А. Артемов

Мониторинг информации в Интернете

А. В. Артемов

Мониторинг информации в Интернете

МОО "Межрегиональная общественная организация "Академия безопасности и выживания""

2014

Артемов А. В.

Мониторинг информации в Интернете / А. В. Артемов — МОО "Межрегиональная общественная организация "Академия безопасности и выживания"", 2014

В данном пособии рассматриваются принципы организации и поиска информации в интернете, приводится описание языков запросов поисковой машины Яндекс, Гугл, Рамблер.<p id="_t65">Предназначено для преподавателей и студентов вузов всех специальностей, руководителей и менеджеров компаний, специалистов по безопасности, а также для широкого круга лиц, работающих в сети Интернет.

Содержание

А. В. Артемов
Мониторинг информации в интернете: учебно-методическое пособие

Рецензент:
кандидат экономических наук, доцент кафедры «Предпринимательство и маркетинг» ФГБОУ ВПО «Госуниверситет – УНПК»
Н.А. Лебедева

А. В. Артемов, кандидат технических наук, доцент кафедры «Электроника, вычислительная техника и информационная безопасность» ФГБОУ ВПО «Госуниверситет – УНПК»

Лекция 1
Принципы организации и поиска информации в интернете

Поиск информации в Интернете проводится двумя основными способами – с помощью каталогов (их еще называют директориями) и с помощью поисковых машин.

Директории обеспечивают контекстный поиск для структурированного просмотра, тогда как поисковые машины, как следует из их названия, контекста не обеспечивают, однако позволяют находить конкретные слова или фразы. Директории можно уподобить оглавлению книги, а поисковые машины – предметному указателю.

Часто поисковые системы объединяют в себе как поисковую машину, так и директории. Это хорошо видно на примере первой страницы Яндекса, где под поисковой строкой размещается список директорий, которые позволяют пользователю уточнять запрос по мере продвижения в глубь каждой из них.

Ввиду того, что принцип организации директорий понятен каждому, кто пользовался библиотечным каталогом – а среди читателей таких, смеем полагать, подавляющее большинство, – мы не будем подробно останавливаться на технике работы с директориями и уделим больше внимания работе с поисковыми машинами. В завершении же разговора о каталогах приведем пример «цепочки», по которой осуществляется поиск каталоге Яндекса: Бизнес > Реклама > Реклама в Интернете.

Все поисковые машины работают по одному и тому же алгоритму и основаны на одних тех же принципах. Различия между ними возникают лишь на уровне технической реализации этих принципов в работе.

Чтобы понять принцип работы поисковой машины, попробуем разделить вопрос на две части: на чем основан поиск и как он реализован.

На чем основан поиск Все поисковые машины базируются на трех основных операторах, лежащих в основе Булевой алгебры (ее также называют Булевой логикой или Boolean). Это логические операторы «И», «ИЛИ» и «НЕ». Работают они следующим образом.

1. Логическое «И». Если между двумя словами в запросе стоит оператор «И», то в результате поиска будут найдены лишь те документы, в которых содержатся оба слова.

Так, например, по запросу собака И кошка будет найден документ, содержащий предложение «собака гналась за кошкой», документов же, состоящих из текста «кошка отдыхала» или «корм для собак», мы не увидим.

2. Логическое «ИЛИ». Если между словами стоит оператор «ИЛИ», то результатом поиска станут документы, в которых содержится хотя бы одно из этих слов. Если мы не сделаем специальных ограничительных оговорок, то материалы, в которых оба эти слова присутствуют, также будут найдены.

По запросу собака ИЛИ кошка мы получим документы, исключенные в прошлом запросе и содержащие текст «кошка отдыхала» или «корм для собак», а также материал с предложением «собака гналась за кошкой».

3. Логическое «НЕ». Если два предыдущих оператора описывали те слова, которые вы хотите включить в запрос, то оператор «НЕ» слова из запроса исключает.

Пользователи, впервые сталкивающиеся с операторами запросов, нередко высказывают удивление: мол, не проще ли и вовсе не включать ненужное слово в запрос? Зачем вводить дополнительный оператор? Увы, нет. Не проще. На самом деле, чтобы понять важность логического оператора «НЕ», имеет смысл вспомнить, что наш запрос не создает в Интернете ничего

нового. Мы лишь выуживаем то, что нам нужно, из имеющегося огромного, но все же конечного массива. При этом необходимо отсечь информационный мусор. Его-то мы и отсекаем с помощью оператора «НЕ». К сожалению, не нам решать, увидим ли мы этот мусор в выдаче. Так, например, по запросу сведений о коньке крыши неизменно появляется информационный мусор в виде документов о Коньке-Горбунке, фигурном катании, хоккее, лошадях и т. п. Без логического «НЕ» тут никак не обойтись.

Давайте рассмотрим примеры работы логического оператора «НЕ». По запросу собака НЕ кошка будет найден документ, содержащий текст «корм для собак», а вот документы со словами «кошка отдыхала» или «собака гналась за кошкой», и даже «корма для собак и кошек» из выдачи будут исключены.

Часто встречается чуть более сложный вариант написания запроса, который содержит все или почти все вышеперечисленные операторы. В этом случае лучше пользоваться таким элементом, как круглые скоби. Скобки позволяют отделять однотипные слова запроса от остальных. Кроме того, самому составителю при этом визуально гораздо удобнее различать отдельные фрагменты запроса. Мы не будем чересчур теоретизировать о скобках, а просто продемонстрируем работу указанного элемента на конкретных примерах. На наш взгляд, так будет понятнее, как и для чего используются скобки. Так, запрос пушистые И (собаки ИЛИ кошки) позволит получить документы, относящиеся как к пушистым собакам, так и к пушистым кошкам – по отдельности или вместе. Скобки при этом «раскрываются» по обычным арифметическим правилам вынесения за скобку общего множителя (для тех, кто не любит математику, поспешим сказать, что болеше углубляться в нее мы не будем). А вот запрос пушистые И (собаки ИЛИ кошки) НЕ (собаки И кошки) выдаст документы, в которых написано про пушистых собак или пушистых кошек, но не будет содержать текстов, где одновременно будут упомянуты и кошки, и собаки.

Еще раз повторимся, все поисковые машины сегодня работают на основе анализа этих трех операторов, хотя нюансы их написания в разных поисковых машинах могут отличаться.

Как поиск реализован. Каждая полноценная поисковая машина располагает собственным штатом роботов, или пауков. Их еще называют краулерами (crawlers) и спайдерами (spiders,). Это программы, которые перескакивают со страницы на страницу и сканируют находящиеся на них тексты, не вникая при этом в их содержание. После чего сбрасывают документы на серверы своих хозяев и идут к следующим страницам. Как паук определяет, куда ему пойти? Он находит так называемую гиперссылку (ту самую, при наведении на которую курсор приобретает вид раскрытой ладони, и при клике по которой происходит переход на другую страницу) и идет по ней. Вот почему, если на страницу не ведет ни одна ссылка, паук на нее тоже не придет. Исключение составляет ситуация, когда владелец страницы вручную сообщит о ней поисковой машине, заполнив специальную форму на сайте поисковой машины.

На сервере поисковой машины текст разбивается на отдельные слова, каждому из которых присваиваются координаты, после чего они заносятся в таблицу сервера вместе со ссылкой на тот адрес в Интернете, по которому текст размещался в момент посещения его пауком. Сам по себе поисковик представляет собой большую локальную сеть, состоящую из мощных компьютеров с огромным объемом дисковой памяти. Эти машины разделены на подгруппы (так называемые кластеры), между которыми распределяется информация, собранная пауками.

Когда поисковая система получает запрос, она ищет ответ именно в своей таблице, а не в Интернете. При этом важно понять, как паук решает, с какой частотой ему следует посещать ту или иную страницу. Выглядит этот алгоритм приблизительно следующим образом. Поработав со страницей, паук возвращается на нее, ну, например, через две недели. И если видит, что никаких изменений не произошло, он планирует следующее посещение через более длительный период – скажем, через месяц. А если и тогда не обнаружит ничего нового, то наведается сюда еще позже, месяца через полтора-два. Вот почему нередко бывает так, что поисковая

машина по запросу результат выдает, а попытка перейти на страницу по полученной ссылке безрезультатна – вероятнее всего, никакой страницы уже просто не существует на прежнем месте, но паук на нее давно не заходил, и, соответственно, поисковая система о ее удалении не знает. Весь комплекс процессов, описанных выше, называется индексацией.

История развития поисковых машин

История эволюции поисковых машин наиболее полно, на наш взгляд, представлена в книге признанных экспертов в области невидимого интернета Криса Шермана и Гарри Прайса «Невидимый Интернет».

До середины 1960-х годов компьютеров было немного. Изолированные друг от друга, они не могли обмениваться информацией.

В 1962 г. профессор Ликлайдер (Licklider) из ведущего технического вуза США – Массачусетского Технологического института – сформулировал концепцию глобальной компьютерной сети «Galactic Network». Идея начала воплощаться в жизнь сотрудником американского министерства обороны Ларри Робертсом (Larry Robberts), который через четыре года после публикации статьи профессора предложил объединить отдельные компьютеры министерства в сеть, описанную Ликлайдером. Таковы предпосылки возникновения сети «ARPANET», которая затем превратилась в то, что сегодня величают Интернетом.

Первый узел «ARPANET» появился в 1969 г., и следующие несколько лет к нему подключались университеты и различные контрагенты, работавшие по заказам военного ведомства США. В 1973 г. американское министерство обороны инициировало новую программу, предполагавшую обеспечивать надежную связь компьютеров между собой с помощью очень большого числа различных соединений. Целью такого решения было повышение устойчивости системы к попыткам массированно нарушить электронные средства коммуникации. Поскольку все это происходило во времена «холодной войны», речь шла об устойчивости к устрашающим последствиям, которыми грозило стратегическое ядерное противостояние.

Поскольку «ARPANET» представлял собой одну-единственную сеть, что на системном уровне понижало его способность сопротивляться разрушениям, возникла идея создания «сети из сетей», которая теоретически могла бы быть бесконечно большой.

Этот проект и назвали «Internetting», а саму сеть «Internet». По мере того, как количество присоединенных к Интернету машин увеличивалось, объективно назрел вопрос о необходимости инструментов, позволяющих легко находить текст и другие файлы на удаленном компьютере, в идеале – на любом, где бы он ни располагался в Сети.

Доступ к файлам на самых ранних этапах развития Интернета осуществлялся в два этапа, каждый из которых выполнялся вручную: специальные команды вводились с клавиатуры. Кстати, тогда компьютеры могли управляться лишь специалистами, способными вводить команды в соответствующую строку. Графического интерфейса, позволяющего комфортно работать с машиной неподготовленному человеку, еще не изобрели. Так вот первым делом с помощью программы Telnet устанавливалось прямое соединение с компьютером, на котором находится нужный файл. На данном этапе лишь налаживалась связь, ничего и никуда в этот момент еще не передавалось. И только затем с помощью специальной программы – FTP – можно было этот конкретный файл взять. Очевидно, что на поиски нужного документа уходила масса времени: требовалось знать точный адрес компьютера, на котором он находится. Между тем файлов становилось все больше, интерес к ним постоянно рос, и для того, чтобы найти адрес одного из них, обычно приходилось обращаться в дискуссионные группы с просьбой о помощи и в надежде на то, что кто-нибудь из собеседников подскажет заветный адрес, по которому хранится нужная информация.

В результате, стали появляться специальные FTP-серверы, которые представляли собой хранилище файлов, организованных в директории, по принципу хранения информации на

персональном компьютере. Такие серверы существуют и по сей день. Первый работоспособный, общедоступный инструмент поиска файлов, хранящихся на FTP-серверах, назывался «Арчи» (Archie) и был создан в 1990 г. группой системных администраторов и студентов старших курсов Университета Мак Джил (McGill) в Монреале. «Арчи» был прототипом сегодняшних поисковых машин, но значительно болеепримитивным и ограниченным в своих возможностях. Он бродил по Интернету, разыскивал файлы на разных FTP-серверах и загружал список директорий каждого найденного сервера на собственный, формируя общий каталог.

Этот каталог затем обрабатывался и хранился в центральной базе данных, внутри которой можно было организовать поиск. Поиск на собственном компьютере к тому моменту существовал уже издавна и, несмотря на то, что тоже требовал ввода команд, трудностей вработе не создавал. Однако без специальной подготовки использовать компьютер полноценно человек не мог. База данных находилась в университете Мак Джилл и обновлялась ежемесячно. В 1991 г. команда Марка Мак Кахилла (Mark McCahill) из Университета Миннесоты создала программу «Голден Гофер» (Golden Gopher – в переводе с английского «золотоискатель» или «старатель»), которая совмещала в себе оба протокола – Telnet и FTP. Все, что нужно было сделать пользователю для получения доступа к нужной информации, – щелкнуть по гиперссылке, приведенной в меню.

Таким образом, впервые в истории вводить какие-либо команды уже не требовалось, так что отныне по ресурсам Интернета люди могли «бродить» и без специальной подготовки.

Программа показывала пользователю последовательно возникающие пошаговые меню, что позволяло ему без проблем идти в глубь базы директорий, все более приближаясь к специфическим документам, которые и составляли цель поиска. Этот алгоритм, по сути, сохранен и сегодня в Каталогах, расположенных в Интернете. Стало возможно получать как текстовые документы, так и графические, и музыкальные, без привязки к какому-то определенному формату. А самое главное, стало в принципе возможно легко найти и получить в Интернете нужную информацию.

Однако проблемы все же оставались. Одна из них, и довольно серьезная, была связана с тем, что компьютеры были построены на разных платформах, которые порой не понимали друг друга. Тут можно провести аналогию с людьми, которые говорят на совершенно разных языках и потому не могут построить более или менее осмысленную беседу. В те времена между собой конкурировали не операционные системы, как сейчас, а производители компьютерного «железа». Сегодня в меньшей степени важно, кто произвел компьютер. Гораздо существеннее, что на нем установлено: Windows, Linux, Mac OS. А тогда именно производители «железа» определяли лицо Интернета.

Объективно назревала идея, согласно которой компьютеры разных платформ должны иметь возможность работать в одном протоколе, позволяющем просматривать страницы вне зависимости от того, на какой конкретно машине эти страницы созданы. Требовалось придумать такой универсальный протокол и сделать его удобным для пользователей.

Чтобы пользователь получил в руки независимый от платформы и при этом простой инструмент, Бернерс-Ли создал HTML. Все Web-документы, отформатированные с помощью тегов HTML, видны совершенно одинаково во всем мире, вне зависимости от типа компьютера, на котором человек открыл страницу сайта. Поэтому и сегодня при переводе файла в формат HTML, например, на машине, работающей под управлением операционной системы MacOS, можно быть уверенным в том, что этот файл будет выглядеть точно так же и на компьютере, работающем под управлением Windows. Затем Бернерс-Ли придумал Universal Resource Identifier – метод стандартизации адресов, при котором компьютерам в Интернете присваиваются уникальные адреса (сегодня мы их называем URL, это то, что в привычном для пользователя виде обычно начинается с «www»). Наконец, изобретатель собрал вместе все эти элементы, создав систему в форме Web-серверов, которые хранят HTML-документы и

предоставляют их другим компьютерам, создавая HTML-запросы о документах по определенным URL. Но Бернерс-Ли хотел видеть Интернет как информационное пространство, в котором можно получить свободный доступ к данным любых типов. На ранних этапах развития глобальной Сети преобладали простые текстовые документы HTML. К тому времени существовали системы поиска информации на локальных машинах, поэтому появилось несколько серверов, которые пытались проиндексировать какую-то часть страниц Web и прежде, чем отправляться за чем-то в Интернет, предлагали поискать необходимые сведения на этих серверах. При этом основная проблема заключалась в том, чтобы отыскать страницы, которые в принципе можно бы было индексировать. Поскольку Интернет лишен централизованной структуры и общего оглавления, единственный способ, позволявший добиться этого, состоял в поиске ссылки на страницу и переходе по этой ссылке, с последующим добавлением найденного ресурса к индексу.

Однако вскоре возникла еще одна проблема. Наиболее популярные страницы посещались пауками чаще остальных, так как на них указывало максимальное количество ссылок.

Пауки, количество и возможности которых были ограничены, «зависали» на таких страницах и впустую расходовали ресурсы, оставляя непосещенным множество других адресов, пока еще менее популярных. Для решения этой проблемы требовалось создать программу, которая позволила бы игнорировать уже проиндексированные страницы и сосредоточиться на поиске новых. Иначе это грозило проблемой с ресурсами.

В 1993 г. студент-физик Массачусетского технологического института Мэтью Грей (Mathew Gray) создал первый широко известный Web-робот, названный «World Wide WebWanderer» или просто «Вандерер», что в переводе с английского означает «скиталец» или «странник». Дело в том, что Грей заинтересовался статистикой. Результатом такого увлечения стало появление «странника»: изобретение было призвано помочь студенту проанализировать размеры Интернета и скорость его роста. «Вандерер» просто приходил на страницу и определял сам факт ее существования, не занося в базу содержимого найденного адреса. Несмотря на то, что создатель робота не преследовал никаких других целей, его детище, фактически дебютировавшее в «забеге» прогрессивных интернет-находок, легло в основу более сложных программ, которые к умению «скитальца» перемещаться по Сети добавили способность сохранять содержимое страниц в базе данных после их посещения.

Случилось так, что 1994 г. стал переломным в истории создания поисковых машин. Студент выпускного курса Вашингтонского университета Брайан Пинкертон (Brian Pinkerton) устал от бесконечной череды электронных писем, которые посылали ему друзья, с информацией о хороших сайтах, найденных ими в Интернете. Безусловно, сайты ему были нужны, однако шквал посланий с их адресами раздражал, а посещение всех страниц отнимало уйму времени. Однако Пинкертон нашел решение проблемы – он создал робота, которого назвал WebCrawler (что-то вроде «вездеход для Интернета»). «ВебКраулер», как и «Вандерер», ползал со страницы на страницу, запоминая при этом весь текст Web-документа и сохраняя его в базе данных, которая была доступна поисковым словам. Изобретатель представил свое детище публике в апреле 1994 г., причем сделал это виртуально – через Web-интерфейс. База данных в тот момент содержала информацию с 6000 самых разных серверов. Уже через неделю она начала расширяться, причем ежедневный прирост составлял более 100 новых серверов. Так родилась первая поисковая машина.

Тогда же был введен в обиход интернетчиков термин «краулер» или «паук», который применяется, как мы уже говорили, и по сей день.

Ну а далее ситуация развивалась еще более стремительно. Крис Шерман и Гари Прайс приводят такую хронологию возникновения и развития современных поисковых машин.

1994 г. – WebCrawler, Lycos, Yahoo!

1995 г. – Infoseek, SavvySearch, AltaVista, MetCrawler, Excite. Появление метапоисковых машин.

1996 г. – HotBot, LookSmart.

1997 г. – NorthernLight.

1998 г. – Google, InvisibleWeb.com.

1999 г. – FAST.

2000 г. и далее – Сотни новых поисковых машин. Русскоязычные поисковые машины появлялись в такой последовательности:

1996 г. – Rambler (www.rambler.ru);

1997 г. – Yandex (www.yandex.ru);

2004 г. – русскоязычная версия Google (www.google.ru) и русскоязычная версия Yahoo! (http://ru.yahoo.com).

Из чего состоит сайт Прежде, чем перейти к описанию языка запросов поисковых машин, рассмотрим, из каких элементов, с которыми предстоит работать пауку, состоит обычно сайт.

Надо сказать, что язык HTML достаточно прост и логичен. Он представляет собой способ разбивки текста с помощью специальных элементов – тегов, которые определяют структуру и внешний вид текста при просмотре его в браузере. О тегах следует знать, что они всегда парные и что они бывают открывающими (обозначают начало определенного форматирования) и закрывающими (обозначают его окончание). Закрывающий тег – такой же по написанию, как открывающий, но перед ним стоит косая черта. Приведем пример очень простого сайта (рисунок 1).

Наверху страницы, изображенной на рисунке, то есть не в тексте сайта, а на верхнем поле рамки страницы, рядом с круглым значком браузера, расположена надпись: «Показываем устройство сайта». Она находится в так называемом заголовке страницы (который заключен между открывающим тегом <TITLE> и закрывающим тегом </TITLE>). Обращаем ваше внимание на то, что это заголовок именно всей страницы, а не текста. Посередине представленного рисунка жирным курсивом выведено: «Это простой сайт». Данная надпись – и есть заголовок текста. Шрифт фразы «Это простой сайт» по размеру.

Наверху страницы, изображенной на рисунке, то есть не в тексте сайта, а на верхнем поле рамки страницы, рядом с круглым значком браузера, расположена надпись: «Показываем устройство сайта».

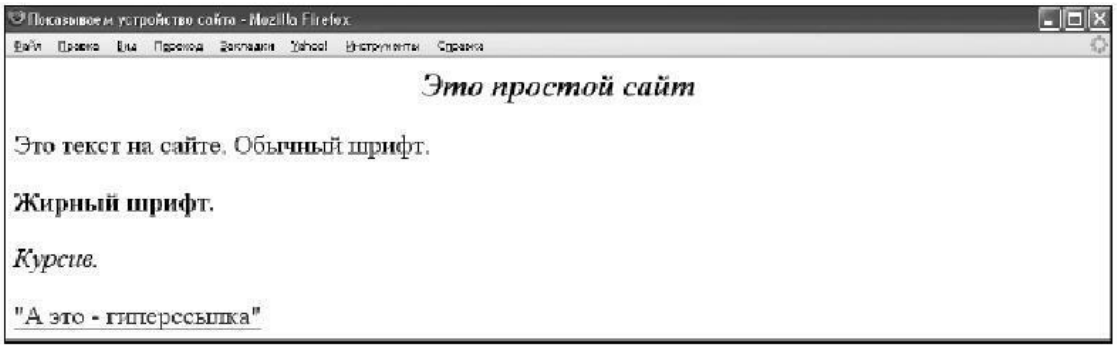

Рисунок 1. ***Пример сайта, как его видно в браузере Мозилла Файрфокс***

Она находится в так называемом заголовке страницы (который заключен между открывающим тегом <TITLE> и закрывающим тегом </TITLE>). Обращаем ваше внимание на то,

что это заголовок именно всей страницы, а не текста. Посередине представленного рисунка жирным курсивом выведено: «Это простой сайт». Данная надпись – и есть заголовок текста. Шрифт фразы «Это простой сайт» по размеру превосходит шрифт текста на сайте, он специально выделен как заголовок текста. При разметке с помощью HTML этот текст расположен ниже тега <TITLE>, но при этом вместе с тегом <TITLE> находится внутри тега <Head>. То есть содержимое, заключенное в <TITLE>, – это часть того, что находится в <Head>.

Такое расположение дает дополнительную возможность пауку лучше определять ключевые слова на сайте. Ведь если слова вынесены в заголовок текста или, тем более, всей страницы, вероятность того, что страница и текст посвящены теме, формулируемой этими словами, повышается. Ниже фразы «Это простой сайт» приведены четыре варианта написания основного текста сайта: обычный, жирный (пишется под тегом <B>), курсив (пишется под тегом <i>), текстовая гиперссылка (пишется под тегом <A HREF=http://www.url. ru>«Текст гиперссылки»</A>).

Основной текст сайта, вне зависимости от того, каким вариантом шрифта он написан, располагается внутри тега <BODY>. Именно содержимое тега <BODY> представляет собой основной объект для паука и рассматривается им как текст страницы (собственно, это действительно текст страницы). Чтобы увидеть внутреннюю разметку сайта, надо в браузере Мозилла Файрфокс навести курсор на любой незанятый текстом участок поля и нажать правую кнопку мыши. В

всплывающем меню следует выбрать пункт «Просмотр исходного кода страницы». Применительно к сайту, который мы рассматривали на рис. 1, этот исходный код будет выглядеть следующим образом:

```
<HTML>
<HEAD>
<TITLE>
Показываем устройство сайта:
</TITLE>
<CENTER>
<B><I>
<SPAN STYLE=«font-size: large»>Это простой сайт</SPAN>
</CENTER>
</B></I>
</HEAD>
<BODY>
<P>
Это текст на сайте. Обычный шрифт.
</P> <P> <B>
Жирный шрифт.
</B> </P> <P> <I>
Курсив. </I> </P>
<A HREF=http://www.url.ru>«А это – гиперссылка»
</BODY>
</HTML>
```

Здесь можно увидеть все элементы, описанные нами выше. Кроме того, в исходном коде видны теги <P>, которые обеспечивают расположение текста в новой строке и с промежутком по отношению к тексту, расположенному в предыдущей строке. Разметка HTML по умолчанию не предполагает переноса текста и его форматирования.

Поэтому текст, не содержащий никаких тегов, воспроизводится подряд, но с соблюдением пробелов между словами. Для того чтобы текст оказался написан не просто в новой строке, а с промежутком относительно находящейся выше строки, используется, как мы уже показали, тег <P>, а для того, чтобы текст был написан в новой строке, но без промежутка между выше– и нижерасположенной строками, применяется тег
.

Начало сайта, созданного с помощью разметки HTML, отмечено тегом <HTML>, а его окончание – тегом </HTML>.

Лекция 2
Описание языков запросов поисковой машины яндекс

Лучшая, на наш взгляд, работа по изучению операторов поисковой машины Яндекс выполнена специалистом из Санкт-Петербурга Денисом Фурсовым. На его ресурсе постоянно проводятся дополнительные исследования, отслеживаются и оцениваются изменения в работе операторов указанной поисковой машины. Ниже речь пойдет о том, как с помощью специальных фильтров, основанных на Булевой алгебре, создавать запросы, максимально соответствующие потребностям специалиста, который ищет информацию в Интернете. При изучении этого вопроса, не следует забывать, что компьютер очень исполнителен, но лишен способности думать, поэтому следует составлять запрос, исходя из того, что он будет обработан компьютером буквально, а не с учетом того, что же на самом деле имел в виду пользователь, создавая свое обращение.

Итак, перейдем непосредственно к операторам запросов Яндекса.

1. Логическое «И».

Яндекс поддерживает три разных оператора, относящихся к логическому «И», что делает его самым гибким из всех поисковиков, работающих с русским языком. Столь развита практически уникальная система операторов поисковых запросов дает возможность предельно точно настроить запрос и сформировать такой фильтр для данных в Интернете который максимально качественно выбирает нужную информацию и отсекает ненужное.

1.1. Пробел. Слова, разделенные пробелом, должны располагаться недалеко друг от друга. Специалисты поясняют, что термин «недалеко» отнюдь не фиксированная величина именяется в зависимости от того, с какими словами указанный оператор в каждом конкретном случае используется. Если они часто употребляются, то «недалеко» – значит на расстоянии нескольких слов друг от друга. Если же они редко встречаются в обиходе, то даже их нахождение в разных концах документа будет восприниматься как «недалеко».

При этом, несмотря на то, что логическое «И» в общем виде Булевой алгебры подразумевает присутствие всех упомянутых слов, Яндекс, тем не менее, действительно выдает сначала те документы, в которых есть все ключевые слова, представленные в запросе. После чего начинает выдавать документы, в которых на одно ключевое слов меньше, чем в запросе, затем – на два слова меньше и так далее.

Запрос: [маркетинг менеджмент]

Результат поиска: страниц – 2 442 393, сайтов – не менее 1456

В выдаче: Маркетинг, Финансы, Реклама, Менеджмент

1.2. Амперсанд (&). Слова, разделенные амперсандом, находятся в одном предложении. Важно: амперсанд должен быть отделен пробелами с двух сторон от любых других слов.

Запрос: [маркетинг & менеджмент]

Результат поиска: страниц – 1 190 379, сайтов – не менее 1093

В выдаче: … Филип Котлер в краткой форме представляет все наиболее значительные и интересные положения самой известной своей работы «Маркетинг менеджмент»…

1.3. Двойной амперсанд (&&). Слова, разделенные двойным амперсандом, находятся в любом месте одного и того же документа.

Важно: между амперсандами не должно быть пробелов, но сам оператор должен быть отделен пробелами с двух сторон от любых других слов.

Запрос: [маркетинг && менеджмент]

Результат поиска: страниц – 3 641 056, сайтов – не менее 1 295

В выдаче, к примеру, будут присутствовать учебные планы вузов, в которых слова «маркетинг» и «менеджмент» находятся в разных частях текста, в том числе – на разных страницах опубликованного в Интернете многостраничного плана занятий.

Чтобы увидеть это наглядно, читатели могут нажать в результатах выдачи гиперссылку «Найденные слова», которая приводится во всех итогах поиска. И тогда слова, которые есть в запросе, будут подсвечены и не придется тратить время на их «отлавливание» в тексте.

2. Логическое «НЕ».

Логическое «НЕ» представлено двумя операторами. Прежде чем рассказать о них, отвечу на вопрос, который часто возникает у людей, впервые приступивших к изучению операторов поиска: «Зачем нужно логическое „НЕ“? Его ведь можно и вовсе не вводить, и тогда оно нам не понадобится!». Отвечаем: если мы сами решаем, что нам вводить, а что нет, то это утверждение справедливо. Но проблема в том, что часто в выдаче принудительно оказывается «мусор» и другого способа избавиться от него, кроме как убрать эти слова при помощи логического «НЕ», у нас нет. Так, например, если вас интересует конек крыши, то по слову «конек» в выдаче окажется информация и о роликовых, и о фигурных коньках, и даже о Коньке-Горбунке. Для таких-то случаев логическое «НЕ» и придумано.

Итак, вернемся к нашим операторам.

2.1. Тильда (~). Знак тильды – это верхняя левая клавиша на буквенно-цифровой клавиатуре. Символ вводится на английском регистре с нажатой клавишей SHIFT. Как и амперсанд, тильда должна быть отделена пробелами с обеих сторон. Часто допускают ошибку, «приклеивая» тильду к следующему за ней слову. Иногда отсутствие пробела между тильдой и последующим словом не влияет на результат, но бывает и наоборот, поэтому лучше внимательно проследить за пробелами вокруг этого знака.

Тильда означает, по аналогии с диаметрально противоположным символом – амперсандом, что слова не должно быть в предложении.

Запрос: [маркетинг ~ менеджмент]

Результат поиска: страниц – 12 604 153, сайтов – не менее 4442

В выдаче: … комплексный подход к услуге интернет-маркетинга, охватывающий все возможности для продвижения интернет-представительств компаний в сети Интернет.

2.2. Двойная тильда (~~). По аналогии с двойным амперсандом, двойная тильда пишется слитно внутри самого этого оператора, но отделяется от остальных слов пробелами с обеих сторон. Она означает, что слова, которое за ней расположено, не должно быть в документе совсем.

Запрос: [маркетинг ~~ менеджмент]

Результат поиска: страниц – 9 675 995, сайтов – не менее 3 976

В выдаче: Форум по маркетингу и рекламе – Маркетинг и Реклама, маркетинговые коммуникации, виды рекламы: реклама в СМИ, наружная реклама, BTL: POS-материалы, У вас есть вопрос по маркетингу и рекламе?

Обратите внимание: в результатах выдачи слова «маркетинг» и «маркетингу» выделены как релевантные, «маркетинговые» же – нет. Это происходит потому, что термин «маркетинг» – существительное, а «маркетингу» – его словоформа, тогда как «маркетинговые» – совсем другая часть речи, а отнюдь не производное от слова «маркетинг». Подобное явление надо учитывать, если вы рассчитываете на способность Яндекса самостоятельно перебирать словоформы. Игнорирование этого факта нередко приводит к искажению результатов выдачи и также является частой ошибкой начинающих специалистов по поиску в Интернете. На самом деле, в Яндексе есть еще один оператор логического «НЕ», который обозначается знаком «минус». По мнению Дениса Фурсова, с которым автор полностью согласен, _ «минус» – это не всегда корректно работающая двойная тильда, поэтому пользоваться им смысла нет. Мы

не знаем наверняка, но предполагаем, что знак «минус» в качестве логического «НЕ» – это способ унифицировать Яндекс с другими поисковыми машинами, поскольку в большинстве своем они обозначают логическое «НЕ» именно этим знаком. Мы не пользуемся оператором «минус» при поиске в Яндексе.

Конец ознакомительного фрагмента.