delete any story content	<input type="checkbox"/>	<input type="checkbox"/>
delete any user_article content	<input type="checkbox"/>	<input type="checkbox"/>
delete own page content	<input type="checkbox"/>	<input type="checkbox"/>
delete own story content	<input type="checkbox"/>	<input type="checkbox"/>
delete own user_article content	<input type="checkbox"/>	<input type="checkbox"/>
delete revisions	<input type="checkbox"/>	<input type="checkbox"/>
edit any page content	<input type="checkbox"/>	<input type="checkbox"/>
edit any story content	<input type="checkbox"/>	<input type="checkbox"/>
edit any user_article content	<input type="checkbox"/>	<input type="checkbox"/>
edit own page content	<input type="checkbox"/>	<input type="checkbox"/>
edit own story content	<input type="checkbox"/>	<input type="checkbox"/>
edit own user_article content	<input type="checkbox"/>	<input checked="" type="checkbox"/>
revert revisions	<input type="checkbox"/>	<input type="checkbox"/>
view revisions	<input type="checkbox"/>	<input type="checkbox"/>

Все. Теперь можете проверить, создайте пользователя, и создайте им статью, она сохранится, пользователь увидит сообщение, но она не будет опубликована, и не будет выведена на главную.

Спасибо за внимание -)

Drupal - Урок 16 - Пишем свой модуль для Drupal.

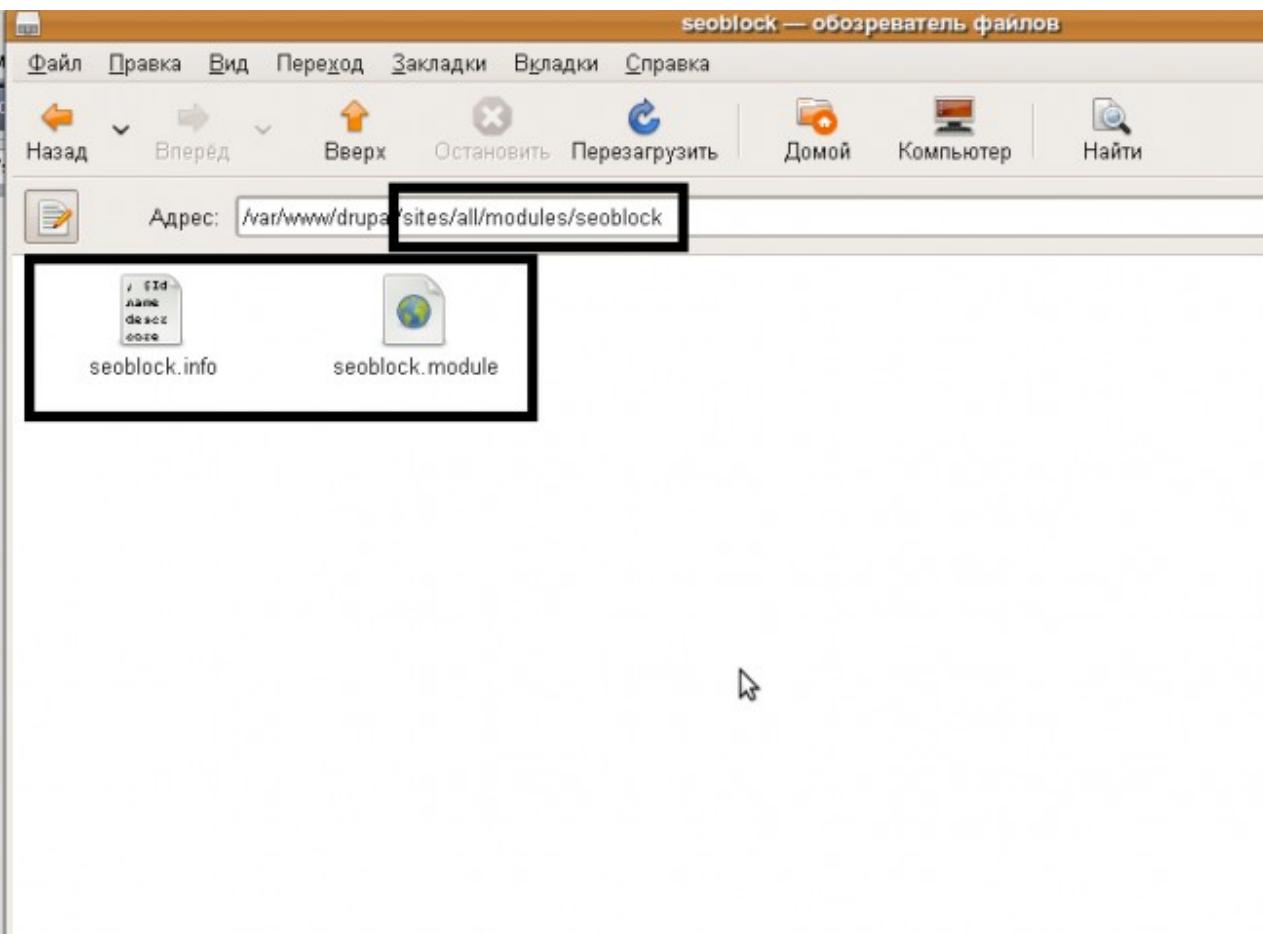
By RemaGe - Posted on 07 Март 2010

Несмотря на огромное количество **модулей для Drupal**, всегда найдутся еще не реализованные идеи. О том как их реализовать в виде модуля для Drupal, мы сегодня и поговорим.

Для того что бы Drupal отобразил ваш модуль на странице modules, необходимо создать папку с названием модуля в **/sites/all/modules** (можно закинуть еще в пару мест, но об этом не сегодня), и создать внутри 2 файла - **имямодуля.info** и **имямодуля.module**. В первом будет информация о модуле, во втором код модуля.

Сегодня мы сделаем модуль-пример, который создаст страницу с настройками, а так же блок, в котором будет выводить Тиц, PR и Alexa rank сайта. Назовем его seoblock.

Для начала создаем папку seoblock в **/sites/all/modules** и там 2 файла **seoblock.info** и **seoblock.module**.



Внутренности файла инфо выглядят так:

```
; $Id$  
name = Seoblock  
description = Display info about site  
core = 6.x
```

Разберем каждую строку поподробнее:

```
; $Id$
```

Эта строка нас не интересует. Если вы будете заливать свой модуль на drupal.org, сюда автоматом вставится информация о модуле.

name (Обязательное)

Имя модуля, которое будет отображаться в списке.

description (Обязательное)

Описание модуля.

core (Обязательное)

Поддерживаемая версия ядра.

dependencies (Не обязательное)

Список модулей, необходимых для работы нашего.

Например:

```
dependencies[] = taxonomy
```

```
dependencies[] = comment
```

package (Не обязательное)

К какому пакету относить модуль. То есть, если вы хотите что бы ваш модуль отображался в списке ССК, то нужно добавить строку:

```
package = "CCK"
```

Если ничего не заданно, то модуль отображается в группе «Other».

Не забываем что файлы должны быть в кодировке utf-8.

На данный момент уже можно установить модуль, но он пока ничего не делает. Перейдем к файлу .module.

По большей части, модуль Drupal состоит из хуков (hook — слово страшное, но на деле все проще чем кажется), специальных функций, каждая из которых предназначена для конкретной задачи. Например, **hook_perm** — определяет права доступа, **hook_help** — создает страницу помощи, **hook_block** — добавляет блок. Хуков много, но нам понадобится всего несколько. Список хуков и информацию о них вы найдете на сайте api.drupal.org, - мегополезный ресурс.

Вернемся к модулю, первым из хуков у нас идет hook_help, который мы уже упоминали чуть выше.

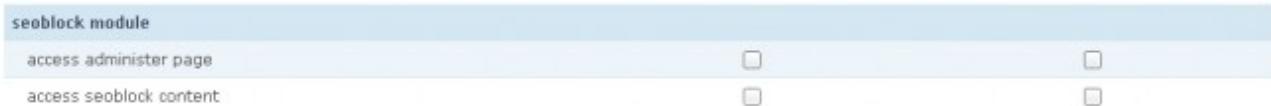
```
function seoblock_help($path, $arg) {  
  $output = "  
  switch ($path)  
  {  
    case "admin/help#seoblock":  
      $output = '<p>'. t("Displays information about site") .'</p>';  
      break;  
    }  
  return $output;  
} // function seoblock_help
```

Слово hook в названии модуля заменено на название модуля, это стандартно для друпал, как в программировании, так и в темизировании (только в темизировании заменяется слово theme, и заменить его можно не только на название темы, но и на название движка темы).

Вторым хуком идет **hook_perm**.

```
function seoblock_perm() {  
  return array('access seoblock content', 'access administration pages');  
} // function seoblock_perm()
```

Он задает разрешения (Permissions) для модуля. Если вы хотите добавить несколько прав, то их можно записать через запятую:



Позже в других хуках можно будет проверять, установлено ли какое-либо разрешение для конкретного пользователя, или нет (не забываем что у админа по умолчанию все права, поэтому, следует проверять работу модуля и под другими пользователями, на всякий -)).

Теперь давайте создадим страницу настроек, на которой можно будет ввести целевой сайт, и указать какие характеристики мы будем выводить. Хук **hook_menu()** позволяет создавать пункты меню, и в зависимости от настроек, выводить их или в меню, или на странице «Управление» (administer). Мы используем его для создания нашей странички.

```
function seoblock_menu()  
{  
  $items = array();  
  $items['admin/settings/seoblock'] = array(  
    'title' => 'seoblock module settings', //Заголовок страницы  
    'description' => 'Description of your seoblock settings page', //Описание  
    'page callback' => 'drupal_get_form', //Функция отображающая страницу, мы  
    создаем  
    //форму, поэтому используем системную функцию  
    'page arguments' => array('seoblock_admin'), //Содержимое формы  
    'access arguments' => array('access administration pages'), //Права доступа,  
    если дать  
    //анонимусам право 'access administration pages', они смогут админить наш  
    модуль.  
    'type' => MENU_NORMAL_ITEM, //Нормальный пункт меню  
  );  
  return $items;  
}
```

Более подробное описание можете посмотреть на api.drupal.ru.

Теперь нужно заполнить страницу, этим занимается функция seoblock_admin, которую мы вызвали в seoblock_menu:

```
'page arguments' => array('seoblock_admin'),
```

Сама функция:

```
function seoblock_admin()
{
    $form = array();
    $form['sitename'] = array(
        '#type' => 'textfield',
        '#title' => t('Sitename'),
        '#default_value' => variable_get('sitename', ''),
        '#size' => 20,
        '#maxlength' => 20,
        '#description' => t('without "http://"'),
    );
    $form['alexa'] = array(
        '#type' => 'checkbox',
        '#title' => t('Display Alexa rank'),
        '#default_value' => variable_get('alexa', false),
    );
    $form['tic'] = array(
        '#type' => 'checkbox',
        '#title' => t('Display Yandex.Tic'),
        '#default_value' => variable_get('tic', false),
    );
    $form['pr'] = array(
        '#type' => 'checkbox',
        '#title' => t('Display Google.PR'),
        '#default_value' => variable_get('pr', false),
    );
    return system_settings_form($form);
}
```

Это всего лишь массив, в котором мы передаем параметры элементов формы. Элементов у нас 3: 'sitename' — название сайта, 'alexa' — alexa rank сайта, 'tic' —

тиц сайта и 'pr' — PR сайта.

'#type' - тип элемента.

'#title' - заголовок.

'#default_value' — значение по умолчанию, в данном случае мы подгружаем (если оно есть) сохраненное значение.

Более подробно, о работе с формами можно прочитать [тут](#).

Наконец выведем сам блок:

```
function seoblock_block($op='list', $delta=0)
{
    if ($op == "list")
    {
        $block = array();
        $block[0]["info"] = t('seoblock');
        return $block;
    }
    else if ($op == 'view')
    {
        switch ($delta)
        {
            case 0:
                $block_content = "";
                $block['subject'] = 'seoblock';
                $tic= variable_get('tic', false); //получаем настройки нашего
модуля
                $alexa= variable_get('alexa', false);
                $pr= variable_get('pr', false);
                if ($tic or $alexa or $pr) //Функции честно скомуниздили в
интернетах -)
                {
                    if ($tic) $block['content'] .=
'Яндекс.Тиц:'.$cy(variable_get('sitename', false)).'<br>';
                    if ($alexa) $block['content'] .= 'Alexa rank:
.alexa(variable_get('sitename', false)).'<br>';
                    if ($pr) $block['content'] .= 'Google.PR:
.getPageRank(variable_get('sitename', false));
'
                }
                break;
        }
    }
    return $block;
}
```

```
    }  
} // end seoblock_block
```

Тут нам интересен параметр \$op, от него зависит, информацию какого рода возвращает функция.

\$op == "list" — выводит список блоков.

\$op == 'view' - выводит содержимое блока. Содержимое какого блока выводить, определяется по переменной \$delta.

Если объединить все вместе, и установить, получим

Настройки

rem

- My account
- ▷ Create content
- ▽ Administer
 - ▷ Content management
 - ▷ Site building
 - ▽ Site configuration
 - Actions
 - Administration theme
 - Clean URLs
 - Date and time
 - Error reporting
 - File system
 - Image toolkit
 - Input formats
 - ▷ Logging and alerts
 - Performance
 - Site information
 - Site maintenance
 - seoblock module settings

seoblock module settings

Sitename:

remage.su

without "http://"

Display Alexa rank

Display Yandex.Tic

Display Google.PR

[Save configuration](#)

[Reset to defaults](#)

Блок

seoblock

Яндекс.Тиц: 20
Alexa rank: 231114
Google.PR: 3

В следующий раз мы продолжим работу над этим модулем. Сделаем кеширование блока, и привяжем его к крону. Отделим дизайн и выкинем его в отдельный tpl.php файл, и может быть создадим отдельную страницу дублирующую наш блок. Оставайтесь на связи -)

Приложение

Drupal 6 - переменные файла page.tpl.php

Переменные файла page.tpl.php

Основные переменные:

\$base_path: URL-путь к папке с Drupal. Покрайней мере, он будет равен */*.

\$css: массив CSS файлов для данной страницы.

\$directory: Путь к папке с текущей темой, например themes/garland или themes/garland/minelli.

\$is_front: Истина, если это главная страница.

\$logged_in: Истина, если пользователь авторизован.

\$is_admin: Истина, если пользователь авторизован под учетной записью администратора.

Метаданные страницы:

\$language: (объект) Язык сайта. \$language->language содержит текстовое обозначение. \$language->dir Содержит директиву языка, принимает значение 'ltr' или 'rtl' .

\$head_title: модифицированная версия заголовка страницы, используется в теге TITLE.

\$head: выводит секцию HEAD (включает в себя мета-теги, keyword tags).

\$styles: CSS стили страницы.

\$scripts: Файлы javascript, используемые на странице.

\$body_classes: множество CSS классов для тега BODY. Содержит переменные определяющие текущую разметку (множественные столбцы, единичный столбец), текущий путь и тд.

Идентификационные данные:

\$front_page: Адрес главной страницы. Так же включает языковой домен, или префикс.

\$logo: Адрес логотипа.

\$site_name: Название сайта, можно отключить в настройках темы.

\$site_slogan: Слоган сайта, можно отключить в настройках темы.

\$mission: Миссия сайта, можно отключить в настройках темы.

Навигация:

\$search_box: HTML код формы поиска, пустой если поиск выключен.

\$primary_links (array): Массив содержащий primary_links.

\$secondary_links (array): Массив содержащий secondary_links.

Содержание страницы (в порядке появления в default page.tpl.php):

\$left: Левая колонка.

\$breadcrumb: «хлебные крошки» для текущей страницы.

\$title: Заголовок страницы.

\$help: Текст справки, в основном используется на страницах администрирования.

\$messages: Статусные сообщения и сообщения об ошибках.

\$tabs: закладки.

\$content: основное содержимое текущей страницы.

\$right: правая колонка.

Переменные подвала:

\$feed_icons: RSS лента текущей страницы.

\$footer_message: Текст подвала сайта.

\$footer : регион footer.

\$closure: Переменная определяющая конец страницы. Она должна быть после остального динамического содержимого.

Drupal 6 - переменные файла node.tpl.php

Переменные файла node.tpl.php

доступные переменные:

\$title: Заголовок.

\$content: тело ноды, или тизер, в зависимости от переменной \$teaser.

\$picture: авторская картинка, выводится из theme_user_picture().

\$date: Форматированная дата создания (используйте \$created чтобы задать свое форматирование с помощью format_date()).

\$links: Ссылки наподобие "Читать далее", "Добавить коментарий", и т. д.

Выводятся из theme_links().

\$name: Имя автора, выводится из theme_username().

\$node_url: Прямая ссылка на текущую ноду.

\$terms: список терминов таксономии, выводится из theme_links().

\$submitted: информация по публикации, выводится из theme_node_submitted().

Другие переменные:

\$node: Объект включающий в себя всю ноду. Содержит небезопасные данные.

\$type: Тип ноды: story, page, blog, и т. д.

\$comment_count: Количество коментариев.

\$uid: идентификационный номер (ID) автора ноды.

\$created: Время публикации ноды, форматированно в Unix timestamp.

\$zebra: выводит "even" или "odd". Может пригодиться для вывода тизеров в виде «зебры» (с чередующимися цветами).

\$id: Позиция ноды. Увеличивается каждый раз когда выводится нода.

Статусные переменные:

\$teaser: Истина, если выводится тизер.

\$page: Истина, если выводится полная страница.

\$promote: Истина, если нода выведена на главную.

\$sticky: Истина, если нода прикреплена.

\$status: Истина, если нода опубликованна.

\$comment: Переменная содержит настройки вывода коментариев у ноды.

\$readmore: Истина, если тизер не содержит в себе всю ноду.

\$is_front: Истина, если это главная страница.

\$logged_in: возвращает истину когда текущий пользователь авторизирован.

\$is_admin: возвращает истину когда текущий пользователь администратор.

Drupal 6 - переменные файла block.tpl.php

Переменные файла block.tpl.php

доступные переменные:

\$block->subject: Заголовок блока.

\$block->content: Содержимое блока.

\$block->module: Модуль который создает блок

\$block->delta: Числовой идентификатор прикрепленый к каждому модулю.

\$block->region: Регион, в котором находится текущий блок.

вспомогательные переменные:

\$block_zebra: Выводит 'odd' и 'even' поочереди, в зависимости от блока в регионе.

\$zebra: выводит то же самое что и \$block_zebra но независимо от региона.

\$block_id: Счетчик зависящий от регионов.

\$id: выводит то же что и \$block_id но не зависит от региона.

\$is_front: возвращает истину, когда вызывается на главной странице.

\$logged_in: возвращает истину когда текущий пользователь авторизирован.

\$is_admin: возвращает истину когда текущий пользователь администратор.

Drupal 6 - переменные файла box.tpl.php

Переменные файла box.tpl.php

\$title: Заголовок.

\$content: содержимое.

Drupal 6 - переменные файла comment.tpl.php

Переменные файла comment.tpl.php

Доступные переменные:

\$author: Автор комментария может быть ссылкой, или обычным текстом.

\$content: Тело комментария.

\$date: Дата и время публикации.

\$links: Ссылки.

\$new: Пометка нового комментария.

\$picture: Аватар автора.

\$signature: Подпись автора

\$status: Статус комментария. Возможные значения: comment-unpublished, comment-published или comment-preview.

\$title: Заголовок-ссылка.

\$comment: Полный объект комментария.

\$node: Объект ноды к который прикреплен комментарий.