

Увеличение скорости загрузки страниц.

На данный момент скорость загрузки страниц влияет на многие показатели. От скорости зависит задержится посетитель на вашем сайте или не дожидается и перейдет на другой. Да и поисковые системы, на данный момент при ранжировании, тоже обращают внимание на скорость загрузки страниц вашего сайта. На SEO оптимизацию этот показатель тоже влияет. Вывод - от скорости загрузки страниц зависит количество пользователей, а от них уже и ваш доход.

В связи с этим хочется дать несколько не сложных советов, для тех кто только начинает свою деятельность на этом поприще. Гениальных идей вы конечно не найдете но десяток полезных рекомендаций никому не помешают.

Уменьшение HTTP-запросов

Загрузка компонентов: скриптов, фотографий, флеш, файлов CSS, занимает около 80% всей загрузки страницы. Увеличить скорость загрузки можно с помощью уменьшения загружаемых компонентов одновременно. Советую, что бы браузер одновременно загружал не более двух компонентов. Таким образом уменьшив количество одновременных запросов посылаемых серверу, увеличиваем скорость загрузки страницы.

У большинства читателей возникает вопрос. Как это сделать без ущерба для дизайна страницы?

Этого можно достичь разными способами:

- 1. Воспользовавшись CSS-спрайтов.** CSS-спрайт – изображение состоящее из нескольких фрагментов. То есть картинка состоит из нескольких небольших. Они в свою очередь используются в нужный момент, с помощью background-image и background-position.
- 2. Использование Inline-картинок.** Inline-картинки с помощью URL-схемы data внедряются непосредственно в страницу. Но при этом увеличиваются объемы HTML-документа. Можно внедрять Inline-картинки в таблицы стилей, в результате также уменьшится количество запросов. Нужно заметить при такой схеме размеры HTML не изменятся.
- 3. Соединение файлов в один.** Это возможно в случае если на странице подключается несколько CSS или JS- файлов, их можно объединить. Сделать это несложно, а результатом будет опять таки уменьшение количества запросов на сервер.

Размещайте CSS файлы в самом начале страницы.

Вся суть состоит в том, что разместив подключение к CSS файлам в хедере вы добиваетесь поэтапного рендеринга страниц. При этом вся страница грузится постепенно: заглавие, лого, навигация и так далее. Визуально это положительно влияет на пользователей, он видит процесс загрузки страницы наглядно. Посетителю не нужно смотреть в пустой экран.

При ситуации когда CSS файлы находятся в низу страницы, рендерить страницу постепенно большинство браузеров не могут. Некоторые элементы страницы после

загрузки могут изменяться, браузер в свою очередь не хочет их прорисовывать. По этому CSS файлам лучше размещать в секции HEAD.

Javascript лучше разместить в конец страницы.

Это даст возможность браузеру, страницу на котором находится контент загружать первой, а после этого уже javascript-файлы. При наличии на сайте нескольких таких файлов нежелательно заставлять посетителей ждать пока все загрузится, пользователям это не очень нравится. Для избежание этого нужно размещать javascript-файлы в конце страницы.

Js файлы блокируют одновременную загрузку. Лучшим вариантом загрузки является не больше двух компонентов параллельно.

При использовании внешних javascript-файлов блокируется одновременная загрузка. Помните, когда рисунки для вашего сайта хранятся на разных хостах, в любом случае вы получаете параллельную загрузку больше двух элементов. При загрузке скриптов, браузер блокирует любые другие компоненты. С других хостов загрузка тоже приостанавливается.

Css и javascript, минимизация.

Когда нужно минимизировать файл, из него удаляются все коды которые не играют существенной роли. Как результат получаете меньшие объемы и быструю загрузку. Удалить можно: табуляцию, перенос строк, не выполняющие важных функций пробелы, комментарии. Зависимость прямо пропорциональная, уменьшаете объем – уменьшается время загрузки страницы. Ответы по вопросам минимизации можно найти в интернете.

Для скачивания пользуйтесь под доменами.

Напомним, что браузеры используют ограничение, загружая параллельно только два компонента сайта с одного хоста. В случае если на вашем сайте используется много графических элементов, их лучше разместить на отдельном под домене или даже на нескольких. Браузер будет воспринимать это, как несколько разных серверов. От количества созданных под доменов зависит число параллельных загрузок, чем их больше, тем больше файлов будут загружаться одновременно и наоборот. Скорость загрузки страницы так же будет больше. Вам нужно будет поменять только адреса графических компонентов. Все довольно просто.

Применяйте кеш.

Кеширование играет важную роль для веб-сайтов, особенно если на них обширное подключение CSS и JS. При посещении сайта пользователем впервые браузер будет загружать все файлы CSS, JS, флеш, графику и т.д. Но с помощью HTTP-заголовка Expires, можно кешировать элементы страниц. Правильно настроив HTTP-заголовок Expires вы добьетесь этого. Результат: при повторном посещении вашего ресурса или переходе на другую страницу, в кеше браузера сохраняются файлы используемые для загрузки следующей страницы. Так повторная загрузка этих файлов не нужна, скорость загрузки страницы само собой увеличивается.

Рекомендуется расставлять HTTP- заголовки Expires там, где только возможно. Лучше выставлять их на перед, от нескольких дней до месяца. Веб – сервер будет отдавать нужные команды HTTP- заголовкам Expires, если разместить в htaccess файл, корневая папка сайта, нужные строки.

Такие как:

```
<IfModule mod_expires.c>
Header append Cache-Control "public"

FileETag MTime Size
ExpiresActive On
ExpiresDefault "access plus 0 minutes"
ExpiresByType image/ico "access plus 1 years"
ExpiresByType text/css "access plus 1 years"
ExpiresByType text/javascript "access plus 1 years"
ExpiresByType image/gif "access plus 1 years"
ExpiresByType image/jpg "access plus 1 years"
ExpiresByType image/jpeg "access plus 1 years"
ExpiresByType image/bmp "access plus 1 years"
ExpiresByType image/png "access plus 1 years"
</IfModule>
```

Этот фрагмент файла в форме Веб-сервера Apache следит за наличием модуля мод экспрес и когда он доступен, начинает работу отдача HTTP-заголовков Expires, от которых зависит период хранения, выше указанных элементов в кэше браузера серверов прокси. С первой загрузки на протяжении одного года.

При большом сроке жизни кэша браузера, иногда возникают проблемы с обновлениями. Решить эту проблему довольно просто, если вы вносите изменения в css и js –файлы, их нужно переименовать. Это необходимо сделать для того, что бы в кэше тоже появились изменения. Не усложняя себе жизнь просто добавьте в название файла его версию.

Применяйте CDN для загрузки JS библиотек.

Например когда на вашем сайте есть популярный js фреймворк, допустим jQuery, подключать его лучше пользуясь CDN.

Content Delivery Network – это большое количество веб – серверов, которые охватывают большую географическую территорию и при этом достигается наибольшая скорость загрузки. На выбор сервера, на котором будет размещаться информация влияют несколько факторов. Первый: предпочтение отдается серверу с самым маленьким количеством промежуточных хопов. Второй: сервер с минимальным временем ответа.

Браузер кэширует js файлы во время загрузки. При повторном посещении сайта где пользуются этим методом, браузер повторно загружать файл не будет. Потому что библиотека уже будет находится в кэше вашего браузера.

Например - Google Libraries один из CDN. Такие CDN предназначены для популярных open-source JS библиотек. При загрузке js фреймворков с Google Libraries дает возможность увеличения скорости при загрузке страниц и снизить трафик сервера.

Оптимизация изображений.

Для начала нужно определить оптимальный формат для графики на вашем сайте. От выбора формата зависит размер файлов, если его подобрать неправильно можно увеличить размеры, а не уменьшить.

GIF – изображение состоящее из нескольких цветов, это может быть логотип и т.п.

JPEG – большое количество цветов, высокая детализация. Например – фото.

PNG – прозрачность, очень высокая степень детализации, высокое качество изображения.

Используются два способа оптимизации графических компонентов. Можно воспользоваться онлайн сервисами, сжав изображение. Или воспользоваться специализированными программами. Во втором случае нужны определенные знания и навыки. А для онлайн сервисов нужен только интернет. Все довольно просто, загружаете изображение, сервер автоматически проводит оптимизацию, а вам нужно только скачать уже готовый файл.

Можете воспользоваться для оптимизации: punypng.com, Smush.it, jpegmini.com.

Не используйте масштабирование изображения.

Для изменения изображения не стоит использовать: width, height теги, CSS. Это может отрицательно сказаться на скорости загрузки. Если у вас рисунок изначально 500x500, а нужна вам 100x100. Изменять размеры лучше оригинала, до того как вы вставите ее на сайт, можете для этого воспользоваться например фотошопом или корелом. От объема картинки зависит скорость ее загрузки.

Gzip- сжатие.

Gzip- сжатие файлов содержащих текстовую информацию, почти в 100% случаев, дает возможность уменьшить время передачи файлов браузеру. Для освобождения соединения в три четыре раза, рекомендуем хранить архивные копии на сервере, прокси сервере или диске.

Веб – клиенты должны обозначить типы сжатия которые они поддерживают. Эта информация указывается в заголовке Accept-Encoding в запросах HTTP.

Accept-Encoding: gzip, deflate.

Это дает возможность веб-серверу считать такой заголовок, использовать нужный тип сжатия для ответа. Используя при этом один из указанных типов сжатия клиентом. При этом клиент получает уведомление о типе сжатия используемом для ответа, с помощью заголовка Content-Encoding.

Content-Encoding: gzip.

Плюсом такого вида передачи будет уменьшение объемов первоначального варианта в пять раз. В этом случае будет только одна негативная сторона, нагрузка на веб – сервер возрастает. Вопрос сервером решаем, по этому не стоит об этом беспокоиться.

Прописав в .htaccess файле сайта ниже указанные коды, вы сможете включить GZIP-сжатие.

```
<IfModule mod_deflate.c>
AddOutputFilterByType DEFLATE text/html
AddOutputFilterByType DEFLATE application/javascript
AddOutputFilterByType DEFLATE text/javascript
AddOutputFilterByType DEFLATE text/css
<IfModule mod_setenvif.c>
BrowserMatch ^Mozilla/4 gzip-only-text/html
BrowserMatch ^Mozilla/4.[0-6] no-gzip
BrowserMatch \bMSIE !no-gzip !gzip-only-text/html
</IfModule>
</IfModule>
```

Есть еще один вариант. Используйте его в случае отказа первого.

```
<IfModule mod_deflate.c>
AddOutputFilterByType DEFLATE text/html text/plain
text/xml application/xml application/xhtml+xml
text/javascript text/css application/x-javascript
BrowserMatch ^Mozilla/4 gzip-only-text/html
BrowserMatch ^Mozilla/4.0[678] no-gzip
BrowserMatch bMSIE !no-gzip !gzip-only-text/html
<ifmodule mod_gzip.c>
mod_gzip_on Yes
mod_gzip_item_include file \.js$
mod_gzip_item_include file \.css$
</ifmodule>
</IfModule>
```

Дать сто процентной гарантии, что эти коды будут работать на вашем сайте нельзя. Если возникнут проблем лучше обратится в службу поддержки хостинга и выяснить этот вопрос.

Из выше изложенного вы узнали о способах клиентской оптимизации. Воспользовавшись ими, вы сможете увеличить скорость загрузки страниц вашего сайта. Попробуйте и пользователи обязательно оценят ваши старания.