



## Оглавление

1. Добро пожаловать в Вордпресс.....	4
Почему Вордпресс удивителен.....	4
Все про установку, настройку и оптимизацию Вордпресс.....	4
Итак, вы никогда ранее не слышали про Вордпресс.....	5
2. Настройка Вордпресс.....	9
Знаменитая установка за 5 минут.....	9
И что теперь?.....	10
Выбираем структуру урл.....	10
Рубрики и Теги.....	12
Пользователи и Администраторы.....	15
Выбор идеального шаблона.....	17
Приступая к работе с плагинами.....	18
3. Анатомия шаблонов Wordpress.....	22
Понимание файлов шаблона.....	22
Понимание различных видов страниц.....	23
Разбираемся с заголовками.....	26
Сайдбар.....	34
Поисковая форма.....	37
Футер.....	38
Функции тем.....	39
4. Дизайн и Разработка Шаблонов.....	41
Настройка цикла loop.....	41
Сайдбары и Футеры.....	48
Меню, Списки архивов и Облака тегов.....	50
Боковое содержимое и полезные элементы меню.....	55
Создаем и используем дочерние темы.....	60
Стилизация шаблонов.....	62
Использование нескольких тем.....	64
Поддержка виджетов.....	65
5. Расширяем функциональность Wordpress.....	67
Расширения.....	67
Использование и уход за плагинами.....	70
Расширяем вордпресс пользовательскими функциями.....	73
Другие способы расширения возможностей.....	76
Wordpress как Система Управления Контентом.....	78
Расширение функционала CMS.....	83
6. Работаем с RSS лентами.....	86
Зачем нужны RSS ленты.....	86
За и Против предоставления фидов посетителям.....	86
Различные типы лент.....	86
Настройки и форматы фидов.....	88
Используем FeedBurner.....	91
Отслеживание и отображение статистики фидов.....	94
Настраиваем ленту.....	96
Валидация фидов.....	104
7. Работаем с комментариями.....	105
Оптимизируем зону комментариев.....	105
Формирование комментариев.....	107
Форматирование зоны комментариев.....	108
Настраиваем отображение комментариев.....	116

=====



Оптимизируем форму комментариев.....	121
Контролируем Спам.....	123
Другие вопросы и методы.....	124
8. Поисковая оптимизация.....	126
Сила и слабость SEO.....	126
Контроль дублированного контента.....	127
Оптимизируем структуру постоянных ссылок.....	132
Ранжируемся в Google.....	134
Отслеживание успешности сайта.....	140
9. Уход за здоровьем сайта.....	143
Бережем здоровье сайта.....	143
Закрываем доступ к важным файлам.....	145
Всегда актуальный Вордпресс.....	155
Оптимизируем вордпресс.....	158
До бесконечности и дальше.....	165

ZZZZZZZZZZZZZZZZZZ

Добро пожаловать в чудесный мир Вордпресса.

Есть много причин для покупки этой книги, но одна из них должна быть та, что вы уже поклонник Вордпресса. Это здорово, потому что я тоже, иначе не стал бы переводить эту книгу :)

Вордпресс делает контроль содержимого сайта легким. Это можете делать вы или научить кого-то еще, потому что административная панель настолько понятна, насколько мощна. Вордпресс дает вам ключи для создания сайтов со всеми современными фишками, которые посетители любят. И т. к. Вордпресс бесплатен, с открытым кодом и имеет гигантское сообщество, вы всегда в надежных руках. Вордпресс изменил подход к размещению контента в сети, и мы все пожинаем плоды этого подхода. По всем этим причинам мы уверенно можем назвать Вордпресс удивительным.

Если же вы купили эту книгу потому, что ваш шеф заставляет вас сделать сайт на Вордпрессе, а вы не очень рады этому факту... Надеюсь к концу книги вы перестанете хмуриться и увидите свет в конце тоннеля.

1. Купить книгу по вордпрессу
2. ???
3. Profit!!!

В этой книге вы узнаете про знаменитую легкую установку и про базовую настройку Вордпресс. Далее мы обсудим различные тонкие настройки, объясняя их и предлагая наше профессиональное мнение по поводу рекомендаций как это стоит делать.

Например, покажется как использовать Вордпресс в качестве полноценной системы управления контентом и попробуем разрушить миф о том, что вордпресс — это «еще один блогový движок». Разберется много полезного и интересного о плагинах, меню, шаблонах, шорткодах и т. п.

Прямо из коробки и при наличии здравого смысла, Вордпресс достаточно безопасная система. Но если вы веб-дизайнер на должном уровне, вы понимаете значимость безопасности и знаете, как важно сделать все возможное для обеспечения защиты сайта. Есть множество мер предосторожности и мы их вам покажем. Речь идет о блокировке важных файлов, проверке прав доступа и использовании различных проявлений принципа «**Безопасность через неясность**»

Кроме этого мы настроим наши сайты и оптимизируем эффективность Вордпресса. Многие из приемов, которые могут сделать Вордпресс быстрее, также сделают любой другой сайт быстрее, как например снижение количества http запросов и оптимизация изображений. Но есть много специфических чисто вордпрессовских приемов, как например кэширование, оптимизация базы данных и снижение количества запросов к базе данных.

*Важно! Обновления Вордпресс не только добавляют новые фишки, но и устраняют ошибки и уязвимости. Популярность Вордпресса делает его лакомой целью для плохих парней. Но*

популярность также позволяет быть уверенным в том, что обновления и заплатки будут выходить очень быстро и безопасность становиться все лучше и лучше.

Не забывайте обновляться на новые версии как только они выходят. Это делается в пару кликов.

По мере роста и развития сайта важно быть уверенным, что сайт счастлив и здоров без напоминаний о необходимости обновления движка и обновления шаблона. И конечно важно следить за логами сервера, бороться против битых ссылок и ошибок несуществующих страниц, и выполнять модераторские обязанности над комментариями.

Все это совершенно не сложно, особенно с оглядкой на жутких конкурентов.

## Итак, вы никогда ранее не слышали про Вордпресс

Если это ваше первое знакомство с Вордпресс, то некоторые вещи могут быть не совсем очевидны. Если же вы все это прекрасно знаете, хорошо, вспомните о временах, когда и для вас это не было очевидным.

## Статические и Динамические сайты

Старые добрые времена...Сайты из страничек на html – это статические сайты. А ваш новый сайт на вордпресс — это динамический сайт. Динамические сайты хранят контент в базе данных и отображают его так, как настроено в шаблоне. Это мощный способ создания сайтов и на первый взгляд может показаться очень сложным, но гибкость предлагаемая динамическими сайтами стоит того, чтобы с этим разобраться.

Динамические сайты:

**Лучше для дизайнеров** — шаблоны легко позволяют вносить глобальные изменения на сайт

**Лучше для разработчиков** — хранение контента в базе данных позволяет отображать то что нужно тогда когда это нужно

**Лучше для редакторов** — публикация и редактирование контента происходит с инструментами, понятными простым людям

**Лучше для посетителей** — потому что ваш сайт просто фантастичен!

Есть две основные группы вордпресс сайтов:

**wordpress.com** – это блогговая платформа сервис, размещающая ваш сайт у себя по адресу **yoursite.wordpress.com** и в основном используется для создания блогов. Не смотря на то, что можно применять различные шаблоны, привязать свой собственный домен — у вас минимум контроля, не можете добавлять различные плагины и не можете размещать свою рекламу. Зато это бесплатно, просто и понятно. Поэтому подойдет для вашей мамы, но не для вас и для настоящих сайтов.

**wordpress.org** – скачиваемая и устанавливаемая на свой хостинг версия вордпресс, в т.ч. и на русском языке. Кроме этого тут находится форум, Codex (официальная вордпресс документация) а также репозиторий с огромным количеством плагинов и шаблонов. Это версия для настоящих разработчиков и дизайнеров, дающая абсолютный контроль над вашим сайтом, размещенном на вашем хостинге/сервере. Эта книга — про настоящий вордпресс.

## Один шаблон — много страниц

Если вы вебдизайнер или разработчик, вы вероятно уже как минимум знакомы с стилями CSS. Это очень прекрасная концепция. Один файл стилей css управляет дизайном множества страниц. Поэтому если в один прекрасный день вы решите что темно-красный фон сайта стоит сделать немного светлее, вам будет нужно сделать всего 1 маленькую правку, которая приведет к изменению на всем сайте и всех его страницах.

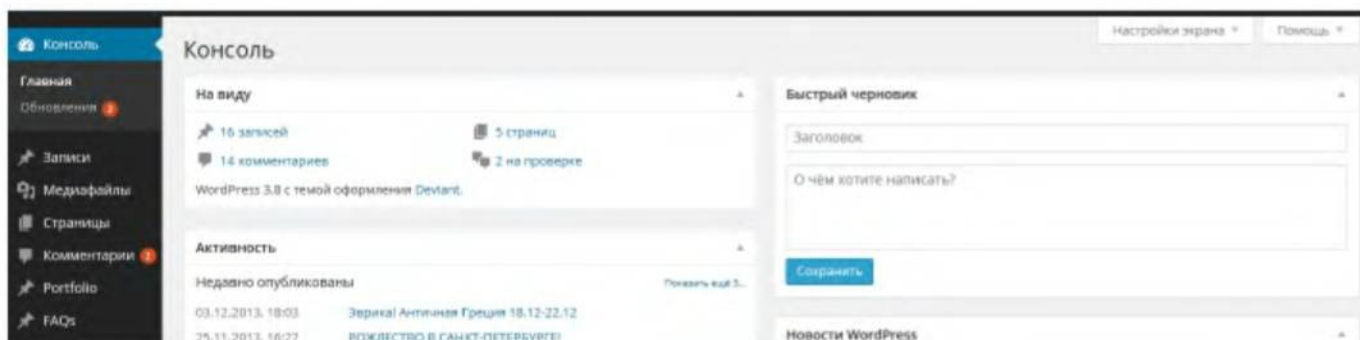
ZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZ





Но несмотря на свою абстрактную сущность, база данных бесспорно самая важная часть вордпресс сайта. Все записи, страницы, комментарии, информация о пользователях, настройки и т. д. хранятся в базе данных. Если ваш сайт испорчен, украден или уничтожен, с резервной копией базы данных вы можете восстановить все с нуля. Вы переставляете вордпресс, находите подходящий шаблон и восстанавливаете из бекапа базу данных- и через минуту у вас снова полнофункциональный сайт.

## Back End



## Front End

Давайте все совместим и посмотрим как фронтенд образуется. Сначала, контент создается в админ части и сохраняется в базу данных. Затем, файлы ядра взаимодействуют с базой данных, чтобы создать сайт для ваших посетителей. И фронтенд — это то, что они видят и то, что делает их счастливыми.

Для начала нам понадобится доступ в интернет, что еще?

Обладать доменным именем все равно что обладать собственным домом. У вас нет хозяина, который бы рассказывал, что вы не можете долбать стены и повесить тут эту картину. В своем доме вы можете делать все что угодно!

**ZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZ**



Т.к. мы работаем с устанавливаемой версией вордпресс, нам нужно место для работы. Вот что значит устанавливаемая — ее нужно куда то установить. Первый шаг — это зарегистрировать доменное имя для нашего сайта. Если вы ранее это не делали, не переживайте, у большинства регистраторов простой и понятный интерфейс, чтобы в этом разобраться.

## Хостинг \ Сервер

Обладать доменным именем лишь пол дела. Теперь нам нужен веб сервер на который доменное имя будет указывать. Т.е. при наборе имени сайта в браузере, будет открываться сайт, который размещен на этом сервере. Зарегистрировать доменное имя и взять хостинг можно как сразу в одном месте, так и в разных компаниях, исходя из ценовой политики и возможностей.

Базовых хостинговых тарифов для малопосещаемых сайтов достаточно с головой, но по мере роста сайта и количества сайтов, стоит подбирать адекватный тариф.

## Редактор кода

Для построения динамических сайтов не нужен WYSIWYG (визуальный) редактор. Мы собираемся запачкать наши ручки в настоящем коде, поэтому нужен правильный редактор.

Для Mac      TextWrangler, jEdit

Для PC Notepad++, jEdit

Для Linux Geany

**FTP клиент**

Для соединения с сервером и заливки файлов с вашего компьютера понадобится ftp клиент.

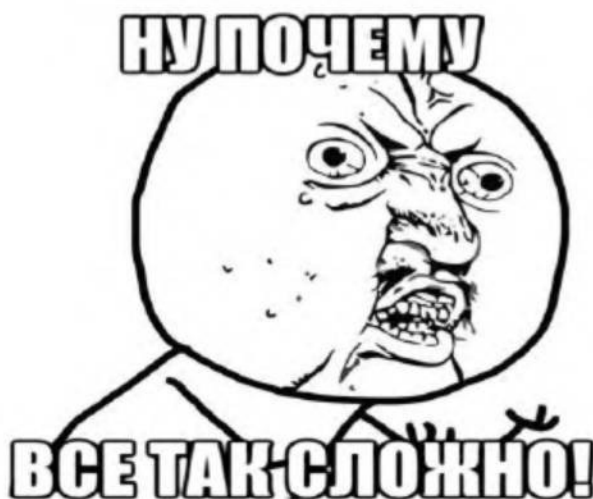
Для Mac Fetch, FileZilla

Для PC      AutoFTP, FileZilla

Для Linux      Geany

Проверяем готовность:

- компьютер с интернетом
- доменное имя
- веб-хостинг
- редактор кода
- ftp клиент
- крутые навыки ← а вот с этим пока проблема...



ZZ

## 2. Настройка Вордпресс

## Знаменитая установка за 5 минут

Установить вордпресс реально просто. Скачать установочный архив, залить файлы, создать базу данных и запустить онлайн установщик с несколькими простыми вопросами. Чем чаще вы это будете делать, тем быстрее это будет получаться. 5 минут говорите? Можно успеть за две с половиной.

Не буду тратить время на расписывание этих шагов, все достаточно подробно указано в Кодексе вордпресса на русском языке [http://codex.wordpress.org/Установка WordPress](http://codex.wordpress.org/Установка_WordPress)

## Где устанавливать?

Когда вы скачиваете инсталляционный архив с [wordpress.org](http://wordpress.org), вы получаете zip файл. После распаковки получаете папку с названием «**wordpress**» набитую файлами. Есть вариант-скопировать содержимое этой папки в корень вашего будущего сайта и начать инсталляцию, но мы предложим совсем другой способ.

Вместо этого переименуйте папку во что-то странное и неочевидное, типа «wrotmnenogi» и загрузите эту папку со всем содержимым в корневую директорию сайта.

Но вы можете возразить:

«Подожди! Я не хочу чтобы мой сайт был по адресу <http://mydomain.ru/wrotmnenogi/> !»

Конечное же не будет. Чтобы сайт загружался из корневой директории, просто скопируйте файлы `index.php` и `.htaccess` из папки `wrotmnenogi` в корень сайта. Если файл `.htaccess` не существует, просто создайте пустой.

Теперь откройте `index.php` и сделайте такую правку:

```
require('./wp-blog-header.php');
на
require('./wrotmnenogi/wp-blog-header.php');
```

Чтобы попасть в админку, нужно будет заходить по урлу

<http://mydomain.ru/wrotmnenogi/wp-admin> , но вордпресс будет работать из корневой директории.

Для этого сразу после инсталляции вордпресса нужно перейти в **Параметры > Общие настройки** и сделать такую настройку

**Адрес WordPress (URL)** указывает на папку с вордпрессом <http://mydomain.ru/wrotmnenogi/>

**Адрес сайта (URL)** указывает на корень сайта <http://mydomain.ru/>

Понятно, что если вы придумаете другую хитрую директорию, то нужно будет указать ее.

**Важно!** Если вы перемещаете вордпресс после инсталляции на уже работающем сайта, то сначала измените **Общие настройки**, и только затем скопируйте index.php и htaccess

### Зачем нужны эти телодвижения?

Преимущества два и они впечатляют:

1. **Безопасность через скрытие.** Любой сканирующий бот на уязвимости не сможет найти ни один файл ядра вордпресс

2. **Чистая корневая директория.** Нет ничего хуже, чем забитая файлами корневая директория.

Разве что взломанный сайт...

zz





ИЗМЕНЕНИЯ.

По умолчанию записывается следующий код

```
# BEGIN WordPress
<IfModule mod_rewrite.c>
RewriteEngine On
RewriteBase /
RewriteCond %{REQUEST_FILENAME} !-f
RewriteCond %{REQUEST_FILENAME} !-d
RewriteRule . /index.php [L]
</IfModule>
# END WordPress
```

Но у ЧПУ есть один небольшой недостаток — они обычно слишком длинные. Поэтому чтобы получить короткий url записи (например для отправки в смс), достаточно нажать на кнопку **Получить короткую ссылку** в редакторе записей. При попытке открыть ее, произойдет редирект на обычный url с пермалинком.

## Какой вариант постоянных ссылок выбрать?

## Вордпресс предлагает несколько вариантов постоянных ссылок

## Общие настройки

- |  |  |
|--|--|
| <input type="radio"/> По умолчанию               | <code>http://worldleaderstravel.com/?p=123</code>                  |
| <input type="radio"/> День и название            | <code>http://worldleaderstravel.com/2013/11/20/sample-post/</code> |
| <input type="radio"/> Месяц и название           | <code>http://worldleaderstravel.com/2013/11/sample-post/</code>    |
| <input type="radio"/> Цифры                      | <code>http://worldleaderstravel.com/archives/123</code>            |
| <input checked="" type="radio"/> Название записи | <code>http://worldleaderstravel.com/sample-post/</code>            |
| <input type="radio"/> Произвольно                | <code>http://worldleaderstravel.com /%postname%/</code>            |

Выбор в большой степени дело вкуса, однако учтите, что потом структуру менять нельзя, особенно при большом количестве проиндексированных записей, чтобы не получить проблем с переиндексацией.

Для написания своей произвольной структуры адреса можно использовать следующие переменные  
 %year%, %monthnum%, %day%, %hour%, %minute%, %second%, %postname%, %post\_id%,  
 %category%, %tag%, %author%.

Например, если вы планируете делать блог в чистом виде с регулярным постингом, то структура `/%year%//%monthnum%//%day%//%postname%/` подойдет лучше всего. Если же записи будут разделяться по тематикам, то будет хорошим вариантом `/%category%//%postname%/`. Ну и в случае неопределенности тематики или нерегулярных публикация остается выбрать урл `/%postname%/`.

При выборе схемы с названием рубрики в урле, после смены рубрики меняется и адрес, поэтому также будет происходить редирект с старого адреса и потребуется время на переиндексацию.

ZZ

## С точки зрения SEO

slug – это специальная строка символов, которая будет представлять каждую запись в ЧПУ, соответственно каждый slug уникален и если будут повторы, то к дублям будут дописываться цифры 2, 3 и т. д. Слаг образуется из заголовка записи (ее титула), поэтому нужно учитывать 2 момента. Во-первых, с помощью специальных плагинов нужно сделать ее транслитерацию в латиницу (плагин Rus-to-Lat). Во-вторых, нужно следить за длиной слага при особенно длинных названиях записей.

**Пример хорошего урл** <http://gizzards.com/how-to-cook-a-gizzard>

**Пример плохого урл** <http://bad-jokes.com/really-dumb-jokes/10/2008/your-momma-is-so-fat-she-sat-on-a-dollar-and-made-four-quarters>

### Пример очень плохого (вордпресс так не делает)

<http://www.surfboardsforyou.com/boards/filename.php?id=F98ZF4&productId=39222&section=wicked&brand=582&template=532>

Slug должен быть

- коротким
- запоминающимся
- понятным и структурированным

## Рубрики и Теги

Как мы знаем, каждая запись и страница имеет свой уникальный ID, slug, и дату публикации. Это помогает с точки зрения организации временного архива, но есть более удобный способ для структурирования. Изначально в вордпрессе нет установленных рубрик и тегов.

Например демозапись по умолчанию «Привет, мир!» находится в рубрике **Uncategorized** и не имеет тегов. Т.о. вам нужно создать свой набор рубрик и добавлять по мере необходимости в дальнейшем.

Давным-давно не было способа для группового назначения или удаления рубрик, поэтому после создания новой нужно было перекликать кучу записей. Сейчас же можно выбрать чекбоксы с списком записей для редактирования, выбрать в выпадающем меню **Изменить**, нажать

**Применить** и получить окошко для массового редактирования

Изменить

Применить

Все даты

Все рубрики

Фильтр

☒ Заголовок

Автор

Рубрики

МАССОВОЕ РЕДАКТИРОВАНИЕ

☒ (без названия)
 ☒ (без названия)
 ☒ 123
 ☒ Newsletter images (do not remove)
 ☒ 567567567
 ☒ 5656y56y
 ☒ Резонанс в українському футболі: центральний матч 12-го туру на лопатях до кінця

Рубрики

☐ Adipiscing elit
 ☐ Aliquet
 ☐ Enim adipiscing
 ☐ Nulla vel diam
 ☐ Sed in lacus ut
 ☐ City
 ☐ Consectetur
 ☐ Donec

Метки

Автор

— Как есть —

Комментарии

— Как есть —

Статус

— Как есть —

Формат

— Как есть —

Отмена

Например можно изменить рубрику для всех выбранных записей, изменить автора, статус и т. п.

////////////////////////////////////





Таким образом для того, кто интересуется автомобилями 2005 года будут доступны 2 обзора. И чем больше будет у вас обзоров, тем больше интересных схожих групп у вас будет получаться.

В целом смысл использования рубрик и тегов в том, чтобы помочь посетителям двигаться по сайту интуитивным путем. Если у вас 350 разных рубрик, то это несколько искажает суть легкости навигации. Мы советуем использовать не более 15 рубрик и не более 100 тегов.

Имейте в виду, что можно в принципе не использовать рубрики и теги. Это очень распространенная практика (записи структурируются на основании календаря) и может быть очень полезной. Это классический пример предпочтения интересов аудитории над техническими возможностями.

Помните, что рубрики и теги это почти одно и то же? Так и есть. Это две различные базовые таксономии, однако вордпресс позволяет создавать дополнительные для задач классификации.

Что мы сделаем — мы создадим новые таксономии для всех этих классификаций. По сути и интерфейсу, они являются тегами.

**Год выпуска: 2013**

Для этого вам нужно зарегистрировать несколько новых таксономий, которые вы создаете добавлением кода в файл шаблона `functions.php`

zz





## Пользователи и Администраторы

Раз уж вы установили вордпресс, значит вы как минимум настроили одного пользователя. Каждый пользователь - это отдельный аккаунт. Когда вы проходили этап инсталляции Вордпресс, вы указали свое собственное пользовательское имя, пароль и емейл.

Это первый, но не единственный пользователь. Вы можете вручную добавлять других пользователей, назначать им права и даже разрешить посетителям регистрироваться на сайте самостоятельно.

Даже если вы отключите или удалите из шаблона возможность отображения вашего имени, оно будет отображаться в RSS ленте. Поэтому позаботьтесь о том, чтобы оно было соответствующим и благопристойным.

Важно! Отображаемое имя может и должно отличаться от реального логина. Это поможет защититься от подбора паролей к вашей учетной записи.



## Добавление новых аккаунтов

Единственное, что вы не можете изменить в созданном аккаунте- это пользовательское имя(логин). Мы рекомендуем не использовать логин admin как небезопасный. Вместо этого придумайте специфическое и необычное.

В меню **Пользователи > Добавить нового** создайте нового пользователя с хорошим паролем (опираясь на индикатор надежности) и ролью администратора. А также придумайте приятное отображаемое имя.

Вордпресс старается быть очень дружелюбным и полезным, поэтому например в окошке авторизации сообщает о неправильном вводе имени пользователя или неправильном пароле. Однако это может быть подсказкой для взломщиков — существует ли данная учетная запись или нет.

Как правило, сразу пытаются подобрать пароль к логину `admin`, выдающемуся по умолчанию до версии 3.0

Мы советуем оставить админский аккаунт активным, но убрать его возможности. Поэтому даже если его взломают — ничего не получают. Для этого нужно создать новый аккаунт с хитрым логином и паролем и админскими правами, перезайти под ним и назначить старому аккаунту `admin` права подписчика. Либо же просто его удалить.

Несколько простых советов относительно сложных паролей:

- не используйте очевидные пароли 1234, qwerty, pass
- используйте сочетание символов и цифр
- минимальная длина 8 символов
- не используйте один пароль на нескольких сайтах одновременно

////////////////////////////////////

## Выбор идеального шаблона

Со всем тем, что вы узнаете о Вордпрессе из этой книги или откуда-то еще, есть надежда, что вы разработаете свой собственный шаблон. Создание своего собственного шаблона позволяет вам добиться чтобы все выглядело и работало именно так, как вы этого хотите, до самой последней мелочи. Но конечно же нет зазорного в том, чтобы использовать готовый шаблон в начале пути. Многие разработчики используют в начале чужие готовые шаблоны, постепенно переделывая их под свои нужды.

## Где искать шикарные шаблоны

Существует два вида шаблонов — бесплатные и платные. Платные шаблоны часто называют премиумными. В общем, платные шаблоны имеют лучший дизайн, лучший программный код и могут предложить приятные опции. Но есть масса шикарных и качественных бесплатных шаблонов, в противовес иногда попадающимся платным, но плохим шаблонам.

Для ориентира бесплатные шаблоны можете посмотреть тут:

Официальный каталог Вордпресс <http://wordpress.org/themes/>

Гугл всемогущий <https://www.google.com/search?q=wordpress+themes>

Места с платными шаблонами:

<http://themeforest.net/>

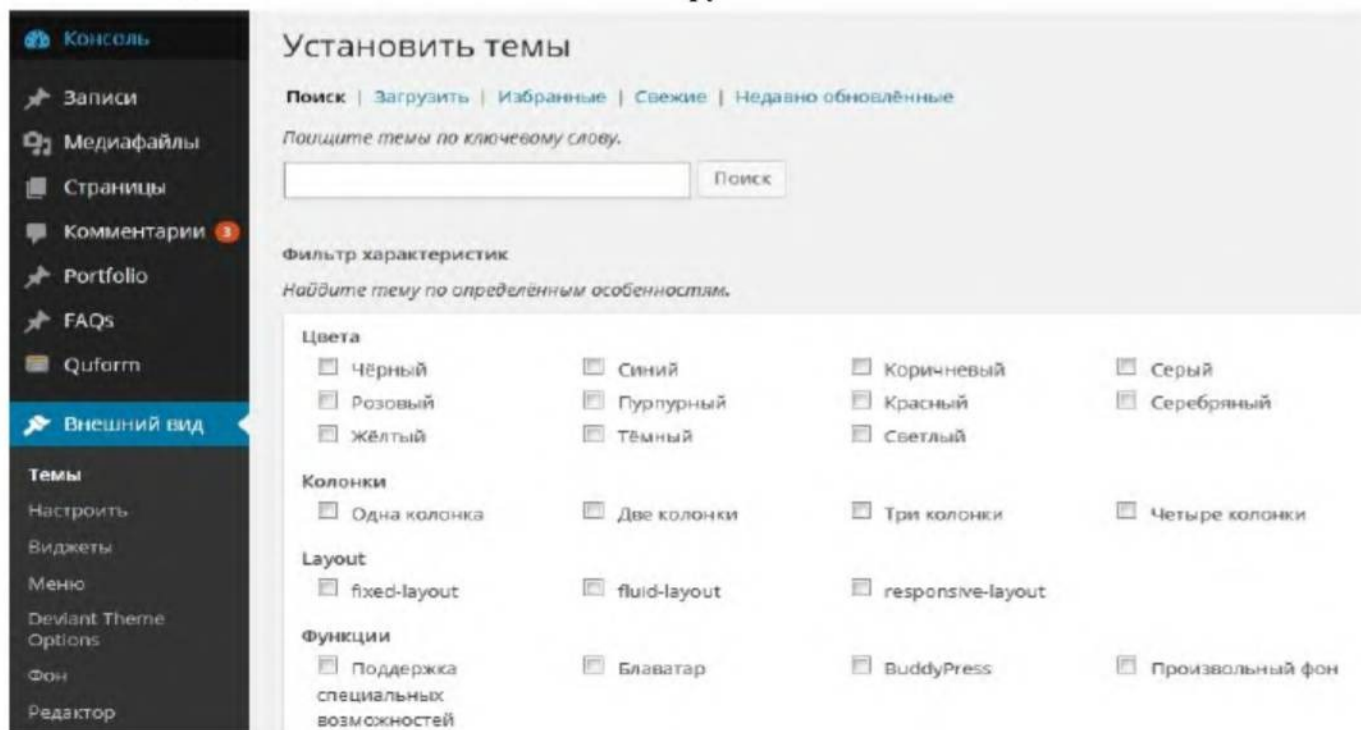
<http://www.woothemes.com/>

<http://www.elegantthemes.com/>

<http://skladchik.com/forums/шаблоны-и-темы/>

## Предпросмотр тем

Многие шаблоны, которые вы найдете в интернете, будут сопровождаться демо, позволяющим вам посмотреть как будет шаблон выглядеть и работать перед установкой. Но даже если у шаблона нет демо, вы можете сделать предпросмотр на своем собственном блоге. Загрузите архив через меню **Внешний вид > Темы > Установить темы > Загрузить**.





В списке загруженных тем вы увидите превьюшки с загруженными темами и краткую информацию о них. Если кликнуть по превьюшке — вы увидите в всплывающем окне как будет выглядеть ваш сайт после установки этого шаблона без какой-либо активации шаблона. Если вас все устраивает — нажимайте кнопку **Сохранить и Активировать**.

### Ключевые критерии при поиске темы

При выборе идеального шаблона для своего сайта вы хотите сфокусироваться на том, как он выглядит и как работает. Будет ли он круто смотреться в браузере? Подаст ли он контент на блюде с голубой каемочкой? Заставит ли он вас закричать и пуститься в пляс? Если это так, значит вы нашли нужный шаблон. Нет причин довольствоваться чем то меньшим, чем просто идеальным, особенно с учетом огромного выбора шаблонов. Но если вы нашли что-то близкое к идеалу, требующее небольших доработок, то помните, что гораздо легче изменить цвета и размер шрифта, чем структуру шаблона и функционал.

А это значит, что нужно определиться с ключевыми критериями поиска идеального шаблона:

## Навигация

Посмотрите на навигацию и меню, предлагаемые шаблоном. Затем подумайте о том, как ваше видение навигации будет лучшим образом работать на вашем сайте. Будут ли видны рубрики? Важны ли страницы? Нужно ли выпадающее меню? Есть ли место для построения дополнительной навигации при необходимости?

## Настройки шаблона

Некоторые шаблоны идут с возможностью менять настройки, реализованными в дополнительном разделе опций в админ части для изменения шаблона. Эти настройки могут быть самые разные, от простых, типа изменения цвета, до сложных, типа интеграции социальных виджетов в шаблон. Иногда набор настроек может быть просто впечатляющим, избавляющим от необходимости что-то править руками в коде или css.

## Поддержка виджетов

Во время разработки шаблона, дизайнер может обозначить некоторые участки как «готовые для виджета». Вы можете быстро и легко добавлять нужные виджеты в эти места, настраивая их появление и функционал. В основном места для размещения виджетов находятся в сайдбаре, но довольно часто их размещают и в футере. В админке в меню **Внешний вид > Виджеты** вы можете брать и перетаскивать нужные виджеты в нужные места, меняя их расположение и порядок. Если вы представляете основные преимущества с функционалом виджетов, убедитесь, что шаблон их поддерживает.

### Дополнительный функционал

Когда речь идет о дополнительном функционале, то нет никаких ограничений, если речь идет о шаблонах Вордпресс. Некоторые темы просто сводят с ума своими запредельными возможностями. Например, шаблон может иметь встроенный сервис фотошаринга, как Flickr, приложение анализа статистики посетителей, как Google Analytics, или интерфейс работы с базами данных, как phpMyAdmin.

## Фреймворки

Фреймворки приобретают все большую популярность. Это некая платформа, требующая

ZZ

дополнительного обучения, но как только вы с ней подружитесь, она значительно облегчит процесс создания шаблонов, предлагая все ключевые возможности и настройки используемые в шаблоне. Стоит поверхностно упомянуть самые популярные фреймворки: Thematic, Hybrid, Thesis.

И один раз разобравшись с фреймворком, можно с легкостью менять и настраивать любые шаблоны, сделанные на нем.

## Исходные файлы

Даже если вы нашли шаблон, который вам очень нравится, вы возможно захотите внести некоторые правки. Если это потребует изменения графических элементов шаблона, то очень хорошим будет получить и исходные файлы. Как правило это Photoshop файлы или векторные исходники, иконки, изображения в высоком разрешении и т. д.

## Приступая к работе с плагинами

Система плагинов — это часть волшебства Вордпресс. Они расширяют и улучшают то, что вордпресс способна делать очень специфическим путем. Любой может написать плагин для Вордпресса, и даже включить его в официальную библиотеку плагинов, известную как Репозиторий плагинов <http://wordpress.org/plugins/>

## Установка и Активация плагинов

Среди файлов и папок на вашем сервере есть особая папка `/wp-content/plugins/` которая содержит все плагины вашего сайта. Проверенный и истинный метод установки плагинов — это скачать его на свой компьютер и затем загрузить его на сервер в эту папку. После этого плагин появится в Админке на странице плагинов. По умолчанию, новые плагины неактивны, поэтому вам нужно вручную активировать их, чтобы они начали работу.

Вы можете в любое время деактивировать плагин. Но учитывайте, что плагины обладают серьезной силой. Каждый раз активируя или деактивируя плагин, вам следует провести небольшое расследование на вашем сайте, чтобы убедиться, что все выглядит и работает так как надо.

Также плагины можно устанавливать непосредственно через админку. Просто зайдите в **Плагины > Добавить новый**. И выполнить поиск по названию плагина в репозитории. В случае непосредственной загрузки плагина из репозитория, нужно убедиться что папка с плагинами открыта для записи на сервере. В идеальном мире, права на запись в директорию абсолютно безопасны, но в враждебной среде современного веба вам стоит определенно задуматься о том, нужны или нет такие права.

Бонусом к загрузке плагина из репозитория вордпресс идет тот факт, что вы можете быть уверены в безвредности этого плагина. Вы конечно можете натворить разных дел с плагином, да и плагины иногда перестают работать как надо, но в основном плагин полученный из репозитория обычно не вредит сайту и не раздражает посетителей, т. к. все плагины проходят проверку.

А вот про плагины доступные для скачивания с любых других сайтов нельзя так сказать, поэтому хорошо подумайте о том, откуда вы берете этот плагин.

## Разница между Деактивацией и Удалением

Деактивация активного плагина выключает его функционал, но не удаляет его физически из

////////////////////////////////////

директории плагинов. Вы можете иметь тысячи деактивированных плагинов, которые ничего не делают кроме того, что занимают место в папке. Фактическое удаление(анинсталлирование) приводит к стиранию всех файлов, связанных с плагином, и если возможно, то к возврату всех изменений, которые плагин мог сделать в базе данных.

В процессе инсталляции и работы, многие плагины автоматически вставляют контент в базу данных вордпресс. Такие плагины могут добавлять новые таблицы или поля, изменять существующие данные и хранить информацию необходимую для работы. Однажды сделанные, эти изменения продолжают существовать и после удаления плагина.

Хорошо сделанный плагин предлагает полную возможность для анинсталлирования, которая выполняет работу по очистке изменений в базе данных. Плагины без такой возможности должны быть очищены вручную. Если возникла ситуация с необходимостью полного удаления плагина, проверьте, что вы точно знаете что вы будете делать перед тем, как начинать изменять базу данных. И не забудьте сделать бекап на случай, если что-то может пойти не так.

### Полезная фишка.

Если вы деактивируете плагин, вызов которого прописан в шаблоне, то сайт перестанет нормально работать. Чтобы этого избежать вместо непосредственного вызова функции плагина нужно делать проверку, есть ли такой плагин или функция

```
<?php if(function_exists('get_poll')) { get_poll(); } ?>
```

## Рекомендованные плагины

Природа плагинов в том то, что они обеспечивают WordPress дополнительными функциональными возможностями, которые не нужны на каждом сайте. Вместо того чтобы пытаться втиснуть миллион функций в ядро WordPress, прикладной функционал приложений оставлен для потрясающих разработчиков в сообществе WordPress. Разработчики видят какую то необходимость (или возможность), создают плагин и отдают его пользователям. Если плагин достаточно популярен и есть смысл интегрировать его в ядро WordPress, волшебники за кулисами видят ситуацию и выполняют.

Тем не менее, остается ряд первоклассных плагинов, которые по какой-то причине до сих пор не воплотились в ядре. Вот некоторые из лучших, которые мы находим полезными для любого типа WordPress сайта.

**VaultPress** <http://vaultpress.com/>

Это плагины и платный сервис от Automattic, создателя Вордпресс. После установки весь ваш блог перемещается на «облачный сервер», включая все файлы и БД. Сервис включает в себя много услуг, в том числе резервное копирование, сканирование на различные проблемы безопасности и т. п.

**Art Direction** <http://wordpress.org/plugins/art-direction/>

Плагин позволяет вставлять дополнительный код (обычно CSS или JavaScript) непосредственно в записи и страницы, что позволяет добиться разного стиля и отображения для разных записей. Иметь полный контроль над стилем и функционалом отдельных страниц создает впечатляющие дизайнерские возможности.

**W3 Total Cache** <http://wordpress.org/plugins/w3-total-cache/>

Ускоряет производительность сайта комбинированием различных технологий: файловое кеширование, кеширование запросов к базе данных, сжатие и комбинирование скриптов и стилей,

ZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZ



интеграция распределенного хранения данных и т. п.

**WP-DBManager** <http://wordpress.org/plugins/wp-dbmanager/>

Нет ничего более важного для сайта на вордпрессе, чем его база данных. Если весь сервер будет со всеми данными уничтожен, но у вас будет последняя копия БД, то все восстановить будет несложно. Этот плагин позволит настроить автоматический бекап, оптимизацию, восстановление БД и даже отправку резервной копии на емейл.

**Posts Per Page** <http://wordpress.org/plugins/custom-post-limits/>

Есть только одна настройка для количества отображаемых записей на странице **Параметры — Чтение — На страницах блога отображать не более...** Но что делать в случае необходимости показывать только одну запись на главной? Это означает, что страница с архивами, результатом поиска и т. д. так же будет отображать одну запись, что глупо.

Этот плагин позволит индивидуально настроить количество отображаемых записей на каждом типа страниц, включая результаты поиска, рубрики, архивы, теги и т. д.

**Post Editor Buttons** <http://wordpress.org/plugins/post-editor-buttons/>

Если вам не хватает стандартного набора кнопок в редакторе записей для форматирования стилей, то вы можете создать свои собственные с нужным функционалом, например для добавления тегов `<h3>`, `<h4>` и `<html>`.

**WordPress SEO by Yoast** <http://wordpress.org/plugins/wordpress-seo/>

Это единственный seo-плагин, который вам понадобится для дальнейшей оптимизации Вордпресса. В нем есть все нужное: анализ содержимого страницы, оптимизация фидов, добавление хлебных крошек и простой импорт из других seo плагинов, типа All in One SEO. Кроме этого доступны тонны фишек и настроек, даже создание XML карты сайта.

**Google XML Sitemaps** <http://wordpress.org/plugins/google-sitemap-generator/>

Если же не хочется использовать предыдущий плагин, то для создания карты сайта можно воспользоваться этим плагином — установка, настройка и вуаля, автоматически создается XML карта сайта, которая соответствует требованиям Google и автоматически обновляется после каждого редактирования или создания записей. Кроме этого, после пересоздания карты происходит пинг поисковика для уведомления о изменениях.

**Clean Notifications** <http://wordpress.org/plugins/clean-notifications/>

Стандартное уведомление по емейлу о новых комментариях несколько уродское — это простой текст, содержащий кучу ссылок. Благодаря этому плагину уведомление получает некоторое форматирование, помогающее выглядеть тексту более читаемым и понятным.

### Subscribe To Comment

<http://wordpress.org/plugins/subscribe-to-comments/>

Этот плагин позволяет легко оставаться на связи в обсуждении в комментариях, добавляя галочку в форму комментирования. После отметки чекбокса и отправки комментария вы будете получать уведомления на емейл о всех новых комментариях опубликованных к этой записи. И самое главное, сможете отписаться от них в любое время.

Что дальше? На данный момент у вас есть установленный и настроенный Вордпресс, заряженный плагинами, и пришло время чтобы погрузиться в сердце сайта — в его шаблон.



ZZ

### 3. Анатомия шаблонов Wordpress

## Понимание файлов шаблона

Давайте посмотрим поблизи на то, как построены и как работают шаблоны.

Все шаблоны разные! Понятно, что все шаблоны выглядят по разному. Но если вы скачаете 5 разных шаблонов и сравните их файлы, вы увидите существенные разные наборы файлов. Есть несколько обязательных файлов, которые должны быть в каждом шаблоне и некоторое количество файлов, которые скорее всего будут совпадать, но в остальном масса отличий.

Например, некоторые шаблоны могут идти с специальной страницей для архивов, потому что структура отображения архивов в этих шаблонах отличается от стандартной. В других шаблонах может отсутствовать страница с результатами поиска по сайту, т. к. нужный функционал встроен в `index.php`

Но в каждом шаблоне должны обязательно присутствовать два файла, `index.php` и `style.css`, являющиеся ядром шаблона и абсолютно необходимые для нормальной работы. Фактически, без этих файлов вордпресс не опознает шаблон как шаблон, даже если он будет находиться в папке `themes`. И формально, можно сделать шаблон, состоящий всего из двух этих файлов. Это будет простой, но функциональный шаблон. Этого может хватить для некоторых задач, но в общем вы вероятно используете вордпресс как раз из-за больших возможностей, которые он предлагает.

Ознакомиться с полной иерархией страниц в вордпрессе можно тут

[http://codex.wordpress.org/Иерархия\\_шаблонов](http://codex.wordpress.org/Иерархия_шаблонов) , но кратко структура выглядит так:

### Обязательные (ключевые) файлы:

index.php обычно это главная «домашняя» страница сайта, но может быть и любой другой  
style.css содержит информацию о стилях шаблона и нужен для его работы, даже если не  
используется

### Стандартные файлы:

404.php страница ошибки для несуществующих адресов

archive.php страница для отображения архивов по выбранной дате, месяцу, году, рубрике, тегу или автору

comments.php отображает комментарии, пингбеки, трекбеки и форму комментариев

footer.php включается в нижнюю часть каждой страницы, содержит коды статистики и копирайты

front-page.php отображает содержимое «домашней» страницы сайта

header.php включается в верхнюю часть каждой страницы, содержит секцию head, навигацию, меню

home.php отображает содержимое «домашней» страницы сайта

page.php шаблон для Страниц — документы отображение статического, неблогового контента

screenshot.png изображение картинке превью вашего шаблона в списке других шаблонов

search.php страница результатов поиска по сайту

sidebar.php включает боковой контент страниц и записей, используется для виджетов

single.php шаблон для Записей, отображает полный контент записи с комментариями

### Специальные файлы:

archives.php страница для отображения всех архивов по рубрикам и месяцам

functions.php файл для добавления специальных возможностей вашего шаблона

image.php отображает отдельно взятые изображения вашего сайта

links.php специальная страница отображения ссылок из блог-ролла

loop.php появился в новых темах, используется для обслуживания настраиваемых и множественных повторяющихся блоков контента записей

ZZ



Большинство шаблонов использует обязательные ключевые и стандартные файлы. Стандартные файлы выполняют все, что нужно от сайта вам и посетителям.

Специальные файлы предлагают нечто, находящееся за пределами стандартных задач. Например, `image.php` может пригодиться, если вы решите использовать стандартную библиотеку медиа для управления картинками, фильмами и т. п., вставляя ссылку на эту специальную страницу. Вы сможете включать дополнительную информацию о авторских правах, копиях и т. п. Не на каждом сайте это необходимо, поэтому их и называют специальными.

Есть еще один файл, мусорный, это `comments-popup.php`, названный так потому что он больше не используется в шаблонах. Даже и не пытайтесь понять зачем он, не будем уделять ему время.

Все эти файлы не работают каждый по отдельности, они взаимодействуют. Например, `index.php` сначала вызовет `header.php`, затем использует `sidebar.php` и в финале вызовет `footer.php`. А `sidebar.php` может в себя включать вызов `searchform.php` для отображения формы поиска. И понятно, что `header.php` будет в секции `<head>` подключать `style.css`.

Такая модульная и динамическая система позволяет раскрыть потенциал Вордпресс. Для тех, кто имел дело только с статическими html сайтами, это крайне привлекательно. Представьте простоту изменения меню, достаточно внести нужные правки в `header.php` и сразу на всем сайте будет отображаться новая навигация. А если пойти еще дальше, то панель меню можно генерировать встроенной функцией вордпресса, т. е. все изменения будут выполняться автоматически по мере изменения структуры сайта. Как только вы добавите новую страницу — в меню сразу появится новая кнопка. Это очень круто — выполнять изменения, обновления и управление сайтом так легко и не задумываясь о мелочах.

А вот полезная информация о том, как правильно оформить свой шаблон.

Во-первых, создайте `screenshot.png` с размерами 600px на 450px, в идеале если это будет превью внешнего вида шаблона, а не просто абстрактный логотип.

Во-вторых, добавьте служебное описание в начале файла style.css

```

/*
Theme Name: Theme Name
Theme URI: http://your-website.com/cool-theme/
Description: Totally awesome WordPress theme by <a href="http://your-website.com/">Yours Truly</a>
Version: 1 (WP2.8.4)
Author: Your Name
Author URI: http://your-website.com/
Tags: super, awesome, cool, sweet, potato nuggets
*/

```

Теперь в списке установленных шаблонов он будет выглядеть просто замечательно.

## Понимание различных видов страниц

На самом деле различных типов страниц (того, что вы видите на сайте) не много:

**Главная/Домашняя страница** — обычно вызывается по корневому адресу вашего сайта, типа site.ru

**Запись** — отображает одну запись в полном виде, принадлежит какой либо рубрике

**Страница** — страница с статическим контентом, не находящаяся в общей ленте записей и не принадлежащая рубрике

**Персональная страница** — страница с настраиваемыми, нестандартными параметрами

**Результаты поиска** — отображает список записей, подходящих под поисковый запросов

**Архив** — отображает список записей выбранной рубрики, тега, даты или автора

ZZ

## Идеология Страниц

Смысл Страниц в том, чтобы отображать некий неизменный контент, который не относится к рубриками, тегам и находится вне хронологии публикуемых записей, страницы не отображаются в RSS ленте. Поэтому шаблон, используемый для отображения Страниц несколько отличается от стандартного, используемого для записей.

Например в файле `page.php` нет необходимости показывать информацию о авторе, дате, рубрике, выводить поле для комментирования и т. п. Вместо этого может понадобится другой дополнительный функционал, типа навигации хлебными крошками или отображения иерархии страниц.

## Использование записей

Файл `single.php` отвечает за отображение единичной записи. Часть его кода будет отвечать за отображение рубрик и метаданных записи, другая часть за отображение комментариев и формы комментирования. Возможно вы захотите чтобы запись выводилась шире или уже чем стандартная ширина.

Давайте сравним:

Страница

Запись



Обычный заголовок  
Нет комментариев  
Свой собственный сайдбар  
Навигация  
Главная — Страница



[Расширенный заголовок](#)  
[Комментарии](#)  
[Свой собственный сайдбар](#)  
[Навигация](#)  
[Главная](#) — [Рубрика](#) — [Запись](#)  
[Дополнительная информация о авторе,](#)  
[дате, тегах и т. п.](#)

ZZ

## Многообразие архивов

Есть много типов архивов, и только один файл `archive.php` заряженный способностью отображать их все. Когда вы смотрите выбранную рубрику, тег, автора или архив на основе выбранной даты, Вордпресс генерирует список записей, их контент и их разметку в соответствии с кодом, содержащимся в `archive.php`

И сразу появляется вопрос «откуда вордпресс знает какой шаблон использовать?» Вы уже знаете, что этот файл не обязательный. Значит ли это, что если в шаблоне нет этого файла, то будет показана пустая страница? Конечно нет, вордпресс последовательно перебирает файлы по своей иерархии, чтобы выбрать подходящий из обязательных файлов, которые, как следует из названия, обязательно будут. Поэтому все пути поиска сойдутся на файле `index.php`, если другие подходящие не будут найдены. Надеюсь, вы теперь понимаете насколько он важен.

И точно также мы можем пройти в обратном направлении. Например мы хотим создать специфический шаблон для просмотра рубрики с id 13. Для этого всего-навсего создадим файл `category-13.php` и при запросе этой рубрики вордпресс автоматически будет его использовать.

Давайте еще раз взглянем на иерархию элементов шаблона

PAGE TYPE	tries first > tries next > tries last
404	404.php > index.php
SEARCH	search.php > index.php
TAXONOMY	taxonomy-{tax}-{term}.php > taxonomy-{tax}.php > taxonomy.php > archive.php > index.php
HOME	home.php > index.php
ATTACHMENT	{mime-type}.php > attachment.php > single.php > index.php
SINGLE	single-{post-type}.php > single.php > index.php
PAGE	{custom-template}.php > page-{slug}.php > page-{id}.php > page.php > index.php
CATEGORY	category-{slug}.php > category-{id}.php > category.php > archive.php > index.php
TAG	tag-{slug}.php > tag-{id}.php > tag.php > archive.php > index.php
AUTHOR	author-{author-nicename}.php > author-{author-id}.php > author.php > archive.php > index.php
DATE	date.php > archive.php > index.php
ARCHIVE	archive.php > index.php

Если нужно отобразить страницу с ошибкой 404, вордпресс будет искать файл 404.php, если не найдет- то использует index.php

Если нужно отобразить архив для тега seo с id5, то будет искать файл tag-seo.php, затем tag-5.php, затем просто tag.php и т. д.

ZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZ







```
<?php wp_enqueue_script('jquery'); ?>
```

Такой вызов имеет массу преимуществ.

1. Это легко, т. к. не требует контроля над местоположением библиотеки и ее версий.
2. Это информативно, т. к. вордпресс будет знать, что требуемый файл успешно загружен.

Если же вы скачаете свою копию jQuery и вставите на нее ссылку в шаблоне, вордпресс даже знать об этом не будет. А если затем активируете плагин, который будет использовать jQuery, он загрузит свою копию, что приведет к бардаку и достаточно часто к конфликтам. Использование `wp_enqueue` позволит избежать дублирования и конфликтов.

Но если вы написали свой собственный скрипт и на 101% уверены, что он уникален и неповторим, то нет необходимости включать его в вордпресс. Его как раз можно загружать и вызывать стандартным способом

```
<script type="text/javascript" src="<?php bloginfo('template_url'); ?>/js/myscript.js"></script>
```

Как видите, в данном примере мы использовали другой параметр функции `bloginfo('template_url')`, чтобы получить точный url адрес расположения шаблона, в подпапке которого находится наш скрипт.

Давайте теперь продемонстрируем, как можно подключать специфический CSS код для определенного архива. Для этого используем логику вордпресс для определения типа загружаемой страницы

```
<?php if (is_page_template('page-archives.php')) { ?>
<link rel="stylesheet" href="<?php bloginfo('template_url'); ?>/css/archives.css" type="text/css"
media="screen" />
<script type="text/javascript" src="<?php bloginfo('template_url'); ?>/js/archives.js"></script>
<?php } ?>
```

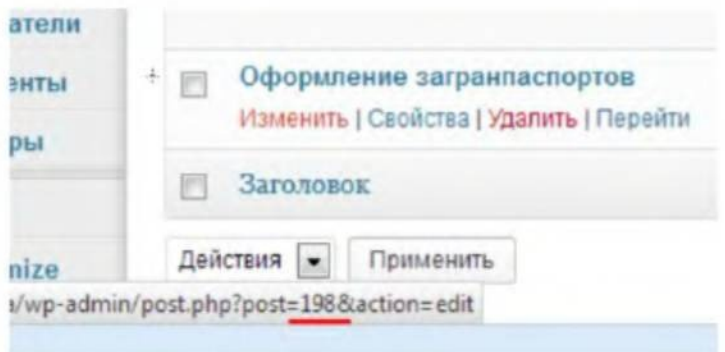
Это работает так — если мы находимся специальной странице архива, описываемой файлом шаблона `page-archives.php`, то подключить два файла. На всех остальных страницах эти файлы загружаться не будут.

А вот пример кода для страницы с определенным id равным 5

```
<?php if (is_page("5")) { ?>
<link rel="stylesheet" href="<?php bloginfo('template_url'); ?>/css/archives.css" type="text/css"
media="screen" />
<script type="text/javascript" src="<?php bloginfo('template_url'); ?>/js/archives.js"></script>
<?php } ?>
```

По непонятной причине id записей, страниц и рубрик явно не отображается в вордпрессе. Чтобы получить эти значения, например id страницы, нужно навести курсор над ссылкой **Изменить** в списке страниц, и в панельке браузера увидим урл содержащий айди этой страницы, в данном примере это 198.

Точно также можно узнать айди рубрики, записи, тега и т. п.



ZZ



## Разберемся с `wp_head()`

Обязательная для всех шаблонов, функция `wp_head()` просто сообщает вордпрессу «вот прямо тут находится секция `<head>`, дальше ты уже знаешь что нужно делать»

Это своего рода метка, по которой сам вордпресс, а также плагины и различные функции добавляют нужный код в хидер.

Например, вы активировали функционал XML-RPC в настройках. Для его работы необходимо добавить определенные `<link>` элементы в `<head>`. В результате работы функции `wp_head()` они будут вставлены в код.

Точно также в хед будет вставлен код функции `wp_enqueue_script`, которая проанализирует требуемые скрипты и подключит только те, которые требуются. А плагины с ее помощью добавляют нужные для работы яваскрипты и файлы стилей.

## Теги шаблона

Есть одна функция, которая может быть в определенных случаях весьма полезна, это `<?php bloginfo(); ?>`

Из нее можно получить много интересного, например такие соответствующие данные:

```
admin_email = jeff@digwp.com
atom_url = http://digwp.com/home/feed/atom
charset = UTF-8
comments_atom_url = http://digwp.com/home/comments/feed/atom
comments_rss2_url = http://digwp.com/home/comments/feed
description = Take Your WordPress Skills to the Next Level!
url = http://digwp.com/home
html_type = text/html
language = en-US
name = Digging Into WordPress
pingback_url = http://example/home/wp/xmlrpc.php
rdf_url = http://digwp.com/home/feed/rdf
rss2_url = http://digwp.com/home/feed
rss_url = http://digwp.com/home/feed/rss
siteurl = http://digwp.com/home
stylesheet_directory = http://digwp.com/home/wp/wp-content/themes/largo
stylesheet_url = http://digwp.com/home/wp/wp-content/themes/largo/style.css
template_directory = http://digwp.com/home/wp/wp-content/themes/largo
template_url = http://digwp.com/home/wp/wp-content/themes/largo
text_direction = ltr
version = 2.8.5
wpurl = http://digwp.com/home/wp
```

Подставляя нужный аргумент, можно получить значение, например `bloginfo('template_url');` вернет ссылку на папку с шаблоном и используется вот так

```
<link rel="stylesheet" href="<?php bloginfo('template_url'); ?>/css/archives.css" type="text/css" media="screen" />
```

Для чего это нужно? Для того, чтобы не прописывать в шаблоне абсолютные ссылки и жесткие параметры, что показывает гибкость шаблона при переносе с одного домена на другой, с сохранением функциональности.

**ZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZ**

Но если стандартных параметров мало, можно использовать функционал произвольных полей. Это атрибуты, которые можно назначать к записям и страницам, а затем получать значения в любом месте шаблона. Кроме этого можно создать глобальные произвольные поля, которые можно использовать не только на страницах записей и страниц.

Например, вы хотите зарабатывать на партнерской программе с своей рефссылкой. Конечно можно руками прописать аффилиатный код везде, где нужно, но что делать, если в один прекрасный день этот код нужно будет поменять?

Вставьте в `functions.php` следующий код (пример для Амазона)

```
<?php add_action('admin_menu', 'add_gcf_interface');
function add_gcf_interface() {
add_options_page('Global Custom Fields', 'Global Custom Fields', '8', 'functions',
'editglobalcustomfields');
}
function editglobalcustomfields() { ?>
<div class="wrap">
<h2>Global Custom Fields</h2>
<form method="post" action="options.php">
<?php wp_nonce_field('update-options') ?>
<p><strong>Amazon ID:</strong><br />
<input type="text" name="amazonid" size="45"
value="<?php echo get_option('amazonid'); ?>" />
</p>
<p><input type="submit" name="Submit" value="Update Options" /></p>
<input type="hidden" name="action" value="update" />
<input type="hidden" name="page_options" value="amazonid" />
</form>
</div>
<?php } ?>
```

Укажите и сохраните в настройках свой реф-айди с амазона.

Теперь где угодно в шаблоне можно с помощью кода `<?php echo get_option('amazonid'); ?>` получить значение рефайди и вставить в партнерскую ссылку на товар.

## Циклы в Вордпресс

Цикл, хорошо известный как петля(loop) это базовое понятие того, как работает Вордпресс. В самом общем виде представить это можно так:

```
<?php // The Loop
if (have_posts()) : while (have_posts()) : the_post();
...
endwhile; else:
...
endif;
?>
```

Сначала проверяется, есть ли в наличии посты\записи для отображения. Если это так, то начинается цикл. Функция `the post()` определяет действия, которые будут выполнены для

////////////////////////////////////

Пока что звучит несколько сумбурно, поэтому давайте рассмотрим более детально.

Есть ли опубликованные записи?  
Начало цикла, работает пока есть записи

Ссылка на запись  
В качестве текста- название записи

Произвольное поле, содержащее id  
картинки-превью записи

Мета-информация: дата публикации  
имя автора

Содержимое записи  
Теги записи  
Название категории записи  
Ссылка на форму комментирования

Конец цикла  
Ссылка на более старые записи  
Ссылка на ранние записи  
Нет записей?  
Сообщаем об этом  
Закончили.

На самом деле цикл хорошо знает, что делать. Достаточно добавить эту строчку кода перед циклом

Если добавить эту строку в файл `single.php`, то увидим что-то типа такого

Что означает, что из базы данных взять и отобразить что-то, опубликованное в феврале 2011 года и называется Version 3 update.

```
posts_per_page=10&what_to_show=posts&orderby=date&order=DESC&category_
name=javascript
```

ZZ



////////////////////////////////////

Это лишь малая часть и приведены для демонстрации принципа работы функций в зависимости от того, где они находятся.

## Комментарии

Комментарии, это вероятно одна из причин, по которой стоит использовать вордпресс. Это мощная система с огромным количеством возможностей. Более глубокое погружение в комментарии произойдет в главе 7, но т. к. функционал комментариев часть анатомии шаблона вордпресс, то познакомимся с ними уже сейчас.

## Файл comments.php

В общем, вордпресс разложил по полочкам функционал комментирования в один файл, который отвечает за следующее:

- Вся необходимая логика для отображения комментариев
- Отображение дополнительной информации
- Отображение формы комментирования

Чтобы выполнять все это, `comments.php` нуждается в комплексной системе логики. Работает она примерно так:

**Защищена ли запись (требуется пароль)?**

Да — вывести предупреждение и остановиться

Нет — продолжать

### Есть ли какие-нибудь комментарии?

Да — вывести комментарии и продолжать

Нет — продолжать

## Открыто ли комментирование?

Да — продолжать

Нет — вывести предупреждение и остановиться

### Требуется ли регистрация?

Да — вывести предупреждение и остановиться

Нет — вывести форму комментариев

И это только базовая логика. Дополнительная логика используется в различных функциях для использования различных условий и настроек.

Например, форма комментариев отображается в виде по умолчанию или при наличии комментария, на который можно написать ответ

```
<?php comment_form_title('Оставить комментарий', 'Комментарий к %s'); ?>
```

### Выборочное включение для различных ситуаций

Красота использования всей этой магии, спрятанной в один файл, заключается в том, что вы можете добавить полный функционал комментирования в любом месте простой вставкой

```
<?php comments template(); ?>
```

И точно также, простым удалением этой строки из файла шаблона вы убираете комментирование с

[illegible]

Данный метод можно распространять на что угодно, например на сайдбар. Допустим вы хотите, чтобы одни записи выводились с сайдбаром, другие- без. Создаете произвольное поле для записи с названием `noSidebar` и заполняете для нужных записей.

```
<?php
if (!get_post_meta($post->ID, "noSidebar", true))
{
get_sidebar();
} ?>
```

Теперь на всех страницах, с установленным полем noSidebar, сайдбар отображаться не будет.

Сайдбары — это вездесущий элемент дизайна, функционал которых встроен в вордпресс для удобного обслуживания. В любой теме можно вставить `<?php get_sidebar(); ?>` там, где вы хотите чтобы появился сайдбар. Эта функция обратится к `sidebar.php` и разместит ее содержимое.

```
<?php get_sidebar('secondary'); ?>
```

Не смотря на весь специфический функционал, сайдбары ни в коем случае не являются обязательными. Некоторые сайты прекрасно обходятся без них. Не создавайте `sidebar.php` и не вызывайте эту функцию.

Сайдбары нужны для разных штучек. Вебсайты конечно же наполнены разными штучками. Да, есть главный контент, навигация. Но кроме этого есть еще куча разных вещей. Мы не хотим размещать это все в футере, потому что им там будет одиноко и никто их не найдет внизу сайта. Поэтому разместим их сбоку. Речь идет о альтернативной навигации, вспомогательном контенте, маленьких подписных формах, описательном тексте, рекламе, ссылках, картинках котиков и т. д.

Сайдбары традиционно уже более чем просто дополнение и могут размещаться не только слева или справа от основного контента, но могут появиться и в любом другом месте. Это все зависит от структуры и стиля вашего шаблона.

Частично требования к вашему сайту должны диктоваться тем, что вы собираетесь поместить в сайдбар. Просто вставить из-за того, что вы можете это сюда вставить или потому, что видели на другом сайте такое облако тегов, поэтому давайте я тоже его сюда засуну — неправильно. Облако тегов на вашем сайте должно появиться только потому, что вы будете использовать много тегов и

////////////////////////////////////





[http://codex.wordpress.org/Template\\_Tags/wp\\_list\\_categories](http://codex.wordpress.org/Template_Tags/wp_list_categories)

- Блогролл

В админке вордпресс до версии 3.5 существовал раздел Ссылки, позволяющий добавлять ссылки на любые сайты. Такой блок ссылок называется Блогролл. Сейчас этот блок отключен за ненадобностью, но если он вам очень нужен, то можете его активировать с помощью плагина Link Manager <http://wordpress.org/plugins/link-manager/>

Для вывода в сайдбаре созданного блока ссылок воспользуйтесь следующим кодом

```
<?php wp_list_bookmarks(array(
    'orderby' => 'name',
    'order' => 'ASC',
    'limit' => -1,
    'title_li' => __('Всеякие Ссылки'),
    'title_before' => '<h2>',
    'title_after' => '</h2>',
)); ?>
```

Полный набор параметров тут [http://codex.wordpress.org/Template\\_Tags/wp\\_list\\_bookmarks](http://codex.wordpress.org/Template_Tags/wp_list_bookmarks)

- Редактируемый текстовый блок

Одна из вещей, которую вы скорее всего захотите добавить в сайдбар — это редактируемый текстовый блок, который можно править через админку вордпресс. Вот как можно это сделать:

Первый — это глобальные произвольные поля, рассматриваемые ранее. Второй — это виджеты.

Виджеты — это способ позволить управлять различными участками через админку, вместо того, чтобы редактировать файлы шаблонов. Один из самых популярных, это простой текстовый виджет. Если ваш сайдбар поддерживает виджеты, то вы можете просто захватить и перетащить на зону сайдбара нужный виджет, записать в нем нужную информацию и сохранить.

Текст: Мы ВКонтакте

Заголовок: Мы ВКонтакте

<!-- VK Widget -->  
<div id="vk\_groups"></div>  
<script type="text/javascript">  
VK.Widgets.Group("vk\_groups", {mode: 0, width: "300",  
height: "200", color1: "FFFFFF", color2: "2B587A", color3:  
"5B7FA6"}, 60200444);  
</script>  
  
<!-- LiveInternet counter--><script type="text/javascript">  
<!--  
document.write("<img src='//counter.yadro.ru/hit?14.6;  
'<br>"+  
escape(document.referrer)+  
(typeof(screen)=="undefined"?<br>:"s"+screen.width+"<br>"+screen.height+"<br>"+  
(screen.colorDepth?<br>screen.colorDepth:screen.pixelDepth))+<br>";

## Виджеты, виджеты, виджеты...

Сайдбар с поддержкой виджетов, или любой другой участок сайта (хидер, футер...) позволяет вам управлять содержимым этой области. Т.к. виджеты стандартизированы, плагины могут использовать их и легко добавлять специальные виджеты для непосредственного контроля из админки. Более подробно виджеты будут рассмотрены позже, а сейчас мы разберем как добавить поддержку виджетов в шаблон:

```
<div id="sidebar">
    <ul>
        <?php if (!function_exists('dynamic_sidebar') || !dynamic_sidebar()) : ?>
        <li><!-- если нет активных виджетов то вывести этот текст --></li>
```

ZZ

```
</div>
</ul>
<?php endif; ?>
```

Теперь в файл `functions.php` добавить следующее

```
if (function_exists('register_sidebar')) {
    register_sidebar(array(
        'before_widget' => '<li id="%1$s" class="widget %2$s">',
        'after_widget' => '</li>',
        'before_title' => '<h2 class="widgettitle">',
        'after_title' => '</h2>',
    ));
}
```

Теперь каждый виджет, который вы добавите в сайдбар, будет появляться внутри тегов `<li>`, а заголовков виджета будет обрамлен тегами `<h2>`. Кроме этого, каждый виджет будет иметь свой уникальный ID и название class, что позволит силами CSS оформить каждый виджет по-разному под свою фантазию.

## Поисковая форма

Вордпресс предлагает встроенный поисковый функционал, который определенно стоит предоставить посетителям. Это ожидаемая посетителями возможность на большинстве сайтов, предлагающая полезный способ для нахождения нужного контента.

В общих случаях Вордпресс имеет файл `searchform.php` Это сделано для того, чтобы поисковую форму можно было встроить в разные файлы темы и в разные места, что еще один из примеров модульности. Как и сайдбар, поисковая форма подключается простым вызовом

```
<?php get_search_form(); ?>
```

Ее можно использовать без каких либо параметров, но если вам хочется сделать свою собственную форму, то подключить ее можно таким способом

```
<?php include(TEMPLATEPATH . '/inc/special-search-form.php'); ?>
```

### Где стоит применять поисковую форму?

На странице 404 ошибки, в сайдбаре и в блоке else цикла loop (если никаких подходящих записей не нашлось).

Если быть честными, то встроенный поиск гадость еще та. Он выдает результат поиска в хронологическом порядке, а подбирает записи исходя из того удалось ли найти в них запрос. Когда вордпресс выполняет поиск по запросу, он смотрит вхождение запроса в заголовке и в контенте записей и страниц. Если ничего не найдено, то ничего не отображается.

Чтобы еще сгустить краски — нет никакого расширенного поискового функционала, т. е. у вас и у посетителей очень мало контроля над найденными результатами, тем как они ищутся и отображаются. Есть несколько способов вмешаться в этот процесс, но они требуют достаточно сноровки.

Гораздо более простой способ улучшить поиск, это или заменить его на поиск от Google или

ZZ



поставить плагин, который сам все сделает за вас.

**Search Everything** <http://wordpress.org/plugins/search-everything/>

Вместо того, чтобы рыться в заголовках и контенте, почему бы не выполнить поисковый запрос непосредственно к базе данных? Этот плагин позволяет искать где угодно — в тексте, черновиках, комментариях, приложениях и т. д.

Google Search <https://www.google.com/cse/>

Система пользовательского поиска по сайту Google позволяет быстро находить релевантные результаты, настроить внешний вид результатов в соответствии с стилем сайта, а также заработать установкой рекламы AdSense для поиска.

**Search API Plugin** <http://wordpress.org/plugins/search/>

Этот плагин предлагает более мощный функционал, нежели стандартный вордпрессовский. Вы можете выполнять поиск различного типа контента, искать в определенных рубриках, использовать операторы поисковых запросов и даже выполнять полнотекстовый поиск по БД сайта. Для тех кому этого мало, плагин позволяет интегрировать пользовательский поиск гугл.

## Футер

Также как хидер или сайдабр, футер это один из вездесущих дизайнерских элементов. Они так часто используются в шаблонах, что вордпресс создал специальный шаблонный тег для вызова футера

```
<?php get_footer(); ?>
```

Эта функция принимает только один параметр, строковый, который работает по аналогичному с сайдбарами принципу. Т.е. если использовать такой код

```
<?php get_footer("mini"); ?>
```

то будет запрашиваться файл шаблона футера с названием `footer-mini.php`

**wp\_footer()**

Помните про `wp_head()`? А `wp_footer()` это тоже самое, только расположенное в самом низу сайта. Он говорит вордпрессу «футер — это тут». Больше ничего он не делает. А вот плагины, использующие этот хук уже добавляют нужный код и скрипты в нижнюю часть сайта.

Например, достаточно общая практика загружать javascript зависящие от содержимого страницы в футере вместо хидера. Это делается для того, чтобы страница точно и полностью была загружена к моменту вызова скрипта.

Поэтому правильным будет разместить хук футера перед закрывающим тегом `<body>`, вот так

```
<?php wp_footer(); ?></body>
```

## Минифутеры и мегафутеры

Как и сайдбары, дизайн футера целиком и полностью зависит от вашего сайта и его аудитории. Если вам нечего сказать в подвале, нет смысла забивать его ненужными плюшками. И в то же время не бойтесь сделать что-то интересное и инновационное, если у вас есть соответствующее содержание для наполнения футера.

[illegible]

Сравните эти два футера:

### Mini footer



## Mega footer



## Функции тем

Шаблоны вордпресс могут включать специальный файл, `functions.php`, который дает много контроля и власти над вещами, которые происходят в шаблоне. Подумайте об этом, как о файле который может делать все, что могут делать плагины, но без плагинов! И во всей красе прелесть этого файла раскрывается в темах, которые должны быть укомплектованы определенными плагинами изначально.

## Функции для специфических шаблонов

Т.к. вы можете выполнять похожие вещи как плагинами, так и встроенными функциями, то есть рациональное зерно в том, чтобы обходиться без плагинов.

Функции, содержащиеся в `functions.php` распространяются только на тот шаблон, к которому этот файл относится, и код зависит от темы, в которой используется. С другой стороны, плагины сохраняют свой функционал независимо от того, какая тема активирована. Т.о. не запикивайте в `functions.php` то, что может понадобиться и с другими шаблонами. И обратно — не используйте плагин там, где нужно что-то сделать в конкретной текущей теме.

Например:

Отключить уведомление о обновлениях Вордпресс  
Добавить кнопку в редактор записей  
Загрузить jQuery с сервера google  
Заменить аватары комментаторов по умолчанию

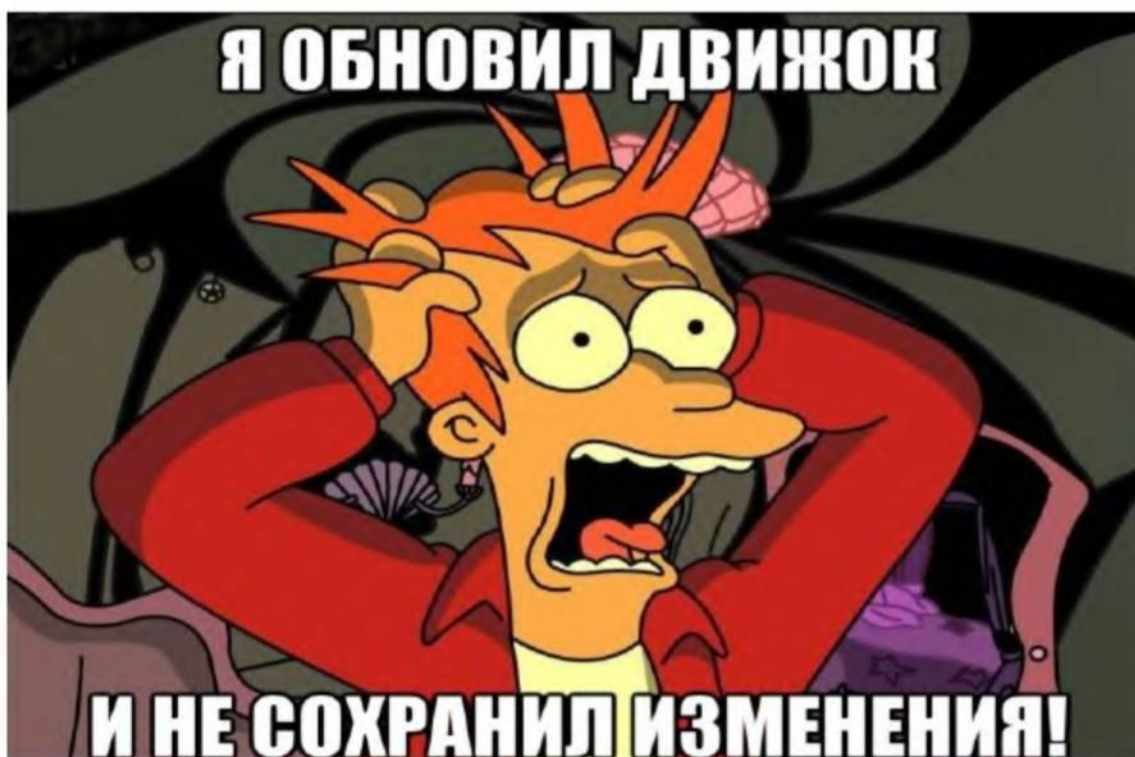
Используй плагин  
Используй плагин  
Используй functions.php  
Используй functions.php

zz



В первых двух примерах, желаемый результат связан с админкой сайта и ничего не делает непосредственно с шаблоном. Вам стоит использовать плагин, т. к. это не будет зависеть от шаблона и не перестанет работать, если его поменять (и соответственно изменится `functions.php`).

## Преимущество над правкой ключевых файлов ядра



Худшая часть этой безумной практики в том, что после нескольких обновлений вы просто забрасывали эти бесконечные правки или же переставали обновляться. Что плохо во всех отношениях по отношению к потенциальным угрозам и неполучению последних и крутых фишек в новых версиях вордпресса.

Но нужно понимать, что вордпресс не всемогущий и кое-какие правки в ядре все же придется сделать, о чем будет позже рассказано несколькими полезными советами.



#### 4. Дизайн и Разработка Шаблонов

## Настройка цикла loop

Как упоминалось ранее, если и есть одна вещь, которую нужно понимать для того чтобы быть реально крутым перцем разработчиком шаблонов, так это как работает цикл и что вы можете с ним делать. Давайте посвятим наше внимание трем способам заставить Wordpress Loop делать все, что нам угодно.

**query\_posts()** [http://codex.wordpress.org/Template\\_Tags/get\\_posts](http://codex.wordpress.org/Template_Tags/get_posts)

Если вы работаете с одиночным циклом записей и просто хотите модифицировать тип записей, которые он возвращает, используйте `query_posts()`. Это идеально для ограничения количества записей, исключения записей из определенной рубрики и т. д.

**WP Query()** [http://codex.wordpress.org/Function\\_Reference/WP\\_Query](http://codex.wordpress.org/Function_Reference/WP_Query)

Для мощной настройки множественных циклов, используйте `WP_Query()`. Настройкой дополнительных экземпляров `WP_Query()` в вашем шаблоне, можно создать любое количество множественных циклов и настроить вывод каждого из них.

**get\_posts()** [http://codex.wordpress.org/Template\\_Tags/get\\_posts](http://codex.wordpress.org/Template_Tags/get_posts)

Последний, но не худший, `get_posts()` позволяет создать дополнительные циклы где угодно в шаблоне. `get_posts()` принимает те же параметры, что и `query_posts()`, и идеален для вывода дополнительных циклов в сайдбаре, футере или где угодно.

Разберемся с ними подробнее.

## Настраиваем query posts()

Типичный цикл по умолчанию выглядит вот так

```
<?php // The WordPress Loop
if (have_posts()) : while (have_posts()) : the_post();
...
endwhile; else:
...
endif;
?>
```

Информация, отображаемая в цикле, зависит от огромного количества факторов, включая запрашиваемый url и соответствующий шаблон, генерирующий страницу. Например, когда кто-то посещает главную страницу сайта, исходный цикл выводит записи из всех категорий в количестве, заданном в админке.

Но что если мы хотим вместо отображения записей из всех категорий, исключить содержимое определенной, чтобы вывести их в отдельном месте страницы. Это идеальная задача для `query_posts()`, которая легко с ней справится:

```
<?php query_posts('cat=-9'); // исключаем категорию с ID = 9
if (have_posts()) : while (have_posts()) : the_post();
...
endwhile; else:
...
endif; ?>
```

ZZ

Но нужно учитывать необходимость внести некоторые модификации в стандартный код вызова: убедиться, что запрос был сделан правильно и сбросить запрос в состояние по умолчанию сразу после выполнения.

```
<?php
global $query_string;
$posts = query_posts($query_string.'&cat=-9');
if (have_posts()) : while (have_posts()) : the_post();
...
endwhile; else:
...
endif;
wp_reset_query();
?>
```

Тег `wp_reset_query` восстанавливает оригинальный запрос `query_posts` и используется при применении в шаблонах нескольких последовательных циклов.

Используя такой шаблон можно легко вносить нужные правки в запрос, например сделав такой

```
$posts = query_posts($query_string.'&cat=-7,-8,-9&posts_per_page=3&order=ASC');
```

что значит вывести 3 записи, исключив 7,8,9 категории и отсортировать записи по алфавиту.

Ключ к успешному использованию `query_posts` заключается в сохранении оригинального формата запроса с помощью `$query_string` переменных. Будучи однажды подключенными, можно добавить столько параметров, сколько нужно ( подробнее тут [http://codex.wordpress.org/Function\\_Reference/WP\\_Query#Parameters](http://codex.wordpress.org/Function_Reference/WP_Query#Parameters) ).

Этот способ хорош, но не лишен недостатков. `query_posts` использует дополнительные запросы к базе данных и может повлиять на работоспособность других условных тегов, таких как `is_page()` и `is_single()`. Поэтому рекомендуется не использовать этот способ, хотя он и работоспособен, а заменить его на `WP_Query` с которым точно проблем не будет.

## Настраиваем WP\_Query

Для полного контроля над любым количеством циклов, `WP_Query` самый лучший способ. Если нужно модифицировать дефолтовый цикл, то это выглядит очень похоже на `query_posts`:

```
<?php // The WordPress Loop - customized with WP_Query
$custom_query = new WP_Query('cat=-9');
while($custom_query->have_posts()) : $custom_query->the_post();
...
endwhile;
wp_reset_postdata(); // reset the query
?>
```

Применяются те же параметры, что и для `query_posts`, поэтому для модификации числа постов, используемых\исключаемых рубрик, порядка сортировки делается все то же самое.

```
$custom_query = new WP_Query('cat=-7,-8,-9'); // исключить категории
$custom_query = new WP_Query('posts_per_page=3'); // ограничить количество
$custom_query = new WP_Query('order=ASC'); // порядок сортировки
```

????????????????????????????????

А также любые нужные комбинации

```
$custom_query = new WP_Query('cat=-7,-8,-9&posts_per_page=3&order=ASC');
```

В дополнение к сферам применения, мы также можем создавать и настраивать дополнительные множественные циклы, вот типовой пример:

```
<?php // Loop 1
$first_query = new WP_Query('cat=-1'); // исключили категорию 1
while($first_query->have_posts()) : $first_query->the_post();
...
endwhile;
wp_reset_postdata(); // reset the query

// Loop 2
$second_query = new WP_Query('cat=-2'); // исключили категорию 2
while($second_query->have_posts()) : $second_query->the_post();
...
endwhile;
wp_reset_postdata(); // reset the query

// Loop 3
$third_query = new WP_Query('cat=-3'); // исключили категорию 3
while($third_query->have_posts()) : $third_query->the_post();
...
endwhile;
wp_reset_postdata(); // reset the query
?>
```

Каждый из этих циклов может быть размещен в любом месте шаблона, не нужно вписывать их последовательно. Например первый цикл может быть в сайдбаре, второй в футере и т. д. И каждый выводит свои собственные списки записей, в соответствии с параметрами.

При желании, можно комбинировать вместе 3 разных способа `query_posts`, `get_posts` и `WP_Query` в множественных циклах. А в версии вордпресса 3.3 появилась функция `is_main_query`, позволяющая модифицировать только главные объекты в `WP_Query`, подробнее на [http://codex.wordpress.org/Function\\_Reference/is\\_main\\_query](http://codex.wordpress.org/Function_Reference/is_main_query)

## Настраиваем get posts

Самый легкий и безопасный способ создать множественные циклы, это `get_posts`. Просто добавляете в нужном месте шаблона

```
<?php
global $post;
$args = array('category' => -9); // исключили категорию
$custom_posts = get_posts($args);
foreach($custom_posts as $post) : setup_postdata($post);
...
endforeach;
?>
```









Глобальная настройка количества отображаемых постов настраивается в админке, но мы можем менять это значение прямо в цикле. Например так мы задаем свое значение вместо значения по умолчанию.

```
$custom_query = new WP_Query('showposts=6&paged='.$paged);
```

Таким образом мы ограничим количество выводимых постов шестью, вместо десяти (значение по умолчанию). Чтобы вывести абсолютно все записи, нужно указать параметр равный «-1».

## Исключаем некоторые категории

Одна из самых популярных задач- исключение записей определенной категории на главной странице сайта. Например у вас есть несколько записей про хомячков, но вы хотели бы убрать их с главной страницы сайта. Этот контент остается в базе данных, отображается в своей рубрике и RSS ленте, но только не на главной. Есть множество способов сделать это, но мы применим самое простое из решений....да, это WP Query class:



```
$custom_query = new WP_Query('cat=-3&paged='.$paged); // нет хомячкам
```

В общих случаях, знак минус означает исключить данный параметр. Поэтому мы исключили третью рубрику (про хомячков).

За что!?

Другие возможности исключить содержимое этой рубрики — использовать php функцию `continue` и пропускать записи определенной категории, скрывать контент с помощью `css` и `javascript`, править файлы ядра вордпресс и использовать плагины. Согласитесь, WP\_Query однозначно лучше.

## Изменяем порядок сортировки

Вордпресс относится к приложениям LIFO типа — last in, first out, т. е. последний пришел, первый вышел. Что означает, что мы видим сначала свежесопубликованные записи, а старички спрятаны глубоко в недрах навигации. Это рационально, т. к. большинство блоговых сайтов стремится первым показать самый свежий контент. Это дает людям смысл периодически возвращаться на сайт, т. к. они смогут найти что-то новенькое прямо на первой странице сайта. Но это не идеальный вариант для каждого типа сайтов.

Допустим вы используете вордпресс для того чтобы публиковать свой роман. Это означает, что вы писали и публиковали его хронологически, главу за главой. Поэтому когда посетители приходят на ваш сайт, вы хотите показать им сначала главу Один, но она является фактически самой старой записью, поэтому вместо этого они увидят самую последнюю опубликованную главу. Но нет проблем, всего навсего изменим порядок сортировки:

```
$custom_query = new WP_Query('orderby=date&order=ASC&paged='.$paged);
```

## Покажем специфические страницы, встраивая страницу в страницу

Другой интересный и полезный цикл, это использовать новый WP\_Query class для отображения только одной определенной страницы. Это может быть полезно из разных соображений, например для встраивания одной страницы в другую. Вот пример запроса, когда содержимое страницы About встраивается в другую страницу:

```
$custom_query = new WP_Query('pagename=about&paged='.$paged);
```

ZZ



Обратите внимание, что в `pagename` нужно использовать не реальное название страницы, а ее слаг.

## Пример использования множественных циклов

Ничто не заставляет вас ограничиваться только одним циклом. Вы можете использовать ровно столько, сколько вам нужно. Фактически, это в целом нормально иметь множественные циклы в шаблоне. Простым примером будет главная страница, которая показывает полное содержание трех новых публикаций. А затем в сайдбаре, в втором цикле показывает выдержки из остальных семи публикаций. И все работает чудесным образом.

А вот пример более сложной реализации. Допустим у нас есть милый четырех колоночный шаблон. В левом сайдбаре мы покажем выдержки из одной категории. Затем в главной центральной части, разбитой на две колонки, мы покажем 10 последних записей по 5 в каждой. И затем в правом сайдбаре, мы покажем еще 10 записей, идущих после первых десяти.

Таким образом у нас получается 4 цикла

1 — Левый сайдбар	2 — Левая колонка	3 — Правая колонка	4 — Правый сайдбар
<pre>new WP_Query( 'cat=7&amp;showposts=3');</pre> <p>3 последние записи из рубрики номер 7</p>	<pre>new WP_Query( 'cat=-7&amp;showposts=5');</pre> <p>5 последних записей, кроме из рубрики номер 7</p>	<pre>new WP_Query( 'cat=-7&amp;showposts=5 &amp;offset=5');</pre> <p>еще 5 последних записей, кроме из рубрики номер 7</p>	<pre>new WP_Query( 'cat=-7&amp;showposts=10 &amp;offset=10');</pre> <p>еще 10 последних записей, кроме из рубрики номер 7</p>

которые будут находиться в `index.php`, но также разместить их можно будет где угодно еще.

Теперь необходимо разобраться с используемыми параметрами, которые помогают отображать именно то, что нам нужно.

### Параметр cat

Цикл 1 — отображает записи только из 7 рубрики (cat=7)

Цикл 2 — отображает записи кроме 7 рубрики (cat=-7)

Цикл 3 — отображает записи кроме 7 рубрики (cat=-7)

Цикл 4 — отображает записи кроме 7 рубрики (cat=-7)

## Параметр showposts

Цикл 1 — отображает только 3 записи (showposts=3)

Цикл 2 — отображает только 5 записей (showposts=5)

Цикл 3 — отображает только 5 записей (showposts=5)

Цикл 4 — отображает только 10 записей (showposts=10)

### Параметр offset

Цикл 1 — не используется

Цикл 2 — не используется

Цикл 3 — пропускает первые 5 записей (offset=5)

Цикл 4 — пропускает первые 10 записей (offset=10)

Таким образом, каждый цикл модифицирован для отображения желаемых записей. Последнее, что останется рассмотреть, это нужна ли нам навигация в одном из этих циклов. В обычных циклах по умолчанию мы используем теги `posts nav link()` для вывода навигационных ссылок на главную,

ZZ



## Сайдбары и Футеры

Сайдбары и футеры настолько типичные элементы дизайна, что обладают своими специальными функциями для решения типовых задач.

Давайте посмотрим на этот пример:



Заметили два сайдбара? Великолепный способ представить огромное количество информации посетителям.

## Множественные сайдбары

Код для создания шаблона, типа показанного выше, выглядит примерно так:

<?php get_header(); ?>	
<?php get_sidebar(); ?>	Первый сайдбар
<div id="main-content">	
// Loop	
</div>	
<?php get_sidebar("secondary"); ?>	Второй сайдбар
<?php get_footer(); ?>	

Обратите внимание, что мы обычно не зовем правый сайдбар «правым», мы называем его «вторым». Если в один прекрасный день мы решим поменять их местами, понятие «правый» потеряет смысл. Вот что означает правильная семантика в веб-дизайне — описывать без указания.

Два тега `get_sidebar()` в этом коде могут размещаться во всех основных файлах шаблона, связанных с стилем полной страницы, содержащей оба сайдбара (`index.php`, `single.php`, `page.php` и т. д.) Вы можете настроить два шаблона сайдбаров, `sidebar.php` и `sidebar-secondary.php` соответственно, с разметкой и тегами по усмотрению, одинаковыми или различными.

Вот типовой пример кода, который может использоваться в этом шаблоне:

```
<div id="sidebar">
<!-- разметка и код первого сайдбара -->
</div>
```

ZZ



```

<div id="main-content">
<!-- разметка и код цикла вывода записей -->
</div>
<div id="sidebar-secondary">
<!-- разметка и код второго сайдбара-->
</div>
<div id="footer">
<!-- разметка и код футера -->
</div>

```

и это нам может дать например такую трехколоночную структуру используя CSS

```

#sidebar { width: 200px; float: left; }
#main-content { width: 500px; float: left; }
#sidebar-secondary { width: 200px; float: right; }
#footer { clear: both; }

```

## Меню, Списки архивов и Облака тегов

Есть множество способов создать динамическую навигацию в Вордпрессе. Это реально удобно, потому что хорошая навигация — это ключ к успеху сайта. Когда мы создаем Записи и Страницы, в момент создания контента мы создаем способ найти этот контент. Давайте посмотрим на два примера.

### • Блог на основе рубрик

Если посмотреть на популярный блог Nettuts+, то мы увидим достаточно традиционную структуру, где записи публикуются день за днем в хронологическом порядке. Главная страница представляет новые записи всех типов, начиная с самых последних. В блогах такого типа, посетители приходят по двум причинам: посмотреть нет ли чего нового или найти что-то нужное. Сайт хорошо отрабатывает обе задачи, отображая заметную форму поиска и выводя первым самый новый контент.

```

wp_list_pages("include=7");
bloginfo('rss_url');
get_search_form();

```

Выбранная страница

RSS Feed

Поиск

```

<ul>
<li><a href="/">Home</a></li>
wp_list_categories();
</ul>

```

Рубрики

Хлебные крошки

```

while (have_posts())
Последние записи

```

Список записей

Список записей

- **Иерархический контент**

Вот как выглядит структура таких страниц с учетом их вложенности в админке:

**Edit Pages**

All (343) | Published (343) | Scheduled (0) | Pending Review (0) | Draft (3) | Private (0)

Bulk Actions ▼ Apply

Displaying 41–60 of 344

Search Pages

<input type="checkbox"/> Title	Author		Date
<input type="checkbox"/> Code Snippets	Chris Coyier		2009/08/06 Published
<input type="checkbox"/> — CSS <a href="#">Edit</a>   <a href="#">Quick Edit</a>   <a href="#">Delete</a>   <a href="#">View</a>	Chris Coyier		2009/08/06 Published
<input type="checkbox"/> — — Scale on Hover with Webkit Transition	Chris Coyier		2009/09/10 Published
<input type="checkbox"/> — — Simple and Nice Blockquote Styling	Chris Coyier		2009/09/04 Published
<input type="checkbox"/> — — Standard CSS Image Replacement	Chris Coyier		2009/08/06 Published
<input type="checkbox"/> — — Sticky Footer	Chris Coyier		2009/09/04 Published
<input type="checkbox"/> — — Style Override Technique	Chris Coyier		2009/08/22 Published
<input type="checkbox"/> — — Text Rotation	Chris Coyier		2009/09/03 Published
<input type="checkbox"/> — — Using @font-face	Chris Coyier		2009/08/10 Published
<input type="checkbox"/> — HTAccess	Chris Coyier		2009/09/04 Published

////////////////////////////////////



А вот как их можно отобразить в браузере с помощью стилей



## Стили меню в зависимости от типа страницы

В предыдущем примере мы видели трех-уровневую вложенную структуру. Каждый уровень имеет различные требования к навигационному меню. Давайте посмотрим подробнее на стили меню в зависимости от типа страницы

- **Родительская страница — главная страница этого раздела**

Родительская страница использует специальный шаблон `page-snippet-cat.php`. Этот уникальный шаблон страницы позволяет вывести все, что нам нужно на этой странице, но он не должен совсем уж отличаться от остальных страниц сайта и содержать те же основные элементы — хидер, сайдбар и футер. Различие между специальной страницей и всеми остальными заключается в специальном заголовке, который отображает дочерние страницы

```
wp_list_pages('title_li=&exclude=3285,3473&child_of='.$post->ID);
```

Эта функция создает всю необходимую разметку для отображения вложенного меню каждой страницы, а с помощью параметров настраивается как и что именно будет отображаться. В параметрах этой функции отключается отображение заголовка страницы, исключается отображение некоторых страниц (с айди 3285,3473)

- **Дочерняя страница — Языки программирования**

Удобно, чтобы все шесть дочерних страниц использовали тот же самый шаблон, что и родительская страница. В любой момент вы сможете сделать шаблон гибким для обслуживания любого нужного количества страниц. Т.к. функция `wp_list_pages` отображает дочерние страницы

////////////////////////////////////





Понятно, что идеальность — очень относительное понятие, и для каждого сайта идеальная страница зависит от конкретного содержания и назначения. Но если вы хотите создать эффективную страницу архивов, то есть хороший и проверенный вариант.

Создайте новый файл шаблона, назовите его `archives.php` и включите нужный тег и приведенного набора, который будет создавать список ссылок на все нужное содержимое.

<?php wp_get_archives('type=yearly'); ?>	Список всех записей, упорядоченный по году написания
<?php wp_get_archives('type=monthly'); ?>	Список всех записей, упорядоченный по месяцам
<?php wp_get_archives('type=daily'); ?>	Список всех записей, упорядоченный по дням
<?php wp_list_authors(); ?>	Список всех зарегистрированных авторов блога
<?php wp_list_categories('title_li='); ?>	Список всех рубрик блога
<?php wp_tag_cloud(); ?>	Список всех тегов блога
<?php wp_list_pages('title_li='); ?>	Список всех страниц блога

Собрав все вместе, можно сделать вот такой вариант шаблона:

```
<?php /* Template Name: Archives */ ?>
<?php get_header(); ?>
<div id="content">
<h2><?php the_title(); ?></h2>
<div class="col">
<h3>Архивы по годам:</h3>
<ul><?php wp_get_archives('type=yearly'); ?></ul>
<h3>Архивы по месяцам:</h3>
<ul><?php wp_get_archives('type=monthly'); ?></ul>
<h3>Архивы по дням:</h3>
<ul><?php wp_get_archives('type=daily'); ?></ul>
<h3>Архивы по рубрикам:</h3>
<ul><?php wp_list_categories('title_li='); ?></ul>
<h3>Архивы по авторам:</h3>
<ul><?php wp_list_authors(); ?></ul>
</div>
<div class="col">
<h3>Все страницы:</h3>
<ul><?php wp_list_pages('title_li='); ?></ul>
<h3>Архивы по тегам:</h3>
<?php wp_tag_cloud(); ?>
</div>
</div>
<?php get_sidebar(); ?>
<?php get_footer(); ?>
```

И если к примеру, вы найдете в интернете симпатичный шаблон, а в нем будет такой код, то для его руссификации достаточно будет перевести «Archives by Year:», «Archives by Month:» (как в этом примере) и все остальное на русский и сохранить.

Но грамотные шаблоны делаются немного по другому. В них используются строковые переменные типа `__` ("string"); и в зависимости от включенного языка, эти переменные будут отображать соответствующее значение, заданное в файле локализации `ro/ro` или `pot`. При смене языка будет меняться отображаемое значение этой переменной. Поэтому для перевода таких шаблонов нужно создать и перевести `po` файл для русского языка.



## Производим впечатление с облаком тегов

Одна из самых крутых штук, которую можно сделать с тегами — это отображать их в виде гигантского облака тегов\ссылок. Облако тегов великолепно не только потому, что оно удобно с точки зрения навигации, но скорее потому, что оно дает посетителю большую картинку о том, о чем весь ваш сайт. В Вордпрессе отображение облака легко настраивается, позволяя каждому создать именно такое облако, как нужно и идеально для его сайта. Можно долго об этом говорить, давайте перейдем к делу:

```
<?php wp_tag_cloud(array(
'smallest' => 10,           // размер шрифта редко используемых тегов
'largest'  => 18,           // размер шрифта часто используемых тегов
'unit'     => 'px',         // размер задается в пикселях
'orderby'  => 'name',       // сортировка по алфавиту
'order'    => 'ASC',        // начиная с А
'exclude'  => 6              // ID исключаемых тегов
)); ?>
```

Полный список параметров можно найти тут [http://codex.wordpress.org/Template\\_Tags/wp\\_tag\\_cloud](http://codex.wordpress.org/Template_Tags/wp_tag_cloud)

## Боковое содержимое и полезные элементы меню

Одна из самых веселых задач — это построить сайдбар сайта, чтобы это не было хаотичным набором блоков. Давайте посмотрим, что может быть полезным и привлекательным для посетителей.

## Отображение свежих комментариев

Есть три способа сделать это, давайте пройдем от самого легкого к сложному:

- **Виджет** — если сайдбар поддерживает вставку виджетов, то просто перетащите на него встроенный виджет «Свежие комментарии», укажите заголовок и количество отображаемых комментариев. Сохраните, готово :)
- **Плагин** — есть огромное количество плагинов для вывода комментариев, например Get Recent Comments (<http://wordpress.org/plugins/get-recent-comments/>), для настройки и вывода комментариев. Он создает виджет, который как мы видели выше, прекрасно заменяется встроенным виджетом. Именно поэтому данный плагин заброшен и не развивается и показан только для примера.
- **Пользовательская функция** — Т.к. комментарии хранятся в базе данных, достать их оттуда труда не составит. Мы создадим свою собственную функцию и добавим ее в functions.php для создания и обработки SQL запроса, который будет собирать самые последние одобренные комментарии. А затем мы создадим цикл для вывода результатов запроса к базе данных и отображения их один за одним

```
<?php // display recent comments
function dp_recent_comments($no_comments = 10, $comment_len = 35) {
global $wpdb;
$request = "SELECT * FROM $wpdb->comments";
$request .= " JOIN $wpdb->posts ON ID = comment_post_ID";
$request .= " WHERE comment_approved = '1' AND post_status = 'publish'
AND post_password = """;
$request .= " ORDER BY comment_date DESC LIMIT $no_comments";
$comments = $wpdb->get_results($request);
if ($comments) {
```

[illegible]



Чтобы использовать эту функцию, достаточно вставить в нужное место

что отобразит 6 последних комментариев.

Давайте так же пройдем от самого простого способа к сложному.

- ```
<?php $home_brew = new WP_Query('showposts=10&orderby=date');
while($home_brew->have_posts()) : $home_brew->the_post();
// здесь поиграемся и настроим вывод заголовков, контента, пользовательских полей и т.д.
endwhile;
wp_reset_postdata();
?>
```

[illegible]

## Отображение популярных записей

Популярность — понятие еще более широкое, нежели «недавность». Как вы определите популярность записи? На основании числа просмотров? Количества комментариев? Обратных ссылок? Если вы хотите собрать все сразу и в одном флаконе, то попробуйте плагин **WordPress Popular Posts** (<http://wordpress.org/plugins/wordpress-popular-posts/>) который как раз и занимается сбором различной информации, чтобы на ее основании определить популярность и вывести список популярных записей в вашем шаблоне:

```
<?php if (function_exists('get_mostpopular')) get_mostpopular(); ?>
```

Если же вы хотите оценивать популярность на основании только одного критерия, к примеру количества комментариев к записи, то этого можно достичь с помощью такого запроса к базе данных:

```
<ul class="popular-posts">
    <?php $popular_posts = $wpdb->get_results("SELECT id,post_title FROM
        {$wpdb->prefix}posts ORDER BY comment_count DESC LIMIT 0,10");
    foreach($popular_posts as $post) {
        print "<li><a href='". get_permalink($post->id) .'>".
            $post->post_title."</a></li>\n";
    } ?>
</ul>
```

### Отображение недавно измененных записей

Можно подумать, что для этого нам опять понадобится какой-то хитрый запрос к базе данных. Мы конечно можем, но все гораздо проще и `WP_Query` с определенным параметром (`orderby=modified`) может все значительно упростить:

```
<?php // display recently modified posts
$home_brew = new WP_Query('showposts=5&orderby=modified');
while($home_brew->have_posts()) : $home_brew->the_post();
    // тут все как обычно- выводим титле и контент
endwhile;
wp_reset_postdata();
?>
```

### Отображение случайных записей

И тут все просто. Опять же, WP\_Query с параметром orderby=rand обеспечивает случайное отображение серии случайных записей блога:

```
<?php // display random posts
$home_brew = new WP_Query('showposts=3&orderby=rand');
while($home_brew->have_posts()) : $home_brew->the_post();
    // тут все как обычно- выводим титле и контент
endwhile;
wp_reset_postdata();
?>
```

Хитрый пример использования этой техники вот какой — можно выводить только одну случайную запись и сохранить функцию в специальный файл шаблона. А затем использовать Ajax для вызова

zzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzz

этого шаблона и отображения, например в сайдбаре. И в результате у вас будет милый блок, в котором будут самостоятельно перегружаться и отображаться разные случайные записи вашего сайта без перезагрузки страницы. (Подробнее тут <http://digwp.com/2009/07/display-a-random-post-with-ajax-refresh/>)

## Отображение записей из Твиттера

А кто сказал, что мы должны ограничиваться только содержимым нашего сайта? Импортировать свежие твиты из Твиттера может оказаться веселым способом общения с посетителями и всегда оставаться актуальным. У Твиттера есть добротная API система для получения и использования данных. Но т. к. твиттер периодически падает и бывает недоступен, нельзя становиться в зависимость от него, т. к. это вызовет задержки в загрузке страниц нашего сайта. Поэтому используя API для получения и отображения различных данных на своих страницах, нужно сделать все так, чтобы это не влияло на загрузку страниц сайта и не выглядело ужасным в результате проблем с API.

Соединение с Twitter API можно целиком сделать с помощью JavaScript, что предпочтительно по нескольким соображениям:

- Соединение с твиттером происходит на клиентской стороне (в браузере посетителя) и не нагружает наш сервер\хостинг
- Если все сделать правильно — это никак не влияет (отрицательно) на скорость загрузки страниц
- Данные обрабатываются и выводятся на страницу только в случае успеха

Т.ч. давайте это сделаем. Мы будем использовать jQuery, что очень удобно и легко.

## Шаг 1. Подключаем jQuery

Добавляем в `functions.php` этот код. Он загрузит библиотеку jQuery на страницу через вставку кода в секцию `<head>`

```
if(!is_admin()) {
    wp_deregister_script('jquery');
    wp_register_script('jquery', ("http://ajax.googleapis.com/ajax/libs/
jquery/1.3.2/jquery.min.js"), false, '1.3.2');
    wp_enqueue_script('jquery');
}
```

## Шаг 2. Загружаем скрипт

Нужно использовать специальную библиотеку, которую скачиваем и подключаем в шаблон

```
<script type="text/javascript" src="<?php bloginfo('template_url'); ?>/js/twitter.js"></script>
```

### Шаг 3. Плагин jQuery

Этот плагин выполняет всю грязную работу по связи с Твиттером и вытягиванием последних твитов. Кроме этого он делает прикольные штучки, типа преобразования сокращенных ссылок в реальные адреса, хеш-тегов в ссылки поиска и @реплай превращает в ссылки на профили.

```
(function($){
    $.fn.lastTwitterMessage = function(username){
    var $base = this;
```

ZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZ



Been getting nailed with WordPress spam  
across the board that use "http://link" as the  
URL. (∞)

## ABOUT THE AUTHOR

Chris Coyier is a web designer from Madison, WI currently living in Chicago, IL and working for Chatman Design.

**CONTACT • ADVERTISING**

## SUBSCRIBE



RSS Feeds



iTunes



iTunes

[illegible]

будет записан последний твит аккаунта `@chriscoyier` и отобразится на странице. Если же не получится — то элемент останется пустым.

А вот тут пример того, как можно встроить на страницу Facebook, Twitter и Google+ <http://wp.smashingmagazine.com/2012/01/19/facebook-twitter-google-wordpress/>

## Импорт и отображение содержимого RSS лент

Ранее в вордпресс был встроен парсер RSS лент MagpieRSS, однако в связи с прекращением его развития, сейчас Вордпресс переключился на SimplePie. И спустя некоторое время разработка SimplePie остановилась. Такое ощущение, что включение библиотеки в Wordpress равносильно поцелую смерти для любого парсера фидов.

Но не будем о грустном, наличие встроенного RSS парсера позволяет получать свежий контент с других сайтов и отображать его у себя. Нужно лишь включить файл `feed.php`, который идет в комплекте Вордпресса по умолчанию на любую нужную страницу, на которой нужно выполнить парсинг, предварительно выполнив следующие настройки.

```
<?php // Import & Display Feed
if(function_exists('fetch_feed')) {
    $feed = fetch_feed('http://digwp.com/feed/'); //адрес для парсинга
    if (!is_wp_error($feed)) :
        $feed->set_cache_duration(1200); // задаем время кеширования
        $feed->init(); // выполняем запрос
        $limit = $feed->get_item_quantity(7); // определяем количество элементов
        $items = $feed->get_items(0, $limit); // создаем массив
    endif;
}
if ($items) { ?>
<ul>
    <?php foreach ($items as $item) { ?>
        <li>
            <div>
                <strong>
                    <a href="<?php echo $item->get_permalink(); ?>">
                        <?php echo $item->get_title(); ?></a>
                    <strong>
                        <em><?php echo $item->get_author(); ?></em>
                    </div>
                    <div><?php echo $item->get_description(); ?></div>
                    <div><?php echo $item->get_date('j F Y'); ?></div>
                </li>
            <?php } ?>
        </ul>
    <?php } ?>
```

Разместив этот код в `sidebar.php` или в любом другом нужном месте, мы сможем наслаждаться несколькими последними записями с другого сайта, отображаемыми на нашем сайте.

Первая часть скрипта подготавливает и выполняет запрос, вторая часть выполняет разметку и отображение результатов запроса. Каждая запись содержит заголовок с ссылкой на оригинальный пост, имя автора, запись и дату публикации.



Конечно же, все настраивается любым нужным способом, документацию по Simplepie можно посмотреть тут <http://simplepie.org/wiki/reference/start>

## Создаем и используем дочерние темы

Вордпресс поддерживает интересную технику создания шаблонов, известную как дочерние темы. Идея заключается в том, что вы задаете некий базовый шаблон как родительский, а затем применяете все правки к нему на дочернем шаблоне, не трогая таким образом родительский шаблон в принципе.

Зачем все это нужно? Почему бы просто не сделать копию оригинального шаблона и не поиздеваться над ним? Есть много разговоров на эту тему. Некоторые ребята утверждают, что дочерние темы- это потеря времени. Поэтому давайте разберемся как следует, чтобы вы самостоятельно пришли к заключению.

Нравится ли вам, чтобы ваш шаблон был всегда актуален и содержал все свежие апдейты и исправления. Нравится ли вам появление в шаблоне новых функций и возможностей? Шаблоны, основанные на фреймворках особенно часто обрастают дополнениями. Но если вы внесли изменения в шаблон, становится достаточно больно обновить такую тему, чтобы не потерять свои правки. А если использовать дочерний шаблон, то это достаточно несложно — всего лишь обновите родительский шаблон, а все правки сохранятся в дочернем. А если вы хотите сделать несколько модификаций шаблона? Сделайте несколько дочерних и переключайтесь между ними.

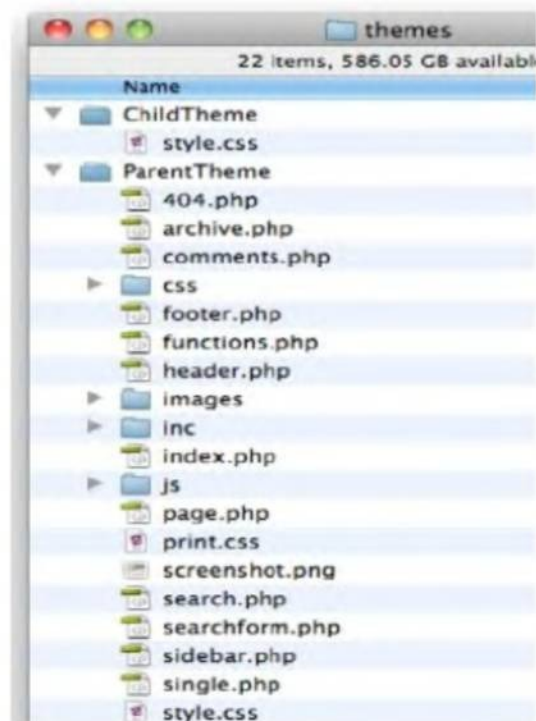
С дочерними темами, вы можете предложить пользователям выбор нескольких альтернатив, чтобы они активировали ту, что им больше понравится. И это без ручных правок или дублирования кода. Создание дочерних тем очень легкое и эффективное и позволяет всегда иметь актуальный и обновленный шаблон.

Чтобы создать дочернюю тему — просто создайте папку по своему желанию в `wp-content/themes` и разместите в ней файл `style.css`

Чтобы тема определялась как дочерняя по отношению к другой, в этот файл нужно дописать специально отформатированный комментарий в самом верху файла, чтобы он корректно идентифицировался и нашел родительскую тему, по отношению к которой будут применяться изменения.

Вот пример заголовка:

```
/*
Theme Name: My Cool Child Theme
Theme URI: http://digwp.com/
Description: Child Theme for Digging Into WordPress
Author: Chris Coyier
Author URI: http://chriscoyier.net/
Template: DigWP
Version: 2.0
*/
@import url("../digWP/style.css");
```



Самая важная строка `Template: DigWP` , которая ссылается на папку с родительской темой. Последняя строка импортирует стили из родительского шаблона, строка опциональная, но

ZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZ



используется для того, чтобы задать базовые стили, которые будут меняться и к которым будут добавляться новые. Все что будет написано ниже- будет перезаписывать и заменять значения и параметры из родительского шаблона.

Например, если изначально цвет ссылок красный, т. е. в родительском шаблоне указано `a { color: red; }`, то после указания в дочернем `a { color: green; }` ссылки станут зеленого цвета, заменив оригинальное значение.

Никакие дополнительные !important правила не требуются, новые стили перекрывают старые.

## Стилизация шаблонов

В любом шаблоне, файл `style.css` обязателен. Фактически, Вордпресс не опознает шаблон как шаблон, если в нем в папке не будет обнаружен этот файл. Он там должен быть обязательно, но вам не обязательно использовать его. И даже не обязательно вызывать этот файл в шапке шаблона. Просто он должен быть в папке и все тут. Но в большинстве случаев этот файл таки используется.

## Различные способы подключения файла стилей

- **Не очень динамичный способ**

```
<link rel="stylesheet" href="/wp-content/themes/Your-Theme/style.css" type="text/css" media="screen, projection" />
```

- **Супер-точный способ**

```
<link rel="stylesheet" href="<?php bloginfo('stylesheet_url'); ?>" type="text/css" media="screen, projection" />
```

- **Динамичный способ**

```
<link rel="stylesheet" href="<?php bloginfo('stylesheet_directory'); ?>/style.css" type="text/css" media="screen, projection" />
```

- В стиле IE спецификации

```
<!--[if IE 6]>
<link rel="stylesheet" type="text/css" href= "<?php bloginfo('stylesheet_directory'); ?>/
css/style-ie6.css" />
<script type='text/javascript' src='<?php bloginfo('template_url'); ?>/js/DD_belatedPNG_0.0.8a-
min.js'></script>
<![endif]-->
```

- **Подключение множественных файлов**

```
<link rel="stylesheet" href="<?php bloginfo('stylesheet_directory'); ?>/style.css" type="text/
css" media="screen, projection" />
<link rel="stylesheet" href="<?php bloginfo('stylesheet_directory'); ?>/forms.css" type="text/
css" media="screen, projection" />
```

- **@import**

Достаточно распространенная техника динамического вызова файлов стилей, путем их подключения к основному. Используется например для вставки типовых блоков стиля, вместо дописывания их в основной файл

```
@import "style/css/reset.css";
@import "style/css/typography.css";
@import "style/css/layout.css";
```

ZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZ

- **Условное подключение дополнительных файлов**

```
<?php if (is_page_template("page-snippet.php")){ ?>
```

```
<link rel="stylesheet" type="text/css" href="<?php bloginfo('stylesheet_directory'); ? /css/snippets.css" />
```

<?php } ?>

## Сбрасывать или не сбрасывать?

Раз уж мы подняли тему стилей и CSS, не лишним будет заострить внимание на одной важной теме, известной как сброс стилей. «Теория Ресета» заключается в том, что удаляются различные настройки стилей по умолчанию, применяемые к элементу правилами браузеров, т. к. эти стили по умолчанию могут быть несколько непредсказуемыми и у каждого браузера свои.

Например, отступы по умолчанию в Firefox отличаются от отступов в Internet Explorer. А к некоторым элементам в разных браузерах могут быть такие правила, которых совершенно нет в других. Поэтому сброс стилей выполняется для того, чтобы устранить эти различия и начать с чистого листа. Если затем нам понадобятся какие-либо свои стили, мы их применим в соответствии с потребностями и с учетом кросс-браузерности (одинакового отображения стилей сайта во всех браузерах).

## Популярные CSS Reset

- Eric Meyer's Reset Reloaded <http://meyerweb.com/eric/thoughts/2007/05/01/reset-reloaded/>
- YUI (Yahoo User Interface) Reset CSS <http://yuilibrary.com/>
- Star Selector Reset `* { margin: 0; padding: 0; }`

Полное собрание сбрасывателей стилей можно посмотреть тут <http://perishablepress.com/a-killer-collection-of-global-css-reset-styles/>

Сбрасывание стилей не обязательно для Вордпресс, но общепринято для всех сайтов. В контексте правильного подключения, это будет выглядеть вот так

```
<link rel="stylesheet" href="<?php bloginfo('stylesheet_directory'); ?>/reset.css" type="text/css" media="screen, projection" />
```

```
<link rel="stylesheet" href="<?php bloginfo('stylesheet_directory'); ?>/style.css" type="text/css" media="screen, projection" />
```

Обратите внимание, что сброс стилей производится до подключения основных стилей. Также применительно к своему сайту или стилям можно использовать онлайн инструмент сброса стилей <http://cssresetr.com/>

## Базовая оптимизация CSS

И теперь после того, как мы показали вам как подключать стили в шаблон, мы скажем вам, что это не самый лучший способ. Ха! Не расслабляйтесь.

Проблема подключения сброса стилей не в том, что он делает, а в том как это сделано. Это отдельный файл, а значит это еще один запрос, который браузер должен обработать и еще один файл, который должен быть загружен. Чем больше запросов выполняется на странице, тем медленнее она загружается и выполняется.



ZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZ



Если вы помешаны на скорости загрузки сайта, вы возможно захотите включить содержимое сбрасывателя в ваш основной `css` файл. В этом случае это будет идеально, т. к. сброс будет загружен на все страницы вашего сайта. Но что делать, если разные стили загружаются в зависимости от разных условий? Или для разных страниц стили отличаются — для главной и для записи. Поэтому не всегда объединение всех стилей в один файл имеет смысл.

Снижение количества css файлов не единственный путь оптимизации, можно применять технику уменьшения самих размеров файлов. В процессе написания стилей, мы их пишем так, чтобы нам было понятно что, где как и зачем, плюс многие места сопровождаются комментариями и пояснениями. Но все эти пробелы, отступы, табуляции, переносы и комментарии добавляют веса файлу, заставляя посетителей загружать файл большего размера.

Красиво отформатированный файл удобен вам, но не браузеру и не посетителям. Поэтому нет важной причины хранить такой файл, особенно когда разработка шаблона завершена и больше частые правки и переделки не планируются. Можно использовать онлайн-утилиту типа <http://www.cssoptimiser.com/> для удаления всего ненужного из файла. Конечно, после такой очистки он превратится в трудно читаемый набор стилей, но никто не запрещает сохранить оригинальную копию для себя. При необходимости вносите в него правки и снова сохраняете на сайте сжатую версию.

Вот пример того, как может выглядеть сжатый css файл

```
#mainNav {position: absolute; bottom: 8px; left: 0; width: 780px; list-style: none; z-index: 100} #mainNav
li {display: inline} #mainNav li a {-webkit-transition: opacity .15s ease-in-out; display: block
;float: left; width: 144px; height: 50px; background: url(images/nav.png) no-repeat; text-indent: -
9999px; margin: 0 10px 0 0; opacity: 0.5; filter: alpha(opacity=50)} #mainNav li.forums a {background: -
144px 0} #mainNav li.videos a {background: -288px 0} #mainNav li.downloads a {background: -432px
0} #mainNav li.snippets a {background: -576px 0}
```

Не очень читаемо? Зато на 43% меньше, чем оригинальный файл.

## Использование нескольких тем

Понятно, что в один момент времени для одного человека может показываться одна тема. Но это не означает, что мы не можем показывать разным посетителям разные темы.

## Зачем это нужно?

- Чтобы дать пользователям возможность выбора шаблонов
- Использовать разные шаблоны для себя, админов, зарегистрированных посетителей и гостей
- Использовать разные шаблоны для разных устройств (десктопы, планшеты, смартфоны)

Давайте рассмотрим разные варианты реализации этих задач.

## Предоставление выбора пользователям

Вероятно, это было бы прикольно позволить посетителям выбирать тему для просмотра вашего сайта. Это не так практично на большинстве сайтов, но в некоторых случаях может стать крутой фишкой. Можно найти сайты, содержащие в сайдбаре или футере менюшку, позволяющую переключить шаблон. И это практически обязательный элемент для сайтов, демонстрирующих и продающих различные шаблоны и дизайны, чтобы можно было в живую увидеть их в работе.

zzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzz



Эту задачу поможет решить плагин **Theme Switcher** (<http://wordpress.org/plugins/theme-switcher/>). Обратите внимание, что нужно обязательно дать возможность посетителям вернуться к теме по умолчанию. Также широко возможности раскрываются путем комбинирования переключения дочерних тем.

## Различные темы для администраторов

Достаточно часто переключение шаблонов используют при разработке шаблонов. Как знает каждый дизайнер, чтобы добиться лучшего результата, нужно работать с реальным контентом, а не с набором *lorem-ipsum lorem-ipsum*. Если вы хотите разрабатывать, править и переделывать шаблон на работающем сайте, то нужно чтобы это было видно только вам, а не посетителям. Такую возможность дает плагин **Theme Switch** (<http://wordpress.org/plugins/nkthemeswitch/>) - не путаем с предыдущим плагином.

После установки и активации плагина у нас появляется возможность выбора отображаемой темы в зависимости от роли на сайте. Т.е. мы сможем задать один шаблон видимый только для админа и другой шаблон для остальных посетителей сайта.

## Различные шаблоны для мобильных посетителей

Другая причина для использования различных тем, это способ прибытия посетителей на сайт. Наглядным примером будет отображение специального шаблона для тех, кто пришел на сайт на мобильном устройстве.

Все что нужно, это использовать плагин, специально созданный для обслуживания мобильных посетителей. Плагин **Wptouch** (<http://wordpress.org/plugins/wptouch/>) обнаруживает мобильные устройства - iPhone, iPod Touch, Android, Blackberry и отображает им специальную удобную и функциональную тему.

Другой способ создания мобильной версии сайта — это бесплатный сервис **Mobify** <http://www.mobify.com/>. Фактически этот сервис не создает шаблон, но конечный результат тот же самый. Он обрабатывает контент вашего сайта и применяет к нему специальные настраиваемые стили для создания мобильной версии, удобной для просмотра на маленьких экранах. Mobify обслуживает только контент с своего домена или специально созданных поддоменов вашего сайта с помощью небольшой правки CNAME записи вашего домена для перенаправления.

Если же вы хотите развернуть свою собственную тему, то нужно подключить детектор мобильных устройств и перенаправлять таких посетителей на другой url. Обычно это JavaScript, загружаемый в <head> секцию сайта. Обратите внимание, чтобы этого скрипта не было по url, на который вы перенаправляете, иначе создадите бесконечный замкнутый цикл перенаправлений.

```
<script type="text/javascript"> if (screen.width <= 699) { document.location = "mobile.html"; } </script>
```

Этот метод может быть скомбинирован вместе с плагином Theme Switcher, т. к. он позволяет отображать для специальных урл отдельные шаблоны.

Адаптивный дизайн сейчас везде, даже в админке вордпресс. Начиная с версии 3.3 адаптивность стала стандартом по умолчанию.

## Поддержка виджетов

Большинство шаблонов, доступные сейчас — уже идут с поддержкой виджетов. Это означает, что они содержат необходимый код, благодаря которому в админке вордпресс в определенные участки

////////////////////////////////////

Авторы тем, желающие чтобы их шаблоны использовали максимально широко, стараются добавлять зоны виджетов, как в сайдбары, так и в футеры. Правда не всегда. Поэтому если вы создаете свой собственный шаблон, будет лучше если вы захотите добавить такие зоны в него.

## Шаг 1. Объявляем зону виджетов

Указанный тут контент отображается только  
если не активирован ни один виджет  
<?php endif; ?>

В файл `functions.php` добавляем код

Обратите внимание, что этот код ссылается на зону виджетов по ее имени и затем задает определенные параметры до и после виджета, а также до и после заголовка виджета. Это позволяет настроить отображение именно так, как нам это нужно. Кроме этого, каждый виджет получит свой собственный уникальный class и ID, что даст возможность стилизовать каждый виджет отдельно или все виджеты сразу.

Изначально в вордпрессе уже идет достаточное количество нужных и полезных виджетов. Просто перетаскивайте их на нужное место и активируйте. Кроме этого многие плагины добавляют свои собственные виджеты. Я думаю, вы уже успели полюбить вордпресс за то, что многие задачи можно сделать без правки кода, а лишь кликами, перетаскиванием и сохранением настроек.

С хорошим пониманием системы шаблонов вордпресс, мы уже готовы к тому, чтобы разобраться как расширить функционал сайта.





## 5. Расширяем функциональность Wordpress

## Расширения

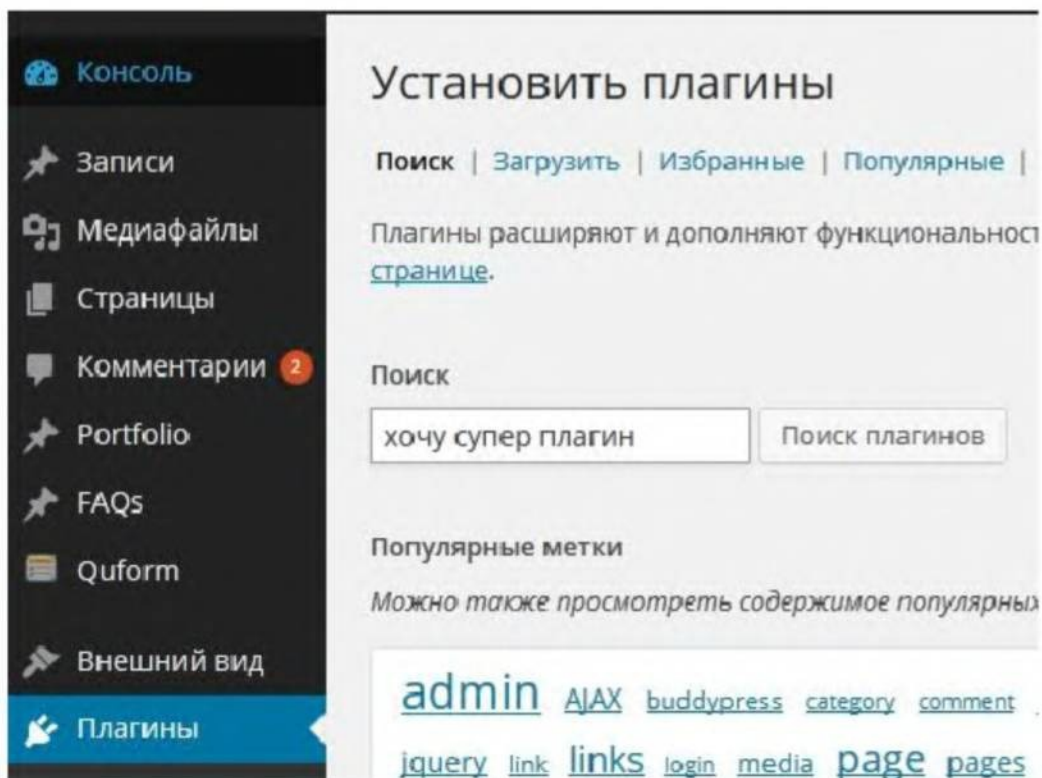
Прямо из архива, Вордпресс мощная платформа для создания динамических сайтов, применяемая для создания любого разнообразия проектов. Заточенный для создания блоговых сайтов, вордпресс предлагает все, что нужно для ведения блога со всеми нужными возможностями, в легкой и удобной админке, которая делает процесс публикации и редактирования контента, а также общения с посетителями и комментаторами, легким как бриз. Но главная причина, почему вордпресс стал так популярен, это тотальная расширяемость.

## Расширяем вордпресс плагинами

Начиная с шаблонов и плагинов и заканчивая пользовательскими функциями, скриптами и даже правками ядра, степень настройки и адаптации движка поразительна. Фактически, с помощью нескольких плагинов и пары встроенных фишек, можно трансформировать Вордпресс из мощного блогового движка в настраиваемую Систему Управления Контентом (CMS). В этой главе мы рассмотрим несколько техник и важных аспектов расширения функционала Вордпресс.

## Плагин для (почти) всего

Самый легкий способ добавления нового функционала заключается в использовании преимуществ тысяч плагинов, доступных в **WordPress Plugins Directory** (<http://wordpress.org/plugins/>), а также в интернете. Даже если не принимать во внимание плагины не вошедшие в официальный каталог, то в нем содержится 28 720 плагинов (на момент перевода книги), а на момент написания было 21 400. С таким количеством плагинов можно решить практически любую, даже очень специфическую задачу. И нужный плагин можно искать непосредственно в админке вордпресса, вводя поисковые запросы для поиска решения.



Изначально вордпресс содержит всего 2 плагина. Один невероятно полезный, а другой практически бесполезный. Первый- это **Akismet**, один из лучших антиспам плагинов. Akismet

ZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZ



защищает ваши комментарии от спама, путем сравнения их с своей базой. Любой комментарий, выглядящий как спам блокируется и отправляется в корзину. Админы затем могут их рассмотреть и восстановить или удалить окончательно. Непроверенные спам-комментарии автоматически удаляются через 15 дней.

К сожалению, другой плагин — это **Hello Dolly**. Он не делает ничего, кроме как отображает случайные кусочки песни “Hello Dolly” в правом верхнем углу админки. Этот плагин был самый первый, созданный для вордпресса, и вероятно поэтому будет включен в инсталляцию до конца дней. Если вы сентиментальный человек, то можете его оставить, во всех остальных смыслах плагин абсолютно бесполезен и должен быть удален первым делом сразу после установки движка.

Из тысяч плагинов, доступных для установки, есть плагины установленные на больших и хорошо известных сайтах, что делает их абсолютно необходимыми для любой инсталляции вордпресс. Эти плагины значительно увеличивают мощь Вордпресса по некоторым важным направлениям.

Когда вы соберетесь скачивать и устанавливать какой-либо плагин, помните, что 99% плагинов создаются независимыми вебмастерами, которые не озадачивались тестами на оптимизацию и максимальную производительность и совместимость. Большинство из них избыточны и не оптимальны, а их функциональность часто достигается более простыми методами. В многих ситуациях вам в принципе плагин не понадобится для достижения своей цели. Поэтому перед тем, как устанавливать еще один плагин, спросите себя - нужен ли тут плагин в принципе?

## Нужен ли вам плагин?

Допустим вы хотите добавить некую крутую фишку на сайт. Перед тем как городить огород с плагином проверьте, нет ли более простого способа? Красота вордпресса в том, что почти всегда есть несколько способов решить одну задачу, особенно если это связано с настройкой шаблона.

Например, чтобы отобразить дату и время последней модификации записи можно установить плагин. А можно добавить одну строку кода в шаблон:

Обновлено `<?php $x = get_the_time('U'); $m = get_the_modified_time('U'); if ($m != $x) { the_modified_time('F d, Y'); } ?>`

Что делает этот код? Он отображает дату\время последнего редактирования записи и размещается например внутри цикла рядом с другими метаданными записи. 7 других интересных хаков можно найти тут <http://perishablepress.com/optimize-wordpress-pure-code-alternatives-for-7-unnecessary-plugins/>

Поэтому если появляются задачи расширения шаблона путем добавления иконок рубрик, случайных картинок, популярных записей, редко требуется именно плагин для реализации этой задачи. Перед поиском плагина — поищите скрипт, виджет или функцию для решения. Пара секунд на поиск сэкономит кучу времени для разбирательства с чужим ненужным, а иногда и криво работающим плагином.

Основная идея в том, чтобы минимизировать число используемых плагинов. Делая это, вы уменьшаете обработку дополнительных скриптов и помогаете оптимизировать производительность сайта. Минимизация числа плагинов упрощает обновление сайта и уменьшает вероятность различных несовместимостей. Чем меньше плагинов работает, тем меньше возможность того, что что-то пойдет не так при переходе на последнюю версию Вордпресс. Особенно если плагин забросили, не развивают и не поддерживают. Все пляшет вокруг облегчения обновления, избегания конфликтов и упрощения ухода за сайтом.

////////////////////////////////////





даже «плагин ... отстой».

9. Проверьте файлы. После того, как вы нашли идеальный плагин, проверьте внимательно его файлы. Ищите все, что может быть не на месте, а также то, чего там быть не должно. Идет ли в комплекте readme и документация.
10. Проверьте код. Если вы разбираетесь в php, html, css, JavaScript - это поможет вам понять все ли в порядке в коде. Ищите непонятные или потенциально вредные функции, спамные ссылки, непонятный код и т. п.

Хотя вы не можете пройти все эти шаги с каждым используемым плагином, держите их в голове во время поиска нового. Опять же, ключ к прокачиванию ваших навыков и опыта заключается в времени, затраченном на исследование и разбирательство. И собрав один раз пакет нужных и полезных плагинов, вы спокойно будете его ставить на все новые сайты.

## Использование и уход за плагинами

Вопреки распространенному мнению, плагины не относятся к категории «установил-настроил-забыл». Насколько они удобны и легки, настолько же они нуждаются в периодической настройке, уходе, обновлении и решении периодических проблем.

## Последовательная установка

Один из самых важных советов, которые мы даем, это во время добавления новых плагинов на сайт делайте это последовательно, один за раз. Установка плагинов по одному дает возможность протестировать сайт на корректную работоспособность после каждой установки. Это позволяет узнать немедленно — совместим ли плагин с набором остальных плагинов на сайте или нет. Установка 20 плагинов за раз и понимание, что сайт упал, приводит к тому, что нужно будет плагины отключать один за другим, чтобы найти виновника торжества. Таким образом вы сэкономите время и нервы, если будете устанавливать плагины тщательно и последовательно.

## Поддерживайте актуальность плагинов

Один из лучших способов убедиться, что все скорее всего в порядке — это обновлять плагины. А самый простой способ убедиться в этом — это периодически заглядывать в админку. Как только появляется обновление плагинов, шаблонов или самого вордпресса — в админке появляется соответствующее сообщение.

После этого у вас есть два пути:

## Автоматическое обновление

Начиная с версии 2.5 администраторы сайта могут обновляться автоматически, нажимая соответствующие ссылки в панели обновлений. Начиная с версии 3.0 стало возможно обновляться пакетно, выделив например все плагины, требующие обновления и нажав **Обновить**. Больше нет никаких оправданий о том, почему же не получается поддерживать актуальность.

## Ручное обновление

Ручной апгрейд дает немного большую степень контроля процесса обновления, нежели автоматический. Контроль процесса обновления важен для плагинов, которые были после установки скорректированы и получили правки в коде, а также для диагностики и поиска решений различных потенциальных проблем, которые могут возникнуть.

Новые версии часто содержат новые файлы, новые опции и настройки или изменения текущих

zzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzz



настроек, а то и их пропажу. В некоторых случаях обновление плагина требует полного анинсталлирования предыдущей версии и даже дополнительных шагов для очистки и удаления следов использования. Ручная установка позволит уменьшить потенциальные проблемы.

Время от времени появляются жалобы на различный вредоносный код, содержащийся в плагинах, особенно если речь идет о плагинах не из репозитория вордпресс и многократно особенно, если речь идет о взломанных и зануленных плагинах. Можно не быть гуру программирования, но все же стоит потратить некоторое время на проверку кода плагина или его файлов антивирусов. В 99% все будет в порядке, но лучше потратить на это немного времени сейчас, чем потом сожалеть и тратить время на восстановление и наведение порядка.

## Подпишись на обсуждение плагина

Отличный способ всегда быть в курсе новостей, связанных с плагином — это подписаться на соответствующий фид, блог автора, форум или обсуждение. Автоматическое обновление работает только для плагинов, установленных из репозитория. Еще один повод следить за новостями авторских плагинов.

## Получение помощи с плагинами

Как только вы погрузитесь в процесс установки и настройки плагинов, неизбежно начнутся вопросы, проблемы и конфликты, которые придется решать. Как только это происходит, первое место куда обращаются- разработчики или автор плагина. Достаточно часто страница разработчиков плагина остается открытой для комментирования, или же предоставляется некоторое подобие форума для вопросов относительно плагина.

Если проблема не решается там способом, не стесняйтесь написать автору плагина лично. Он создал плагин и таким образом будет более чем счастлив помочь людям с его настройкой и использованием.

Дальше стоит поискать на [wordpress.org](http://wordpress.org), где можно зарегистрироваться и создать топик с своей бедой. Но перед этим убедитесь, что вы тщательно искали в интернете, иначе вас отправят назад в гугл. Большинство модераторов терпеть не могут пользователей, которые не тратят время на домашнюю работу перед тем, как напечатать свой вопрос.

Также нужно понимать, что бесплатная помощь — бесплатна. Большинство новичков делают ошибку, ожидая немедленную и исчерпывающую помощь и поддержку по всем вопросам, которые они задают. Это работает немного не так. Руководители Вордпресс получают радость от того, что помогают людям учиться, но будьте уверены, что потратили достаточно времени и внимания, чтобы разобраться в своей ситуации, прежде чем просить о помощи. Будьте корректными, понятными, вежливыми и уважайте тех, кто тратит время помогая вам. Это все карма! :-)

## Диагностика конфликтов плагинов

Диагностика конфликтов довольно часто гораздо легче, чем это могло бы показаться. С учетом того, что вы устанавливали плагины **последовательно** (ведь так?), проблемы возникают с конкретно взятыми плагинами. А зная проблемный плагин — мы находимся в выгодной позиции при поиске решения. Иногда стоит просто отключить плагин и подождать обновления — если правильно выбрать плагин, то его автор не станет затягивать время и выпустит исправление.

## Деактивация и удаление плагинов

Так как вы работаете с плагинами, нужно четко понимать разницу между деактивацией и

////////////////////////////////////

деинсталляцией(удалением) плагина. В общем, деактивация плагина означает что он не работает, но присутствует в папке плагинов. Гораздо важнее, что для деактивированного плагина все настройки все еще присутствуют в базе данных. С другой стороны, при удалении плагина, он удаляется из папки плагинов и все связанные настройки из базы данных удаляются (если анинсталлирование плагина сделано автором плагина правильно).

Помните, что много плагинов добавляет информацию в базу данных. После активации плагин может изменить и добавить информацию в различные таблицы, в основном, но не только, в таблицу «options». Новые таблицы могут быть добавлены и наполнены данными. Поэтому если вы пробуете и добавляете новые разные плагины, они все оставляют после себя следы различных добавление и изменений. Правильно спроектированный плагин после деинсталляции обеспечивает уборку мусора за собой и удаляет все ненужное из базы данных.

Но часто бывают ситуации, когда нужно быстро отключить один или несколько плагинов из-за конфликта или проблем. Т.к. обычно деактивация происходит через админку, то вероятны ситуации невозможности в нее попасть. Есть несколько способов отключить плагины, и самый быстрый и простой — это переименовать плагин, его папку или всю папку `wp-content/plugins` целиком (чтобы деактивировать сразу все плагины). Переименовав директорию в `wp-content/plugins-old` вы деактивируете (но не удалите!) все плагины. После того как вы все наладили и готовы вернуть все назад — верните папке ее оригинальное имя. Все настройки плагинов остались неприкосновенны, останется только их заново активировать.

Если же вас не пугает перспектива поковыряться в phpMyAdmin, то есть более изящный способ деактивации плагинов. Но обязательно перед всеми манипуляциями сделайте резервную копию юазы данных!

Сделайте простой запрос `SELECT * FROM wp_options WHERE option_name = 'active_plugins'`, который выведет список всех активных плагинов.

Вы увидите что-то типа такого

```
a:31:{i:0;s:13:"AddMySite.php";i:1;s:19:"akismet/akismet.php";i:2;s:23:"all_in_
one_seo_pack.php";i:3;s:16:"authenticate.php";i:4;s:28:"breadcrumb-navigation-xt.
php";i:5;s:18:"codeautoescape.php";i:6;s:37:"contact-coldform/contact_coldform.
php";i:7;s:32:"custom-query-string-reloaded.php";i:8;s:30:"customizable-post-listings.
php";i:9;s:33:"dd-sitemap-gen/dd-sitemap-gen.php";i:10;s:20:"download-counter.
php";i:11;s:13:"feedcount.php";i:13;s:15:"get-weather.php";i:14;s:36:"google-
sitemap-generator/sitemap.php";i:15;s:13:"gravatars.php";i:16;s:19:"kill-admin-
nags.php";i:17;s:18:"landingsites13.php";i:18;s:30:"nofollow-free/nofollowfree.
php";i:19;s:17:"ol_feedburner.php";i:20;s:16:"plugins-used.php";i:21;s:22:"popularity-
contest.php";i:22;s:39:"search-everything/search_everything.php";i:23;s:27:"simple-
tags/simple-tags.php";i:24;s:26:"simple_recent_comments.php";i:25;s:18:"simple_
twitter.php";i:26;s:25:"subscribe-to-comments.php";i:27;s:24:"the-excerpt-reloaded.
php";i:28;s:18:"theme-switcher.php";i:29;s:9:"top10.php";i:30;s:16:"wp-db-backup.php";}
```

Весь этот массив представляет собой каждый активный плагин на вашем сайте. Таким образом для быстрой деактивации всех активных плагинов сохраните этот код в надежном месте и удалите его из БД. После этого плагины станут неактивны, но все еще установлены и не теряют свои настройки. После того как вы готовы активировать их заново — вернитесь обратно в поле `active_plugins` и верните назад его предыдущее значение.

Либо же вместо того, чтобы удалить все активации, можно выборочно деактивировать плагин, вырезав его название из этого списка и оставив примерно так

////////////////////////////////////

Также быстро деактивировать все плагины можно запросом к БД

```
UPDATE wp_options SET option_value = '' WHERE option_name = 'active_plugins';
```

```
UPDATE wp_options SET option_value = 'a:0:{}' WHERE option_name = 'active_plugins';
```

## Поделитесь своим опытом

Также стоит поделиться важной информацией с авторами плагинов, чтобы они могли внедрить это исправление в обновление плагина или знали что советовать остальным новичкам. Идея в том, что разделяя информацию и опыт с другими пользователями, вы помогаете им лучше понимать с чем им придется иметь дело.



Просто потому что вы можете, не означает, что вам следует это делать. Как ранее уже обсуждалось, есть много способов расширить функциональность вордпресс, особенно если вы комфортно себя чувствуете ковыряясь в коде.

## Плагины против functions.php

- Зона действия - плагины обычно работают на всем сайте, функции работают только когда тема активна.

////////////////////////////////////



- Обновления — плагины обычно обновляются периодически. Функции темы обычно не обновляются, особенно те, которые мы сами добавили вручную.
- Админка — плагины всегда показываются в админ-панели. Функции темы могут и не проявлять себя в админке, в зависимости от их функции и скриптов.

Когда же используем возможности `functions.php`? Если функции привязаны к шаблону, если это пользовательские функции шаблона, если это маленькие и простые скрипты и функции.

## Примеры полезных функций тем

Если вернуться в далекие времена Wordpress 1.5, то разработчикам даже и не снились возможности специфических возможностей шаблонов. Все изменения делались непосредственно в файлах ядра вордпресс, то, что вы сейчас скорее всего уже и не будете делать. Сейчас, в большинстве случаев functions.php позволяет сделать любые вещи без изменения файлов вордпресса.

Например:

## Как отключить уведомления о обновлениях

В админке вордпресс напоминает о доступных обновлениях ядра и плагинов. Напоминание обновления плагинов показывается только на странице плагинов, а вот уведомление обновления ядра показывается на каждой странице. Давайте отключим, добавив в `functions.php` этот код

```
add_action('admin_menu', create_function('$a', "remove_action('load-plugins.php', 'wp_update_plugins');"));
add_filter('pre_option_update_plugins', create_function('$a', "return null;"));
add_action('init', create_function('$a', "remove_action('init', 'wp_version_check');"));
add_filter('pre_option_update_core', create_function('$a', "return null;"));
```

### Дополнительные кнопки комментирования

Добавок к кнопкам администрирования комментариев (редактировать, удалить, отправить в спам) в админке, удобно иметь такие же кнопки на страницах сайта, чтобы заниматься модераторскими делами не заходя в админку. Вот простой способ добавить кнопки «Удалить» и «Пометить как спам», которые будут выводиться после каждого комментария, если вы авторизованы как администратор. Простым посетителям они не будут видны и доступны.

```
function spam_delete_links($id) {
global $id;if (current_user_can('edit_post')) {
echo ' | <a href="'.get_bloginfo('wpurl').'/wp-admin/comment.php?action=cdc&c='.$id.'">Удалить</a>';
echo ' | <a href="'.get_bloginfo('wpurl').'/wp-admin/comment.php?action=cdc&dt=spam&c='.$id.'">Спам</a>';
}}
}
```

После размещения вышеуказанного кода в `functions.php`, останется только вставить в нужное место шаблона вызов этой функции

```
<?php spam_delete_links(get_comment_ID()); ?>
```

И это все!

~~zzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzz~~

## Шорткоды для сайта

Шорткоды вордпресс- это ссылки на часто используемый контент, такой как ссылки, картинки, заголовки. Например, вместо того, чтобы писать полную ссылку на главную страницу сайта, этот шорткод будет выглядеть как [home]. И везде, где он размещен в тексте, он будет заменен на ссылку. Шорткоды могут быть весьма значительны и выполнять разнообразные задачи.

Как и ожидается, функционал шорткодов будет добавляться в `functions.php`. В этом примере, функция для преобразования `[home]` в ссылку на главную страницу сайта будет выглядеть так

```
<?php
function myHomePage() {return '<a href="http://supersite.ru/" >Ссылка!</a>';}
add_shortcode('home', 'myHomePage');
?>
```

И если вдруг нужно будет изменить ссылку везде, где она использовалась, достаточно просто исправить ее адрес в функции. Более подробно про шорткоды можно почитать тут [http://codex.wordpress.org/Shortcode\\_API](http://codex.wordpress.org/Shortcode_API)

## Перемещение плагинов в functions.php

Этот файл настолько крутой, что может заменить собой все плагины. Это хороший способ изолировать функциональность плагинов внутри специального шаблона. По умолчанию, плагины влияют на все темы, но если включить их в `functions.php`, то при смене шаблона плагины работать не будут. Легче всего переместить те плагины, которые состоят из одного файла — нужно просто скопировать содержимое этого файла в `functions.php` и удалить. Остальные плагины могут потребовать дополнительных маневров для корректной работы и переноса, но в теории- это возможно.

## Создание плагинов из функций

Переместить плагин в `functions.php` так же легко, как и создать плагин из пользовательской функции. Как и в прошлом примере, нужно просто скопировать и вставить пользовательскую функцию в новый `php` файл и сохранить.

Но перед заливкой в папку плагинов нужно указать специальный заголовок с требуемой информацией. Вот пример правильного заголовка, который можно использовать как шаблон для создания своих плагинов:

```
<?php
/*
Plugin Name: My Awesome New Plugin
Plugin URI: http://example.com/plugins/awesome-new-plugin/
Description: Provides awesome new functionality for WordPress.
Version: 1.0
Author: The Awesome Plugin Maker
Author URI: http://example.com/
*/
// разместите тут крутой код функции
?>
```

Как и говорили, лучше размещать простые функции внутри `functions.php`, а не как отдельный плагин, но представление в виде плагина также имеет свои преимущества. Например, работа для

всех шаблонов сайта и возможность активировать и деактивировать функционал через админку. В случае с `functions.php` пришлось бы удалять этот код из файла или превращать его в комментарии.

Эти простые примеры показывают только вершину айсберга и лишь познакомили с широкими возможностями, которые можно воплотить в `functions.php`

Но кроме плагинов и функций есть еще способы расширить возможности Вордпресс. Давайте продолжать дальше изучение.

## Другие способы расширения возможностей

Итак, мы уже изучили возможности расширения с помощью плагинов и пользовательских функций. В дополнению к этому мы можем расширить возможности непосредственно в файлах шаблона. Это гибкий и удобный способ настроить шаблон под себя.

Допустим, вы хотите поставить ссылку на какую-то специфическую страницу.

`<a href="/contact/">Contact</a>` - не круто

`<a href="<?php echo get_permalink(12); ?>">Contact</a>` - гораздо лучше

В чем фокус? Если урл страницы (ее слаг) изменится, ссылка останется рабочей. Если изменится структура пермалинка, страница станет вложенной или сайт переместится в поддиректорию, то ссылка все равно останется рабочей.

Используя функцию `get_permalink` и ID записи или страницы мы всегда будем точно ссылаться на нужную нам запись или страницу не зависимо от того, что происходит на сайте. Но это работает только в шаблоне и не работает из текста записи. В этом случае стоит использовать шорткод.

```
function permalink_thingy($atts) {extract(shortcode_atts(array('id' => 1,'text' => "" ), $atts));
if ($text) {
    $url = get_permalink($id);
    return "<a href='$url'>$text</a>";
} else {
    return get_permalink($id);
}}
add_shortcode('permalink','permalink_thingy');
```

Использовать шорткод можно двумя способами:

**Базовый** [Текст ссылки](#)

Указав айди требуемой страницы и текст ссылки, на сайте в записи будет отображена соответствующая ссылка. Кроме этого можно указать дополнительный стиль для такой ссылки.

С представление текста [permalink id=49 text='Текст ссылки']

Таким способом вернется ссылка с заданным текстом.

Но самые горячие головы могут воспользоваться плагином **Exec-PHP** (<http://wordpress.org/plugins/exec-php/>) который позволяет вставлять исполняемый php код прямо в текст записи. Но с ним нужно быть очень внимательным и осторожным.

ZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZ





- Абсолютно точно понимать, что, как и зачем будет делаться
- Создать файл «Прочитать перед обновлением» в котором описать все детали сделанных правок, на случай полной потери памяти

Еще раз подытожим. Перед правкой все хорошенько изучить, знать что ты делаешь и быть абсолютно уверенным, что по другому не получится. Также все хорошенько задокументировать на случай грядущих обновлений. И вы заранее не знаете, что интересное раскопаете, залезая с головой под капот Вордпресса.

## Wordpress как Система Управления Контентом

Как мы видели, возможности вордпресса практически бесконечны. То, что начиналось как скромная блогговая платформа, названная b2/cafelog в 2001 году, сейчас крепкая и гибкая система для управления контентом (CMS).

Главное различие между блогом и CMS заключается в гибкости базовой программы и ее способности создавать и управлять множеством пользователей и различных цифровых данных — тексты, изображения, аудио, видео и т. д. С каждым новым апдейтом, Вордпресс становится все лучше для выполнения этих задач и продолжает развиваться.

Прямо из коробки, Вордпресс предлагает много фишек, которые помогают любому количеству пользователей публиковать и редактировать контент, а с добавлением плагинов возможности значительно растут. В этом разделе мы впервые посмотрим на вордпресс с точки зрения использования как системы управления контентом

## Пользовательские поля

Пользовательские поля раскрывают весь потенциал Вордпресс. Они позволяют пользователям назначить любой контент и информацию к записи. В результате запись перестает быть просто текстом с заголовком.

Для лучшего понимания того, как работают пользовательские поля, давайте рассмотрим вариант, когда нам нужно связать картинку с записью, но при этом чтобы картинка не была частью содержимого записи и ее можно было отобразить например в сайдбаре.

Для этого создается пользовательское поле, называемое “thumbnail”, в которое записывается урл изображения <http://supersite.ru/images/thumbnail-01.jpg>. Таким образом мы привязали этот адрес к записи, не вписывая его непосредственно в текст записи.

### Неправильно



### Правильно





Имея на руках это пользовательское поле и его значение, мы готовы внести некоторые изменения в функционал вордпресса и отображать нужную картинку нужным способом. Функция, которая будет использована для получения этой картинки — `get_post_meta`, которая используется для получения значений из пользовательских полей. Она оперирует тремя параметрами:

В целом это выглядит вот так `<?php get post meta($post id, '$key', $single); ?>`

И готовая функция будет выглядеть как `<?php get_post_meta($post->ID, 'thumbnail', true); ?>`

<?php echo get\_post\_meta(\$post->ID, 'thumbnail', true); ?>  
но не в конкретно таком виде — так как это отобразит  
ури картинки, а внутри тега `img`

Будучи размещенный внутри цикла, этот код будет выводить картинки-превью для всех записей.

```
.postThumb {
float: left; clear: none;
margin-right: 10px;
}
```

Как было разобрано ранее, в главе 4.1, `get_posts()` идеально подходит для такой задачи. Поэтому после добавления кода вывода изображений, этот дополнительный цикл в сайдбаре будет выглядеть так

```
<?php // второй цикл для превьюшек
global $post; $args = array('showposts=10&offset=0'); $custom_posts = get_posts($args);
foreach($custom_posts as $post) : setup_postdata($post); ?>
<a class="postThumb" href="<?php the_permalink() ?>"><img src="<?php echo get_post_meta($post-
```

## Catching a Train

Lorem ipsum dolor sit amet,  
 consectetur adipiscing elit. Donec  
 odio. Quisque volutpat mattis eros.  
 Nullam malesuada erat ut turpis.  
 Suspendisse urna nibh, viverra  
 non, semper suscipit, posuere a,  
 pede. Donec nec justo eget felis.

Lorem ipsum dolor sit amet,  
 consectetur adipiscing elit. Donec  
 odio. Quisque volutpat mattis eros.  
 Nullam malesuada erat ut turpis.  
 Suspendisse urna nibh, viverra  
 non, semper suscipit, posuere a,  
 pede. Donec nec justo eget felis.

Lorem ipsum dolor sit amet,  
consectetur adipiscing elit. Donec  
odio. Quisque volutpat mattis eros.  
Nullam malesuada erat ut turpis.  
Suspendisse urna nibh, viverra  
non, semper suscipit, posuere a,  
pede. Donec nec justo eget felis.



```
>ID, 'thumbnail', true); ?>" />
</a>
<?php endforeach; // конец второго цикла ?>
```

Обратите внимание, какие параметры были переданы в `get_posts` - `showposts` определил количество записей, `offset` определил что сдвига не будет и будут выбраны те же 10 записей, что были использованы в первом цикле.

Этот пример показывает как можно с помощью пользовательских полей преобразовать Вордпресс. Вместо публикации статических страниц и ленты записей, можно выражать свои замыслы с помощью различных изображений, медиа, дополнительной информации, полей и меню. Т.е. не ограничиваться одним только содержимым поля `the_content()`

Подытожим — пользовательские поля рулят!

## Пользователи, роли и разрешения

Вордпресс предлагает отличную поддержку нескольких пользователей. Каждый пользователь блога может обладать определенной ролью с определенным набором прав и возможностей.

Понятие пользователи относится ко всем, кто зарегистрирован на сайте. Что определяет пользователя, так это различные вещи, которые ему разрешено делать. Вещи, которые пользователи могут делать, называются ролями, по умолчанию включая следующие:

**Суперадминистратор** — всемогущий пользователь с полным контролем над всем, существует только в Мультисайт установках Вордпресса.

**Администратор** — суперадминистратор для обычного, одиночного вордпресс, с полным доступом и правами ко всему что есть на сайте.

**Редактор** — может публиковать и редактировать любые записи на сайте.

**Автор** — может публиковать и редактировать только свои записи на сайте.

**Участник** — может писать и редактировать свои записи, но не публиковать их.

**Подписчик** — зарегистрированный посетитель сайта без каких либо прав.

Более подробная раскладка приведена тут [http://codex.wordpress.org/Roles\\_and\\_Capabilities](http://codex.wordpress.org/Roles_and_Capabilities)

Каждая из ролей может быть назначена на любое количество пользователей. И каждая роль обладает определенными возможностями, т. н. Разрешениями. Получая какую-либо роль, пользователь получает определенные права, типа редактирования записей, публикации контента, модерации комментариев и т. д. Особые роли можно назначать с каким угодно сочетанием прав и возможностей.

Некоторые особые возможности для подписчиков можно создать. Функция `is_user_logged_in()` опознает зарегистрированного подписчика, поэтому таким способом можно, например, отображать им дополнительный контент, скрытый для простых посетителей сайта.

В общих настройках вы можете разрешить или запретить посетителям регистрироваться на сайте, а также определить, какая роль по умолчанию будет им назначаться. Пока вы не определились с концепцией сайта, будет лучше запретить регистрацию, или назначать всех свежесозданных пользователей подписчиками.

Мультисайт возможности Вордпресса позволяют создать отличную платформу для контента, генерируемого пользователями, поэтому можно будет гибко назначать для них соответствующие роли и разрешения.

ZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZ





**Страница Архивов Тегов** — отображать облако тегов вместе с популярными и недавно добавленными тегами.

```
<?php /*
Template Name: No Sidebar No Title
*/ ?>
```

## Страницы, Рубрики и Теги

Пример сайта с вложенной структурой контактной информации — в соответствии с структурой отделов\команд.

Таким способом можно организовать тысячи страниц, без хаоса, а с ясной и понятной многоуровневой системой иерархии.

zzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzz



## Динамическое меню

Каждый сайт нуждается в твердой, хорошо структурированной навигационной системе. К счастью, вордпресс предлагает массу мощных функций, помогающих создать любой вид и тип меню, какой только можно представить.

**wp\_list\_pages()**

Автоматически создает список страниц сайта. Вы можете сортировать, включать/исключать нужные страницы, ограничивать уровень вложенности и формат заголовков, настраивать ссылки и многое другое.

## wp\_get\_archives()

Автоматически создает списки записей сайта по годам, месяцам или дням. И также включает большое количество настроек.

**wp\_list\_categories()**

Отображает список рубрик с большой свободой настроек. Есть параметры для всего — включение/исключение рубрик, порядок сортировки, добавление специфических ссылок.

## wp\_tag\_cloud()

Отображает коллекцию ссылок, известную как облако, для всех тегов использованных на сайте.

А кроме этого, начиная с версии 3.0 предлагается удобный визуальный драг-н-дроп инструмент создания меню в админке вордпресс. Его функционала достаточно для большого количества задач, но всегда есть место для улучшений. Вот несколько бесплатных плагинов, которые помогают улучшить и расширить базовые навигационные возможности:

- **WP-PageNavi** <http://wordpress.org/plugins/wp-pagenavi/>

Заменяет архивную навигацию (раньше/позже) на постраничную, выводя строку с ссылками на все страницы сайта, помогая посетителям и поисковым ботам быстро получить доступ к любой записи.

- **Wordpress Navigation List Plugin NAVT** <http://wordpress.org/plugins/wordpress-navigation-list-plugin-navt/>

Добавляет полный контроль над созданием, стилизацией и наполнением навигации сайта с легким драг-н-дроп интерфейсом

- **WP-dTree** <http://wordpress.org/plugins/wp-dtree-30/>

Создает навигационное дерево для страниц, записей, ссылок и рубрик с впечатляющими визуальными эффектами.

- **Sub Pages widget** <http://wordpress.org/plugins/subpages-widget/>

Отображает дочерние страницы по отношению к текущей.

## Расширение функционала CMS

Хорошая новость в том, что есть много способов существенно расширить функционал. Плохая в том, что из книги эту главу вырезали, переместив на сайт авторов книги\*.

\*Примечание переводчика: *Но я ее оттуда вернул назад :)*

Представленные плагины иногда очень сильно меняют базовые возможности вордпресса с целью преобразовать его в многопользовательскую систему управления разными видами контента. А самое интересное — это все бесплатные плагины.

ZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZ

- **Административные инструменты**

**WP-CMS Post Control** <http://wordpress.org/plugins/wp-cms-post-control/> настраивает возможности редактора записей и страниц для разных пользователей, отключая ненужные элементы, удаляя ревизии постов и даже добавляя персональные сообщения

**WP-CMS** <http://wordpress.org/plugins/wp-cms/> меняет админку вордпресса, смещая акцент с создания записей на создание страниц. Блоговый функционал может быть полностью отключен, создано для неопытных пользователей вордпресса, для которых делается готовое решение в виде CMS, а не блога

**Supple Forms** <http://wordpress.org/plugins/supple-forms/> создает пользовательские панели, вставляя дополнительный контент в записи с помощью шорт-кодов

**Custom Write Panel** <http://wordpress.org/plugins/custom-write-panel/> позволяет создавать пользовательские панели с различным содержанием- чекбоксы, формы ввода, выпадающие меню и т. п.

**Pods** <http://wordpress.org/plugins/pods/> - это мощнейший фреймворк, позволяющий создавать, редактировать и управлять различным пользовательским контентом и данными, введенными в поля разнообразных форм

- **Управление ролями пользователей**

**Members** <http://wordpress.org/plugins/members/> комплексный плагин управления ролями, пользователями и создаваемым ими контентом

**s2Member** <http://wordpress.org/plugins/s2member/> предлагает набор инструментов для создания сайта с платным доступом, включая закрытый контент для разных пользователей

**Role Scoper** <http://wordpress.org/plugins/role-scoper/> создает различные специфические группы пользователей с заданными правами и возможностями

**Disclose-Secret** <http://wordpress.org/plugins/disclose-secret/> создает скрытые записи, видимые только пользователям с определенными критериями

**Page Restrict** <http://wordpress.org/plugins/pagerestrict/> скрывает страницы от неавторизованных пользователей

- **Сортировка, фильтрация и ограничение контента**

**AstickyPostOrderER** <http://wordpress.org/plugins/astickypostorderer/> создает любые произвольные порядки вывода записей на заданных страницах

**Advanced Category Excluder** <http://wordpress.org/plugins/advanced-category-excluder/> определяет разрешенные и запрещенные рубрики для вывода на определенных страницах, в фидах и результатах поиска

**Custom Post Limits** <http://wordpress.org/plugins/custom-post-limits/> независимые настройки для количества выводимых записей на любых страницах сайта

////////////////////////////////////



- **Функционал коммерческих сайтов**

**eShop** <http://wordpress.org/plugins/eshop/> добавляет функционал корзины, оформления и оплаты покупок

**WP e-Commerce** <http://wordpress.org/plugins/wp-e-commerce/> обеспечивает все необходимое для совершения цифровых покупок

**Quick Shop** <http://wordpress.org/plugins/quick-shop/> выводит в сайдбаре корзину покупок и обеспечивает легкий процесс шоппинга

**Cart66 Lite** <http://wordpress.org/plugins/cart66-lite/> обеспечивает все необходимое для совершения покупок цифровых и реальных товаров

**WooCommerce** <http://wordpress.org/plugins/woocommerce/> вероятно, самая комплексная и навороченная система совершения покупок на базе вордпресс

- **Подписные емейл листы**

**PHPList Form** <http://projects.jesseheap.com/all-projects/wordpress-plugin-phplist-form-integration> дает возможность пользователям легко подписываться на новости и RSS ленты

**WPnewsman** <http://wpnewsman.com/> позволяет собирать базу подписчиков, используя подписную форму

**MailChimp List Subscribe Form** <http://wordpress.org/plugins/mailchimp/> форма подписки сервиса рассылок MailChimp

- **Полезные дополнения**

**ProjectManager** <http://wordpress.org/plugins/projectmanager/> дает возможность управлять проектами с перекрестными и повторяющимися наборами данных. Очень удобно для организации различных каталогов и сборников

**WP-PostRatings** <http://wordpress.org/plugins/wp-postratings/> создает систему голосов и рейтингов для записей блога

**User Submitted Posts** <http://wordpress.org/plugins/user-submitted-posts/> создает возможность для пользователей публиковать записи из любого места сайта

- **Используем вордпресс как форум**

<http://bbpress.org/> простой, легкий и быстрый форумный движок на базе вордпресс

<http://simple-press.com/> еще один трансформер вордпресса в форум с массой возможностей

Но разве это все?

ZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZ



Честно говоря, нельзя просто так взять и закончить разговор о путях расширения функционала. В этой главе основной акцент был сделан на плагинах, файлах шаблона, пользовательских функциях и расширенном функционале самого вордпресса.



Но даже с этими возможностями можно наворотить такое, что достаточно раскроет потенциал вашего сайта. Если хотите еще больше погрузиться в способы прокачки вордпресса, вам будет интересна книга «Тао Вордпресса» <http://wp-tao.com/> - полное руководство для начинающих, админов и пользователей.

////////////////////////////////////

## 6. Работаем с RSS лентами

## Зачем нужны RSS ленты

После того, как вы опубликовали контент — он сразу же появляется на сайте. Но кроме этого Вордпресс формирует различные фиды (RSS ленты), содержащие информацию о последних публикациях на сайте.

Прямо из коробки, Вордпресс создает фиды каждого типа страниц: рубрики, теги, архивы по датам, фиды комментариев и т. д. Для типичного блога большинству посетителей будут интересны только фиды, содержащие основные публикации на сайте и возможно лента последних комментариев. Они забирают эти два фиды для того, чтобы быть в курсе всех событий вашего сайта, и добавляют фиды в фид-ридер.

Как способ публиковать и распространять контент, фиды революционным образом изменили то, как информация доставляется по интернету. Вместо того, чтобы заставлять посетителей постоянно посещать сайт, чтобы узнать не появилось ли чего нового, они могут просто открывать свой ридер и смотреть где какие новые публикации появились на их любимых сайтах, добавленных в ридер. Легкость этой технологии позволяет всегда быть в курсе достаточно большого объема информации из различных источников.

## За и Против предоставления фидов посетителям

Кто угодно может посетить ваш сайт. Точно также, кто угодно может подписаться на ваш сайт. В результате получается интересная и неоднозначная ситуация. Для сайтов, источником дохода которых является реклама, потеря посетителей это потеря доходов, т. к. подписчики не видят рекламу в rss-лентах.

В целом последствия таковы:

- Потеря трафика, меньше просмотров, кликов по рекламе и т. п.
- Потеря визуального брендинга, т. к. в фид-ридере все публикации выглядят однотипно и нет узнаваемого стиля сайта
- Воровство контента различными парсерами и грабберами рсс-лент

К счастью, есть несколько решений этих проблем. Но даже не взирая на все недостатки, рсс-ленты великолепный способ построить и расширить аудиторию сайта. Не будем пока забегать вперед и рассмотрим основные типы фидов, доступных в Вордпресе.

Важное замечание — не используйте относительные ссылки в своих записях!

Относительные ссылки вида «/category/post-123.html» отлично работают на вашем сайте, но становятся нерабочими когда включены в фиды. Т.к. фиды могут открываться в приложениях, браузерных читалках и различных десктопных программах, относительные ссылки и относительные адреса картинок приводят к неправильному формированию адресов. Если вы не можете сами исправить все адреса на своем блоге, то воспользуйтесь плагином **URL Absolutifier**

<http://scott.yang.id.au/pages/url-absolutifier.html> , который отфильтрует содержимое фидов и преобразует все относительные ссылки в абсолютные, содержащий полный адрес вашего сайта.

## Различные типы лент

Вордпресс генерирует кучу рсс-лент для всего, что только можно. Кроме этого можно создавать свои собственные ленты, в которые будут включены или наоборот исключены определенные теги и рубрики. И кроме этого, для поддержки большинства типов программ и читалок, вордпресс предлагает все фиды в множестве форматов: RSS, RDF, RSS2, и Atom. Все это может несколько

[illegible]



обескуражить, но давайте разберемся что к чему.

## Лента записей

Лента записей вашего блога или главная лента является основным источником контента для подписчиков. Если не ограничивать плагином или скриптом, каждый опубликованный пост появляется в этой ленте. И только содержимое записей появляется в ней. Обычно она доступна по следующему адресу <http://supersite.ru/feed/> и именно этот адрес подписчики добавляют в свои ридеры.

Вот небольшой обзор основных ридеров.

Выбрать есть из чего — различные онлайн сервисы, мобильные приложения и десктопные программы. Все они делают одну и ту же задачу — собирают и отображают содержимое фидов на которые вы подписались.

**Google Reader** некоторое время назад был самым популярным сервисом, однако по понятным лишь Гугл причинам был закрыт. На освободившееся поле боя ринулись конкуренты, одним из наиболее активно развивающимся стал Feedly <http://feedly.com>, позволяющий сделать импорт подписок из аккаунта гуглридера.

**Thunderbird** <http://www.mozilla.org/ru/thunderbird/> десктопное почтовое приложение, также позволяющие делать подписки на фиды любых форматов.

**Firefox Live Bookmark** <http://www.mozilla.org/ru/firefox/new/> Firefox не просто отличный браузер с кучей плагинов и возможностей, но и замечательно подходит для подписки и чтения лент с любимых сайтов, на которых нужно будет только выбрать «Подписаться на ленты новостей» в меню Закладки — и готово, вы всегда в курсе всех новостей.

## Лента комментариев

Лента с комментариями вашего сайта отображает в хронологическом порядке все опубликованные комментарии. Она доступна по адресу <http://supersite.ru/comments/feed/>

Многие сайты дают возможность подписаться на комментарии для того, чтобы вы не только читали содержимое сайта, но и вступали и поддерживали имеющиеся дискуссии и обсуждения.

### Индивидуальная лента комментариев записи

Для каждой записи на вашем сайте, открытой для комментирования, создается соответствующая лента комментариев, чтобы можно было подписаться и отслеживать дискуссию конкретно по этой записи. Каждая индивидуальная лента комментариев записи содержит все комментарии этой записи и находится по адресу типа такого

<http://supersite.ru/2009/10/billy-mays-fan-club/feed/>

Такие ленты тоже бывают достаточно востребованными, но достаточно часто вместо рсс-подписки на новые комментарии посетители предпочитают подписаться на новые комментарии на email. Далее будет рассказано как это сделать.

## Лента рубрик и тегов

Каждая отдельная рубрика или тег имеют свою собственную ленту. Этот фид включает в себя все записи соответствующей рубрики или тега. В общем виде адрес этих лент выглядит вот так

<http://supersite.ru/category/football/feed/>

ZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZ





рода война форматов с целью выявить окончательного победителя. К счастью, сейчас нет необходимости беспокоиться о том, в каком именно формате создается лента, т. к. популярные ридеры, а именно потому они и стали популярны, поддерживают все основные форматы. Большинство сайтов, которые мы посещаем и добавляем в фидридеры создают ленты в RSS 2.0 формате, а в формате Atom раздают совсем уж незначительное количество.

И как определить, в каком именно формате раздается лента? Ответ заложен в ее адресе:

- RSS 2.0 формат - <http://supersite.ru/feed/>
- RSS 2.0 формат - <http://supersite.ru/feed/rss2/>
- RSS 0.92 формат - <http://supersite.ru/feed/rss/>
- RDF/RSS 1.0 формат - <http://supersite.ru/feed/rdf/>
- Atom формат - <http://supersite.ru/feed/atom/>

Точно так же создаются ленты и для всех остальных типов фидов, путем дописывания своего типового адреса в урл. Например вот так будут выглядеть адреса лент в различных форматах для выбранной рубрики:

- <http://supersite.ru/category/coffee/feed/>
- <http://supersite.ru/category/coffee/feed/rss2/>
- <http://supersite.ru/category/coffee/feed/rss/>
- <http://supersite.ru/category/coffee/feed/rdf/>
- <http://supersite.ru/category/coffee/feed/atom/>

Можно заметить, что RSS 2.0 формат отдается в двух одинаковых версиях, что сделано для совместимости, но фактически создает дубли. Чтобы этого избежать, необходимо выполнить каноникализацию этих двух лент, перенаправив в одну. С точки зрения seo это очень хорошо, и делается довольно просто, используя редирект в htaccess файле:

RedirectMatch 301 \rss2\ http://supersite.ru/feed/

Это правило все урлы фидов, содержащие `rss2`, будет перенаправлять на основной фид. В 8 главе мы рассмотрим другие полезные приемы каноникализации ссылок.

Теперь мы знаем, что у нас на сайте вордпрессом созданы тысячи рсс-лент. Давайте разберемся как это все настроить и оптимизировать. Для начала стоит определиться какие же мы будем отдавать записи в ленту — полные или сокращенные.

## Полнотекстовые ленты

В настройках админки (Параметры — Чтение) есть опция - отдавать полный текст в ленту или только анонс. Если выбрать «Анонс», то в ленту будет отдавать текст записи, обрезанный по тегу `<!--more-->`, который разбивает текст з выводимую в блоге при клике по ссылке «Подр текст записи будет обрезан по определенное ко поля «Цитата записи», если оно заполнено.

Для каждой статьи в ленте отображать

☒ Полный текст  
☐ Анонс

С другой стороны, включив опцию «Полный текст» вы будете отдавать в ленту и, соответственно, читателям полный вариант записей, что делает ленту более привлекательной для подписчиков. Но

[illegible]



как говорилось выше, такую ленту проще «грабить», копируя в автоматическом режиме все содержимое и публикуя на других сайтах, наполняя интернет дублями текстов. Множество сайтов специально создано для того, чтобы грабить контент из чужих рсс-лент и зарабатывать размещая рекламу.

Как уже упоминалось, посетители не будут заходить на ваш сайт, если смогут прочитать весь текст статьи или новости в ридере. Если вы зарабатываете на рекламе, то это будут убытки. Несмотря на это, многие блогеры продолжают отдавать в ленты полные версии, т. к. распространение информации является для них ключевым моментом.

## Урезанные ленты

Для других же блогеров вышеприведенные недостатки перевешивают преимущества, поэтому они отдают урезанные версии рсс-лент. Обычно анонсы состоят из пары предложений и их задача пробудить интерес у читателя и заставить перейти на сайт, если для них это окажется нужным. Посещая ваш сайт они видят и кликают по вашей рекламе, поэтому уровень дохода не падает из-за раздачи информации в ленту.

Для различных граберов несколько предложений с вашего блога особого интереса не представляют, т. к. их сложно будет использовать для создания и автоматического наполнения сайтов. Конечно, используя определенные сервисы и программы, можно получить полный вариант текста, но обычно грабберы достаточно ленивы, чтобы этим заниматься, и им проще найти другой сайт для воровства.

Давайте подведем итоги. Показывать или не показывать полные версии — зависит только от вашего контента и модели заработка. Если вы зарабатываете на контекстной или баннерной рекламе, то для вас лучше, чтобы посетители приходили на сайт- из поиска или читалок. Если же для вас гораздо важнее распространение информации, которая построит растущий список подписчиков, а заработок происходит из публикуемых текстов (статьи с рефссылками, продажи различных товаров в обзорах и т. п.), то в этом случае выгоднее раздавать полные записи.

### Количество записей

Там же в админке, в настройках Чтения, есть настройка **«В RSS-лентах отображать последние ...»** позволяющая определить количество записей, видимых в ленте. По умолчанию стоит 10, однако это число можно настроить исходя из особенностей своего блога. Если вы публикуете часто и много, то лучше чтобы это число было больше. Если же обновления публикуются редко, то и количество одновременно видимых записей можно уменьшить.

Также необходимо учитывать вот еще что — чем больше записей будет отдаваться в ленту, тем больше трафика и создаваемой нагрузки на сервер будет больше при загрузке такого фида в ридер.

## Форматы ссылок лент

Если в блоге включены Постоянные ссылки — ЧПУ, то адрес фида будет выглядеть так

<http://supersite.ru/feed/> - RSS 2.0 format

<http://supersite.ru/feed/rss2/> - RSS 2.0 format

<http://supersite.ru/feed/rss/> - RSS 0.92 format

<http://supersite.ru/feed/rdf/> - RDF/RSS 1.0 format

<http://supersite.ru/feed/atom/> - Atom format

[illegible]



также доступен при включенных ЧПУ)

<http://supersite.ru/wp-rss2.php> - RSS 2.0 format

<http://supersite.ru/wp-rss.php> - RSS 0.92 format

<http://supersite.ru/wp-rdf.php> - RDF/RSS 1.0 format

<http://supersite.ru/wp-atom.php> - Atom format

Альтернативным способом обращения к лентам при выключенных ЧПУ будут такие адреса

<http://supersite.ru/?feed=rss2> - RSS 2.0 format

<http://supersite.ru/?feed=rss> - RSS 0.92 format

<http://supersite.ru/?feed=rdf> - RDF/RSS 1.0 format

<http://supersite.ru/?feed=atom> - Atom format

Чтобы указать какой именно стандарт будет использоваться в качестве стандарта по умолчанию, в хидере шаблона нужно указать тег

```
<?php bloginfo('rss2_url'); ?>
```

Для комментариев доступны ленты только в формате RSS 2.0, и фиды доступны по трем альтернативным адресам

<http://supersite.ru/comments/feed/>

<http://supersite.ru/wp-commentsrss2.php>

<http://supersite.ru/?feed=commentsrss2>

Чтобы указать ссылку по умолчанию на фид комментариев, нужно в хидер шаблона добавить тег

```
<?php bloginfo('comments_rss2_url'); ?>
```

### Фиды отдельных записей

По умолчанию каждая запись генерирует свой отдельный фид для своих комментариев. Чтобы отобразить ссылку для подписки на такой фид, в шаблон нужно вставить в основной цикл или в цикл комментариев

```
<?php comments_rss_link('Подписаться на комментарии по RSS'); ?>
```

В результате для каждой записи будет выведена своя ссылка.

## Фиды рубрик

Фид рубрик доступны по следующим адресам в зависимости от настроек ЧПУ

<http://supersite.ru/category/categoryname/feed/>

<http://supersite.ru/wp-rss2.php?cat=33>

В итоге — рсс лент в блоге не просто много, а очень много :)

zzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzz

Есть много способов отследить статистику фидов. Возможно самый популярный — это бесплатный сервис FeedBurner, который в свое время был куплен гугл. Миллионы владельцев сайтов и блогов перенаправляют свои фиды на фидбернер, который доставляет содержимое лент подписчикам, анализируя статистику, просмотры, клики и добавляет много различных дополнительных опций и возможностей.

Первое главное преимущество фидбернера- это его популярность. Хотя уже были прецеденты закрытия популярных сервисов с огромной аудиторией. Фидбернер основан в 2003 году и принадлежит гугл с 2007 года (ориентировочная сумма покупки 100млн. долларов) и пока нет тенденций к прекращению работы. За время использования периодически были падения сервиса и глюки, но гугл прилагает все усилия для поддержания его работоспособности. Одним из самых неприятных являются периодические слеты статистики подписчиков, нужные как для самих вебмастеров, так и для посетителей, которым нравится видеть большие числа на различных счетчиках — это добавляет сайтам авторитетности.

Analyze

Optimize

Publicize

Troubleshootize

VIEW

Feed Stats

Subscribers

Item Use

Map Overlay

Uncommon Uses

Export: Excel • CSV

SERVICES

Configure Stats

Feed Stats Dashboard

Date	Subscribers	Item Use
Dec 11, 2013	35	45
Dec 12, 2013	30	40
Dec 13, 2013	45	45
Dec 14, 2013	35	45
Dec 15, 2013	35	50
Dec 16, 2013	35	55
Dec 17, 2013	40	50
Dec 18, 2013	40	60
Dec 19, 2013	35	50
Dec 20, 2013	25	60
Dec 21, 2013	35	55
Dec 22, 2013	30	50
Dec 23, 2013	30	45
Dec 24, 2013	35	40
Dec 25, 2013	30	45

Thursday, January 9, 2014

♦ 32 subscribers ⓘ

♦ 15 reach ⓘ

- Впечатляющие инструменты анализа — обширная статистика подписчиков, кликов и другой активности
- Интеграция рекламы AdSense — гугл позволяет встраивать рекламу адсенс прямо в фиды
- Универсальный формат фидов — стандарт SmartFeed идеально работает с любыми читалками фидов
- Тонны оптимизационных настроек для настройки и улучшения фидов
- Инструменты публикации фидов — анимация заголовков, подписчики различных сервисов, встраивание логотипов, интеграция с другими сервисами и многое другое
- Легкий экспорт данных — статистика экспортируется в формат Экселя или CSV для анализа и изучения

ZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZ

лидером.

## Настройка и конфигурация аккаунта

Чтобы «зажечь» свой фид, нужно сделать следующее:

1. Зарегистрировать аккаунт Google (единый аккаунт для всех его сервисов)
2. Ввести урл своего фида, генерируемого блогом
3. Скопировать урл фидбернера, созданного из вашего исходного фида
4. Заменить или перенаправить исходный урл фида сайта на урл фида Фидбернера

Для выполнения последнего шага есть несколько способов и инструментов., включая перенаправление с помощью плагинов, htaccess или php. Смысл перенаправления — чтобы подписчикам отдавался фид фидбернера, даже если они будут использовать стандартный адрес стандартного фида Вордпресса.

## Перенаправление с помощью плагина

Можем предложить плагин **FD FeedBurner** <http://wordpress.org/plugins/feedburner-plugin/>, который будет выполнять автоматический редирект основного фида и при дополнительной настройке — фида комментариев. Скачайте, активируйте, укажите url фидбернер ленты и готово.

## Перенаправление с помощью htaccess

Один из недостатков использования плагинов заключается в том, что мы не контролируем, какая именно лента будет перенаправляться, а также не сможем перенаправлять несколько разных лент. При необходимости отдавать через фидбернер нескольких разных лент (например, фиды разных рубрик) плагины не помогут.

Допустим, по концепции сайта мы хотим раздавать и перенаправлять на фидбернер следующие фиды:

**Главный фид** <http://supersite.ru/feed/>=> <http://feeds.feedburner.com/main-feed>

**Фид комментариев** <http://supersite.ru/comments/feed/>=> <http://feeds.feedburner.com/comments-feed>

**Фид рубрики Бизнес** <http://supersite.ru/category/business/feed/> => <http://feeds.feedburner.com/business-feed>

**Фид рубрики Развлечения** <http://supersite.ru/category/pleasure/feed/> =>  
<http://feeds.feedburner.com/pleasure-feed>

На данный момент ни один из плагинов не может сделать так. Но с помощью директив размещенных в htaccess файле можно легко сделать сколько угодно редиректов

```
RewriteCond %{HTTP_USER_AGENT} !.*(FeedBurner|FeedValidator) [NC]
RewriteRule ^feed/?.*$ http://feeds.feedburner.com/main-feed [L,NC,R=302]
```

```
RewriteCond %{HTTP_USER_AGENT} !.*(FeedBurner|FeedValidator) [NC]
RewriteRule ^comments/?.*$ http://feeds.feedburner.com/comments-feed [L,NC,R=302]
```

```
RewriteCond %{HTTP_USER_AGENT} !^(FeedBurner|FeedValidator) [NC]
RewriteRule ^business/feed/?.*$ http://feeds.feedburner.com/business-feed [L,NC,R=302]
```

```
RewriteCond %{HTTP_USER_AGENT} !^.*(FeedBurner|FeedValidator) [NC]
RewriteRule ^pleasure/feed/?.*$ http://feeds.feedburner.com/pleasure-feed [L,NC,R=302]
```

[illegible]



Обратите внимание, что даже для перенаправления всего одного главного фида вам не нужен плагин. А любые новые правила будут добавляться простым копированием. С такими правилами все запросы будут перенаправляться на соответствующие адреса фидов в фидбернере. Ну а минимальное количество плагинов- это как мы с вами помним очень хорошо!

## Перенаправление с помощью RNR

Если правила и директивы htaccess бросают вас в дрожь — есть отличный альтернативный способ редиректов с помощью php.

Для начала нужно добавить в файл вашей темы `functions.php` следующий код:

```
function custom_feed_link($output, $feed) {
    $feed_url = 'http://feeds.feedburner.com/your-feedburner-feed';
    $feed_array = array(
        'rss' => $feed_url,
        'rss2' => $feed_url,
        'atom' => $feed_url,
        'rdf' => $feed_url,
        'comments_rss2' => "
    );
    $feed_array[$feed] = $feed_url;
    $output = $feed_array[$feed];
    return $output;
}
add_filter('feed link','custom feed link', 1, 2);
```

Сразу после размещения этот код будет перенаправлять все варианты ссылок главного фида во всех форматах на один соответствующий url фида в фидбернере. Точно также можно перенаправлять любые другие типы лент. Например этот код будет перенаправлять фид определенной рубрики

```
function other_feed_links($link) {
$link = 'http://feeds.feedburner.com/your-feedburner-feed';
return $link;
}
add_filter('category_feed_link', 'other_feed_links');
```

Если нужно перенаправить фид автора, то нужно использовать в последней строке фильтр `add filter('author feed link', 'other feed links');`

Если фид тега, то фильтр  
add filter('tag feed link','other feed links');

Ну и нужно еще раз понять, что данный код не заменяет стандартные ссылки. Формально они остаются прежними, но для браузеров и приложений по чтению фидов сообщается, где находится адрес фида который им нужен, и этот новый адрес — урл фидбернера.

## Отслеживание и отображение статистики фидов

Как только вы создадите и правильно настроите ленту, вы захотите предложить ее посетителям. Есть много способов сделать это, включая текстовые ссылки, кнопки-картинки и даже иконки фидбернера или счетчики, которые показывают общее количество подписчиков вашего фида.

ZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZ

Вместе с числом подписчиков, сервис отслеживает множество другой полезной статистической информации.

## Типы статистики, предлагаемой фидбернером

Как говорилось, фидбернер в одном флаконе предлагает массу полезной информации:

- Subscriber counts – сумма за день хронологическая история изменения подписчиков
- Feed reader applications — использование приложений для просмотра ленты
- Uncommon uses — нестандартное использование, включая граббинг
- Reach – число уникальных просмотров или кликов по содержимому фида
- Item views – отслеживание доступа к указанным элементам фида
- Item link clicks — оптимизация для поисковиков
- Item enclosure downloads — скачивания с помощью фида

Фидбернер собирает много статистики, особенно учитывая, что это подразделение гугл. Многое из данных включено по умолчанию, еще больше доступно опционально, путем дополнительных настроек при необходимости.

## Отображение статистики Фидбернера



Многие из пользователей фидбернара на самом деле используют его только ради виджета счетчика подписчиков, не подозревая или не нуждаясь в дополнительных функциях. В очень большом количестве сайдбаров или футеров можно увидеть эти знакомые разноцветные прямоугольники.

В зависимости от дизайна сайта, может понадобится особенный способ расположения и цветовой гаммы виджета. Кроме цветного счетчика, также есть текстовый способ отображения, позволяющий применить к нему свои собственные стили и разметку.

### Метод 1. Используем PHP

С помощью FeedBurner's Awareness API и щепотки PHP можно вывести счетчик фидбернера как текст. Вот как это делается.

В нужном места шаблона размещается следующий код для вызова и отображения счетчика

```
<?php // display FeedBurner count as plain text
$feed = "https://feedburner.google.com/api/awareness/1.0/GetFeedData?uri=digwp";
$curl = curl_init();
curl_setopt($curl, CURLOPT_RETURNTRANSFER, 1);
curl_setopt($curl, CURLOPT_URL, $feed);
$feed = curl_exec($curl);
curl_close($curl);
$xml = new SimpleXMLElement($feed);
$feedburner = $xml->feed->entry['circulation'];
echo $feedburner;
?>
```

Вместо `uri=digwp` укажите свой логин в аккаунт фидбернера

Все! Или вы думали что будет что-то еще :) Останется только применить нужные стили по своему вкусу.

[illegible]



## Метод 2. Используем плагин

Если вам проще скачать и активировать, чем отредактировать и вставить, то вот вам плагин **Feed Count** <http://www.mapelli.info/feed-count-12> . С помощью этого плагина можно легко настроить текстовое отображение счетчика и украсить его по желанию.

## Альтернативы фидбернеру

Обратной стороной популярности фидбернера является огромнейшая база его пользователей, создающих соответствующую нагрузку. Из-за периодической недоступности сервиса, некоторые пользователи ищут альтернативы. Одно дело видеть нули на счетчике, вместо реальных цифр подписчиков. И совсем другое дело, если эти подписчики не видят ленту с сообщениями сайта.

## Сервисы-аналоги фидбернера:

**FeedBlitz** <http://www.feedblitz.com/feeburner-alternative-blitz-your-rss/>

**RapidFeeds** <http://www.rapidfeeds.com/>

Плагин статистики фидов **FeedStats** <http://wordpress.org/plugins/feedstats-de/> анализирует общее число подписчиков на различные фиды блога.

Плагин мониторинга фидов **Feed Statistics** <http://wordpress.org/plugins/wordpress-feed-statistics/> также наблюдает и докладывает о читателях: откуда и как пришли, что смотрели, где кликали.

## Настраиваем ленту

После того, как фид создан, есть много способов настроить его внешний вид и функционал. Если вы используете фидбернер, то достаточно просто войти в аккаунт и сделать нужные правки. А если нет?

Ничего страшного, своими силами мы можем сделать еще лучше, еще больше, еще круче.

## Форматируем изображения фидов

Изображение, вызываемые тегом `` по сути являются строчным элементом. Изображение имеет высоту и ширину, но оно не разрывает строку и идет сразу после текста, а не с новой строки. А после изображения текст продолжает идти дальше в ряд, а не обтекает его. Поэтому не зависимо от стиля на сайте, в rss читалке запись будет выглядеть не так как на сайте.

**На сайте:**



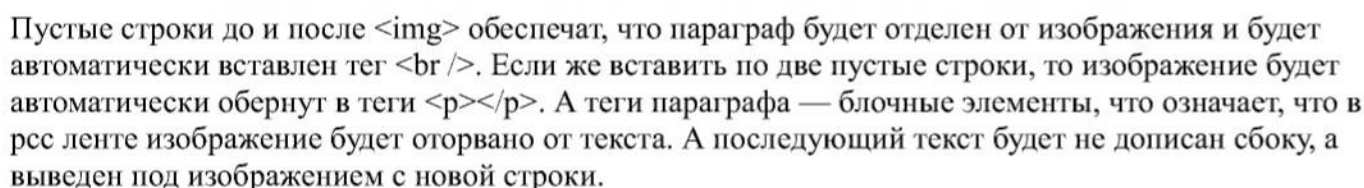
**В читалке:**



zzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzz



Вот пример того, о чем идет речь:



[Go to the top](#)

## Добавляем свое изображение для ленты

```
function mwp_feedImage() {
    echo "<image>
    <title>Digging Into WordPress</title>
    <url>http://digwp.com/images/feed-icon.png</url>
    <link>http://digwp.com/</link>
    <width>150</width><height>150</height>
    <description>Digging Into WordPress</description>
    </image>";
}

add_action('rss2_head', 'mwp_feedImage');
```

zzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzz

Брендинг логотипом или характерной картинкой добавит узнаваемости и запоминаемости для вашего сайта. Пользователи фидберна могут добавить свое изображение в меню Feed Image Burne.

Т.к. вордпресс по умолчанию исключает комментарии из главного фида, для некоторых сайтов может оказаться нужным выводить в одном месте и записи, и комментарии к ним. Простым и оригинальным способом сделать это будет использование сервиса [Yahoo! Pipes](#). Звучит устрашающе, но на самом деле все просто.

- 
- Fetch Feed
- URL
- urner.com/DiggingInto Wordpress
- http://digwp.com/comments/feed/
- Add your Posts Feed URL and Comments Feed URL to the Fetch Feed Badge
- Sort
- Sort by
- item.pubDate in ascending order
- Specify sort order as "item.pubDate" in the Sort Badge
- Pipe Output
- Click on the Pipe Output Badge to see the output results
- Time taken: 0.132856s [Refresh](#)
- WordPress Debug Theme
- Ideas for Plugins I'm Too Lazy To Write
- Comment on New Poll: How Many Plugins do You Use?
- Comment on New Poll: How Many Plugins do You Use?
- Comment on Plugin Wishes Come True by Latz
- Poll Results: Goodbye Dolly

**ZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZ**



И все. У вас есть объединенный фид, состоящий из самых последних записей вперемешку с последними комментариями в хронологическом порядке. Конечно, с помощью Yahoo! Pipes можно создавать очень сложные и разнообразные сервисы, при этом совершенно бесплатно. Поэтому изучайте его возможности и экспериментируйте.

## Создаем пользовательские фиды

**Исключаем рубрики, авторов и все остальное ненужное**

где sat означает, что мы исключаем категорию, а x,y,z номера исключенных рубрик. После этого записи этих рубрик не будут попадать в главный фид. Точно так же можно исключить любой другой ненужный элемент

В процессе создания таких пользовательских лент верным будет самому на них подписаться, чтобы убедиться что фильтр работает как надо. А также проверить отображение содержимого фида в разных читалках.

ZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZ





## Стилизация фидов

Как выше показывалось, самый простой способ добавить огня — это вписывать стили непосредственно в текст записи. Плагин **Feedstyler** <http://www.devlounge.net/general/style-your-feeds-with-wp-feed-styler-v10> позволит все это сделать быстро и легко.



## Удаляем версию Вордпресс

```
function killVersion() { return "; }  
add filter('the generator','killVersion');
```

```
remove action('wp head', 'wp generator');
```

ZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZ

Чтобы удалить остальную ненужную или нежелательную информацию из секции `<head>` сайта, также можно добавить следующее

```
remove_action('wp_head', 'rsd_link');
remove_action('wp_head', 'wp_generator');
remove_action('wp_head', 'feed_links', 2);
remove_action('wp_head', 'index_rel_link');
remove_action('wp_head', 'wlwmanifest_link');
remove_action('wp_head', 'feed_links_extra', 3);
remove_action('wp_head', 'start_post_rel_link', 10, 0);
remove_action('wp_head', 'parent_post_rel_link', 10, 0);
remove_action('wp_head', 'adjacent_posts_rel_link_wp_head', 10, 0);
```

Однако по мере выхода новых версий вордпресс часть этих функций может потерять актуальность.

## Отключение и перенаправление нежелательных форматов лент

Как мы знаем, вордпресс генерирует огромное количество рсс лент разных форматов. Чтобы навести немного порядка, стоит в `functions.php` добавить следующее

```
function redirectFeed() {
wp_die();header("Location: http://digwp.com/feed/");exit();}
```

Укажите урл, на который вы бы хотели, чтобы перенаправлялись вызовы ненужных форматов лент. А теперь задайте, какие именно форматы будут перенаправлены

```
add_action('do_feed', 'redirectFeed', 1); // все фиды
add_action('do_feed_rdf', 'redirectFeed', 1); // RDF (RSS 0.92)
add_action('do_feed_rss', 'redirectFeed', 1); // RSS 1.0 feed
add_action('do_feed_rss2', 'redirectFeed', 1); // RSS 2.0 feed
add_action('do_feed_atom', 'redirectFeed', 1); // Atom feed
```

## Добавляем пользовательскую информацию в фид

Добавлять свой собственный дополнительный контент очень просто. Это может быть текст, стили и даже скрипты, которые будут отображаться в тексте записи, в rss-ленте или в любом месте сайта. По желанию, можно задать контент, который будет отображать или в фиде, или в записи на сайте, или в обоих местах одновременно.

Хорошим примером может служить добавление информации о авторе или копирайтах в футер фида и рекламу до или после каждого поста в фиде. Возможности реально неограниченны.

Чтобы добавить пользовательский контент в фид, разместите в нашем любимом `functions.php` следующее

```
function insertContent($content) {
    $content = $content . '<p>А тут мы написали что-то интересное!</p>';
    return $content;
}
add_filter('the_excerpt_rss', 'insertContent');
add_filter('the_content_rss', 'insertContent');
```







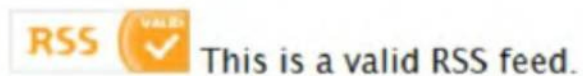
фи́ды создаются и доставляются такими, как надо. Меньше всего вам хотелось бы тратить время на то, чтобы писать контент, который не будет отображаться в рсс-ленте из-за ошибок. Есть огромное количество онлайн сервисов, которые проверяют фи́ды на наличие ошибок.

Вот некоторые из них:

<http://feedvalidator.org/>

<http://validator.w3.org/feed/>

Просто вставьте урл своего фида и проверьте. Если все в порядке — вы увидите оптимистичную надпись и картинку

[illegible]



## 7. Работаем с комментариями

## Оптимизируем зону комментариев

Комментарии — это то, что оживит любой сайт. И в зоне комментариев есть очень много того, что можно улучшить и оптимизировать, поэтому выделим этому целую главу. Изначально вордпресс предоставляет все необходимое для создания комментирования и обратной связи, надежную систему для обслуживания больших объемов комментариев, пингбеков и трекбеков. Однако, есть над чем поработать: отображение, функционал и безопасность.

**Добро пожаловать в зону комментариев**

Блогосфера живет за счет своей интерактивности. Публикуя контент с технологиями типа Вордпресс, очень легко дать возможность миллионам людей выражать себя, распространять новости и создавать огромное количество различного контента. Интерактивность, даваемая вордпрессом, соединяет авторов, пользователей и поисковые машины. Зона комментариев с помощью нескольких технологий, пары плагинов и некоторых знаний легко трансформируется в интерактивную систему обратной связи.

Это нереально круто для удобства пользования сайтом, роста его популярности и привлекательности.

Чтобы вы не публиковали, вордпресс автоматически делает это доступным для посетителей и читателей, предоставляя возможность отреагировать несколькими способами.

Изначально, есть три способа для читателей сайта. Они могут или оставить комментарий непосредственно на вашем сайте, сделать трек-бек из статьи на своем сайте или просто написать о вашем посте в своем блоге.

Все это вместе составляет типовое содержимое зоны комментариев.



## Комментарии, Пингбеки, Трекбеки

Вордпресс поддерживает три разных типа комментариев. Самый распространенный вид комментариев происходит так: посетитель заходит на ваш сайт, читает ваш пост и имеет что сказать по этому поводу. Он находит соответствующую форму, обычно размещаемую под статьей. В ней он вбивает свое имя, емейл, ссылку на сайт и собственно текст высказывания. После этого нажимает на кнопку «Отправить» и это вот и есть классический комментарий, который будет опубликован на сайте автоматически или после прохождения модерации админом.

Второй тип комментариев называется пингбек, и по своей сути больше ответ, чем комментарий. Допустим, вы написали заметку про свинину и какой-нибудь другой кулинарный сайт решил рассказать своим читателям о вашей шикарной статье. Они публикуют у себя свою заметку, которая содержит ссылку на вашу, чтобы их читатели могли перейти на ваш сайт и прикоснуться к прекрасному. В этом случае вордпресс обнаруживает такую входящую ссылку и взаимно публикует в виде комментария обратную ссылку на этот сайт. Обратная ссылки обычно отображается в виде заголовка ссылающегося поста в качестве анкера.

zzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzz

Пингбек требует фактического наличия ссылки на ваш сайт с другого сайта. Другой сайт ссылается на вас, вы в ответ автоматически ссылаетесь на него. Логика такая, мол посмотрите, про нас написали где-то еще- можете перейти по ссылке и посмотреть что там.

Третий тип отклика называется трекбек, который больше является именно откликом, чем настоящим комментарием. Для иллюстрации, допустим на кулинарном сайте упомянули о вашей чудо-статье про свинину, но не поставили ссылку на ваш сайт. Если они создадут трекбек на вашу статью (через админку в редакторе записей, заполнив соответствующее поле ссылкой), вы автоматически опубликуете обратную ссылку на эту статью. По своей сути пингбек и трекбек очень близки, но трекбек не требует наличия реальной обратной ссылки в тексте статьи. Но результат одинаков, и пингбек и трекбек ссылается на сайт, на котором сослались на нас.

В теории, это очень крутая технология, изучение сути которой несколько выходит за рамки книги. Важно запомнить и понять следующее о различных типах реакции на ваш сайт:

- **Комментарий** — оставляется посетителями непосредственно на вашем сайте
- **Пингбек** — автоматически создаваемая обратная ссылка на чужой блог
- **Трекбек** — автоматически создаваемая обратная ссылка, не требующая фактического размещения ссылки на вас

Технически, трекбеки и пингбеки не являются комментариями, однако множество сайтов отображают и учитывают их вместе. Поэтому несколько странновато выглядят статьи с примечанием «Эта запись содержит 15 комментариев». Более правильно было бы назвать «Эта запись содержит 15 откликов».

## Анатомия зоны комментариев

Есть четыре основных поля и кнопка, присутствующих на большинстве сайтов. Имя, емейл, сайт, комментариев и Отправить. На первый взгляд выглядит достаточным, однако давайте расширим функциональность и возможности этой формы.

Создатели вордпресс предполагали, что создание формы комментариев у многих может вызвать определенные трудности, поэтому упростили все настолько, насколько это возможно. Например, весь код, необходимый для создания системы комментирования размещен в одном файле `comments.php`, а для добавления формы комментариев на страницу достаточно вписать всего одну строчку

```
<?php comments template(); ?>
```

Этот тег обычно находится в файле `single.php`, т. к. в комментировании нуждаются прежде всего одиночные публикации, но его можно включить на любую нужную страницу.

Теперь мы знаем где находятся комментарии и как включаются, давайте разберемся с зоной комментирования — что и как выводится в ней:

**ZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZ**

Leave a Comment

<input type="text"/>	Name *
<input type="text"/>	E-mail *
<input type="text"/>	Website

☐ Notify me of followup comments via e-mailSubmit



Зона отображения комментариев, или дерево комментариев, может быть настроена для отображения комментариев в хронологическом порядке (самые старые комментарии отображаются первыми, свежие- последними) или в обратном порядке, что определяется настройками

Классическим порядком размещения комментариев считается «сначала поздние». Логика такая — посетителю будет удобнее сначала прочитать с чего началась дискуссия, а затем читать дальше что наотвечали другие участники. Размещение сначала ранних комментариев имеет смысл в том случае, если комментарии не являются дискуссией и ответом один на другой, а каждый не зависит один от другого. Тогда удобнее читать сначала новые комментарии.

В зоне комментариев могут отображаться комментарии, трекбеки и пингбеки как вместе вперемешку, так и разделенные на свои участки или вкладки.

Граватары — это такие маленькие иконки, связанные с каждым автором, оставившим комментарий. По умолчанию вордпресс может отображать граватары и для каждого пользователя выводится случайная картинка, если он не зарегистрирован в сервисе <http://ru.gravatar.com/>. В этом случае опознается емейл, на который зарегистрирован аккаунт граватара и привязана своя собственная аватара, и рядом с комментарием отображается именно она.

Для большинства форм комментирования обязательными являются имя, емейл и текст комментария. Поле емейл дает возможность автору блога связаться с комментатором, однако из соображений безопасности явно нигде на странице он не публикуется. Поле ссылка обычно нужна для того, чтобы комментатор указал свой сайт и не является обязательной, т. к. свой сайт есть не у всех.

Так как вы публикуете то, что никого не может оставить равнодушным — на ваш сайт обрушиваются тонны комментариев, а также трекбеков и пингбеков. Поскольку общение развивается, дайте возможность читателям и комментаторам быть в курсе развития дискуссии, публикуя свежие комментарии в рсс-ленту. По умолчанию большинство шаблонов поддерживает такую возможность, но если нет, давайте разберемся как это добавить.



## Главный фид комментариев

Как мы помним, фиды есть разные и их есть много. Для комментариев доступны три основных фида

<http://supersite.ru/comments/feed/> - формат чпу

<http://supersite.ru/wp-commentsrss2.php> — формат по умолчанию

<http://supersite.ru/?feed=commentsrss2> — формат запроса

Но зачем иметь фид со всеми комментариями? Это потому что вордпресс пытается предоставить всю доступную и возможную информацию. Но нам как пользователям следить за потоком несвязных комментариев, собранных со всего сайта, будет крайне неудобно. Возможно какой-нибудь супер-фан и будет в восторге от того, что он следит за всем на вашем сайте, но большинству необходимо отслеживать именно ту дискуссию, которая им интересна. Поэтому нужны фиды с привязкой к конкретно взятым записям.

### Лента комментариев отдельных записей

Каждый пост имеет свою ссылку, а чтобы получить ссылку на его фид достаточно приписать к урлу /feed/ - очень просто и легко запомнить.

Например так:

<http://supersite.ru/alligator-tacos/> - ссылка на запись

<http://supersite.ru/alligator-tacos/feed/> - ссылка на фид комментариев

Таким образом можно давать возможность посетителям подписывать на конкретно взятый отдельный фид. Они доступны для всех, даже если посетитель не оставил ни одного комментария и возможно и не собирается оставлять.

При использовании стандартных ссылок — без чпу, ссылку на фид получают следующим образом

<http://supersite.ru/?p=123&feed=rss2>

Хорошим местом для размещения ссылки подписки на фид комментариев является место перед или после формы комментариев. Это лишний раз напомним комментатору о возможности подписаться на обсуждаемую тему.

В дополнение к подписке на фид, можно предложить возможность подписаться на новые комментарии через емейл. Часто получение уведомлений о новых комментариях на емейл легче, чем процесс подписки на фид и периодический просмотр читалки. Изначально такая возможность в вордпресс отсутствует, но хорошие ребята сделали плагин **Subscribe to Comments** <http://wordpress.org/plugins/subscribe-to-comments/>

Оставляя комментарий, посетитель имеет возможность поставить галочку в чек-боксе для того, чтобы получать уведомления на почту о всех последующих комментариях сразу же, как только они будут опубликованы.

[illegible]

## Форматирование зоны комментариев

Лезть в код или не лезть? Раньше мы уже говорили, что форма комментариев немного сложна, и поэтому создатели все максимально упростили.

Функция вызова комментариев простая и не имеет параметров `<?php wp_list_comments(); ?>`

Выглядит просто, но делает очень много. Фактически она создает огромное количество html кода и использует кучу php кода. Т.к. эта функция генерирует весь готовый html код, то нам придется иметь с ним дело без возможности как то отрегулировать или настроить использованием параметров. Это может быть определенной проблемой, с учетом того что мы не очень можем этот html код изменить. Не нравятся классы div? Не хотите использовать граватары? Желаете изменить порядок вывода строк?

Со всем этим придется смириться и скрывать ненужные вещи с помощью `css`. С этой точки зрения вам нужно будет определиться- по какому пути предпочтений вы пойдете:

**wp\_list\_comments**

- легко использовать
- встроенный функционал комментирования
- встроенный функционал разбития на страницы
- встроенный функционал древовидного отображения
- css классы по умолчанию для базовых задач

## пользовательская функция комментариев

- абсолютный полный контроль над всем
- стандартные и пользовательские CSS классы
- наличие плагинов для добавления нужного функционала

Выбор метода формирования комментариев зависит от множества факторов. В общем, мы рекомендуем использовать `wp_list_comments` и не морочить голову. Если же вам нужно сделать что-то свое, особенное и специфическое, с желаемым функционалом и возможностями, и это желание подкрепляется знаниями, то использование пользовательских циклов даст возможность это все воплотить.

```

class="commentlist">
  <div class="comment even thread-even" id="div-comment-2083">
    <div class="comment-author">
      
      <cite class="fn"><a href="http://www.gravatar.com/avatar/00000000000000000000000000000000" data-retina="true" data-size="40" data-default="mm-gravatar" data-cs="2" data-kind="parent" data-rs="2" data-bbox="100 100 140 140" data-kind="ghost">
    </div>
    <div class="comment-meta">
      <p>This just happened to me</p>
    <div class="reply">
      <a href="#" rel="nofollow" class="comment-reply-link">
    </div>
  </div>
  <ul class="children">
    <li class="comment byuser">
      <div id="div-comment-2084">
        <div class="comment-author">
          
          <cite class="fn"><a href="http://www.gravatar.com/avatar/00000000000000000000000000000000" data-retina="true" data-size="40" data-default="mm-gravatar" data-cs="2" data-kind="parent" data-rs="2" data-bbox="100 100 140 140" data-kind="ghost">
        </div>
        <div class="comment-meta">
          <p>That's the whole point</p>
        <div class="reply">
          <a href="#" rel="nofollow" class="comment-reply-link">
        </div>
      </div>
    </li>
  </ul>
</div>

```

////////////////////////////////////

Кроме этого есть старый добрый метод вывода комментариев с помощью цикла `foreach`, поэтому нам еще есть с чем разобраться.

**wp\_list\_comments()** или пользовательский цикл?

До версии 2.7 цикл вывода комментариев был на базе запросов к базе данных и отображался несколько извилистым, но хорошо понятным путем.

```
<?php foreach ($comments as $comment) : ?>
    <?php comment_author_link(); ?>
    <?php comment_text() ?>
<?php endforeach; ?>
```

Затем, комментарии были обогащены дополнительным функционалом — постраничной навигацией и древовидной вложенностью. Эти возможности активировались через админку и цикл `foreach` был заменен на уже показанную функцию `wp_list_comments`. Однако этот цикл по-прежнему остается работоспособен, правда по понятной причине не поддерживая дополнительный функционал.

Хотя `wp_list_comments` и не выглядит как цикл, но она делает все тоже самое, повторяет раз за разом запросы и вывод комментариев, пока они не закончатся, размечая их соответствующими тегами.

Обычно готовый HTML код одного комментария выглядит примерно так:

```
<ol class="commentlist">
<li class="comment even thread-even depth-1" id="comment-1">
<div id="div-comment-1" class="comment-body">
<div class="comment-author vcard">

<cite class="fn">
<a href='http://wordpress.org/' rel='external nofollow' class='url'>Имя комментатора</a></cite> <span
class="says">says:</span>
</div>
<div class="comment-meta commentmetadata">
<a href="http://localhost/283/2009/08/hello-world/comment-page-1/#comment-1">August 13, 2009 at
3:08 pm</a>
</div>
<p>Спасибо, у вас супер сайт!</p>
<div class="reply"></div>
</div>
</li>
</ol>
```

Как видно, цикл вывода комментария содержит картинку-граватар, несколько дивов, много разных CSS классов, ссылку на автора и собственно текст комментария. Если это вас устраивает — то это хорошо, больше ничего делать и не нужно.

В этом случае вы можете протянуть свои руки только к следующим доступным параметрам wp list comments :

- размер аватары - по умолчанию 32 пикселя

~~zzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzz~~



- **СТИЛИ CSS**— используя их названия можно изменить параметры на нужные

Это немного, но в большинстве случаев достаточно для минимально необходимых модификаций. Если нужно больше, то придется использовать пользовательский цикл либо вмешиваться в php код, написанный в `comments.php`, либо добавлять нужный функционал в `functions.php`.

## Шаманим в functions.php

Для начала укажем в `comments.php`, что мы будем использовать свою пользовательскую функцию

```
<ul class="commentlist">
<?php wp_list_comments('type=comment&callback=mytheme_comment');?>
</ul>
```

Теперь создадим в `functions.php` свою пользовательскую функцию

```
<?php function mytheme_comment($comment, $args, $depth) { $GLOBALS['comment'] = $comment; ?>
<li <?php comment_class(); ?> id="li-comment-<?php comment_ID(); ?>">
<div id="comment-<?php comment_ID(); ?>">
<div class="comment-author vcard">
<?php echo get_avatar($comment,$size='48',$default='<path_to_url>');?>
<?php printf(__('<cite class="fn">%s</cite>
<span class="says">says:</span>'), get_comment_author_link()); ?>
</div>
<?php if ($comment->comment_approved == '0') : ?>
<em><?php _e('Your comment is awaiting moderation.');?></em>
<?php endif; ?>
<div class="comment-meta commentmetadata">
<a href="<?php echo htmlspecialchars( get_comment_link( $comment->comment_ID ) ) ?>">
<?php printf(__('%1$s at %2$s'), get_comment_date(), get_comment_time()) ?>
</a><?php edit_comment_link(__('(Edit)'),',,') ?></div>
<?php comment_text(); ?>
```

ZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZ





И здесь есть один подвод. Возникает известная seo проблема дублирования контента с постраничными комментариями. Это происходит из-за того, что создаются одинаковые урлы для каждой страницы комментариев

http://yoursite.com/ghost-dad-rocked/  
http://yoursite.com/ghost-dad-rocked/comments-page-1/  
http://yoursite.com/ghost-dad-rocked/comments-page-2/  
http://yoursite.com/ghost-dad-rocked/comments-page-3/

Понятно, что на этих страницах разными будут только комментарии. Что еще больше ухудшает общую картину — каждый комментарий имеет свою собственную уникальную ссылку с временем публикации, что еще добавляет пачку дублей. Поэтому поисковый бот обнаружит один и тот же контент под огромным количеством разнообразных ссылок.

Чтобы избежать дублирования, есть несколько решений:

1. Не использовать постраничные комментарии
2. Использовать мета-теги для создания канонических урл для каждой записи
3. Использовать плагин **SEO for Paged Comments** <http://wordpress.org/plugins/seo-for-paged-comments/>

Если ваш сайт супер-популярный или собирается таким стать, то конечно лучше использовать канонические урлы. Если же бурные обсуждения не планируются, то проще использовать традиционные простынки комментариев без разбития на страницы.

SEO for Paged Comments применяет еще более радикальный способ — убирая содержимое записи с страниц комментариев и заменяя на цитату или урезанную версию записи.

## Реализуем вложенные комментарии

Вложенные древовидные комментарии позволяют посетителям писать ответ выбранному существующему комментарию, создавая развернутое обсуждение-диалог. После активации этой возможности в настройках, возле каждого комментария появится ссылка «Ответить», позволяющая высказаться по конкретному комментарию.

До появления такой возможности существовал стиль ответа выбранному автору, используя в тексте комментария его имя и знак @, типа «@Вася ты не прав!» С некоторой трансформацией этот способ прижился в многих приложениях в интернете и живет до сих пор, например в твиттере. А в современных версиях вордпресса такое упоминание не нужно использовать.

Теперь достаточно просто нажать на соответствующую ссылку «Ответить» и высказаться. После одобрения текста комментария он появится сразу же под комментарием, на который был написан.

Если ваш шаблон не поддерживает такой вид комментирования, есть легкий способ добавить такую возможность:

1. Делаем резервные копии файлов, что по идее должно уже войти в привычку
2. Включаем вложенные комментарии в настройках Параметры- Настройки обсуждения — Разрешить древовидные комментарии

ZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZ




3. Добавляем следующий код в header.php сразу же перед wp\_head()  
`<?php if (is_singular() AND comments_open() AND (get_option('thread_comments') == 1))  
wp_enqueue_script('comment-reply'); ?>`
4. Добавляем в comments.php непосредственно перед закрывающим тегом `</form>`  
`<?php comment_id_fields(); ?>`
5. Проверяем, что `<textarea>` шаблона содержит атрибут `id="comment"`
6. Проверяем, что форма комментариев заключена в `<div>` с атрибутом `id="respond"`
7. Добавляем ссылку отмены комментария прямо перед формой комментариев  
`<div><?php cancel_comment_reply_link(); ?></div>`
8. В comments.php заменяем старый цикл на обновленный

Ну и собственно все :) Традиционно есть триллион различных способов, чтобы настроить и переделать под свои нужды, но суть я думаю вы уловили. Останется только не забыть настроить стили для вложенных комментариев

## Разделяем комментарии, пингбеки и трекбеки

Это один из самых старых трюков и есть много способов его воплотить, вот вам пример простого разделения путем применения параметров `wp list comments()`:

ZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZ

 **#35 — Sunny Seneviratne**  
I am trying to find a technique  
characters..If any of you know  
let me know. Thanks!

## Trackbacks / Pingbacks



```
<?php wp_list_comments('type=comment'); ?>
```

## Удаляем wp-trackback.php

Это очень простой и эффективный способ — физическое удаление файла `wp-trackback.php` из корневой директории вордпресса. Это абсолютно безопасный способ, но файл восстанавливается после апгрейда вордпресс и его нужно будет удалить заново.

## Отключаем с помощью плагинов

Это можно сделать с помощью плагина **Disable Trackbacks** <http://wordpress.org/plugins/disable-trackbacks/>.

Сам по себе плагин очень простой и состоит всего из одного файла, поэтому его содержимое можно просто скопировать и вставить в файл `functions.php`

```
class DisableTrackbacks {
function DisableTrackbacks() {add_action('pings_open', array(&$this, 'pings_open'));}
function pings_open($open) {return ('1' == get_query_var('tb')) ? FALSE : $open;}
}
```

```
add_action('plugins_loaded', create_function("", 'global $DisableTrackbacks; $DisableTrackbacks = new
DisableTrackbacks();'));
```

После размещения, функция `DisableTrackbacks` уничтожает все трекбеки. А пингбеки остаются.

## Контролируем комментарии непосредственно в базе данных

Вордпресс предлагает несколько способов контроля над тем, открыта ли запись для комментирования или нет. В админке можно полностью запретить на всем сайте оставлять комментарии к будущим записям. Также в отдельно взятой записи в редакторе можно выбрать в блоке Обсуждения будет ли к ней открыто комментирование или нет.

Но прямо над этими способами находится плагин **Auto Moderate Comments** <http://wordpress.org/plugins/ozh-auto-moderate-comments/> позволяющий все старые комментарии по истечении заданного срока скрывать, отправляя на модерацию. И без всяких плагинов включать\отключать комментирование можно простыми запросами к базе данных.

## Глобально разрешаем пинги для всех пользователей

```
UPDATE wp_posts SET ping_status = 'open';
```

## Глобально запрещаем пинги для всех пользователей

```
UPDATE wp_posts SET ping_status = 'closed';
```

## Глобально запрещаем пинги до определенной даты публикации

```
UPDATE wp_posts SET ping_status = 'closed' WHERE post_date < '2009-09-09' AND post_status = 'publish';
```

Такие запросы позволяют периодически проводить чистку. А вот пример запросов для глобального разрешения или запретов комментирования

ZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZ



////////////////////////////////////



## Изменяем стиль комментариев

Другой способ улучшить общее восприятие и стилистику зоны комментариев, это изменить отображение чередующихся комментариев, выделяя четные или нечетные другим фоном. В зависимости от каждого отдельно взятого дизайна, этот метод контрастного выделения цветом соседних комментариев может значительно повысить читаемость и улучшить восприятие.

Процесс стилизации чередующихся комментариев сильно зависит от того, какие комментарии считать четными, а какие нечетными. Вордпресс «автоматически» помечает комментарии классами `odd`(нечетный) и `even`(четный), снимая головную боль по всем вычислениям. Остается только определить нужные нам стили, например вот такой вот очень контрастный стиль

```
.odd {background: white;color: black;}
.even {background: black;color: white;}
```

И все. Реально все :)

## Отдельный стиль комментариев для зарегистрированных авторов

Похожая техника используется для стилизации комментариев, оставленных авторами и зарегистрированными пользователями. Как и в случае чередующихся стилей, выделение текстов комментариев авторов позволяет указать нужные акценты и выделить их. И точно также вордпресс проделывает для нас всю грязную работу.

Для авторов записей вордпресс добавляет класс “bypostauthor”, а для зарегистрированных пользователей класс “byuser”. Соответственно, к этим классам можно применить любые нужные нам стили и выделения.

Старые версии вордпресс таких возможностей не имеют, поэтому придется немного поколдовать. Данный код проверяет, не является ли автор комментария автором записи и если да, то присваивает ему заданных класс

```
<?php if ($comment->user_id == $post->post_author) { echo 'class="comment-author"'; } ?>
```

Для простейшего цикла вывода комментариев полный код выглядит вот так

```
<?php foreach ($comments as $comment) : ?>
<div<?php if ($comment->user_id == $post->post_author) {echo ' class="comment-author"'; } ?>>
<p><strong><?php comment_author_link(); ?></strong></p>
<?php comment_text(); ?>
</div>
<?php endforeach; ?>
```

а в стили добавляем например такое

```
.comment-author {font-weight: bold;color: red;}
```

Преимуществом данного кода будет то, что он опознает все комментарии всех авторов сайта. Недостатком то, что он не различает одного автора от другого, а значит все они будут в одинаковом стиле. Давайте устраним данный недостаток.

```
<?php if ($comment->comment_author_email == "chris@digwp.com") echo 'class="author-chris"'; ?>
```

zzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzz



Теперь по емайлу будет опознан конкретный автор и ему будет присвоен конкретный стиль.

```
<?php if ($comment->comment_author_email == "chris@digwp.com") echo 'class="author-chris"'; elseif ($comment->comment_author_email == "jeff@digwp.com") echo 'class="author-jeff"'; ?>
```

## Стилизуем комментарии с граватарами

Теперь останется только добавить функцию вставки изображения внутри цикла вывода комментариев

Также в качестве аватара можно использовать аватары Твиттера, для этого забирайте плагин **Twitter** отсюда <http://wp.smashingmagazine.com/2009/01/08/twitter-avatars-in-comments-wordpress-plugin/> и вставляйте в комментарии функцию

Базовыми настроек аватар остаются их размеры — по умолчанию квадрат со стороной 50 пикселей, который может быть увеличен до 96 пикселей и изображение по умолчанию, которое отображается если аватар не задан. Эта функция вернет тег изображения `img` с ссылкой на аватар пользователя или на картинку по умолчанию с вашего сайта.

```
<?php $author_email = get_the_author_email(); echo get_avatar($author_email, '96'); ?>
```

**Добавление сообщения «Ваш комментарий ожидает модерации»**

Хорошая новость в том, что если использовать в шаблоне `wp_list_comments`, то особенно ничего больше делать и не нужно, система уведомлений встроена в эту функцию. А для пользовательских циклов комментариев в `functions.php` добавляется небольшой код

ZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZ









```
<?php while (have_posts()) : the_post(); ?>
<?php comments_template(); ?>
<?php endwhile; ?>
```

Один из лучших способов расширения функционала комментирования вордпресс, это добавление тегов комментирования. Быстрые теги — это кнопки, созданные с помощью JavaScript, предлагающие ссылки на различные наборы элементов разметки. Во время набора текста комментариев, пользователь сможет сделать текст жирным или наклонным, вставить код или цитату.

Или его аналог **Comment Form Quicktags** <http://wordpress.org/plugins/comment-form-quicktags/>

Соответствующий уход за комментариями требует время. Но разница между сайтами уделяющими внимание комментариями и не уделяющими бросается в глаза, поэтому это время стоит того. И большую часть этого времени будет занимать борьба со спамом, т. к. распространители Виагры будут использовать любую возможность, чтобы загадить каждый комментарий.

## Встроенный анти-спам функционал

- Разрешить оставлять комментарии. Если вам не нужны комментарии на сайте- отключите их. Полное отключение комментариев как нельзя лучше будет бороться со спамом.
- Пользователи должны быть зарегистрированы и авторизованы для комментирования. Хороший способ разрешить комментарии и заблокировать 90% спама.
- Перед появлением комментария — комментарий должен быть одобрен вручную и автор должен иметь ранее одобренные комментарии.
- Автоматически закрывать обсуждение статей старше .. дней — удобно, если на сайте слишком много комментариев, чтобы контролировать только свежие записи.
- Модерация комментариев — если комментарий содержит указанное слово или айпи, он будет отправлен на модерацию.

[illegible]

- Черный список - если комментарий попал под фильтр, он будет отправлен в спам.

Здесь <http://perishablepress.com/wordpress-spam-battle-3-seconds-that-will-save-you-hours-of-time/> можно посмотреть подборку англоязычных стоп-слов для заполнения черных списков.

## Антиспам плагины

- **Akismet** <http://akismet.com/> один из самых лучших плагинов, идет в стартовом комплекте вордпресс.
- **Peter's Custom Anti-Spam** <http://wordpress.org/plugins/peters-custom-anti-spam-image/> Плагин на основе капчи, заставляя всех комментаторов разгадать случайное слово.
- **reCAPTCHA Plugin** <http://wordpress.org/plugins/wp-recaptcha/> В качестве капчи отображает отсканированные слова из старых книг. Результаты отгаданных капч используются для распознавания книг для цифровых библиотек.
- **WP-HashCas** <http://wordpress.org/plugins/wp-hashcash/> Капча плагин, декларирующий 100% эффективность.
- **Simple Trackback Validation** <http://wordpress.org/plugins/simple-trackback-validation/> плагин блокирующий трекбек спам.

## Другие вопросы и методы

В дополнение ко всему вышеперечисленному хотелось бы еще кое-что рассказать про улучшение системы комментирования. Вот вам два отличных плагина:

- **Keyword Luv** <http://wordpress.org/plugins/keywordluv/>  
Этот плагин делает небольшой подарок комментаторам, отображая их имя отдельно от ключевого слова, указанного для их сайта. Это позволяет комментаторам оставлять ссылку на свой сайт по отдельно указанной фразе, которую они продвигают. С точки зрения seo-это очень круто и привлекает толпы комментаторов.
- **CommentLuv** <http://wordpress.org/plugins/commentluv/>  
Плагин создает ссылку на самый последний пост с сайта комментатора. Т.е. посетитель оставляет комментарий, указывает свой сайт, а плагин забирает с него ссылку на свежий пост. Что также привлекает комментаторов.

## Nofollow ссылки

Вот вам пример нолоуу ссылки `<a rel="nofollow" href="http://supersite.ru/" >Ссылка</a>`

Ссылки помечаются ноболоу как знак того, что они не одобряются автором. Делается это для соответствия следующим правилам поисковиков:

1. По таким ссылкам не следует переходить поисковой роботу
2. Такая ссылка не будет включена в расчет передаваемого веса страницы
3. Анкор такой ссылки не будет учитываться при определении релевантных ключевых слов целевой страницы.

[illegible]



В основном, нофолоу ссылки игнорируются Бинг и Яху и частично игнорируются Гугл (т. к. по таким ссылкам боты все равно лезят и кое-какие ссылки могут учитывать, например из твиттера).

Разработчики вордпресс решили, что будет лучше пометать атрибутом нополоу все ссылки комментаторов и пинги. Но не все поддерживают такую стратегию. Главный аргумент заключается в том, что обратная ссылка является платой за то, что посетитель оставил комментарий. Заклучая ссылку в нополоу мы убираем стимул комментировать и уменьшаем награду.

Поэтому многие блоги включаются в так называемое дуфолоу движение и удаляют этот тег из ссылки комментатора. Вот вам подборка из 11 плагинов, позволяющих это сделать

<http://perishablepress.com/comprehensive-reference-for-wordpress-no-nofollow-dofollow-plugins/>

## Интеграция Твиттера

Твиттер является одной из самых популярных социальных сетей, поэтому не будет лишним добавить его интеграцию на сайт.

- **TweetSuite** <http://wordpress.org/plugins/tweetsuite/> добавляет автоматическую отправку записей в твиттер
- **Tweet This** <http://richardxthripp.thripp.com/tweet-this/> добавляет кнопку отправки записи в твиттер
- **TwitterLink Comments** <http://wordpress.org/plugins/twitterlink-comments/> добавляет ссылку на твиттер в профиль комментатора
- **TwitThis** <http://twitthis.com/> позволяет читателям отправить в твиттер ссылку на понравившуюся запись
- **Twitter Tools** <http://wordpress.org/plugins/twitter-tools/> комплексный набор инструментов для интеграции вордпресса и твиттера

В этой главе, плотно упакованной кодом и плагинами, мы исследовали систему комментирования на немаленькой глубине. С впечатляющей гибкостью и невероятными возможностями настроек, мы можем создать красивый, оптимизированный и дружелюбный блог.

## НЕ САЙТ, А ПРЕЛЕСТЬ!



Но какой бы не был шикарный наш сайт, об этом никто не узнает, если не сможет найти наш сайт в интернете. В следующей главе мы погрузимся в искусство поисковой оптимизации и разберем много полезных и эффективных seo техник.

zzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzz



## 8. Поисковая оптимизация

## Сила и слабость SEO

Изначально вордпресс дает достаточную гибкость организации и управления контентом блога. Большая часть этого достигается с помощью рубрик и тегов. Каждый пост вашего блога может быть закреплен за любым количеством рубрик и тегов.

Рубрики предназначены для классификации контента в соответствии с широкими определениями, тогда как теги используются для классификации более узко. Например, запись о любимых фильмах может находиться в рубрике «Любимые фильмы», и при этом помечена тегами определяющими суть текста «Звездные войны», «Матрица», «Бегущий по лезвию».

Помимо этого центрального организующего принципа, вордпресс вместе с ним дает как сильные, так и слабые стороны такой организации контента и для посетителей, и для поисковиков. Давайте для начала разберемся с этими сторонами

## Сильный акцент на содержании

Контент, как все уже в курсе, король. Интернет существует благодаря ему. Пользователи ищут его. Поисковые системы созданы для этого. И для того, чтобы преуспеть в интернете, ваш сайт должен быть прежде всего сфокусирован на доставке полезного контента. Реально, одна из главных целей вордпресса — это сделать публикацию контента максимально легкой.

После установки вордпресса сразу же становится возможной публикация, а сотни качественных шаблонов дают возможность организовать и представить этот контент в лучшем виде.

## Встроенные нѳолоу ссылки

Возможно это работает не так как изначально придумывалось, но атрибут nofollow для ссылок комментариев позволяет усилить seo-качества вордпресс сайтов. Этот атрибут позволяет препятствовать поисковым роботам следовать по размещенным ссылкам, оставаясь на вашем сайте. Изначально это использовалось для контроля распределения веса Пейджранка PR, однако т. к. он становится все меньшим фактором в ранжировании, nofollow ссылки становятся все большим спорным вопросом. Мы еще вернемся к этому вопросу.

## Проблемы дублированного контента

Организационная сила рубрик и тегов дается за высокую цену: дублирование контента. Дублирование контента означает, что идентичный текст появляется на более чем одной странице. Поисковики, такие как Гугл и Яндекс, штрафуют сайты за дублированный контент, выбрасывая сайты из индекса или понижая позиции в выдаче. Если вернуться к примеру статьи про любимые фильмы, то идентичные копии можно найти на нескольких разных урлах:

- оригинальная статья -> <http://example.com/blog/my-favorite-movies/>
- архив рубрики фильмы -> <http://example.com/blog/category/favorite-movies/>
- тег звездные войны -> <http://example.com/blog/tag/star-wars/>
- тег матрица -> <http://example.com/blog/tag/the-matrix/>
- тег бегущий по лезвию -> <http://example.com/blog/tag/blade-runner/>

Но и это еще не все! Было бы не так плохо, если бы не было еще хуже:

- архив по определенной дате -> <http://example.com/blog/2009/02/02/>

ZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZ

- архив за месяц -> <http://example.com/blog/2009/02/>
- архив за год -> <http://example.com/blog/2009/>
- архив автора -> <http://example.com/blog/author/your-name/>

В зависимости от шаблона и настроек, может быть даже еще хуже. И с точки зрения поисковика это не есть хорошо. Особенно если вы любите активно создавать различные теги.

## Контроль дублированного контента

К счастью, есть несколько способов и технологий, чтобы побороть дубли контента:

- `тег meta nofollow`
- `тег meta noindex`
- атрибут `nofollow`
- директивы `robots`
- `тег canonical`
- использование цитат

Разберемся с каждым по отдельности.

## Теги meta noindex и nofollow

Эти теги принадлежат к секции `<head>` и размещаются соответственно в `header.php`

```
<meta name="googlebot" content="index,archive,follow" />
```

В данном примере поисковому роботу гугл сообщается, что вся страница целиком должна быть проиндексирована, по ссылкам с нее можно переходить и ее можно добавлять в архив поисковика. Это самый лучший способ разметить страницу записи.

Понятно, что большинство блогеров хочет, чтобы их статьи попадали в индекс, но при этом не возникало дублей, поэтому для страниц архивов оптимальной будет такая конфигурация

```
<meta name="googlebot" content="noindex,noarchive,follow" />
```

В данном случае мы сообщаем поисковику, что не хотим чтобы он индексировал эту страницу, но в то же время говорим, что будет нормально проследовать по всем ссылкам, находящимся на этой странице. Это предотвращает появление этой страницы в поисковых результатах, но позволяет проиндексировать остальные страницы.

И конечно мы можем сообщить поисковику, что данная страница не должна индексироваться и не нужно следовать по любым ссылкам с этой страницы

```
<meta name="googlebot" content="noindex,noarchive,nofollow" />
```

Учитывая эти теги, мы можем лучшим способом настроить борьбу с дублированным контентом автоматически. Для этого внесем условия в `header.php`, в соответствии с которыми страницы будут

ZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZ





## Атрибут Nofollow

Еще одна возможность запретить индексировать ненужный контент — использовать спорный атрибут ссылок nofollow

<a href="http://domain.tld/path/target/" rel="nofollow">Эта ссылка запрещена для перехода  
ботам</a>

По ссылкам содержащим нополоу атрибут поисковый бот не должен переходить (хотя все равно делает это). И целевая страница точно будет проиндексирована, если на нее есть другая незакрытая ссылка (с вашего или чужого сайта).

Можно рассмотреть в качестве примера страницу автора, на которой размещаются все те же посты блога. Индексировать страницу автора смысла нет никакого, поэтому все ссылки на эту страницу в шаблоне стоит пометить атрибутом nofollow

<a href="http://domain.tld/author/author-name/" rel="nofollow">Автор записи</a>

Но как выше говорилось, это не 100% метод запрета индексирования подобных страниц, поэтому для максимального эффекта нужно комбинировать несколько способов — запретить индексирование в мета тегах `head`, проставить тег `noindex` на ссылки на эту страницу, а также запретить индексирование в директивах `robots`

## Директивы robots.txt

Директивы robots еще один эффективный и часто недооцененный способ для борьбы с дублированным контентом. Файл robots.txt находится в корневой директории сайта и доступен по адресу <http://domain.tld/robots.txt> При размещении в любых других местах или с другим названием он работать не будет.

Суть его в том, чтобы сообщать поисковым роботам что можно и что нельзя делать на сайте. Большинство белых роботов, таких как google и yandex принимают их во внимание, а различные спам боты и парсеры- игнорируют.

Имейте в виду, что страницы запрещенные для посещения и индексирования в роботе, все равно поглощают ссылочный вес. Поэтому при неправильной организации и перелинковке сайта можно терять достаточно много ссылочного веса, который потерян и не учтен.

Кроме этого, если посещение страницы запрещено для бота, это не означает, что поисковый бот ее не посетит, вероятно из любопытства — что же там ему запретили. И иногда появляется в индексе — правда с отображением только одного лишь урл страницы без отображения контента в снипете.

Давайте разберем пример оптимальной конфигурации директив robots для предотвращения индексирования ненужных страниц. Для этого нужно освежить в памяти структуру файлов и папок вордпресс.

Этим двумя строками мы запрещаем индексировать любой урл, начинающийся с wr- и любой урл содержащий .php на конце. Таким образом все папки и системные файлы не должны индексироваться

Disallow: /wp-\*

Disallow: \*.php

ZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZ



zzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzz





<http://digwp.com/index.php?p=123>

преображаясь в постоянную ссылку

<http://digwp.com/post-name>

Замена динамических ссылок с параметром в запросе на текстовый формат оптимизируют сайт для поисковика и делают удобным для восприятия людьми.

Нужно обратить на один момент — постоянная ссылка формируется из текста заголовка записи, поэтому название статьи на русском языке трансформируется в кириллический урл. Чтобы урл был латиницей, нужно воспользоваться плагином типа **Rus to Lat** <http://wordpress.org/plugins/rustolat/> который будет автоматически транслитерировать урлы записей.

## Стандартные урлы и ЧПУ

Один из важных факторов оптимизации сайта- это оптимизация его ссылок. ЧПУ

(человекопонятные урлы) потому так и называют, что из адреса записи понятно о чем она, какие ключевые слова фигурируют в заголовке и обозначена тема. Тогда как стандартный урл содержит айди записи или страницы, хотя на самом деле поисковику все равно, какой урл индексировать.

Ранее уже приводились примеры оптимально созданных шаблонов формирования урлов записей, приведем еще раз варианты

<http://domain.tld/name-of-post/>

<http://domain.tld/name-of-page/>

<http://domain.tld/category/boots/>

<http://domain.tld/author/fonzi/>

<http://domain.tld/2008/08/08/>

### Настройки постоянных ссылок

По умолчанию WordPress использует ссылки со знаком вопроса и цифрами, однако у вас придать вашим ссылкам эстетичность, удобство использования и обеспечить совместимость с примерами.

#### Общие настройки

<input type="radio"/> По умолчанию	<a href="http://domain.tld/?p=123">http://domain.tld/?p=123</a>
<input type="radio"/> День и название	<a href="http://domain.tld/2014/01/14/sample-post/">http://domain.tld/2014/01/14/sample-post/</a>
<input type="radio"/> Месяц и название	<a href="http://domain.tld/2014/01/sample-post/">http://domain.tld/2014/01/sample-post/</a>
<input type="radio"/> Цифры	<a href="http://domain.tld/archives/123">http://domain.tld/archives/123</a>
<input checked="" type="radio"/> Название записи	<a href="http://domain.tld/sample-post/">http://domain.tld/sample-post/</a>
<input type="radio"/> Произвольно	<a href="http://domain.tld/%postname%">http://domain.tld/%postname%</a>

Разные варианты легко переключаются в настройках. Но игра с настройками должна быть только на начальном этапе создания сайта, так как резкая смена схемы настройки постоянных ссылок потребует полной переиндексации всего сайта, а это может вызывать определенные проблемы.

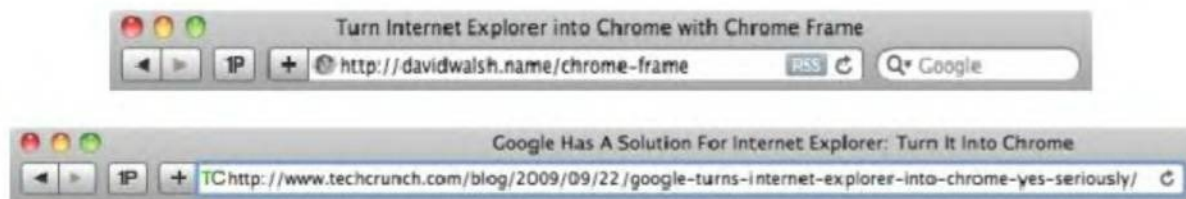
## Держите ссылки короткими

После принятия решения о использовании постоянных ссылок можно сделать выбор из нескольких вариантов, но главной рекомендацией будет — чем короче, тем лучше. Использование структуры, построенной на упоминании рубрики в урле хорошо скажется на поисковых позициях, т. к. сегментирует записи сайта по темам.

Но не стоит слишком увлекаться детализацией, сравните визуально эти две ссылки

<http://domain.tld/2008/08/08/>





Поэтому, чем короче- тем лучше.

После того, как вы опубликовали запись, сформировалась ее ссылка, по которой она доступна в интернете. Если вы измените ссылку - адрес записи, то ее не смогут обнаружить на своем месте и нужно будет сделать редирект для обеспечения корректного перехода на нее из поиска или закладок.

Но зато после создания записи можно смело менять ее название-заголовок, т. к. он никоим образом не связан с урл после его создания. Т.е. на основании заголовка формируется урл, но потом становится независим — это значит что можно переименовывать записи без каких либо последствий для индексации или корректных переходов по ссылкам.

## Максимизируем ключевые слова

Одно из seo - преимуществ постоянных ссылок- возможность использовать в урле ключевые запросы. Одно время наличие в названии домена или в урле ключевого слова сильно улучшало позиции этой страницы по этому запросу. Сейчас этот вес несколько снижен, но ключевое слово в урле все равно выделяется жирным при отображении сайта в результатах поиска.

Сравните привлекательность этих двух ссылок

<http://yoursite.com/index.php?p=123>

<http://yoursite.com/search-engine-optimization/>

При всех прочих равных, даже для людей вторая ссылка более привлекательна, особенно если искать статью про поисковую оптимизацию. Таким образом урл помогает пользователям ориентироваться в выдаче, выбирая наиболее интересные варианты сайтов.

Поэтому при необходимости вносите правки в слаг — ссылку записи, затачивая ее под определенные запросы, а не простое повторение заголовка статьи.

Все это вместе приводит нас к таким выводам:

- правильная ссылка должна быть короткой,
- не должна содержать союзы, предлоги и мусорные или малополезные слова,
- содержать в себе целевые ключевые слова и фразы.

Все это вместе даст максимум отдачи как для посетителей, так и для поисковиков.

Ранжируемся в Google

В дополнение к сео приемам, описанным тут, есть много других способов создать сео оптимизированный сайт. В данной главе мы рассмотрим некоторые идеи и объясним как они нам могут помочь.

////////////////////////////////////

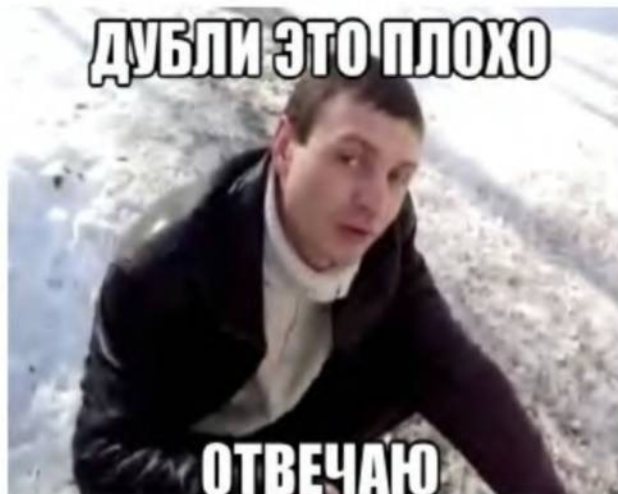


## Боремся с дублями контента

Чтобы определить, как наш сайт проиндексирован в гугл, достаточно сделать запрос вида

site:yoursite.com

который выведет список всех страниц, проиндексированных поисковиком. Для начала сравните количество проиндексированных страниц и общее фактическое число страниц на вашем сайте, они не должны сильно отличаться. Если страниц существенно меньше- значит проблема с индексацией, если страниц найдено существенно больше- проблема с дублями. В найденных результатах можно покопаться и поискать неправильные варианты, а также порадоваться если все в порядке.



Во-вторых, стоит поискать конкретные примеры дублированных страниц, для этого сделайте запрос с фрагментом url записи

site:yoursite.com inurl:urlzapisi

или же

site:yoursite.com поисковый запрос

## И еще про контент

Уделяя внимание разным нюансам оптимизации и настройки, не забывайте про главный критерий поискового ранжирования — контент. Убедитесь, что каждая страница предлагает максимальное содержание с оптимальным количеством слов. Также, при разработке шаблона убедитесь, что контент размещается максимально близко к верху страницы в коде. Из этих соображений размещение сайдбара справа более выгодное. Или же пусть он визуально находится слева, но фактически в коде идет после кода основной части вывода контента. И избегайте размещения кучи разных скриптов и кода перед контентом в шапке сайта.

Хороший способ реализации этой стратегии, это представить как видит страницу вашего сайта поисковый робот. Посмотрите исходный код страницы, какое количество кода и служебной информации он содержит, какую часть его составляет текст записи и полезный контент, сколько кода и данных придется пропустить, прежде чем робот доберется до сути страницы? Чтобы почувствовать себя поисковым роботом, воспользуйтесь утилитой

<http://tools.seobook.com/general/spider-test/>

И придерживайтесь максимально высокой пропорции соотношения контента к коду.

## Ищите дублированный контент

Уже не раз говорили о способах обнаружения и предотвращения дублирования контента. Кроме этого есть масса утилит позволяющих проанализировать ваш сайт или страницу, как например <http://www.virante.org/seo-tools/duplicate-content>





## Оптимизируем заголовки

Другой важный момент в оптимизации контента сайта, это оптимизация элементов заголовка. Речь идет про теги, с помощью которых размечают различные заголовки <h1> - <h6>. Например, довольно часто тегом h1 обозначают название сайта, тегом h2 описание сайта, тегом h3 название каждой записи, h4 элементы заголовков в сайдбаре и т. д.

Проблема такого подхода прежде всего в seo. Поисковый робот видит, что самый важный с точки зрения алгоритмов тег помечает на всех страницах одно и то же- название сайта. Поэтому гораздо правильнее будет помечать тегом h1 заголовок записи, h2 использовать для подтитлов записи, а название сайта писать простым текстом, отформатированным с помощью стилей.

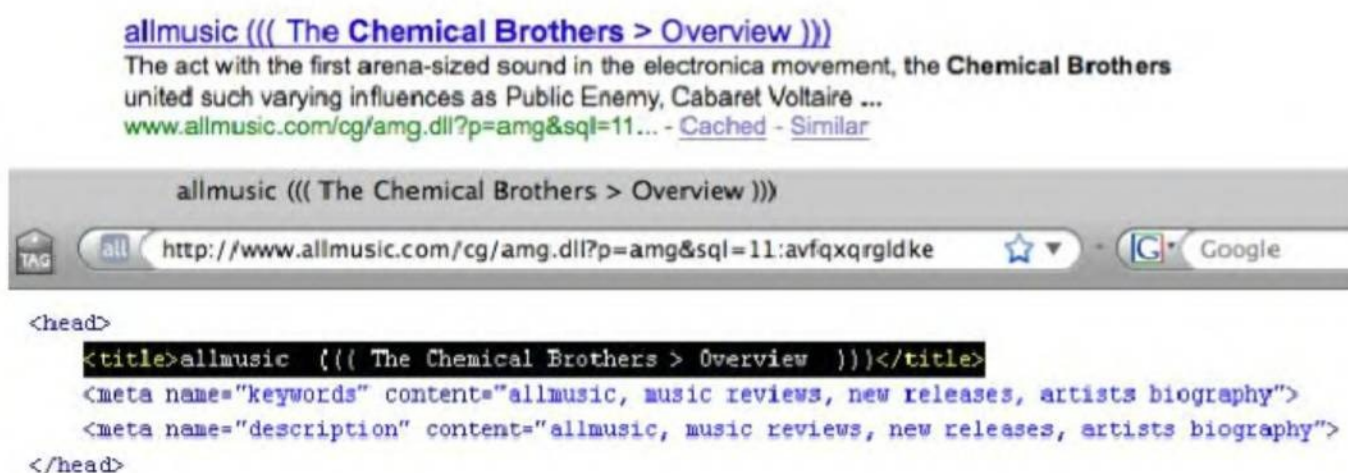
Но точно также, как не все однозначно с атрибутом nofollow, не все так просто и с этими тегами. Использование тега h1 для названий записей может привести к санкциям со стороны поисковика за манипуляции, т. к. на архивных страницах и страницах рубрик будет выведено несколько заголовков с тегом h1, что не совсем корректно с точки зрения семантики.

Для простоты понимания есть правило, что количество вставок тега на странице определяется по его номеру, для тега h1 допустимо 1раз, для тега h2 2 раза и т. д. С другой стороны есть сторонники, считающие корректным использование тега h1 только для названия сайта, поэтому не будем слишком заострять на этом свое внимание.

Все дело в том, что новый стандарт HTML5 допускает для документов структуры с множественными заголовками любых типов любое нужное количество раз, поэтому классической можно считать схему, в которой тег h1 используется и для названия сайта, и для названия статьи.

## Оптимизируем названия статей

Названия статей — следующий шаг оптимизации сайта. Потенциально стоит улучшить заголовок, декларируемый тегом `<title>` в секции `<head>`, т. к. он отображается в поисковых результатах, сохраненных закладках и на панели браузера. И в общем, не обязательно, чтобы этот заголовок полностью совпадал с названием статьи.



В текущей версии вордпресс, заголовок формируется тегом шаблона `wp_title()`, в зависимости от типа страницы он выводит следующую информацию

zzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzz





Ну и как говорили, нофолоу ссылки остаются полезными для управления индексацией дублированным контентом, не пускать ботов на определенные страницы и защищать блог от комментариев с ссылками, ведущими на сомнительные сайты.

Каждый блогер знает, как быстро меняется интернет. Мертвые ссылки появляются все время и могут существенно повлиять на качество вашего сайта. Если посетители и поисковые боты переходят с вашего сайта по ссылкам на несуществующие сайты и страницы, то какого мнения они будут о вас и о вашем сайте. Будь входящая или исходящая, ссылки должны быть актуальными и рабочими.

Для проверки одной страницы сайта можно воспользоваться сервисом <http://validator.w3.org/checklink>

А вот альтернативный вариант — адд-он для Мозиллы <http://frayd.us/extensions/linkchecker/> проверяющий ссылки на загруженной странице за 1 клик.

## Использование карты сайта

<http://mindfeed.org/>  
<http://mindfeed.org/family-hatred/>  
<http://mindfeed.org/im-loving-it/>  
<http://mindfeed.org/just-nod-if-you-can-hear-me/>  
<http://mindfeed.org/interpretation-is-relative/>  
<http://mindfeed.org/doom-and-gloom/>  
<http://mindfeed.org/give-up-dont-give-up/>

Также карту сайта можно создать и без плагина, с помощью онлайн сервисов типа <http://www.xml-sitemaps.com/> и десктопных программ. Созданные с их

ZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZ



## SEO плагины

Вордпресс можно расширять практически бесконечно. Пользователи этого движка постоянно создают новые и усовершенствуют старые плагины. И применительно к сео ситуация точно такая же. Вот пример некоторых самых полезных и выгодных плагинов:

- **All-in-One SEO** <http://wordpress.org/plugins/all-in-one-seo-pack/> оптимизирует вордпресс, позволяя применить большинство описанных тут техник и приемов
- **HeadSpace2 SEO** <http://wordpress.org/plugins/headspace2/> оптимизирует сайт настройкой различных метаданных
- **Redirection** <http://wordpress.org/plugins/redirection/> управляет редиректами, отслеживая недоступные страницы и страницы, изменившие свои адреса
- **Robots Meta** <http://wordpress.org/plugins/robots-meta/> помогает настроить разрешенные и запрещенные для индексирования страницы
- **SEO Smart Links** <http://wordpress.org/plugins/seo-automatic-links/> помогает создать внутреннюю перелинковку по заданным вручную и автоматически фразам

Эти плагины позволяют разработать и воплотить оптимальную стратегию продвижения. Но это еще не все — много интересных плагинов можно найти в репозитории вордпресс

<http://wordpress.org/plugins/tags/seo>

## Отслеживание успешности сайта

После того, как вы оптимизировали сайт для поисковиков, жизненно важно мониторить и следить за развитием. Есть много различных аналитических и статических инструментов на любой выбор:

## Плагины статистики

- **WP-ShortStat** <http://wordpress.org/plugins/wp-shortstat2/> предлагает базовую статистику посетителей, просмотров, реферреров, поисковых запросов и т. д. Это легкий плагин, тихонько работающий в фоне и собирающий статистику, доступную через админку.
- **StatPress** <http://wordpress.org/plugins/statpress/> очень мощный плагин, собирающий массу информации о посетителях, роботах и ботах и отображающий ее в реальном времени.
- **Google Analytics Dashboard** <http://wordpress.org/plugins/google-analytics-dashboard/> дает возможность просматривать информацию из аккаунта гугл аналитикса прямо в админке вордпресса.
- **Analytics360** <http://wordpress.org/plugins/analytics360/> также собирает и выводит массу информации гугл аналитикс, объединяя ее с данными сервиса MailChimp

Также тут <http://stylizedweb.com/2010/06/03/5-rocking-google-analytics-plugins-for-wordpress/> можно посмотреть обзор 5 плагинов интегрирующих функционал гугл аналитикс в вордпресс.

И это лишь вершина айсберга, на самом деле плагинов статистики просто море

<http://wordpress.org/plugins/tags/statistics>

ZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZ



Правда у многих плагинов статистики есть темная сторона — собранные данные очень быстро увеличивают размер базы данных, поэтому нужно время от времени проводить чистку, удаляя старые и неактуальные данные.

К примеру, для плагина WP-ShortStat можно быстро удалить выбранное количество ненужных записей прямо из базы данных

DELETE FROM `wp\_ss\_stats` ORDER BY `id` ASC LIMIT n, где n – количество записей для удаления

## Статистика Mint

Mint — это самостоятельное приложение, позволяющее создать свой собственный сервис статистики. Вы получаете полный контроль над различными собранными данными — хитами, визитами, реферрами и т. п. Кроме базового функционала можно получить дополнительные расширения. Приложение платное, подробнее можно узнать тут <http://haveamint.com/>

## Google Analytics

Гугл аналитикс — один из самых популярных сервисов статистики, который отслеживает практически ВСЕ. Для того, чтобы данные начали собираться, нужно зарегистрироваться в сервисе и разместить в футере сайта код типа этого

```
<script type="text/javascript">
var _gaq = _gaq || [];
_gaq.push(['_setAccount', 'UA-XXXXXX-XX']);
_gaq.push(['_trackPageview']);
(function() {
var ga = document.createElement('script'); ga.type = 'text/javascript'; ga.async = true;
ga.src = ('https:' == document.location.protocol ? 'https://ssl' : 'http://www') + '.google-
analytics.com/ga.js';

var s = document.getElementsByTagName('script')[0]; s.parentNode.insertBefore(ga, s);

})();
</script>
```


Для выполнения этой простой задачи есть масса плагинов похожего функционала, но проще всего сделать это самостоятельно. Ну а для совсем ленивых можем порекомендовать плагин **Google Analytics for WordPress** <http://wordpress.org/plugins/google-analytics-for-wordpress/>

Одним из главных недостатков аналитикса является то, что он требует наличие JavaScript для своей работы. Это означает, что те, у кого отключен ява, не будут учтены в статистике. Для того, чтобы определить насколько много таких посетителей, есть плагин <http://remysharp.com/2009/10/15/the-missing-stat-noscript/>

И как альтернативу гугл аналитикс, следует рассматривать статистику Яндекс Метрики. Многие блогеры используют на одном сайте сразу несколько статистик, добавляя еще и счетчик Liveinternet.

////////////////////////////////////

Одна из самых сложных частей сео стратегии — это создание высококачественных и релевантных ссылок на ваш сайт. Без этих ссылок сайт для поисковиков практически невидим. Если представить интернет в виде множества потоков, то становится понятным, что сайты с большим количеством потоков на них и есть самые популярные сайты.



**БЕЛОЕ СЕО?**

**ОДОБРЯЮ**

И третья часть — это оптимизация мета информации, связанной с ссылками, изображениями и другими ресурсами. В процессе написания разметки этих элементов, рекомендуется включать атрибуты alt, title и desc, чтобы максимально точно и конкретно их описать. Это все значительно усилит ваш контент и его релевантность для поисковиков. Добавление правильной мета информации в процессе создания сайта станет очень полезной и выигрышной привычкой.

И заключительный, но не последний этап — это производительность сайта, многократно влияющей на количество и качество получаемого трафика. Медленно загружаемые страницы, битые ссылки, пропавшие изображения — это красный флаг и для людей, и для ботов. Теперь пришла очередь рассмотрения стратегий оптимизации производительности сайта, чтобы предоставлять контент быстро и аккуратно.

Нужно понимать, что нет единого рецепта для всех, поэтому каждому нужно будет разработать свою собственную стратегию. Неважно, что вы до этого слышали, обучение искусству seo не происходит в одночасье, поэтому давайте начнем с основ и будем их использовать как фундамент для роста и развития своих навыков.

И не будет лишним напомнить — контент это король, ссылаться на другие сайты это хорошо и немного здравого смысла ведут по длинному пути принятия правильного решения.

ZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZ



## 9. Уход за здоровьем сайта

## Бережем здоровье сайта

Очень легко на базе вордпресс создать блог, интернет-магазин и развитую систему управления контентом, но еще остается огромное количество работы, которую нужно сделать в порядке обеспечения здоровья и процветания сайта.

Для любого сайта сильная стратегия безопасности просто абсолютно необходима. Будучи самым популярным блоговым приложением, вордпресс огромная цель для вредоносных атак. К счастью, сообщество вордпресс также очень сильно и тысячи разработчиков постоянно работают над развитием и обновлением плагинов, шаблонов и функций. Если объединить все это вместе с методологией и информацией, то можно создать безопасный и защищенный сайт против атак и спама.

Первое, что стоит сделать, это обеспечить защиту файлов сайта. Традиционно есть много способов для этого, включая настройку соответствующих прав доступа к файлам, отмену просмотра директорий, блокирование восприимчивых файлов и постоянное обновление движка.

## Настраиваем безопасные права доступа

Обычно ваш хостинг уже имеет оптимальную настройку прав доступа, поэтому стоит просто в этом убедиться. Это можно сделать как на самом сервере, просматривая список файлов и директорий и их права, так и через ftp клиент.

<a href="#">dev</a>	<dir>	Aug 26, 2009 01:20:58 PM	drwxr-xr-x	root	root
<a href="#">etc</a>	<dir>	Sep 17, 2009 03:03:00 PM	drwxr-xr-x	root	root
<a href="#">home</a>	<dir>	Jul 23, 2009 04:58:06 PM	drwxr-xr-x	root	root
<a href="#">lib</a>	<dir>	Aug 26, 2009 02:09:35 PM	drwxr-xr-x	root	root
<a href="#">media</a>	<dir>	Mar 29, 2007 01:00:12 PM	drwxr-xr-x	root	root
<a href="#">mnt</a>	<dir>	Mar 29, 2007 01:00:12 PM	drwxr-xr-x	root	root
<a href="#">opt</a>	<dir>	Mar 29, 2007 01:00:12 PM	drwxr-xr-x	root	root
<a href="#">proc</a>	<dir>	Aug 26, 2009 01:20:56 PM	dr-xr-xr-x	root	root
<a href="#">root</a>	<dir>	Aug 14, 2009 02:15:23 PM	drwxr-x---	root	root
<a href="#">sbin</a>	<dir>	Aug 26, 2009 02:07:53 PM	drwxr-xr-x	root	root
<a href="#">selinux</a>	<dir>	Mar 29, 2007 01:00:12 PM	drwxr-xr-x	root	root
<a href="#">srv</a>	<dir>	Mar 29, 2007 01:00:12 PM	drwxr-xr-x	root	root

Есть три базовых вида доступа к файлу

- **Чтение** — позволяет читать содержимое файла или имя файла
- **Запись** — позволяет изменять содержимое файла или имя файла
- **Выполнение** — позволяет выполнять файл или обрабатывать содержимое файла

На вашем сервере файлы вордпресс должны принадлежать вашему аккаунту и быть открытыми для записи. В большинстве случаев у вас не будет поводов беспокоиться о правах доступа и вопросы будут возникать только если будут появляться проблемы, связанные с настройкой файлов или директорий для работы плагинов или обеспечения безопасности.

**ZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZ**





просмотр содержимого директории, и директория не содержит индексные файлы, типа index.html, index.php и т. п., то при открытии этой директории в браузере показывается все ее содержимое, что крайне небезопасно и неосмотрительно.

Вот типичный пример такой директории

## Index of /wp-content

	<u>Name</u>	<u>Last modified</u>	<u>Size</u>	<u>Description</u>
[DIR]	<u>Parent Directory</u>		-	
[DIR]	<u>themes/</u>	26-Oct-2009 02:20	-	
[DIR]	<u>plugins/</u>	26-Oct-2009 02:20	-	
[DIR]	<u>images/</u>	26-Oct-2009 02:20	-	
[DIR]	<u>backups/</u>	26-Oct-2009 02:20	-	

Получив доступ к директории и файлам, будет совсем легко получить остальные доступы.

К счастью, решается все довольно быстро и просто, добавлением в htaccess конфигурации

## Options -Indexes

Альтернативным вариантом будет загрузить во все важные директории пустой файл под именем index.html. Многие директории вордпресса изначально уже имеют такой файл для защиты.

И тогда вместо содержимого каталога будет показан пустой белый экран.

## Закрываем доступ к важным файлам

## Защищаем wp-config.php

Кроме запрета просмотра директорий, мы помешаем прямому внешнему обращению к критически важным файлам. Первый и самый главный, это `wp-config.php`, содержащий настройки сайта, а также информацию о доступе к базе данных.

Закроем к нему доступ в файле htaccess

```
<Files ~ wp\~config\php>  
Order Deny,Allow  
Deny from all  
Allow from 123.456.789  
</Files>
```

где 123.456.789 ваш айпи (узнать можно тут <http://www.whatismyip.com/>)

А если нужно разрешить нескольким айпи доступ, то напомним вот так

```
<Files ~ wp\~config\php>
Order Deny,Allow
Deny from all
Allow from 123.456.789
Allow from 456.789.123
Allow from 789.123.456
```

[illegible]

```
# additional IP addresses
</Files>
```

## Защищаем директорию wp-admin

Следующим шагом будет настроить защиту файлов, находящихся в папке `wp-admin`, что является критическим для вашего сайта.

Самый простой способ, это ограничить доступ к папке по айпи. Необходимо внутри папки разместить файл htaccess с содержимым

```
<FilesMatch ".*.*">
Order Deny,Allow
Deny from all
Allow from 123.456.789
</FilesMatch>
```

Как и ранее, будет запрещен доступ всем, кроме вашего айпи. Если на сервере настроен `mod_rewrite` Apache, то синтаксис запрета будет немного другой

```
<IfModule mod_rewrite.c>
RewriteEngine On
RewriteCond %{REMOTE_ADDR} !^123\.456\.789
RewriteRule ^(.*)$ - [F,L]
</IfModule>
```

Для всех попыток обратиться к этой директории будет выдаваться сообщение об ошибке 403 Доступ запрещен.

Вместо ошибки, можно перенаправлять на определенную страницу, например на главную страницу сайта

```
RewriteRule ^(.*)$ http://domain.tld/ [R,L]
```

Но учтите, такой способ защиты может привести к непредсказуемым ошибкам, если например какому-то плагину для работы понадобится доступ в эту директорию и к ее файлам. Один из примеров- плагин **Subscribe to Comments**, перестанет работать его функция отписки от комментариев, т. к. пользователю для этого понадобится доступ в административную часть.

## Защищаем вход в админку

Вход на сайт — лакомый кусочек для хакеров. Это своего рода виртуальная дверь в админку. Есть несколько эффективных стратегий защиты:

- Сложный пароль
- Частое изменение пароля
- Запрет доступа на вход

Не будьте админом с ником **admin**! В прежних версиях вордпресс по умолчанию предлагался логин `admin` для регистрирующегося админа, поэтому взломщику оставалось только подобрать пароль. В новых версиях при регистрации нужно уже самостоятельно придумывать логин, поэтому

ZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZ



будьте оригинальны и неочевидны.

Чтобы помешать подбору пароля на сайт, помогут плагины

**Login LockDown** <http://wordpress.org/plugins/login-lockdown/> или **Limit Login Attempts** <http://wordpress.org/plugins/limit-login-attempts/>

Путем настройки максимального количества попыток ввода пароля, мы существенно снижаем шансы на его подбор. При превышении числа ошибок, данный айпи будет заблокирован на определенное время.

Другой способ улучшить безопасность, это убрать сообщение об ошибке входа на сайт, т. к. она содержит информацию о том, что было введено неправильно — логин или пароль. Добавив в `functions.php` следующий код

```
add filter('login errors',create function('$a', "return null;"));
```

мы полностью отключаем это уведомление.

Следующий вариант — плагин **WP-Adminprotection** <http://wordpress.org/plugins/wp-adminprotection/> который определяет разрешенные айпи для авторизации. Для всех остальных доступ будет заблокирован.

Еще один интересный способ — плагин **AskApache Password Protect** <http://wordpress.org/plugins/askapache-password-protect/> создающий дополнительный пароль для доступа к папке админки. Т.е. чтобы войти на сайт, нужно знать пароль доступа к папке и пароль доступа к админке.

Если же вы оказались в ситуации, когда забыли пароль к сайту, а доступа к почте нет, чтобы восстановить пароль, можно создать свой новый пароль прямо в базе данных сайта. Для этого:

1. используя phpMyAdmin зайдите в свою базу данных
2. найдите таблицу wp\_users и выберите своего пользователя
3. в поле user\_pass вы увидите свой старый зашифрованный пароль, замените его на свой новый пароль
4. выберите опцию MD5 из выпадающего меню и сохраните изменения

Таким образом в базе данных будет записан ваш новый пароль в зашифрованном виде. Имейте в виду, что MD5 абсолютно безопасный метод хранения паролей, т. к. он работает только в одну сторону и расшифровать зашифрованный код невозможно.

## Блокируем доступ на страницу логина

Блокирование доступа- отличный способ защиты, однако это мешает простым пользователям зарегистрироваться и авторизоваться на сайте. Специфика вордпресс такова, что код авторизации и регистрации на сайте находится в одном файле wp-login.php, а для регистрации на сайте нужно открывать урл

<http://domain.tld/wp-login.php?action=register>

ZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZ



возникают неожиданные ошибки, то просто удалите этот код из файла htaccess

Кроме этого для дополнительной защиты стоит использовать 8 опций ключей. Есть специальный генератор, создающий уникальные и неповторимые защитные ключи  
<https://api.wordpress.org/secret-key/1.1/salt/>

которые нужно скопировать и вставить в файл `wp-config.php` вместо тех, что прописаны там.

## Остальные файлы, нуждающиеся в защите

В дополнение к вышеперечисленным файлам, в защите нуждаются следующие файлы, которые не всегда связаны непосредственно с вордпрессом, но присутствуют практически на каждом сайте и потенциально уязвимы:

- .htaccess
- .htpasswd
- php.ini
- PHP скрипты
- .fla, .psd файлы — файлы флеш и фотошопа
- лог файлы

Как и остальные, эти файлы участвуют в конфигурировании сервера, выполнении скриптов или являются источниками контента. Защита этих файлов легко выполняется блокированием доступа по определенным типам файлов.

```
<FilesMatch "\.(htaccess|htpasswd|ini|php|fla|psd|log)$">
Order Deny,Allow
Deny from all
Allow from 123.456.789.0
</FilesMatch>
```

При настройке правил htaccess учитывайте тот факт, что его правила распространяются не только на ту директорию в которой он размещен, но и на все вложенные в нее.

## Удаляем номер версии Вордпресс

Следующий шаг — удаление номера версии вордпресса, который может подсказать взломщику потенциальные уязвимости, существующие для нее. По умолчанию, вордпресс включает номер версии в секции <head> с примечанием «*оставьте это для статистики, пожалуйста*». Статистика конечно вещь хорошая, типа знать какую долю интернета захватили сайты на вордпрессе, но безопасность прежде всего.

Вот код который выводит номер версии

```
<meta name="generator" content="WordPress <?php bloginfo('version'); ?>" />
```

Соответственно для предотвращения вывода этой информации либо физически удаляем эту

ZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZ



функцию из файла `header.php`, либо если она генерируется функцией `wp_generator` из глубины ядра вордпресса, то удаляем через вставку кода в `functions.php`

```
<?php remove_action('wp_head', 'wp_generator'); ?>
```

## Защищаем Базу Данных

Не устаем напоминать, как важна и ценна база данных сайта. Поэтому также озаботьтесь ее защитой. Стандартный префикс базы данных `wp_`. Миллионы сайтов используют этот префикс и огромное количество хакеров знает об этом. Многие SQL инъекции и автоматические эксплойты используют этот префикс в процессе атаки. Поэтому изменив префикс, мы обломаем хороший процент ломателей.

Вот подходящие плагины для этой цели

**Change Table Prefix** <http://wordpress.org/plugins/change-table-prefix/>

**Acunetix WP Security** <http://wordpress.org/plugins/wp-security-scan/>

Если конечно у вас руки чешутся все проделать самостоятельно, то есть два способа это сделать в зависимости от того, успели или еще нет вы установить вордпресс.

Меняем префикс перед установкой вордпресс — для этого в процессе инсталляции выбираете нетипичный и неочевидный вариант, например `sdfo_`

## Меняем префикс для существующего сайта

1. Бэкап!
2. Меняем в файле `wp-config.php` значение `$table_prefix` на новое.
3. Выполняем запрос к базе данных через phpMyAdmin

Запросы должны быть следующие (вместо xxx укажите свой новый префикс)

```
rename table wp_comments to xxx_comments;
rename table wp_links to xxx_links;
rename table wp_options to xxx_options;
rename table wp_postmeta to xxx_postmeta;
rename table wp_posts to xxx_posts;
rename table wp_terms to xxx_terms;
rename table wp_term_relationships to xxx_term_relationships;
rename table wp_term_taxonomy to xxx_term_taxonomy;
rename table wp_usermeta to xxx_usermeta;
rename table wp_users to xxx_users;
```

Также проверьте какие еще есть таблицы и в них тоже измените префикс. Затем откройте таблицу `options` и в колонке `option_name` найдите `wp_user_roles` и замените его на `xxx_user_roles`. Кстати, после установки вордпресс содержит 11 таблиц, но в процессе эксплуатации это число может вырасти значительно.

Откройте таблицу `usermeta` и в колонке `meta_key` найдите все поля с префиксом `wp_` и замените на новые.

ZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZ

////////////////////////////////////



```
RewriteCond %{REQUEST_FILENAME} -f
RewriteCond %{REQUEST_FILENAME} \.(gif|jpe?g?|png)$ [NC]
RewriteCond %{HTTP_REFERER} !^https?://([^.]+\.)?domain\. [NC]
RewriteCond %{HTTP_REFERER} !^http://www.feedburner.com/.*$ [NC]
RewriteCond %{HTTP_REFERER} !^http://feeds.feedburner.com/example-feed$ [NC]
RewriteCond %{HTTP_REFERER} !^http://feeds.feedburner.com/example-feed-comments$ [NC]
RewriteRule \.(gif|jpe?g?|png)$ - [F,NC,L]
</IfModule>
```

Ну и понятно, что вместо domain укажите домен своего сайта.

## Еще больше помощи в защите

На любом сайте всегда есть, что улучшить. Для большей информации и большего набора инструментов проверьте следующие плагины

- **Secure WordPress** <http://wordpress.org/plugins/secure-wordpress/> предлагает массу различных настроек безопасности и инструментов защиты
- **WP Security Scan** <http://wordpress.org/plugins/wp-security-scan/> сканирует вордпресс на известные уязвимости и предлагает методы их решения
- **WordPress File Monitor Plus** <http://wordpress.org/plugins/wordpress-file-monitor-plus/> сканирует файлы вордпресс на наличие вредного кода и уведомляет о результатах проверки. Если файл пропал, изменился или переместился- вы сразу об этом узнаете.
- **Ultimate Security Check** <http://wordpress.org/plugins/ultimate-security-check/> сканирует на сотни различных проблем и предлагает улучшения безопасности.
- **BulletProof Security** <http://wordpress.org/plugins/bulletproof-security/> создает защиту сайта на основе хитрых конфигураций .htaccess, блокируя XSS, RFI, CRLF, CSRF, Base64, Code инъекции и SQL инъекции
- **WordPress AntiVirus** <http://wordpress.org/plugins/antivirus/> эффективное решения для защиты сайта от различных заражений
- **WordPress Firewall** <http://www.seoegghead.com/software/wordpress-firewall.seo> блокирует потенциальные угрозы на основе списка подозрительных параметров
- **Stealth Login** <http://wordpress.org/plugins/stealth-login-page/> защищает админку сайта
- **Exploit Scanner** <http://wordpress.org/plugins/exploit-scanner/> сканирует файлы, базу данных и плагины на различные подозрительные вещи
- **Block Bad Queries** <http://wordpress.org/plugins/block-bad-queries/> блокирует слишком длинные запросы к сайту, а также различные потенциально опасные

## Останавливаем спам в комментариях

Спам в комментариях — это чума, которая пытается поглотить каждый форум или сайт с

ZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZ



открытыми комментариями, и вордпресс не исключение. Но нет повода для печали- нам помогут несколько плагинов и полезных настроек.

- **Akismet** <http://akismet.com/> идет в комплекте вордпресс, легко настраивается и эффективно работает
- **Bad Behavior** <http://wordpress.org/plugins/bad-behavior/> анти-спам защита и дополнительный функционал защиты блога
- **WP-SpamFree** <http://wordpress.org/plugins/wp-spamfree/> уничтожает автоматический спам без помощи капч
- **NoSpamNX** <http://wordpress.org/plugins/nospamnx/> добавляет скрытые поля для блокировки плохих ботов

## Настраиваем опции в админке

Ранее мы уже подробно обсуждали безопасные и надежные опции в меню **Параметры — Настройки обсуждения** позволяющие полностью отключить комментирование. Если же возможность оставлять комментарии нужна, то стоит настроить тотальный контроль и модерацию всего публикуемого контента. Это позволит как минимум избежать опубликованного и видимого на сайте спама, т. к. он будет висеть в очереди на проверку.

## Используем встроенные инструменты модерации

Модерация комментариев и черный список слов позволят по мере набора статистики блокировать наиболее употребляемые слова и понятия, а также айпи адреса заядлых и неутомонных спамеров. А потом собранную базу можно скопировать и вставить на другой свой сайт.

Но будьте осторожны с черными списками — запретное слово может оказаться частью другого, вполне нормального слова. Поэтому лучше использовать полные названия, без отбрасывания окончаний. Хотя одно хорошо известное слово, будучи включено в список блокировки, заблокирует другое невинное слово «застрахуй».

И так как спам делается с целью получения обратной ссылки, то будет правильным настроить автоматическую отправку на модерацию комментариев, содержащих более 1 ссылки.

## Отключаем комментарии в старых записях

Спаммеры стараются атаковать или самые свежие публикации, чтобы в индекс попала страница сразу с размещенной их ссылкой, или самые старые публикации, которые уже находятся в индексе и могут иметь накопленный вес и пейдж ранк. Поэтому по мере роста количества записей, вам придется иметь дело с все большим и большим числом спама и другого бесполезного хлама.

Легкое и простое решение- автоматически закрывать обсуждение статей старше заданного числа дней. В зависимости от типа публикуемого у вас контента и скорости его устаревания, это число стоит варьировать. Например, если у вас публикуются новости, то их можно закрывать уже через 14 дней, если тематические статьи, то через 30-60 дней.

## Блокируем доступ запросам

Многие скрипты спамботов целятся непосредственно на скрипт комментариев вордпресса, пропуская заполнение формы комментариев на странице. Легкий способ вычислить такое

ZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZ





практически любую активность и действия на сайте — вы будете в курсе всего, что происходит у вас

- **Post Logger plugin** <http://www.village-idiot.org/archives/2008/04/16/postlogger-for-wordpress/> показывает подробности всех запросов \$POST к вашему сайту, позволяя следить за всеми комментаторами
- **TTC WordPress Tripwire Tool** <http://wordpress.org/plugins/ttc-tripwire-plugin/> отслеживает изменения файлов за заданный период времени
- **InspectorWordpress** <http://wordpress.org/plugins/inspector-wordpress/> следит и сохраняет в журнал все запросы к вашему сайту

## Сервисы онлайн мониторинга

Важный момент в создании и развитии сайта, это постоянный контроль над тем, что он всегда доступен для всех посетителей. В идеальном мире, ваш сайт доступен 100% времени, но из-за проблем на серверах, конфликтов программного обеспечения, вирусных атак, взломов, ддосов и т. п. ваш сайт будет недоступен время от времени.

Т.к. вы не можете полностью предотвратить периоды неожиданных падений сайта, вы можете улучшить свою систему мониторинга и реакции на падения сайта. Онлайн сервисы мониторинга обычно занимаются тем, что с заданной периодичностью опрашивают ваш сайт и сообщают вам сразу же, как только он становится недоступен. Есть масса платных и бесплатных сервисов с разным набором возможностей. Вот некоторые из них:

- **Are My Sites Up?** <http://aremysitesup.com/> платный сервис с возможностью бесплатного мониторинга в течение 15 дней
- **Pingdom** <https://www.pingdom.com/pricing/> платный сервис с возможностью бесплатного мониторинга в течение 30 дней
- **Mon.itor.us** <http://mon.itor.us/en/free-monitoring-features/website-uptime-monitoring> бесплатный сервис мониторинга с рассылкой писем в случае недоступности сайта, проверяет каждые 5 минут
- **Montastic** <http://www.montastic.com/> бесплатный сервис мониторинга с рассылкой писем в случае недоступности сайта, проверяет каждые 10 минут
- **BasicState** <http://basicstate.com/> бесплатный сервис мониторинга с рассылкой писем в случае недоступности сайта, проверяет каждые 15 минут
- **Яндекс-Метрика** <https://metrika.yandex.ru> эта система аналитики и мониторинга пользователей также предлагает возможность информирования на емейл в случае недоступности сайта

zzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzz



## Всегда актуальный Вордпресс

Конечно же, один из лучших способов сохранять свой сайт в безопасности, это иметь всегда актуальный вордпресс. Будучи в админке блога, всегда следите за уведомлениями о обновлениях движка, шаблонов и плагинов. Постоянные обновления будут давать вам все самые последние безопасные заплатки, фиксы глюков и улучшения.

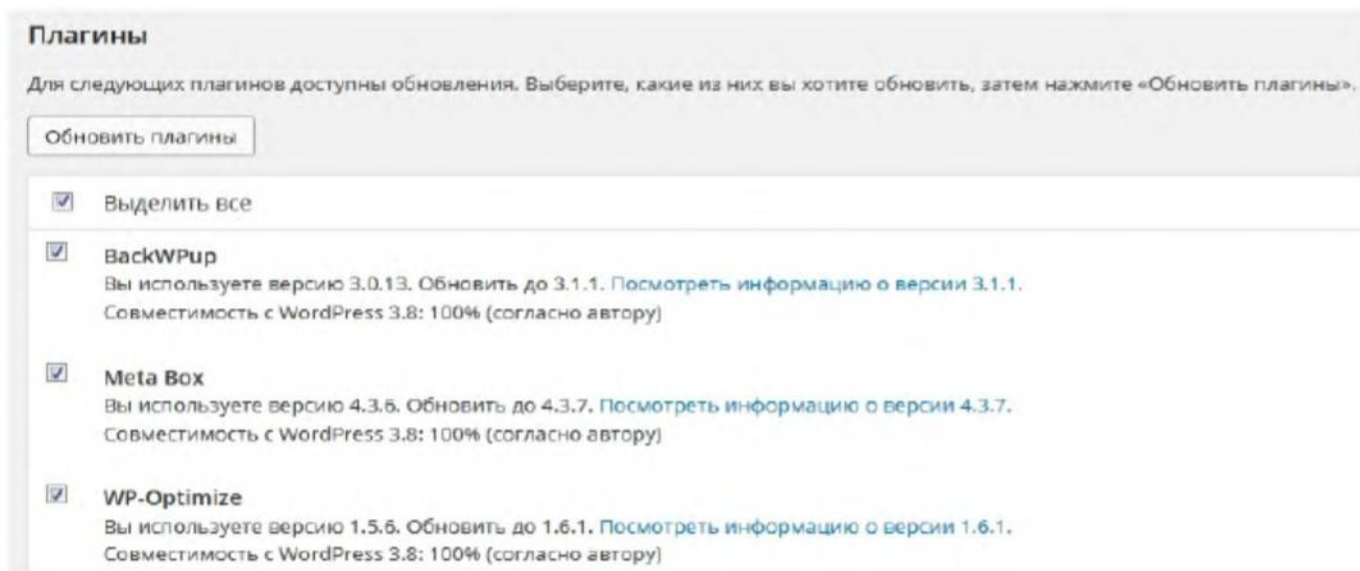
Также, комплексное обновление и плагинов, и движка, обеспечит их полную совместимость и предотвратит возможные глюки в дальнейшем. Продвинутые пользователи вордпресс также читают все новости, которые публикуются в виджете на главной странице админки, там сообщается о изменениях и планах по развитию как самого движка вордпресс, так и о различных популярных плагинах.

Если вы обнаружили какую-то ошибку, вы можете отправить запрос на странице кодекса вордпресс [http://codex.wordpress.org/Reporting\\_Bugs](http://codex.wordpress.org/Reporting_Bugs) . А в случае обнаружение серьезной проблемы или уязвимости, можно написать непосредственно команде, занимающейся вопросами безопасности на емейл [security@wordpress.org](mailto:security@wordpress.org)

## Обновление вордпресс

Новые версии и обновления движка выходят достаточно часто — вордпресс очень динамически развивается. В среднем, 4 раза в год выходят новые версии, а в течение года — несколько промежуточных версий с исправлениями и дополнениями к релизу.

Обновляться очень просто — достаточно кликнуть по ссылке Обновить в уведомлении и система автоматически выполнит все необходимые действия. Точно также обновляются и плагины — их можно обновлять как по одному, так и массово выделить все и обновить группой.



Перед обновлением плагинов, шаблонов и движка рекомендуется сделать резервную копию базы данных, а также вспомнить — не вносили ли вы ручные правки в какие либо из файлов, т. к. при обновлении они будут перезаписаны и потеряны.

ZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZ

Все обновления завершены.

zzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzz







- **W3 Total Cache** <http://wordpress.org/plugins/w3-total-cache/>

На данный момент это три самых основных конкурента по популярности. Перед использованием рекомендуется изучать документацию и интерфейс, но в целом в большинстве ситуаций они равнозначны. И уж тем более не рекомендуется устанавливать одновременно несколько кеширующих плагинов одного типа. Комбинирование допускается в случае, если один плагин занимается файловым кешированием, а другой кеширует запросы к базе данных.

А для наблюдения за кэшированием можно использовать **WP Cache Inspect** <http://wordpress.org/plugins/wp-cache-inspect/>, который будет выводить подробную информацию о том, какие и сколько файлов кэша создано, как они используются и сколько раз вызываются.

И возможно самый лучший способ добиться максимальной производительности и скорости загрузки сайта — это выбрать правильный хостинг. Ничто так не влияет на скорость и постоянную доступность сайта, как быстрый сервер, поэтому держитесь подальше от дешевых хостингов и различных распродаж, когда на один сервер забивается огромное количество пользователей с своими сайтами, создающими обильную нагрузку.

Понятно, что хороший хостинг стоит дороже. Но не всегда более дорогой хостинг означает лучший. Поэтому только остается посоветовать искать, изучать и пробовать.

## Метод сжатия файлов

Другой способ улучшения производительности и экономии трафика, это использование сжатия файлов и страниц сайта. Сжатие не означает сжатия файлов в прямом смысле этого слова, файлы и страницы остаются такими какие есть, но сжатие происходит на сервере перед передачей их посетителю. Как только страница загружена в браузер, она распаковывается и отображается корректно. В результате это приводит к уменьшению размеров передаваемых файлов и передаваемого трафика, и соответственно к уменьшению времени загрузки страницы.

Вот небольшая экскурсия в используемые техники сжатия, к которым вы точно обратитесь по мере достижения определенного опыта в настройке сайтов и серверов

1. Сжатие файлов Apache's gzip module — для серверов, работающих под управлением старых версий Апаха, позволяет директивами `mod_gzip` в `htaccess` файле указать что и как сжимать
2. Сжатие файлов Apache's deflate module — для серверов, работающих под управлением новых версий Апаха, настраивается директивами `mod_deflate`
3. Сжатие методами PHP, используя сжатие выходного буфера. Обычно реализуется вставками специальных скриптов в файл `header.php`, подробнее можно почитать тут <http://perishablepress.com/fast-effective-php-compression/>
4. Ручное сжатие файлов — для всех статических файлов типа `css`, `javascript` и т. п. Возможно существенное уменьшение размера путем выполнения архивации файлов в `gzip` архив
5. Уменьшение CSS и JavaScript файлов — в отличие от упаковки в архив, речь идет о удалении из файлов всей служебной и ненужной информации — комментариев, пробелов, переносов и т. п. Автоматическое уменьшение таких файлов поддерживают некоторые кеширующие плагины

Собственно для минимизации файлов можно использовать плагин **WP Minify** <http://wordpress.org/plugins/wp-minify/> или **PHP-Speedy** <http://aciddrop.com/php-speedy/>, которые выполняют уменьшение размеров, комбинирование и объединение нескольких

ZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZ

файлов, а также кеширование.

Кроме этого можно воспользоваться онлайн сервисами и скриптами для минимизации размеров CSS и JavaScript файлов:

- **YUI Compressor** <http://yui.github.io/yuicompressor/>
- **Dean Edwards Packer** <http://dean.edwards.name/packer/>
- **JavaScript Compressor** <http://javascriptcompressor.com/>
- **JavaScript Compressor** <http://www.xmlforasp.net/JSCompressor.aspx>
- **Styleneat.com** <http://www.styleneat.com/>
- **JSMin** <http://crockford.com/javascript/jsmin>


В то время как есть много разных способов выполнить компрессию, ваша реализация прежде всего зависит от возможностей вашего сервера. Если можно, то включите `mod_gzip` или `mod_deflate` и забудьте об остальных способах.

Если же эти варианты недоступны, то тогда уже используйте другие методы.

## Оптимизируем CSS и JavaScript

Если все эти серверные штучки и настройки сжатия-пережатия вызывают потерю сознания, то расслабьтесь — большой вклад в оптимизацию можно сделать простой концентрацией на коде и контенте, используемых для создания пользовательских интерфейсов. Оптимизируя код JavaScript, CSS, и даже HTML страниц, вы можете существенно снизить размер файлов и время загрузки.

Вот вам несколько ключевых стратегий оптимизации кода:

- Пишите понятный код! Убирайте ненужные комментарии и пометки, избегайте путанных и сложных конструкций.
  - Уменьшайте http запросы до минимума. Это также очень сильно влияет на время загрузки. Каждый вызываемый файл, будь то скрипт, изображение или файл стиля создает свой http запрос и снижает время загрузки. Объединяя несколько css и javascript файлов в один и используя технологию спрайтов, когда несколько изображений объединяются в одно, вы можете значительно уменьшить число создаваемых запросов.
  - Используйте спрайты. Это изображение, состоящие из нескольких склеенных других изображений. Размещая их в одном файле, мы снижаем задержку на загрузку файла и уменьшаем число http запросов. Также спрайты вместе с css используются для создания различных эффектов и сменяющихся при наведении изображений, создания фоновых изображений и т. п. Для полуавтоматического создания спрайтов воспользуйтесь <http://spriteme.org/>
  - Загружайте файлы стилей вверху страницы. Все файлы стилей должны загружаться в секции <head> для того, чтобы браузеры отрисовывали содержимое страницы постепенно по мере загрузки.
- 





- Загружайте JavaScript файлы внизу страницы. Все файлы внешних скриптов должны загружаться непосредственно перед закрывающим тегом `<html>` , чтобы обеспечить полноценную загрузку страницы и выполнения скриптов на ее полностью готовом коде.
- Проверяйте свой код. Это самый простой способ убедиться, что все работает и отображается так как надо и соответствует всем требованиям и стандартам языка. После проверки, валидатор сообщит о всех найденных ошибках и покажет рекомендации по их устранению.

W3C (X)HTML Validator <http://validator.w3.org/>

W3C CSS Validator <http://jigsaw.w3.org/css-validator/>

А теперь об этом поподробнее.

## Снижение числа http запросов

Каждый вызов отдельного файла — это отдельный запрос к серверу, своему или внешнему. Есть определенные ограничения на количество одновременных запросов к одному серверу, поэтому все невыполненные запросы остаются в очереди, увеличивая и без того большое время загрузки страницы. Это хорошо видно на сайтах с огромным количеством подключаемых `css` и `javascript` файлов.

Вот основные методы борьбы:

1. Уничтожайте ненужные файлы. Вызывайте только те файлы и только тогда, когда они действительно нужны
2. Объединяйте CSS файлы. Вместо того, чтобы обращаться к пяти css файлам, объедините их в один. Особенно много разных файлов появляется при использовании плагинов, т. к. почти каждый плагин подключает свой файл стилей.
3. Объединяйте JavaScript файлы. Точно так же как и в случае с css, нужно группировать и объединять скрипты в один.
4. Используйте спрайты изображений. Многие шаблоны имеют большое количество мелких деталей- иконки социальных сетей, кнопки, элементы дизайна, а это все отдельные файлы и отдельные запросы. Чтобы объединить это все в спрайты и разобраться с этой технологией потребуется некоторое время, но оно того стоит — это очень эффективный прием.
5. Комбинированное сжатие. С помощью различных плагинов и скриптов достигается максимальный эффект сжатия, когда файлы сначала минимизируются, затем объединяются, комбинируются, и упаковываются в архив.
6. Прогрессивная загрузка изображений. Вместо того, чтобы одновременно пытаться загрузить все изображения на странице, делайте загрузку для тех изображений, что не попали в поле зрения. Это особенно эффективно на длинных страницах с большим количеством картинок. По своей сути это способ обратной предзагрузки — когда в фоновом режиме загружается тяжелая страница, зато при переходе на нее- она открывается моментально. При быстром проматывании страницы с прогрессивной загрузкой можно наблюдать небольшую задержку в появлении изображений, но это дает очень огромный выигрыш в производительности. Используйте для этой цели плагин Lazy Load <http://www.appelsiini.net/projects/lazyload>

[illegible]



ZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZ

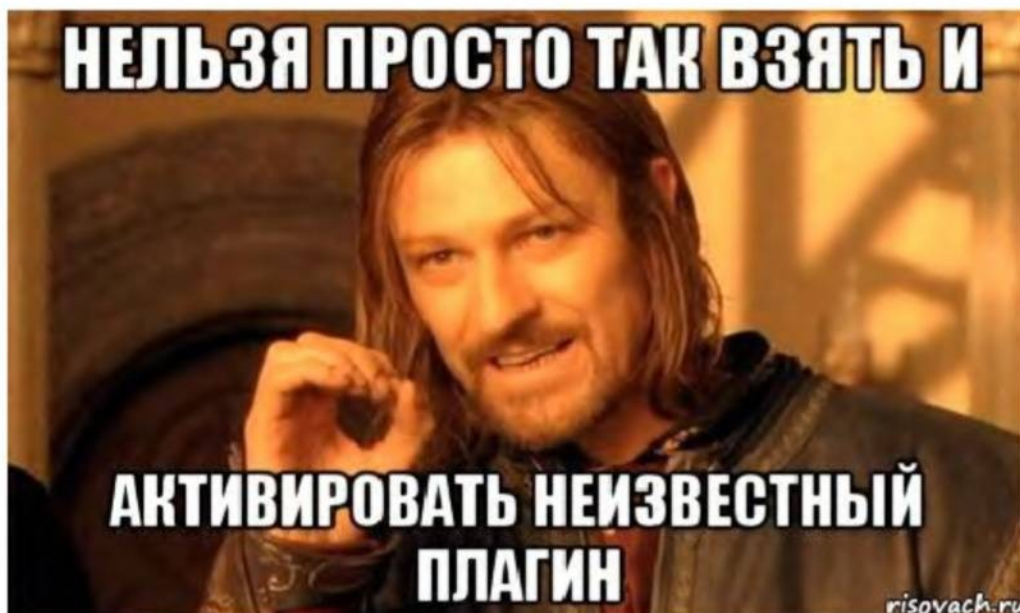
Если покопаться в этих отчетах, можно найти очень много полезного и интересного.

## Уход за плагинами

Много статей в интернете рекомендует отключать неиспользуемые плагины для оптимизации сайта. Но это лишь половина правды. Все начинается тогда, когда вы выбираете какой плагин устанавливать. Делайте некоторые исследования и принимайте осознанное решение, вместо того чтобы ставить все подряд. Каждый установленный плагин потребляет ресурсы и может снизить производительность.

Также возможен накопительный эффект, когда сочетание двух определенных плагинов может привести к непредсказуемым последствиям. Но все же, вполне бывают сайты с активированными и прекрасно и шустро одновременно работающими 50 плагинами. Поэтому просто контролируйте то, что вы делаете.

Далее, после того, как выбранный плагин был установлен, проверьте- все ли работает как надо?



Если вы замечаете глюки или тормоза, поменяйте свое решение и выберите другой плагин. В большинстве случаев всегда есть масса альтернатив.

Далее, чтобы взять под контроль весь имеющийся арсенал плагинов, имейте в виду, что следует избегать плагинов про запас, которые могут понадобиться когда-нибудь, но совершенно бесполезны сейчас. Деактивируйте и деинсталируйте абсолютно все ненужные и неиспользуемые плагины. Каждый раз, когда загружается страница вашего сайта, все плагины загружаются в фоновом режиме. Поэтому любой плагин, который можно убрать — нужно убрать.

Ну и как многократно говорилось, регулярно обновляйте плагины. Вполне вероятно, что кроме устранения старых глюков, в плагине могут появиться новые, но в большинстве случаев обновления приносят улучшения, повышенную безопасность и новые фишки.

И наконец, как только вы решаете отключить какой-либо плагин- удаляйте его полностью. Не оставляйте его занимать место в админке в выключенном виде. Аниинсталируйте его и убедитесь, что все следы его пребывания также удалены.

////////////////////////////////////





```
ExpiresActive On
ExpiresByType image/gif A2592000
ExpiresByType image/png A2592000
ExpiresByType image/jpg A2592000
ExpiresByType image/ico A2592000
```

Имейте в виду, что с таким кешированием при необходимости изменить изображение, нужно его переименовать, чтобы изменения увидели посетители имеющие закешированную копию.

## Уменьшаем размер изображений

Очевидно, что огромные изображения требуют большого времени для загрузки, чем маленькие. Есть миллионы статей на эту тему, объясняющих как и зачем оптимизировать изображения, поэтому будем краткими.

- Онлайн-сервисы оптимизации, такие как **Smush.it** <http://developer.yahoo.com/yslow/smushit/> или **punypng** <http://www.punypng.com/> выполняют свою работу так- вы загружаете неоптимизированное изображение, а взамен скачиваете оптимизированное. Лучше всего они работают с JPEG, GIF и PNG. Оба сервиса используют технологию сжатия без потери качества, поэтому вы не потеряете ни грамма удовольствия от просмотра сжатых изображений.
- Плагин **WP Smush.it** <http://wordpress.org/plugins/wp-smushit/> который на лету выполняет оптимизацию и сжатие уже загруженных на сайт изображений
- Офлайн-оптимизационные программы, типа **OptiPNG** <http://optipng.sourceforge.net/> или **PNGcrush** <http://pmt.sourceforge.net/pngcrush/>. Каждая из этих программ удаляет всю ненужную служебную информацию из png изображений, не ухудшая качества.
- Оптимизация изображений в процессе создания. Возможно самый лучший способ оптимизации- создавайте изображения именно в том размере\разрешении, в котором они будут использованы на сайте без масштабирования, сохраняйте изображения в требуемом уровне качества, т. к. визуально не видно разницы между 100% и 80% качеством сжатия, а вот размер файла может отличаться в разы.

## До бесконечности и дальше

Как мы увидели- есть масса способов, методов, плагинов и техник. Прямо из коробки вордпресс работает достаточно хорошо, но нет предела совершенству, поэтому даже с небольшим количеством усилий можно добиться впечатляющего функционала и производительности. Начиная с простой установки пары кеширующих плагинов и заканчивая вручную написанными функциями, безопасность, производительность и работоспособность может развиваться вами настолько, насколько вы этого захотите.

Вам показалось этого мало — встречайте нашу новую книгу, Тао Вордпресса\*, самое полное руководство для начинающих, студентов, админов, пользователей и всех, кто хочет свежие и простые подходы для того, чтобы рулить в Вордпрессе.

\*От переводчика: При наличии желания, вы также можете получить эту книгу в моем переводе.

Надеюсь, было интересно и полезно.

Спасибо!

[illegible]