Создание сайта на Drupal 6

Практический вводный курс

С. Шапошникова (plustilino)

Лаборатория юного линуксоида
http://younglinux.info

ноябрь, 2011

Пояснительная записка

Учебное пособие "Создание сайта на Drupal 6. Практический вводный курс" включает:

- Описание установки движка (CMS – системы управления контентом) Drupal на локальном сервере. В качестве программной среды для локального сервера используется Denwer и Windows; возможно в будущем курс будет включать описание установки Drupal в Linux.
- Создание сайта-блога на основе Drupal.
- Основы администрирования сайта.
- Установка темы оформления и ее правка.
- Знакомство с понятием таксономии в web.
- Работа с модулями Views, CCK, Image, Geshifilter и др.

При создании учебного сайта, описанного в курсе, в качестве его контента (наполнения) использовались статьи и изображения с сайта http://younglinux.info и http://inf1.info. Подготовленный материал для работы с данным пособием содержится в архиве. Однако вы можете использовать собственный контент при создании учебного сайта с помощью этого пособия.

Курс предполагает, что обучаемый знаком с основными тегами HTML и как минимум знает о назначении языка CSS.

Желательно, чтобы в операционной системе было установлено два браузера.

Цикл уроков “Создание сайта на Drupal 6. Практический вводный курс” первоначально публиковался на сайте http://younglinux.info в период с сентября по ноябрь 2011 года.

Содержание

Урок 1. Установка Drupal, первое знакомство с системой управления контентом

Установите Denwer, скачав установочный пакет с официального сайта программы. На системном диске компьютера (там, где установлена Windows) должен появиться каталог WebServers. В нем есть каталог home. Зайдите в него и создайте папку; присвойте ей доменное имя вашего будущего сайта. Доменное имя потом можно менять, т.е. менять название папки. Содержимое сайта от этого не меняется и не теряется. Поскольку мы создаем сайт на локальном компьютере, а не в Интернет, то пусть его доменное имя будет, например, таким: developer.home. Итак, в папке home должен быть создан каталог developer.home. Зайдите в него и создайте там каталог www. На разных web-серверах используется свой способ организации файловой структуры. Так, например, в Denwer требуется создавать каталог www, и только там размещать файлы сайта.

Скачайте архив Drupal 6-й версии с сайта drupal.org или drupal.ru. Распакуйте его, все файлы из распакованного архива переместите в каталог www.

Если во время установки Denwer на рабочем столе были созданы ярлыки, то запустите сервер с помощью ярлыка *Start Denwer*. Если ярлыков нет, то зайдите в WebServers\denwer и запустите файл *Run.exe*.

Теперь в браузере перейдите по адресу localhost/Tools/phpMyAdmin/. Localhost – это сетевое имя вашего локального хоста (компьютера), а phpMyAdmin — это инструмент (программа) для управления базами данных, описанными на языке программирования MySQL. Системы управления контентом, в том числе и Drupal, обычно работают с данными, сохраненными в базах данных. Вообще база данных для сайта имеет ключевое значение, ее потеря или порча может "убить" сайт насовсем. Файлы же самого движка Drupal и его модулей всегда можно загрузить снова. Но есть одно но. В процессе жизни сайта на него загружаются картинки, архивы, иные файлы. Нужно знать, в каком(их) каталоге(ах) они сохраняются, и также, как и для базы данных, регулярно делать их резервные копии.

С помощью интерфейса phpMyAdmin создадим базу данных, назовем ее *developer*. Для этого в поле *Create new database* следует вписать имя и нажать кнопку *Create*. Запомните, как вы назвали базу данных; эти данные понадобятся при установке Drupal. Также заметьте, что база данных пуста — в ней нет таблиц.

Перейдите в браузере по адресу developer.home. Запустится установщик Drupal, он будет на английском языке. Будет предложено либо продолжить установку на английском, либо узнать, как инсталлировать Drupal на иных языках. Выберем пункт *Install Drupal in English*, а русифицируем систему позже, уже после установки. Далее должно появиться сообщение об ошибке, которая должна быть устранена, чтобы инсталляционный процесс мог продолжиться. Ошибка эта заключается в том, что должен быть файл с настройками *settings.php*, и во время установки у него должно быть установлено разрешение на возможность записи. После установки Drupal должно быть оставлено только разрешение на чтение для владельца. На локальном сервере обычно можно не возиться с разрешениями на

файлы, но при размещении сайта в Интернет будьте внимательны. Заходим в каталог sites\default движка Drupal (в каталоге www), делаем копию файла *default.settings.php* и называем ее *settings.php*. После чего в браузере нажимаем F5.

Далее происходит переход к настройкам базы данных. Наша база называется *developer*. На локальном сервере обычно пользователь базы данных — root, пароль отсутствует. При размещении сайта на хостинге вам обычно самим надо будет создавать пользователя и связывать его с определенной базой данных. Но не здесь, так что пока вписываем в поле *Database name* слово *developer*, а в поле *Database username - root*; продолжаем установку.

После установки сайта предлагается выполнить первоначальную его настройку. Все, что здесь вписывается, в последствии может быть исправлено. Пусть сайт называется "Путь программиста". Далее вписываются данные администратора сайта. Администратор — это самый важный человек для сайта. Причем не надо путать пользователя базы данных и администратора сайта — это разные вещи! Вам надо придумать имя и пароль для администратора и хорошо их запомнить. Забудете — ваша проблема. Да, имя пользователя и пароль всегда можно будет исправить, но только если вы знаете старые имя и пароль.

После первоначальной настройки сайта появляется сообщение, что установка Drupal завершена и можно посетить ваш новый сайт. Сделайте это.

На главной странице сайта, помимо небольшого меню слева, вы увидите приветствие и четыре пункта того, что желательно сделать. Предлагается сделать примерно следующее: 1 — настроить сайт, 2 — подключить дополнительные возможности, 3 — выбрать оформление, 4 — приступить к написанию содержания сайта. Но не будем спешить.

Давайте сначала посмотрим, что случилось с базой данных. Зайдите в phpMyAdmin, убедитесь, что у базы данных *developer* появилось около 50 таблиц. Именно в них будет храниться содержимое сайта, а сам движок лишь управляет им. В последствии число таблиц может увеличиваться.

Вернемся к сайту. В меню присутствуют четыре пункта: *My account* (аккаунт), *Create content* (создать материал, т.е. контент, т.е. содержимое сайта), *Administer* (администрировать, управлять), *Log out* (выйти). Нажмите *Log out*, посмотрите, как изменится меню, войдите в систему обратно, введя имя пользователя и пароль.

Нажмите *Administer*, произойдет переход на страницу, где открывается доступ ко всем настройкам. В разделе *Site configuration* выберите пункт *Site information*. Давайте придумаем какой-нибудь слоган для нашего сайта. Например, "Разработка и web-дизайн". Сохраните конфигурацию. Ничего не изменилось, но это пока.

Теперь попробуем изменить оформление сайта. Нажимаем *Administer* и в разделе *Site building* выбираем *Themes*. Попробуйте уже установленные там темы, в конце снова вернитесь к теме Garland.

Обратим внимание на комментарии под названиями тем. Там перечисляются ключевые особенности построения тем оформления, то есть того, как эти темы созданы и какими

свойствами обладают. Не вдаваясь в подробности, можно сказать, что тема представляет собой описание расположения, типов и свойств элементов страницы на языках HTML и CSS. Выбор темы — это не только выбор оформления сайта, но и в определенной степени выбор его поведения. Кроме того, оттого как организована тема, сайт может лучше или хуже индексироваться поисковыми системами (Yandex, Google и др.). Например, считается, что лучше выбирать бестабличные темы. Так темы Bluemarine и Chameleon основаны на таблицах, т.е. разные области страницы представляют собой ячейки таблицы. Темы Garland и Minnelli бестабличные. Но первая имеет плавающую ширину (fluid), а вторая – фиксированную (fixed). При fluid-темах ширина страниц сайт изменяется в зависимости от рабочей области окна браузера. Чем шире монитор, окно, тем больше сайт растягивается в ширину. Фиксированные по ширине сайты остаются одного размера не зависимо от рабочей области. Поскольку мы не знаем ширину монитора пользователя (а он может быть большим, и читать содержимое сайта будет не удобно), возможно лучше выбирать fixed-темы.

Измените размер окна браузера и отметьте, как меняется ширина сайта. Строки контента (не меню) становятся либо длиннее, либо короче. Включите тему Minnelli и также понаблюдайте за поведением сайта (длина строки теперь не меняется). Остановим свой выбор на Minnelli.

Возможно, нас не устраивает ширина сайта. Мы хотим, чтобы он был фиксированного размера, но немного шире, чем есть. Как быть? Бывает, что настройка ширины есть в интерфейсе настройки темы в системе Drupal, а бывает, что ее там нет, и приходится править файлы *.*css* темы вручную.

Исследуем настройки темы Minnelli. Для этого перейдите по ссылке *configure*, которая находится в конце строки с названием темы. У каждой темы могут быть свои специфические настройки. В данном случае, можно выбрать цветовую схему (из готовых шаблонов или настроить по своему усмотрению), включить определенные элементы страницы, загрузить логотип и иконку сайта, но изменить ширину сайта через интерфейс возможности нет. Поэтому оставим эту задачу на потом. Пока выберите понравившуюся вам цветовую схему или настройте свою; сохраните конфигурацию.

Обратите внимание, что флажок *Site slogan* в настройках темы снят. Именно поэтому, когда мы написали слоган и сохранили его, он не отобразился. Включите этот флажок и сохраните настройки. Теперь в шапке сайта помимо его названия должен красоваться и слоган. Очень часто бывает, что тема не совсем корректно его отображает. Например, в одной строке с названием, как в данном случае. Получается не красиво, если слоган начинается с большой буквы. Давайте немного подправим слоган, пусть он будет выглядеть так: "- разработка и web-дизайн" (начинается с тире). Чтобы исправить слоган, перейдите к настройкам *Site information*.

А нужен ли вообще слоган? На этот вопрос нельзя ответить однозначно. Бывает, что нет, если название сайта, его стиль или узнаваемый брэнд уже в достаточной степени сообщают о его содержании. Слоган как бы дает краткую характеристику сайта, описывает его цели. Желательно, чтобы он содержал ключевые (важные) для сайта слова. Так, если сайт о программировании, то поисковые системы лучше отреагируют на слова типа

"программирование", "разработка", "код" и будут озадачены, если в названии сайта или его слогане будут красоваться не связанные с тематикой слова, например, "рецепты", "планета" и т.п. Поэтому если вы хотите назвать сайт как-то оригинально, будьте осторожны: поисковые роботы еще не в силах понять красоту фраз и их переносный смысл. Хотя жертвовать действительно оригинальным названием ради этого, не стоит.

Урок 2. Перевод интерфейса, обновление и блоки сайта

Перевод интерфейса

Не очень удобно работать с системой, интерфейс которой на иностранном языке, если не очень хорошо знаешь язык и саму систему. Поэтому начинающим пользователям Drupal желательно установить перевод интерфейса на русский язык, так сказать, русифицировать систему. Перевод можно загрузить с сайта drupaler.ru. В верхней части этого сайта нажимаем кнопку *Download*, выбираем *Russian*. В поле *Проект* набираем *Drupal* и нажимаем *Выбор проекта*. Далее следует выбрать релиз (версию) пакета перевода, он должен соответствовать версии системы Drupal, которую вы установили. В разделе *Формат* переключите радиокнопку на вариант *Все в одном файле*. Установить перевод интерфейса можно по-разному, но выберем самый простой. После нажатия кнопки *Экспорт* будет предложено сохранить файл (с расширением **.po*). Сохраните его пока в любом месте на вашем компьютере.

Теперь запустим Denwer и перейдем на страницу администрирования сайта. Если там поискать что-нибудь связанное с языками (что-то типа language) и переводом интерфейса (translate), то мы ничего не найдем. Следовательно, требуется включить дополнительную функциональность для сайта. Дополнительная функциональность обеспечивается модулями. Сам Drupal представляет собой в основном ядро, а разные модули как бы "прикручиваются" к нему, как какие-нибудь фичи, придающие сайту те или иные свойства и особенности. Модулей существует огромное количество, лишь малая часть из них входит в поставку Drupal, остальные можно загрузить с сайта drupal.org. Каждый конкретный сайт использует ограниченное число модулей (что разумно), в зависимости от своих задач; но поскольку модулей много, то сайты, сделанные на Drupal, могут очень сильно отличаться между собой по своим возможностям. В этом заключается гибкость и настраиваемость CMS Drupal.

Итак, пока нам требуется включить возможность переводить интерфейс. Модуль, который это обеспечивает, уже входит в пакет Drupal. Поэтому сразу заходим в Modules и ищем там модуль Local, включаем его и сохраняем конфигурацию. Теперь на главной странице управления сайтом в разделе *Site building* появился пункт *Translate interface*, а в разделе *Site configuration* — пункт *Languages*.

Заходим в *Translate interface* и выбираем *Import*. В открывшейся форме указываем файл с переводом, загруженный ранее с сайта drupaler.ru, а также указываем язык перевода (*Русский*) и нажимаем кнопку *Import*. После чего в *Site configuration* → *Languages*

включаем радиокнопку *Default* напротив русского языка. После сохранения конфигурации интерфейс станет русским.

Cron, update.php

Система управления содержимым Drupal регулярно требует проверки на наличие обновлений для самой системы, ее модулей и тем. С этой целью надо с определенной периодичностью запускать так называемый Cron. Если он давно не запускался, то вы увидите сообщение на странице администрирования. Запустить Cron можно со страницы *Модули*, либо *Отчет о состоянии*. Свою работу Cron выполняет самостоятельно, ничего дополнительного вам делать не требуется. Зайдите в *Отчет о состоянии* и кликните по ссылке *запустить выполнение регулярных процедур*. Обычно этого достаточно.

Однако бывают ситуации, когда выходит новая версия модуля, самой системы Drupal или темы оформления. Тогда требуется удалить старую версию, загрузить новую и разместить ее в соответствующем каталоге (на хостинге или локальном компьютере), где установлен Drupal. Если обновляется сам Drupal, то надо быть аккуратными при удалении; не удалить ничего лишнего, что не входит в пакет Drupal (сторонние модули, темы, загруженные изображения и др.). Поэтому настоятельно рекомендуется всё, что загружает пользователь (он же администратор сайта), хранить в отдельном каталоге CMS Drupal – sites. Там же (в sites/default) хранится такой важный файл, как *settings.php*, с которым мы уже сталкивались при установке системы. В нем прописаны имя базы данных, имя пользователя базы данных и его пароль. Поэтому, размещая сайт в Интернете, устанавливайте для этого файла лишь право на чтение для владельца.

После включения новых и обновления старых компонентов сайта рекомендуется обновлять базу данных. Делается это с помощью скрипта *update.php*, который расположен в корневой директории CMS Drupal. Найдите его в каталоге WebServers\home\developer.home\www. Чтобы его запустить, надо обратиться к нему через адресную строку браузера, например, так: developer.home/update.php. Можно прописать адрес вручную или перейти к этому файлу через страницу *Модули* (developer.home/admin/build/modules), щелкнув там по ссылке *update.php* (всегда читайте пояснения на страницах администрирования Drupal, система сама по себе хорошо документирована).

Также желательно запускать обновление базы данных перед ее экспортом. Это связано с тем, что после обновления размер базы уменьшается за счет удаления из нее временной информации.

Обновлять базу данных имеет право только администратор сайта. Чтобы в этом убедиться, выйдите из своего аккаунта (пункт *Выйти* в меню) и попытайтесь запустить скрипт по адресу developer.home/update.php. Drupal вам сообщит, что доступ запрещен (access denied). Вернитесь в свой аккаунт и снова попробуйте запустить этот скрипт. Должно все получиться (на открывающихся страницах следует нажать *Continue*, затем *Update*).

Блоки

Как вы должно быть уже знаете, страницы сайта состоят из ряда областей, выполняющих свое определенное назначение. Так почти всегда у каждого сайта есть так называемая "шапка" - область вверху, где отображается его название и логотип; есть меню (только справа или только слева или по обеим сторонам); есть "подвал" - область внизу сайта; и есть самая большая центральная область, где располагают контент (текст) страниц. Бывает более сложная структура страниц сайта. Обычно шаблон страниц, также как и тема, применяется для всего сайта. И это правильно: было бы странно и неудобно ориентироваться на сайте, если бы на одной странице меню располагалось слева, а на другой вдруг переезжало вправо.

Чтобы увидеть какие структурные части (области) сайта есть, какие элементы (блоки) в них находятся, какие блоки можно еще добавить, надо перейти на страницу *Блоки* (developer.home/admin/build/block).

Количество областей сайта, а также их названия определяются включенной темой сайта. Однако обычно всегда есть *Содержимое*, *Левая колонка*, *Заголовок* и *Подвал* (хотя они могут называться не обязательно именно так). Перемещать блоки между областями можно с помощью мыши (путем перетаскивания) или с помощью выпадающего списка. *Блоки* — это элементы сайта (меню, текстовые части, кнопки и др.). Блоки можно создавать и настраивать; некоторые модули автоматически создают блоки. Новые блоки сначала отключены.

Посмотрим, как обстоят дела у нас. В левую колонку помещены три блока: *Вход в систему*, *Навигация* и *Создано на Drupal*. Блок *Вход в систему* мы видели, когда покидали свой аккаунт. *Навигация* — это меню, настроенное изначально так, что его видит только администратор сайта; а *Создано на Drupal* представляет собой кнопку со ссылкой на сайт drupal.org. Также существует ряд отключенных блоков, которые нам пока не нужны, а возможно будут не нужны никогда.

Давайте переместим кнопку *Создано на Drupal* с левой колонки на правую, чтобы посмотреть, как изменится внешний вид сайта. Сохранив изменения, перейдите на главную страницу. Появилась правая колонка, но интересно отметить, что размер области, где располагается содержимое страницы, не изменился. Сайт стал шире, хотя на нем fixed-тема; просто fixed-размер стал другим, или, можно сказать, что он установлен только для области содержимого. Однако чаще бывает, что использование дополнительной области сужает область содержимого. Особенно критично это может быть как раз для fixed-тем.

Урок 3. Добавление материалов, их типы. Фильтры HTML

Отметим на будущее, что будем создавать web-сайт типа "блог". Блог представляет собой регулярно обновляемый сайт статей, отсортированных в хронологическом порядке, и предполагающий комментарии читателей.

Первая страница, которую мы создадим, будет называться "О сайте". Обычно все приличные

сайты имеют такую страницу; на ней может содержаться информация о том, какой теме посвящен данный сайт, его структура, принципы, цели, предположительная аудитория, авторские права, контакты и др.

Для того, чтобы добавить контент в Drupal, нажмите в меню *Создать публикацию* (или *Создать материал*). Материалы сайта бывают разным по своему назначению и содержанию. Это может быть запись, основное назначение которой информировать посетителей сайта, ставить их перед фактом; или это может быть запись, предполагающая коллективное обсуждение (запись блога, например); или материал, главной частью которого является изображение или видеозапись; также бывает материал, представляющий собой коллекцию ссылок на другие материалы, объединенные по тому или иному признаку. От типа материала зависит, какие части могут быть у содержимого страницы. Поэтому форма создания каждого типа материала может иметь свои нюансы и особенности, но не обязательно. По умолчанию в системе Drupal включены только два типа материала: *Page* (страница) и *Story* (заметка, рассказ). По своей структуре эти материалы ничем не отличаются, но имеют разное смысловое назначение. *Page* предназначен для более "статичного" материала: того, который реже меняется. Такой тип как раза больше всего подходит для создания страницы "О сайте".
После выбора *Page* перед вами откроется форма для заполнения. В поле *Title* вписывается название статьи, в поле *Body* – ее текст. Сочините и напишите два-три абзаца о вашем сайте. Теперь развернем раздел *Параметры меню*. Название ссылки в меню может быть таким же, как ее заголовок, хотя может и отличаться (если заголовок статьи очень длинный, то ссылки в меню стараются сделать немного короче). Сложнее обстоит дело с выбором места расположения ссылки. На сайте может быть несколько меню. Есть те, которые уже созданы самой системой Drupal (*Primary links, Secondary links*), однако большинство меню вы будете создавать сами. Любое меню представляет собой блок, и как любой блок может быть отключено или расположено в любой области сайта. По умолчанию меню Primary links включено, его расположение зависит от темы (обычно где-нибудь вверху), а мы его не видим лишь потому, что на данный момент в нем нет ни одной ссылки. Оставим в списке *Родительский пункт* вариант *Primary links* и сохраним материал.

После этого вы увидите, как выглядит готовая страница. Обратите внимание на адресную строку браузера, там будет значиться developer.home/node/1. Это адрес данной страницы. В системе Drupal принято статьи называть нодами (node). Не трудно догадаться, что следующая статься будет иметь адрес developer.home/node/2. Перейдите на главную страницу сайта (заметьте, что она осталась прежней) и затем щелкните по ссылке "О сайте", которая должна была появиться в верхней его части.

При написании статьи, скорее всего, вы создавали новые абзацы одним нажатием Enter. Так происходит во всех текстовых процессорах, но не в системе Drupal. Здесь один Enter означает разрыв строки (
), а два Enter – формирование абзаца ( ). Если вы посмотрите исходный код страницы, на которой расположена нода "О сайте", и найдете там свой текст, то увидите, что он весь обрамлен единственным контейнером … , внутри которого встречаются один-два тега
. Отредактируйте статью, нажав на кнопку *Редактировать*

(Изменить) около ее названия: вставьте между абзацами пустые строки. Теперь снова посмотрите исходный код страницы, отметьте разницу.

Добавленная нами статья не появилась на главной странице сайта. Это связано с настройками типа материала *Page*: по умолчанию статьи туда не выводится. Но это вовсе не значит, что мы не может поместить на главную страницу какую-нибудь избранную статью. На странице редактирования материала раскройте раздел *Параметры публикации* (он находится в конце). Установите флажок *Помещать на главную* и сохраните материал. Теперь перейдите на главную страницу сайта; она изменилась: приглашение исчезло, появилась статья в урезанном виде (если конечно, она не слишком маленькая, тогда вы увидите ее целиком). Надо понимать, что это не сама нода, а лишь ее "представление", ссылка на нее. По мере добавления материала на сайт на главной странице сверху будут появляться ссылки и усеченные версии новых статей (если, конечно, для них установлен флажок *Помещать на главную*), а более старые будут опускаться вниз.

Нода "О сайте" нам на главной странице не нужна, поэтому отредактируйте материал снова, убрав соответствующий флажок.

На странице *Управление* в разделе *Управление содержимым* выберите пункт *Содержимое*. Перед вами откроется форма, позволяющая видеть, какие материалы есть на сайте, искать (фильтровать) материалы определенного типа, осуществлять с ним ряд действий. Эта страница бывает весьма полезна, если надо снять с публикации несколько статей или разместить их на главной странице. Редактировать каждую из них было бы достаточно трудоемко. Пока, как мы видим, на сайте существует только одна статья, так что пока нам здесь делать нечего.

Сайт будет блогом. Запись в блоге — это особый тип материала, это не Page и не Story. Включить тип материала *Запись в блоге* можно на странице *Модули* (модуль Blog). Теперь при попытке создать материал, помимо страниц и историй, вам будет предложена возможность создать блоговую запись.

Создадим свою первую запись в блоге. В качестве материала можете взять статью "Краткая история языков программирования" (1). Поместите ее заголовок в поле *Заголовок*, а все остальное в *Содержимое*, разделите содержимое на абзацы и сохраните материал. Убедитесь, что усеченный материал статьи появился на главной странице, а адрес самой статьи — developer.home/node/2.

Статья получилась неоформленной, по-идее в ней должны быть подзаголовки (<h2> или <h3>) и выделение терминов курсивом (). Ничего нам не мешает отрыть статью на редактирование и прописать соответствующие теги. Однако сначала надо определиться с уровнем подзаголовков. Понятно, что они должны быть младше, чем основной заголовок статьи. Чтобы выяснить, какого он уровня, посмотрим исходный код страницы. Если заголовок статьи второго уровня, то логично будет выделять подзаголовки третьим уровнем. Заходим в редактирование, обрамляем подзаголовки в контейнер <h3> … </h3>, а термины — в … .

Сохранив статью, видим, что курсив сработал, а заголовки — нет. Об этом же свидетельствует и исходный код страницы: там нет тегов <h3>. Почему? Давайте вернемся на страницу редактирования и раскроем там раздел *Формат ввода*. Здесь переключатель по умолчанию стоит на варианте *Filter HTML*, а среди допустимых тегов нет тега <h3>. Именно поэтому он не был интерпретирован. Переключитесь на вариант *Full HTML* и сохраните изменения. Теперь заголовки третьего уровня есть. Хотя их вид может оставлять желать лучшего, но они все-таки есть.

Однако предпочтительнее использовать именно формат ввода Filter HTML, что связано с безопасностью сайта. Поэтому сделаем немного по-другому. Сначала снова включим вариант Filter HTML для статьи, а затем перейдем на страницу *Управление → Форматы ввода* (находится в разделе *Настройка сайта*). Здесь выберем *Настроить* напротив *Filter HTML*, а далее следует щелкнуть на кнопке *Настройки* вверху. В поле *Допускаются только следующие HTML теги* через пробел надо дописать тег <h3> и сохранить изменения. Тем самым мы изменили сам фильтр.

Добавьте еще пару записей в блог: "Языки программирования. Общая характеристика" (2) и "Некоторые причины и тенденции развития языков программирования" (3). При этом разметьте соответствующие части статей с помощью тегов HTML (списки - , <ul>, усиление -).

Посмотрите, как выглядит главная страница.

Урок 4. Установка и настройка HTML-редакторов в Drupal

Как можно было заметить на прошлом уроке, при создании статей не удобно размечать текст тегами HTML вручную, особенно если текст большой и содержит много структурных элементов. Куда удобней, когда можно просто нажимать на кнопки, после чего в текст добавляются соответствующая разметка и оформление так, как это бывает, например, в текстовых процессорах. Действительно, в Drupal можно подключать модули, которые позволяют использовать определенные HTML-редакторы, или сам модуль является реализацией редактора. В web широкое распространение нашли несколько редакторов: например, CKEditor, BUEditor, TinyMCE и др.

Текстовые редакторы делят на два класса — это плоские редакторы и так называемые WYSIWYG-редакторы ("что видишь, то и получишь"). В последних пользователь не вписывает и не видит теги (коды) разметки и оформления, он сразу видит конечный результат. То, каким образом достигается такое представление, скрыто от глаз пользователя.

Мы рассмотрим пару разных редакторов.

CKEditor

CKEditor представляет собой HTML-редактор типа WYSIWYG. Скачать его можно с сайта

ckeditor.com (страница download, далее выбрать загрузку обычного CKEditor). Сделайте это, сохраните и распакуйте архив в любом месте (например, на рабочем столе).

CKEditor – это отдельный проект, не имеющий отношения к Drupal. Поэтому, чтобы подключить данный редактор к системе Drupal был разработан специальный модуль, который также называется CKEditor. Любые модули Drupal загружаются с сайта drupal.org. Чтобы быстро найти необходимый модуль, на главной странице этого сайта в форме вверху следует вписать его название и ниже выбрать вариант *Modules*, после чего нажать *Search* (поиск). Далее появится страница с ссылками подходящими под запрос поиска. Скорее всего, CKEditor будет первым. Перейдя на страницу модуля, вы можете загрузить подходящую версию (ссылки на архивы находятся в конце страницы, зеленым цветом помечены стабильные версии). Загрузите и распакуйте модуль. При распаковке будьте осторожны: не перепутайте модуль и сам редактор, т.к. их каталоги называются однаково.

Установим редактор в модуль:

1. Откройте каталог, где находятся файлы модуля. Там есть одноименный каталог ckeditor. Зайдите в него, здесь вы должны увидеть текстовый файл *COPY_HERE.txt*.
2. Переместите сюда каталог ckeditor с редактором.

Теперь необходимо переместить каталог-модуль ckeditor в файловую структуру системы Drupal (туда, где установлен сайт). Но куда именно? В каталоге WebServers/home/developer.home/www есть директория под названием *modules*. В ней находятся файлы всех модулей, изначально включенных в систему Drupal. Теоретически, если мы сюда поместим новый модуль, то система его увидит, и он будет работать. Однако так делать не следует. Все сторонние модули, загруженные пользователем, правильней помещать в каталог sites/all/modules. Папки modules в каталоге all в Drupal 6 по умолчанию нет, поэтому создайте ее и переместите сюда ранее сформированный каталог с модулем CKEditor.

Теперь в браузере перейдите на страницу модулей вашего сайта. Там должен появиться модуль CKEditor, включите его. После этого на главной странице управления сайтом в разделе *Настройка сайта* появится ссылка на страницу конфигурации редактора CKEditor. Данный редактор имеет множество настроек, которые мы не будем рассматривать. Как вы можете заметить, модули устанавливаются на английском языке; для их русификации следует посетить сайт drupaler.ru. Пока мы этого тоже делать не будем, т.к. далее не будем использовать CKEditor.

Создайте новый материал типа *Запись в блог*. Когда откроется страница добавления материала, вы увидите, что к полю редактирования текста прикреплена панель инструментов, подобно тому, что мы наблюдаем в любом текстовом процессоре. Попробуйте набрать произвольный текст и оформить его с помощью кнопок. Имейте ввиду, что здесь абзац формируется единичным нажатием Enter, а не двойным, как в "голом" Drupal. Если вам неудобно набирать текст в узком поле, то нажмите кнопку *Максимизировать* и редактор развернется на все окно. Если требуется посмотреть, какие теги HTML вставляет редактор, то

следует нажать на кнопку *Источник*. Здесь можно редактировать материал вручную.

Назовите созданный вами материал Test1 и сохраните его.

Поскольку мы будем использовать другой редактор, зайдите на страницу модулей и отключите CKEditor.

BUEditor

BUEditor представляет собой более простой редактор, хотя назвать его "плоским" сложно. Теги вставляются с помощью кнопок, хотя пользователь видит именно теги разметки, а не результат их интерпретации. Поэтому BUEdior не является wysiwyg-редактором. Преимущество этого редактора заключается в том, что он легко настраивается под нужды пользователя.

Для того, чтобы установить в систему BUEditor достаточно загрузить с сайта drupal.org и распаковать в каталог sites/all/modules модуль *bueditor*. Далее его надо включить на странице модулей сайта (если эта страница уже была открыта, то перезагрузите ее).

Создайте публикацию Test2, исследуйте при этом возможности редактора. BUEditor не переопределяет установки Drupal: абзац формируется двойным нажатием Enter.

Теперь перейдем на страницу конфигурации редактора (Управление → Настройка сайта → BUEditor). Мы не будем русифицировать этот модуль, т.к. здесь не так уж много настроек и в большей степени они интуитивно понятны. Страница настроек BUEditor разделена на две части — *Available editors* (доступные редакторы) и *Role-editor assignments* (назначение редакторов ролям). В последних версиях BUEditor по умолчанию существует четыре разновидности редактора – BBCode, Commenter, Default и Lab. Версию Default мы уже видели, когда создавали материал Test2. Commenter представляет собой урезанную версию Default. BBCode использует специфичный язык разметки похожий на HTML; перед отображением страницы система Drupal преобразует теги BBCode в теги HTML. В Lab многие кнопки вызывают функции, осуществляющие те или иные действия (например, поиск, замену, отмену действия и т.д.).

В разделе *Role-editor assignments* мы видим три роли, которые существуют на данный момент в системе Drupal. Это user #1 (администратор системы), зарегистрированный и анонимный пользователи. Анонимный пользователь — это любой посетитель сайта, не зарегистрированный в системе. По умолчанию в Drupal анонимные пользователи не имеют даже прав оставлять комментарии. Зарегистрированный пользователь — это пользователь, представившийся системе, через форму *Вход в систему* (которую мы видели, когда выходили из своего аккаунта). Перед этим пользователь однажды должен был пройти процедуру регистрации самостоятельно или с помощью администратора системы. Как видно, зарегистрированному пользователю не назначен никакой редактор. Назначьте ему Commenter.

Поменяйте для user #1 редактор с Default на Lab. Откройте на редактирование ранее

созданную статью (Test1 или Test2). Оцените возможности такого варианта редактора. После этого установите для администратора снова редактор Default.

Ранее было отмечено, что BUEditor легко настраивается под нужды пользователя. Давайте выясним, как это делается. Допустим, нам надо отредактировать вариант редактора Default. Для этого перейдем по ссылке *Редактировать* напротив его названия. Перед нами откроются настройки редактора Default. Наибольший интерес представляет раздел *Buttons*. Назначение столбцов понять не сложно. Создадим еще одну кнопку в этом редакторе, например, Table, генерирующую теги таблицы, с вложенными в нее тегами одной строки и двух ячеек. Внизу в строке подсвеченной красным цветом в столбце *Заголовок* пишем Table. В строке содержимого должен быть код HTML или вызов функции языка JavaScript.

Напишем для таблицы понятный нам код:

```
<table>
  <tr>
    <td></td>
    <td></td>
  </tr>
</table>
```

Поскольку для таблицы нет пока иконки, то вместо ее имени файла пропишите слово Table. Сохраните изменения. В поле *Demo* должна появиться кнопка *Table*, посмотрите, как она работает.

Теперь осталось подготовить иконку (рисунок *.png обычно размером 20x20 пикселей) и положить ее в каталог /sites/all/modules/bueditor/icons/. После этого перезагрузите страницу настроек редактора Default и в выпадающем списке иконок выберите ваш рисунок, сохранив после этого изменения. Следует отметить, что кнопки можно перемещать относительно друг друга, настраивая тем самым их последовательность над формой ввода текста.

Удалите две созданные на этом уроке статьи, используя форму на странице *Управление* → *Управление содержимым* → *Содержимое*.

Урок 5. Адреса страниц. Регистрация пользователей и создание ролей

Использование модуля Path

Адреса страниц, которые мы создавали до этого, выглядели так: доменное_имя/node/№_страницы (например, developer.home/node/4). Для поисковых роботов это вполне приемлемый адрес, а вот людей он может отталкивать своей неинформативностью. Поэтому лучше, чтобы в адресах страниц был какой-нибудь намек на то, что содержится на данной странице. Например, если на странице речь идет о программировании, то логично, чтобы ее

адрес выглядел примерно так: developer.home/programming.

Для того, чтобы можно было переименовывать адреса страниц, в системе управления содержимым Drupal надо включить модуль Path. После его включения на главной странице управления сайтом появится ссылка *Синонимы* (в разделе *Конструкция сайта*). Задавать собственные имена ссылкам можно здесь, но лучше это делать сразу при создании материала на странице его редактирования.

Перейдите на страницу *Управление → Управление содержимым → Содержимое*. Здесь вы должны увидеть созданные вами ранее четыре страницы. Нажмите на ссылку *Изменить* напротив страницы *О сайте*. Откроется форма ее редактирования, где должен появиться новый раздел *Параметры адреса*; раскройте его. В поле следует вписать новое имя страницы. Будьте внимательны, не надо писать полный адрес, developer.home/ следует опустить. Впишите в поле слово about и сохраните изменения. Аналогичным образом переименуйте остальные страницы; задайте для них такие имена - history-programming, property-lang, evolution-lang. Перейдите на какую-нибудь из этих страниц, чтобы убедиться, адрес выглядит теперь иначе.

На самом деле у нас есть еще пара страниц, которые мы не создавали, но Drupal создал их автоматически. Поскольку мы создавали записи в блог, то была сгенерирована страница, где в хронологическом порядке перечисляются все записи в блог от всех пользователей. Ее адрес — developer.home/blog. Также должны существовать страницы блогов для каждого отдельного пользователя. Перейдите на главную страницу сайта и кликните по ссылке *Блог пользователя — имя_пользователя*. Адрес страницы будет выглядеть следующим образом: developer.home/blog/1. Давайте переименуем этот адрес со страницы *Управление → Синонимы*. Нажмите кнопку *Добавить синоним*, далее в поле *Существующий системный путь* напишите blog/1, а в поле *Синоним пути* — blog/имя_пользователя (вместо имя_пользователя впишите логин вашего пользователя). Сохраните изменения и перейдите на страницу блога пользователя, чтобы отметить изменения в адресе страницы.

Пользователи

Посетители сайта, не зарегистрированные в системе Drupal, называются анонимными пользователями. По умолчанию они не могут оставлять даже комментарии. Чтобы убедиться в этом, достаточно покинуть аккаунт администратора. Если в вашей операционной системе стоит два и более браузеров, то не выходите из аккаунта, а запустите другой браузер и откройте сайт. Здесь вы окажетесь в роли анонимного пользователя.

Разрешим всем посетителям сайта оставлять комментарии, но во избежание спама комментарии должны будут проходить модерацию (т.е. предварительный просмотр администратором сайта). Для этого перейдите на страницу *Управление → Управление пользователями → Разрешения ролей*. В столбце *анонимный пользователь* в разделе *модуль comment* проставляем флажки *иметь доступ к комментариям* и *размещать комментарии*, далее сохраняем изменения. Теперь в окне другого браузера (где мы играем роль анонимного пользователя) перезагрузите страницу и оставьте комментарий к любой

статье. (Обратите внимание, сначала происходит просмотр комментария, а уже потом его сохранение.) В результате вы получите сообщение, что ваш комментарий был отправлен на проверку, а сам комментарий вы не увидите.

Администратор системы, решивший заняться модерацией новых поступивших комментариев, должен перейти на страницу *Управление содержимым → Комментарии → Очередь на подтвержение*. Здесь каждый комментарий следует просмотреть (ссылка *изменить*). Комментарии, содержащие спам и другие недопустимые вещи, следует удалить, остальные опубликовать. Опубликуйте пока единственный комментарий. В другом браузере перезагрузите страницу и отметьте его появление.

Анонимный пользователь по умолчанию может зарегистрироваться в системе Drupal. Попробуйте сделать это. Перед вами откроется форма, где нужно будет вписать имя и адрес электронной почты. Сайты Drupal, расположенные в Интернет на нормальных хостингах, высылают по почте пароли для пользователей. К сожалению Denwer этого не сделает. Но давайте все равно зарегистрируемся в системе, а пароль подправим со стороны администратора сайта на странице *Управление пользователями → Пользователи* (надо перейти по ссылке *изменить* и вписать в поле *Пароль* новый пароль). Теперь в другом браузере войдите в систему под только что зарегистрированным пользователем.

Зарегистрированные пользователи могут оставлять комментарии без предварительной модерации. Это не совсем хорошо, т.к. зарегистрироваться в системе может каждый, даже спамер или робот. Обычно с этим борются либо включая модерацию для зарегистрированных пользователей, либо устанавливая так называемую капчу, которая требует, например, при регистрации дополнительных действий от посетителя сайта. На странице У*правление пользователями → Разрешения ролей* снимите флажок *размещать комментарии без проверки* для зарегистрированного пользователя. Но и после этого на реально функционирующем сайте лучше ставить капчу, иначе в системе количество эфемерных пользователей будет расти в геометрической прогрессии. Удалять их можно со страницы *Управление пользователями → Пользователи*. Можно запретить посетителям регистрироваться самостоятельно вообще. Для этого нужно на странице *Настройка пользователей* установить переключатель в позицию *Только администраторы могут создавать новые учетные записи пользователей*.

Когда регистрация пользователей на сайте не предполагается, бывает лучше отключить блок *Вход в систему*. Отключите его и выйдите из своего аккаунта. Как теперь войти под своим логином на сайт? В CMS Drupal есть специальная страница с формой входа в систему: домен_сайта/user. Перейдите на эту страницу вашего сайта и войдите в систему через нее.

Роли

Посетители (пользователи) сайта могут выступать по отношению к сайту в разных ролях. Пока мы видели только три роли, присутствующие в Drupal по умолчанию. Администратор сайта — это особенная роль, которой позволительно все, что вообще возможно в CMS Drupal. Данная роль может принадлежать лишь одному аккаунту (одному человеку).

Зарегистрированных и анонимных пользователей может быть любое количество, между собой они могут различаться своими возможностями (разрешениями) на сайте, которые настраиваются через страницу *Разрешения ролей*.

Но что делать, если существующих трех ролей недостаточно? Что, если на сайте должны быть как зарегистрированные пользователи, так и особые пользователи, осуществляющие проверку их комментариев и записей (модераторы). Или на сайте может содержаться материал, доступ к которому должен предоставляться только некоторым пользователям. В случае, когда возникают подобные задачи, администратор Drupal может создавать новые роли и назначать им соответствующие права.

Пусть наш сайт будет чем-то вроде многопользовательского блога. Допустим, два программиста ведут личные блоги на одном сайте. Требуется создать специальную роль, которая позволяла бы вести на сайте блог, а также создать пользователей, которым принадлежит эта роль.

Создать новую роль можно на странице *Управление пользователями → Роли*. Добавьте здесь роль "блогер", далее кликните по ссылке *изменить права*. Разрешите блогеру следующее:

- создавать записи в блоге
- удалять свои записи в блоге
- редактировать свои записи в блоге
- иметь доступ к комментариям
- размещать комментарии без проверки
- доступ к содержанию сайта
- создание синонимов адресов страниц
- иметь доступ к профилям пользователей
- изменять свое имя пользователя

Поскольку блогеры могут создавать записи, скорее всего, им понадобится HTML-редактор. Поэтому на странице настроек BUEditor следует назначить для роли "блогер" вариант редактора, например, Default.

У нас в системе уже есть один пользователь (помимо администратора) в роли зарегистрированного пользователя. Поменяем ему роль. Для этого на странице *Пользователи* перейдите по ссылке *изменить* напротив его имени. Откоется страница настроек этого пользователя, установите там флажок *блогер* и сохраните изменения.

Теперь поменяем автора у всех ранее опубликованных статей, кроме страницы "О сайте". Пусть администратор сайта только управляет им и размещает информационные страницы, а блоговые записи делают только блогеры. Для этого на странице редактирования каждой

статьи в разделе *Информация об авторе* в поле *Автор* впишите имя пользователя-блогера.

Создайте еще одного пользователя-блогера (*Пользователи → Добавить пользователя*). Зайдите под его аккаунтом и добавьте пару статей, например, "Что такое ядро операционной системы" и "Особенности ядер Unix-подобных операционных систем". Не забудьте назначать адресам страниц синонимы.

Урок 6. Организация навигации по сайту. Создание меню и подшивок

Количество статей на сайте с течением времени будет увеличиваться. Чтобы было легко ориентироваться и без проблем находить то, что требуется, следует продумать удобную навигацию по сайту. То, как будет организовано меню сайта, как страницы сайта будут между собой связаны с помощью ссылок, играет важную роль, как для пользователей, так и для поисковых роботов. Для посетителей сайта важна понятность навигации, ее однозначность, а для поисковых систем — хорошая перелинковка. На каждую страницу сайта должна быть ссылка или из меню, или хотя бы с других страниц сайта.

Самое очевидное, что приходит в голову, это организовать доступ к материалам сайта с помощью меню. Пусть у нас будет меню со ссылками на блоги пользователей. На прошлом уроке мы забыли назначить синонимы для страниц блогеров. Адрес блога конкретного пользователя должен выглядеть так: developer.home/blog/имя_пользователя, где вместо имя_пользователя следует вписать логин блогера. Как вы помните, синоним можно добавить на странице *Конструкция сайта → Синонимы*.

Создание меню

Добавить новое меню на сайт можно на странице *Конструкция сайта → Меню*. Здесь следует нажать на кнопку *Добавить меню* и в открывшейся форме вписать уникальное "машинное" имя меню (на английском языке) и его название (можно на русском). Обычно название меню отображается при его размещении на сайте. Также по желанию можно добавить описание меню. Создадим меню под названием "Блоги" (машинное имя: mnu-blogs).

После сохранения меню открывается страница со списком его пунктов. На данный момент здесь ничего нет. Добавим два пункта — ссылки на блоги пользователей. Нажимаем кнопку *Добавить пункт*, в поле *Адрес* вписываем blog/имя_пользователя, в поле *Название ссылки в меню* — логин первого блогера. Вес влияет на последовательность ссылок в меню. Чем он меньше, тем ссылка выше. Если же у ссылок одинаковый вес, то они располагаются по алфавиту. Также добавьте в меню блог второго пользователя.

Несмотря на то, что меню было создано, на сайте оно так и не появилось. Когда создается меню, то автоматически вместе с ним создается блок для него. Именно размещение блока в определенной области сайта, открывает доступ к этому меню. Поэтому, чтобы все-таки меню

появилось, нужно перейти на страницу *Блоки*. Здесь в разделе отключенных блоков вы должны увидеть только что созданный блок меню. Разместите его в правой колонке. Кнопку *Создано на Drupal* можно отключить. После сохранения изменений новое меню должно появиться. Перейдите по его ссылкам, чтобы удостовериться, что все было сделано правильно.

Следует отметить, что чаще ссылки на страницы добавляются в меню не так, как мы сейчас делали. Обычно это происходит в момент создания материала на странице его редактирования. При этом само меню уже должно быть создано заранее. Вспомните, как мы создавали ссылку на страницу "О сайте".

Подшивка

В CMS Drupal меню — это далеко не единственный способ организация навигации по материалам сайта. Одним из способов является использование так называемой подшивки, когда ссылки на определенные страницы располагаются вместе на главной для них странице. Обычно имеет смысл группировать таким образом ссылки на статьи, которые связаны между собой одной темой и последующая страница является в каком-то смысле продолжением предыдущей.

Поскольку у нас на сайте слишком мало материала, будем использовать подшивку не совсем по ее назначению, а лишь с целью знакомства с возможностями. Пусть через подшивку будут организованы месяцы: на каждой странице подшивки будут объединены блоговые записи, которые были сделаны в одном определенном месяце.

Подшивка — это определенный тип материала. Чтобы он был доступен, надо включить модуль Book. После этого при создании публикации вы сможете выбрать пункт *Страница подшивки*.

На странице управления в разделе *Управление содержимым* появился пункт *Подшивки*. Откройте настройки подшивки. В разделе *Разрешенные в подшивке типы материалов* установлен флажок на пункте *Страница подшивки*. Это значит, что в подшивку мы можем объединять другие подшивки и создавать целые деревья подшивок. Нам этого не надо. Поэтому устанавливаем флажок только на пункт *Запись в блог*. Также запись в блог пусть будет дочерней страницей по умолчанию. Сохраните настройки.

Теперь создайте пустое меню "Архив подшивок" (mnu-archive) и расположите в правой колонке сайта под меню "Блоги". Оно не появится, т.к. пока в нем отсутствуют пункты меню.

Далее следует создать материал подшивки (*Создать публикацию → Страница подшивки*). В заголовок и название ссылки в меню впишем месяц, когда производились публикации на сайте. (Если публикации были добавлены в течение двух месяцев, то придется создать две подшивки.) Родительский пункт меню — *Архив подшивок*. Ниже в разделе *Оглавление подшивки* нужно выбрать *Создать новую подшивку*. В качестве адреса страницы пропишите, например, book/september. Появится пустая страница и справа отобразиться новое меню.

Как администраторы системы Drupal давайте откроем любую статью на редактирование. Там в разделе *Оглавление подшивки* выбираем только что созданную нами подшивку и сохраняем изменения. После чего на странице developer.home/book/september должен появиться пункт-ссылка на эту блоговую запись.

Если мы войдем в систему в роли блогера, то не сможем добавлять материалы к подшивке. Администратор сайта должен отредактировать разрешения ролей и позволить блогерам добавлять материалы к подшивке. Сделайте это. После этого зайдите на сайт под разными пользователями и отредактируйте материалы, добавив их в подшивку. В результате в подшивке месяца (если он один) у вас должно быть пять ссылок на статьи.

В конце добавьте от каждого пользователя еще по одной записи в блог (например, "Типы файлов в Linux" (6) и "Тестирование и отладка программ" (7)). При этом добавляйте записи со страницы подшивки, кликнув по ссылке *Добавить дочернюю страницу*. В этом случае в разделе *Оглавление подшивки* страницы редактирования материала сразу будет установлена подшивка текущего месяца.

Отметим недостатки такого способа организации материалов для нашего сайта (надо понимать, что в иных случаях подшивка может быть наилучшим выбором). Чтобы пользователи могли добавлять свои записи к подшивке, она заранее должна быть создана администратором сайта. Но и при этом блогер может забыть добавить статью к подшивке. Обычно сайты существуют длительное время (больше года). Даже если каждую подшивку называть более конкретно (например, "Сентябрь 2011", "Октябрь 2011"), их количество будет сильно увеличиваться. Конечно, можно организовать месяцы в подшивку года, создав тем самым иерархическую структуру. Однако при этом пришлось бы решать еще ряд проблем.

Урок 7. Таксономия и облако тегов

Таксономия

Навигацию по сайту можно организовать и через так называемую таксономию. Вообще таксономия — это учение, которое занимается классификацией и систематикой; изначально оно никак не связано с web, но термин и его значение прижились во Всемирной паутине.

Любую статью можно отнести к какой-нибудь теме. Если на сайте предусмотрена таксономия, то при создании статьи (например, записи в блог) пользователь обязан (или ему предлагается) выбрать из уже существующих или вписать новую тему, которой будет принадлежать данная запись. Для статей каждой темы движок Drupal формирует отдельные страницы, после чего на них можно устанавливать ссылки из меню. Т.е. если на прошлом занятии мы пытались классифицировать материал по месяцу публикации, то на этом уроке будем классифицировать по темам. В принципе правильно на сайтах-блогах предусмотреть обе классификации, т.к. в определенных случаях бывает удобнее искать материал по дате публикации, а в иных — по теме.

Допустим, на нашем сайте освещаются следующие темы: программирование, Python, Linux, web-технологии. Бывает строгая таксономия, когда статья может принадлежать только одной теме, но пусть у нас любая запись в блог может принадлежать или одной или сразу нескольким темам. При этом запись обязательно должна принадлежать хотя бы одной теме.

Также следует определиться с тем, может или нет пользователь (блогер, в нашем случае) сам добавлять новые термины таксономии при создании материала, если ему недостаточно существующих. Пусть у нас будет так, что он не может этого делать; термины таксономии будет создавать только администратор сайта.

Итак, перейдите на страницу *Управление содержимым → Таксономия*, нажмите кнопку *Добавить словарь*. Заполните открывшуюся форму следующим образом:

- *Название словаря:* Темы
- *Типы содержимого:* Запись в блог
- *Настройки:* Множественный выбор, Обязательный

Сохраните изменения. Теперь кликните по ссылке *Добавить термины* напротив только что созданного словаря. По очереди добавьте термины (программирование, Python, Linux, web-технологии). После чего щелкните по кнопке *Список* вверху, вы должны увидеть список созданных терминов.

Наведите мышку на термины и посмотрите их адреса URL в строке состояния внизу. Там значится примерно следующее: developer.home/taxonomy/term/1. Номера в конце адреса не очень информативны, поэтому назначим адресам страниц синонимы, заменив цифры на ключевое слово темы. Чтобы было удобно переименовывать, откройте несколько вкладок в браузере. Адреса страниц должны выглядеть так:

- developer.home/taxonomy/term/linux
- developer.home/taxonomy/term/python
- developer.home/taxonomy/term/programming
- developer.home/taxonomy/term/web

Если вы перейдете на любую из этих страниц, то увидите сообщение "В этой категории нет материалов". Когда мы добавляли статьи, то словаря таксономии еще не существовало, поэтому уже добавленные материалы не принадлежат ни одному термину словаря. Это несложно исправить. Поочередно откройте статьи на редактирование и выберите в списке *Темы* один или несколько терминов, которым может принадлежать заметка. Чтобы выбрать несколько тем, следует зажать Ctrl. Большинство ранее добавленных статей относятся к теме "Программирование". Чтобы перейти на эту страницу и посмотреть, как она выглядит щелкните по ссылке "программирование" внизу страницы любой из статей, которая принадлежит этой теме. Вы окажетесь на странице taxonomy/term/programming.

Облако тегов

Далее можно создать меню и привязать к нему страницы терминов, но мы поступим по-другому. Воспользуемся специальным модулем, который создает так называемое облако тегов. Облако тегов - это тоже меню, но своеобразного вида. Понятие "тег" совпадает или очень близко по значению с понятиями "термин, тема" и т.п.

С сайта drupal.org загрузите модуль Tagadelic и распакуйте его в каталог sites/all/modules системы Drupal. На странице модулей включите его. Если на сайте существуют словари таксономии, то Tagadelic автоматически создает блоки для них. Перейдите на страницу *Блоки* и разместите блок *Tags in Темы* в правой колонке сайта.

Скорее всего результат будет выглядеть не очень красиво. Облако тегов подразумевает, что термины, у которых больше статей, имеют более крупный шрифт. Поскольку у нас пока терминов мало, то "программирование" будет выглядеть гипертрофированно. Кроме того, как-то странно выглядит надпись над блоком (Tags in Темы). Кликните по ссыке *настроить* напротив блока и впишите в поле *Заголовок блока* одно слово — Темы.

Добавим еще ряд статей от имен двух блогеров. Это может сделать и под администратором сайта: при редактировании материала в разделе *Информация об авторе* следует вписывать имя одного из существующих блогеров. При оформлении материала встречающийся программыный код помещайте в контейнер code, используйте таблицу там, где это необходимо (вспомните, мы создали кнопку в BUEditor для добавления табличных тегов; пользуйтесь ей; кроме того, следует добавить теги <table>, <tr> и <td> в формат ввода Filter HTML). Примеры статей для добавления:

- Направления развития и "эры" технологий Web (8)
- Всемирная паутина и язык HTML (9)
- Данные и операции над ними (10)
- Условный оператор (11)
- Права доступа к файлам в Unix-подобных операционных системах (12)

При добавлении материала вы должны были заметить, что облако тегов не обновилось (новые термины не появились). Чтобы исправить ситуацию, достаточно запустить крон (например, со страницы *Отчет о состоянии*).

Урок 8. Установка и настройка тем оформления сайта

До сих пор мы использовали поставляемую с системой Drupal тему с установленными по умолчанию настройками. Однако любой владелец web-ресурса желает, чтобы его сайт был оригинален и неповторим. Отчасти это достигается за счет web-дизайна. На этом уроке мы рассмотрим, как устанавливать и настраивать темы в CMS Drupal.

Темы, поставляемые с системой Drupal, хранятся в каталоге themes верхнего уровня его файловой структуры. Сторонние темы, которые администратор сайта загружает самостоятельно, настоятельно рекомендуют распаковывать в каталог sites/all/themes. В Drupal 6 каталога themes в папке all нет, поэтому необходимо его создать.

В большинстве тем предусмотрена возможность изменения некоторых стандартных настроек через интерфейс CMS Drupal. Когда администратор сайта меняет их, то в каталоге sites/default/files появляется папка, в которой сохраняются эти новые настройки. Обычно там среди прочего обязательно присутствует файл style.css. Проблема в том, что если вы будете вручную править этот файл, то потом в случае любого изменения настоек через интерфейс Drupal, он будет удален. Изменения, которые вы вручную вносите в файлы, содержащиеся в каталоге, где установлена сама тема, не изменяются при изменении настроек через интерфейс. Но тут появляется другая проблема — при обновлении темы (когда выходит ее новая версия), вы теряете все ваши изменения. Поэтому, запомните два правила:

1. Перед тем как вручную править тему, установите все необходимые вам настройки через страницу настроек темы в CMS Drupal.
2. В какой файл темы вы бы не вносили изменения, сохраняйте его, скопировав на локальный компьютер.

Обо всем этом можно не заботиться, если не менять вручную никакие файлы темы, но в реальности такие ситуации бывают редко.

Откройте каталог sites/default/files. Если вы не меняли настройки темы Minnelli (которую мы используем), то каталог не должен содержать связанного с ней подкаталога. Теперь в браузере перейдите на страницу настроек темы Minnelli (на самом деле она является разновидностью темы Garland) и измените цветовую схему. Отметьте, что в каталоге files появилась папка color, а в ней каталог с измененными файлами темы. Еще раз измените цветовую схему, в color появится еще один каталог (первый можно удалить).

На нашем сайте используются логотип (картинка в верхнем левом углу сайта) и фавикон (иконка в адресной строке) самой системы Drupal. Очевидно, каждому приличному сайту требуется иметь свои личные логотип и фавикон, которые будут в сети Интернет отличать его от других.

Щелкните правой кнопкой мыши по логотипу сайта и выберите в контекстном меню пункт *Свойства изображения*. Обратите внимание на его размеры: 64 x 73 пикселей. Примерно такое по размеру оригинальное изображение вам нужно будет создать самостоятельно в любом графическом процессоре. Вообще размер логотипа зависит от темы; часто бывает, что его размер не столь принципиален, т.к. тема способна "адаптироваться" под размеры изображения. Однако бывает, что изображение должно быть строго определенного размера. В данном случае вы можете строго не придерживаться оригинальной ширины картинки. Изображение лучше всего сохранить в формате PNG.

Для создания фавикона можно воспользоваться сайтом favicon.ru.

На странице настроек темы окончательно определитесь с цветовой схемой, снимите флажки *Использовать логотип по умолчанию* и *Использовать иконку по умолчанию*, укажите свои изображения и сохраните изменения.

Теперь в каталоге colors зайдите в только что сгенерированную папку. Откройте файл style.css на редактирование. Допустим, мы хотим изменить размер шрифта статей и немного подправить внешний вид заголовка третьего уровня. Найдите и исправьте следующие строки:

Исходное описание стиля	*Исправленное описание стиля*

`h3 {`
`  font-size: 140%;`
`}`
	`h3 {`
	`  font-size: 140%;`
	`  font-weight: bolder;`
	`}`

`p {`
`  margin: 0.6em 0 1.2em;`
`  padding: 0;`
`}`
	`p {`
	`  margin: 0.6em 0 1.2em;`
	`  padding: 0;`
	`  font-size: 1.2em;`
	`}`

Если был увеличен размер шрифта абзацев, то, следовательно, должен быть увеличен размер шрифта маркированных и нумерованных списков, встречающихся в статьях. Однако изменение свойства font-size для всех списков приведет к искажению списков в меню. Чтобы изменить только списки в ноде, надо добавить в файл style.css, например, следующее определение стиля:

```
.node ol, .node ul {
      font-size: 1.2em;
}
```

Файл style.css очень большой (более 1000 строк), кроме того, не всегда описание всех стилей сайта находится в одном файле. Чтобы найти описание внешнего вида того или иного элемента, пользуются специальными инструментами, например, дополнением Firebug для браузера Mozilla Firefox. Чтобы его установить, в браузере надо выбрать пункт меню *Инструменты → Дополнения*, найти firebug и установить его. Далее, открыв в браузере сайт, кликнуть по странице правой кнопкой мыши и в контекстном меню выберите пункт *Анализировать элемент*. Внизу справа вы будете видеть стили и файлы, в которых они описаны.

Допустим, нас не устраивает ширина центральной части сайта (места, где располагается статься на странице). В Firefox щелкните правой кнопкой мыши по области сайта и в контекстном меню выберите *Анализировать элемент*. (Обратите внимание, у сайта должно быть меню справа и слева.) Найдите тег `<div id="container" class="clear-block">`. Справа вы увидите его ширину в 980px, файл и место в нем, где находится это описание. В

данном случае оказывается, что ширина сайта определена в файле minnelli.css, расположенном в каталоге themes/garland/minnelli. Найдите этот файл, увеличьте все три свойства ширины (width) в нем, например, на 200px.

Теперь с сайта drupal.org (ссылка *Themes*) скачайте несколько понравившихся вам тем. Обращайте внимание на версию темы, которую вы скачиваете. Если у вас Drupal 6, то тема для Drupal 7 на нем работать не будет. Далее создайте каталог themes в папке sites/all и распакуйте темы туда. Протестируйте их по очереди (страница *Темы оформления*). Обратите внимание на то, какие области сайта присутствуют в каждой теме, является ли тема бестабличной и т.д.

После опытов снова примените для сайта тему Minnelli.

Урок 9. Вставка изображений в статьи. Установка модуля для синтаксической подсветки кода

Способы вставки изображений

Публикуемые на сайтах статьи могут содержать различные изображения (фотографии, рисунки, скриншоты и т.д.). До сегодняшнего дня мы размещали на сайте лишь текст; рассмотрим, как добавлять в него изображения.

На языке HTML разметка изображения осуществляется тегом img с атрибутом src, у которого в качестве значения используется URL места размещения изображения. Возникает вопрос, где должно находиться изображение: в сети Интернет (на любом другом сайте) или на вашем собственном сайте (хостинге, где размещены файлы сайта)? Когда изображение размещено на других ресурсах, то все достаточно просто: копируем адрес рисунка (чтобы получить адрес изображения, в браузере щелкают по нему правой кнопкой мыши и выбирают *Свойства изображения*) и вставляем его в тег img статьи на своем сайте. Если же мы хотим загрузить изображение на наш сайт, то существует как минимум два варианта:

- заранее загрузить "вручную" на сайт (например, по протоколу ftp или с помощью модуля upload), затем посмотреть адрес изображения и использовать его при добавлении тега img;
- вставлять изображения в момент редактирования статьи, загружая их с помощью специального модуля в файловую структуру CMS Drupal.

Второй вариант предпочтительнее, если вы не единственный, кто публикует материалы на сайте. Рассмотрим оба варианта.

Добавьте на ваш сайт от имени любого из блогеров статью, которая содержит два-три изображения (но пока без них), например, "Цикл While" (13). Первый рисунок (while.png) разместим копированием/перемещением в файловую структуру Drupal; но куда? По логике вещей, если модули и темы, которые загружает администратор сайта, помещаются в каталог

sites/all, то хорошо бы здесь же размещать изображения. Создадим каталог sites/all/images и переместим сюда один из рисунков.

Теперь откроем статью на редактирование, установим курсор в место, где должно быть это изображение и вставим тег img (можно воспользоваться соответствующей кнопкой BUEditor). Лучше указывать относительный адрес, т.к. в этом случае при смене доменного имени адрес все равно будет правильно указывать на местоположение рисунка. В результате тег должен выглядеть примерно так:

```
<img src="sites/all/images/while.png" width="605" height="209" alt="Цикл While" />
```

Когда вы сохраните статью после редактирования, то не увидите изображения, т.к. <img> не является допустимым тегом для формата ввода Filtered HTML. Добавьте <img> в настройках этого фильтра.

Второе изображение будем добавлять с помощью специального модуля — IMCE. Загрузите его с сайта drupal.org и распакуйте в каталог sites/all/modules, после чего включите на странице *Модули* CMS Drupal. На странице *Настройка сайта → IMCE* назначьте для роли "блогер" профиль User-1. Теперь зайдем на сайт под блогером и попробуем вставить в статью второе изображение.

Нажмите на кнопку вставки изображения редактора BUEditor, там должна появиться кнопка *Browse*. Нажмите ее, откроется окно файлового браузера IMCE. IMCE настроен так, что открывает просмотр каталога sites/default/files/, куда многие модули сохраняют сгенерированные и загруженные с их помощью файлы. Конечно, лучше если администратор создаст здесь дополнительный каталог и настроит IMCE так, чтобы открывался именно он.

Для загрузки изображения в окне IMCE следует нажать кнопку *Загрузить*, выбрать и загрузить изображение. После этого изображение появиться в основном окне IMCE, надо выделить его и нажать кнопку *Send to editor*.

Установка и использование GeSHi

На сайте, который мы создаем, некоторые статьи содержат примеры программного исходного кода. Код принято помещать в контейнер тега code. Внешний вид, который принимает исходный код после этого, зависит от применяемого к нему стиля CSS. Если вы посмотрите на любую статью сайта на Drupal, содержащую пример исходного кода, то заметите, что шрифт внутри тега code отличается от основного шрифта. Т.е. сама тема CMS имеет описание стиля для этого тега. Несмотря на это есть ряд проблем. Исходный код легче читается, если выполнена его синтаксическая подсветка. Реализовывать это только с помощью языков HTML и CSS весьма трудоемкая задача. В исходном коде бывает важным наличие отступов от левого края строки. Например, в языке программирования Python отступы имеют синтаксический смысл. Как мы видим, тег code их не интерпретирует.

Проблему можно решить с помощью установки специального модуля, который отвечает за

то, как отображается содержимое контейнера code. В системе Drupal широко используется модуль GeSHi Filter for syntax highlighting. Этот модуль позволяет использовать библиотеку GeSHi, которая должна быть также загружена в файловую структуру CMS Drupal.

Загрузите с сайта drupal.org модуль GeSHi Filter for syntax highlighting и распакуйте его в каталог sites/all/modules. С sourceforge.net/projects/geshi/files/geshi/ загрузите саму библиотеку. Каталог geshi (библиотека) должен быть помещен внутрь каталога модуля geshifilter.

Откройте страницу *Модули* сайта и включите только модуль Geshi Filter (GeSHi field и GeSHi node нам пока не нужны). Теперь его нужно настроить на странице *Настройка сайта → GeSHi Filter*. Здесь в поле *Generic syntax highlighting tags* содержатся теги, которые будут "обрабатываться" модулем GeSHi Filter. В поле *Default highlighting mode* (вариант подсветки по умолчанию) следует выбрать язык программирования, для которого будет выполняться подсветка синтаксиса, если используется тег code (или любой другой из указанных в поле *Generic syntax highlighting tags)* без каких-либо атрибутов. Когда необходимый язык отсутствуют, то его можно включить на странице *Настройка сайта → GeSHi Filter → Языки → Отключено*. Перейдите на эту страницу и включите Pascal, после чего на странице *Включено* пропишите тег для этого языка (<pascal>). Но на странице общих настроек GeSHi filter выберите в качестве варианта подсветки по умолчанию язык программирования Python и сохраните изменения.

Если вы теперь посмотрите на ранее созданную статью, содержащую тег code, то не увидите изменений. Дело в том, что для формата ввода Filtered HTML не включен фильтр GeSHi filter. Исправить этот недочет можно на странице *Настройка сайта → Форматы ввода → Filtered HTML*. После этого подсветка кода сработает.

Когда в тексте статьи вам потребуется подсветка синтаксиса отличная от установленной по умолчанию, вместо тега <code> следует прописать тег для требуемого языка (например, <pascal>). При этом язык программирования должен быть предварительно включен на странице настроек GeSHi filter.

Урок 10. Модуль Image. Создание галереи

На многих сайтах присутствуют так называемые галереи изображений, когда фотографии или рисунки группируются по категориям, а также организуется их удобный просмотр. Нечто подобное можно реализовать в CMS Drupal с помощью модуля Image.

Установите данный модуль в систему Drupal. На странице *Модули* включите его подмодули Image и Image Gallery. Первый добавляет возможность создания еще одного типа материалов, а второй — возможность создавать галереи.

На главной странице управления сайтом выберите *Управление содержимым → Галерея картинок*. Добавим здесь несколько галерей, например, "Алгоритмы", "Python", "Linux".

Просматривая список, обратите внимание на адреса страниц галерей. Поменяйте их на странице *Синонимы*: image/linux, image/python, image/algorithm.

Помимо страниц только что созданных галерей автоматически была сгенерирована еще одна страница, где перечислены все существующие галереи: developer.home/image. Посмотрите на нее. Добавим ссылку на эту страницу в меню Primary links через *Меню* системы Drupal.

Теперь следует установить разрешения для ролей по отношению к модулю Image. Заходим на страницу *Разрешения ролей* и устанавливаем для всех возможность смотреть оригиналы картинок, а для блогера — возможность создавать картинки, редактировать и удалять только свои картинки.

Галереи картинок создают отдельный словарь таксономии. В этом можно убедиться на странице *Управление содержимым → Таксономия*. Словарь Image Galleries содержит термины только что добавленных вами галерей изображений. На странице настроек данного словаря установите флажок *Обязательный*, чтобы пользователь не смог забыть причислить загруженное им изображение к какому-нибудь термину.

Кроме того, благодаря модулю Tagadelic появляется блок Tegs in Image Galleries. Разместите этот блок в одной из колонок сайта, в качестве заголовка блока пропишите "Галереи изображений". Блок не появится, т.к. еще ни одно изображение не было добавлено.

От имени пользователей-блогеров сайта добавьте ряд изображений, например, эти. При создании публикации следует выбирать тип материала *Image*. Каждое изображение желательно кратко описать одним-двумя предложениями в разделе *Содержимое* формы редактирования.

Примечание. При загрузке изображений в формате GIF возникает предупреждение.

Посмотрите галереи изображений, перейдя по ссылке в главном меню. Обратите внимание на ссылку *Оригинал* под некоторыми изображениями. Она позволяет открывать изображение в его исходном размере. Чтобы появился блок тегов, надо запустить Cron.

При включении модуля Image в разделе *Настройка сайта* появилась одноименная ссылка на страницу настроек данного модуля. Помимо прочего здесь можно поменять размеры создаваемых модулем миниатюр и картинок для предпросмотра. Однако делать это надо до загрузки изображений, т.к. рисунки сжимаются до необходимо размера в момент загрузки и при изменении настроек "разжаться" или "дожаться" уже не смогут. Так если, например, размер миниатюры будет увеличен уже после загрузки изображений, то все миниатюры будут выглядеть расплывчато.

Урок 11. Создание видов

Страницы, содержащие не конкретные статьи, а коллекции материалов, по умолчанию в Drupal организованы наподобие дневников, когда более новые записи появляются вверху, а старые уходят вниз и на следующие страницы. Таким образом, например, организован внешний вид главной страницы и страниц терминов таксономии. Та начальная часть каждой статьи, которая отображается на странице-коллекции, называется аннотацией, анонсом и т.п. Длину анонса можно поменять на странице *Настройки публикации*. Также здесь можно изменить количество публикуемых материалов на каждой странице-коллекции.

Часто бывает, что подобный внешний вид не устраивает владельцев сайта, они хотят или им требуется что-то особенное для их web-ресурса. Именно для таких случаев предназначен модуль Views (Виды). Загрузите и установите его в файловую структуру сайта; также с сайта drupaler.ru загрузите и установите файл перевода для данного модуля, т.к. сам модуль достаточно сложен и работа с ним и без того обычно малопонятна начинающим. Если вы экспортируете перевод в формате пакета, то после его разархивации переместите содержимое в папку модуля Views.

Включите следующие подмодули: Views и Views UI. Перейдите на страницу *Конструкция сайта → Представления*. Здесь уже есть несколько готовых представлений (включенные или нет). Включим представление frontpage, которое предназначено для формирования главной страницы, и перейдем на страницу его редактирования.

Перед вами откроется сложная форма с множеством настроек. Слева будут три вкладки — *Default* (по умолчанию), *Page* (Страница) и *Feed*. Выберем Page, т.к. нам требуется настроить вид именно целой страницы. Затем в разделе *Основные настройки* формы щелкните по значению *нет* свойства *Заголовок*. Внизу появится текстовое поле, куда можно вписать заголовок, который будет отображаться на странице. Впишите туда фразу "Новое на сайте". Обратите внимание на кнопку *Переопределить*. Если ее не нажимать, то изменения затронут все представление, а не только Page. В данном случае нас это устраивает. Поэтому сразу нажимаем на кнопку *Обновить отображение по умолчанию*.

Установите стиль в значение *Таблица*, щелкнув по фразе *Без форматирования*.

На странице надо что-то отображать. Это что-то определяется с помощью полей. Поле можно представить как определенную часть каждой статьи (материала). Например, поля — это заголовок, дата создания, автор, термин таксономии, к которому относится статья и т.д.

Щелкните по плюсу рядом с разделом *Поля*. Внизу появится список флажков — это поля, которые могут быть отображены в виде. Выберите следующее:

- Материал: Заголовок,
- Материал: Дата создания,
- Материал: Содержимое.

Нажмите на кнопку *Добавить*. В разделе Поля появятся три поля, а ниже перед вами

поочередно будут открываться настройки этих полей. Пока просто удалим из них метки и нажмем на кнопки *Обновить отображение по умолчанию*.

В разделе *Настройка страницы* адрес указан как frontpage. По умолчанию главная страница имеет адрес node. Можно либо поменять в настройках сайта, что главная страница будет не node, а frontpage. Либо непосредственно на странице редактирования представления поменять frontpage на node. Предпочтем второй вариант. После этого сохраните представление и посмотрите на главную страницу сайта (не уходите со страницы редактирования вида, лучше откройте вторую вкладку). Страница преобразилась, но вряд ли мы хотели именно такого вида: дата идет впереди заголовка, заголовок ничем не выделяется, а статьи представлены целиком. Продолжим редактировать вид Page представления.

Нажмите на кнопку с двумя стрелками рядом с разделом *Поля*. Ниже появится форма *Перестроить поля*. Поднимите заголовок выше даты создания, после чего нажмите на кнопку *Обновить*. Теперь щелкните поле *Материал: Заголовок*, в форме его редактирования установите флажок *Заменить выводимое полем значением* и пропишите в многострочном текстовом поле <h3>[title]</h3>. Кроме того установите флажок *Связать это поле с его материалом*. Сохраните изменения. Щелкните по полю *Материал: Содержимое* и в его настройках установите флажок *Установить для этого поля максимальную длину*. Пусть максимальная длина будет равна 400 символам.

Обратите внимание, что внизу вы можете видеть, как будет теперь выглядеть главная страница.

В основных настройках вида нажмите на кнопку с шестеренкой рядом со свойством *Стиль*. Поменяйте колонку даты создания на *Материал: заголовок*.

В свойстве *Элементов на страницу* укажем значение 6. Сохраните представление и посмотрите, как выглядит главная страница.

Ранее мы решили кроме прочего организовать коллекции материалов по месяцам. Для этого пришлось создавать подшивку. На самом деле куда правильней использовать специальное представление. Поскольку многие сайты коллекционируют материалы по месяцам, то модуль Views уже включает в себя представление archive. Включите его и перейдите на страницу редактирования.

У вида Page представления archive меняем стиль на табличный и добавляем следующие поля:

- Материал: Заголовок,
- Материал: Дата создания,
- Пользователь: Название,
- Материал: Тип
- Таксономия: Все термины.

Устанавливаем последовательность полей такую, как приведена в списке. Сохраняем вид.

Откройте страницу *Блоки* системы Drupal. В отключенных блоках появился новый блок *Archive list*. Разместите его на сайте вместо блока *Архив подшивок*. После этого посмотрите, как выглядят архивы месяцев.

Немного подредактируем представление. Щелкните по полю *Материал: Заголовок* и установите флажок *Связать это поле с его материалом*. Для *Пользователь: Название* в метку впишите слово "Автор", а также снимите флажок *Ссылаться на пользователя*. В основных настройках свойству *Элементов на страницу* присваиваем значение *Не ограниченно* (для этого следует прописать 0). Сохраните представление и посмотрите страницы архива.

Итак, модуль Views предоставляет возможность организовать материалы сайте не стандартным образом. Хотя в данном уроке мы использовали готовые представления, ничего не мешает вам создавать собственные. Следует знать, что возможности модуля Views намного превосходят рассмотренные на этом уроке.

Урок 12. Модуль CCK – пакет для конструирования содержания материалов

При создании новой статьи пользователь сайта на Drupal определяет, к какому типу материала она будет относиться. Некоторые материалы, такие как Page и Story, включены изначально, другие активируются при включении соответствующих модулей, например, Book, Blog, Image. Однако ничего не мешает пользователю-администратору создавать собственные типы материалов под какие-нибудь специфические виды статей.

До этого мы видели, что формы редактирования всех типов материалов содержат примерно одинаковые поля: заголовок, меню, содержимое и другое. Хотя, например, тип "Картинки" имеет дополнительное поле для загрузки изображения. Но для какого-нибудь особенного типа материала этих полей может быть недостаточно. Например, может потребоваться поле, где будет указываться уровень статьи, какие-нибудь числовые значения, может потребоваться загрузка нескольких изображений и т.п.

В CMS Drupal можно добавлять собственные поля к материалам, как созданным собственноручно, так и тем, которые активируются при включении модулей. Для создания дополнительных полей используется модуль CCK (Content Construction Kit). Загрузите его с сайта drupal.org, установите в систему, также загрузите файл перевода с drupaler.ru.

Перейдите на страницу *Модули* и включите следующие подмодули CCK – *Content, Option Widgets, Text*. Также следует включить подмодуль модуля GeSHi – *GeSHi field*, – который позволяет создавать поле исключительно для программного кода с подсветкой синтаксиса.

Допустим, на нашем сайте будут публиковаться решения задач по программированию. Для подобного материала хорошо бы создать собственный тип (не Story и не блоговую запись), т.к., скорее всего, он потребует поля нестандартного типа. Переходим на страницу

Управление → Типы содержимого → Добавить тип содержимого. В открывшейся форме в поле *Название* вписываем "Задача", а в поле *Тип* - "task", при желании можно заполнить поле описания типа. Далее раскройте раздел *Свойства формы*, в поле *Название поля текста* замените слово "Содержимое" на слово "Пояснение". Сохраните новый тип материала, после чего вы окажетесь на странице *Управление → Типы содержимого*.

Теперь необходимо настроить поля только что созданного типа. Для этого надо перейти по ссылке *управлять полями* типа материала *Задача*. К существующим стандартным полям добавим новое поле, в котором будет указываться язык программирования, на котором решена задача. Внизу страницы есть форма для добавления новых полей. Заполните ее следующим образом: в поле *Метка* впишите "Язык программирования", в *Имя поля* - "lang", тип данных и элемент — текст, текстовая строка. Нажмите на кнопку *Сохранить*, откроются настройки данного поля. Здесь установите флажок *Обязательный* и еще раз сохраните. После этого вы снова окажетесь на странице управления полями типа материала "Задача", где теперь в списке полей появится новое поле.

Следующее поле, которое нам потребуется, будет показывать, решена ли задача или еще нет. Добавляем такое поле: метка - "Задача решена", имя - "solved", тип — текст, элемент — флажки/радиокнопки. В настройках данного поля указываем: обязательный, список допустимых значений — "Да", "Нет" (каждое значение на отдельной строке).

Хотя исходный код решения задачи можно вставить в поле содержимого, но мы создадим специальное поле для кода. Метка - "Исходный код", имя — "source", тип — GeSHi field. Сохраним настройки поля как есть.

Теперь надо правильно выстроить последовательность полей, которую блогеры будут видеть в форме редактирования материала. Поля без проблем можно перемещать относительно друг друга. Расположите поля так, чтобы после *Параметры меню*, сразу следовали поля в следующей последовательности *Язык программирования, Задача решена, Пояснение, Исходный код*. Сохраните изменения.

Далее следует установить разрешения на странице *Разрешение ролей*. Блогеру должно быть позволено create task content, delete own task content, edit own task content.

Кроме того, не забываем, что у нас на сайте поддерживается таксономия. Заходим на страницу таксономии. Здесь два есть два словаря - "Image Galleries" и "Темы". Нас интересует только второй, нажимаем *изменить словарь* и в типах содержимого указываем помимо записи в блог еще и задачу. Сохранив изменения, добавляем еще один термин - "Решение задач". На странице списка терминов узнайте адрес страницы добавленного термина, после чего задайте для нее синоним taxonomy/term/tasks.

Наконец, создадим материал "Задача" от имени одного из блогеров. Примеры задач есть в архиве к данному курсу (файл tasks.odt). Обратите внимание, для поля GeSHi field можно выбрать подходящий язык для синтаксической подсветки. Также в данном поле не надо использовать тег code.

После сохранения статьи может не понравиться то, что слова "Pascal" и "Да" расположены ниже заголовков "Язык программирования" и "Задача решена", а не рядом. Чтобы исправить это, перейдите на страницу *Управление содержимым → Задача → Отображение полей*. Здесь в столбце *Метка* полей "Язык программирования" и "Задача решена" установите значение *В линию*.

Также можно обособить эти два поля от всего остального текста с помощью объединения их в отдельную группу. Для этого на странице модулей в CCK следует включить подмодуль Fieldgroup. Далее в управлении полями типа "Задача" создать новую группу, например "Информация о задаче" (taskinfo). Потом следует поднять только что созданное поле над полями "Язык программирования" и "Задача решена", а их как бы вставьте внутрь, немного сместив вправо. Чтобы название группы не отображалось, на странице отображения полей поставьте полю "Информация о задаче" метку *Скрыто*.

Добавьте еще несколько задач. Запустите cron, чтобы увидеть ссылку на задачи из облака тегов.

Заключение

Если бы сайт на локальном компьютере создавался не в учебных целях, и вы в последствие планировали бы разместить его в сети Интернет, то сайт нужно было бы правильно перенести на хостинг.

Самое ценное в сайтах, работающих на основе любой CMS, - это база данных. Ее потеря равноценна потери сайта. Поэтому важно всегда иметь ее свежую резервную копию. Чтобы экспортировать базу данных сайта надо зайти на панель управления phpMyAdmin (для Denwer – это localhost/Tools/phpMyAdmin/), выбрать в списке слева необходимую базу, а затем вверху найти вкладку или кнопку *Export*. Вид интерфейса зависит от версии phpMyAdmin. Во первых базу данных нужно сохранять в файл (Save as file); если она большая, выбрать способ сжатия (например, "zipped" или "gzipped"). Остальные настройки обычно трогать не надо. Далее следует непосредственно выполнить экспорт, нажав на кнопку, допустим, *Go*. В результате вы получите файл с расширением sql, gz или др.

Следует иметь в виду, что, если перед экспортом базы данных выполнить через интерфейс Drupal скрипт ее обновления (update.php), то размер базы уменьшится.

На хостинге данные надо импортировать. Для этого предварительно требуется создать базу данных и пользователя. Обычно это делается на панели управления хостингом. На данном этапе важно запомнить имя пользователя, его пароль и название базы данных. Далее следует перейти на панель phpMyAdmin (обычно ссылка на нее есть в панели управления хостингом), выбрать базу данных и через вкладку *Import* загрузить ранее экспортированный файл данных. Могут быть проблемы с допустимым форматом для импорта, а также размером файла. Бывает, что при импорте базы данных может возникнуть ошибка, но такое бывает редко.

Сами файлы сайта на CMS Drupal также надо загрузить на хостинг по FTP или через менеджер файлов панели управления хостингом. При этом сам Drupal всегда можно взять с сайтов drupal.org и drupal.ru, а вот "пользовательские" файлы имеют особую ценность. Они также как и база данных составляют контент вашего сайта. Поэтому их надо также регулярно сохранять. Обычно в CMS Drupal все сторонние для него файлы находятся в каталоге sites, именно в один из его подкаталогов мы загружали дополнительные модули и темы. Также там хранятся изображения.

После загрузки файлов на хостинг надо изменить запись о базе данных и пользователе в файле sites/default/settings.php (т.е. эта строчка уже есть, надо ее найти и изменить, а не вписывать новую строчку). Например, если на хостинге вы создали пользователя myuser с паролем 3qi8cdp и базу данных mybase, то соответствующая строчка в файле должна принять следующий вид:

```
$db_url = 'mysql://myuser:3qi8cdp@localhost/mybase';
```

Также на хостинге для файла settings.php важно оставить только право на чтение для владельца.