

GREP В ПОМОЩЬ!

В 2013 году я попробовал изложить свои знания grep в виде небольшой книги. 20 страниц формата А4. Просто хотелось для себя систематизировать знания, а вариант сделать это в виде книги казался самым лучшим. К тому же была надежда, что будут отклики, а обратная связь в таком деле хороший помощник. Отклики были, и самым неожиданным было суждение, что вот тот подробный пример с оформлением телефонных номеров, там было всё понятно, а остальные далеко не все понятны. Да, в той книге объяснение, как оформить запись телефонного номера, растянулось на полторы страницы. И я тогда переживал, нужно ли так подробно. Оказалось, что именно подробно и стало самым полезным материалом.

С тех пор прошло много времени, и теперь мне хочется не просто рассказать о grep, а сделать платный обучающий курс.

Да по грепу море информации!

Я понимаю, как это, возможно, для кого-то дико звучит «платный курс», когда в сети столько материала, что жизни не хватит всё прочесть. Да, я с этим согласен. Материала в сети запредельно много. У меня на диске больше сотни мегабайт информации на эту тему. У кого-то, возможно, и больше. А что толку? Все эти

информационные источники не ставят целью научить нас пользоваться грепом. Я думаю, что большинство этих источников было сделано с той же целью, для чего я писал первую книгу — объяснить себе.

Но у нас нет нескольких жизней на изучение премудростей grep. У нас, честно говоря, и времени-то на учёбу нет. Ведь прими решение всё бросить и заняться освоением grep, первый вопрос «А по какому из многих источников учиться? Что относится именно к индизайну, а не просто к grep, коих есть несколько диалектов? Как не оглохнуть в этом информационном шуме?»

Вот поэтому я уверен, что лучше заплатить и за одну-две недели сделать эту доселе недостижимую премудрость своим знанием, чем продолжать годами чертыхаться, делая поиск/замену, вздыхая, что был бы опыт работы с grep, всё было сделано намного быстрее.

Трудности роста

Я не утверждаю, что самому, без чьей-то помощи, grep освоить нельзя. Можно. Но сложность самостоятельного освоения grep в том, что процентов семьдесят выкроенного на это дело времени вы будете не изучать, а «разбираться». Если вы, разговаривая и думая на русском языке, работаете в английской версии индизайна, то вы будете ломать голову, например, к чему там

слово *negative*. В публикациях на английском языке о грер есть понятие *Negated Character Class*, и в русских переводах всякие вариации на тему классов негативных символов. Но переводить надо не слова, а смысл. И смысл тут такой «класс неучитываемых знаков». Т.е. *negated* (инвертированный) тут для указания, что символы этого класса грер при обработке пропускает.

Автоматический перевод документации ведёт только к потере времени. Вот образчик перевода на русский язык (ссылка рабочая):

www.macprovideo.com/article/indesign/indesign-understanding-grep-styles/ru

Жестокое быть тупым: ГРЭП является инструментом для полудурков. И GREP Стили являются инструментом полудурков, кто действительно хотел InDesign. Тем не менее, его удивительно полезным инструментом, и это может оказаться весьма кстати, так как она позволяет искать сложной (или простой) последовательность символов, и применить стиль символов к только те символы автоматически. Пусть, что раковина в немного, то читайте дальше.

Так будет GREP?

Глобальный регулярных выражений Parser позволяет найти почти все, что угодно, используя либо сложную серию кодов, или (к счастью) серию выпадающих меню. Его не только для конкретного слова, хотя; Вы можете искать масок, для некоторых пункт повторяется один или несколько раз, за фразу вы хотите повторно стиль.

Это большая проблема — адекватно перевести термин, и я понимаю людей, пытавшихся сленг программистов *negative look-behind* осмысленно перевести на русский язык:

www.skillz.ru/dev/php/article-Regulyarnye_vyrazheniya_dlya_chaynikov.html

Рассмотрим сначала заглядывание вперед с отрицанием

<https://habr.com/post/159483/>

Опережающие и ретроспективные проверки в регулярных выражениях

Работая в русской версии, мы столкнёмся с ошибками локализации. Там их пять, они показаны восьмой странице той первой книги.

Но самое главное — это то, что какой сетевой ресурс ни возьми, нет никакой без гарантии, что найденное на учёбу время будет потрачено с пользой, и грер будет твоим инструментом. Достоинство только одно — бесплатно.

Но с бесплатным источником несистематизированных сведений о грер вы тратите своё невосполнимое время. И возможно, останетесь примерно на том же уровне некоторого представления о его возможностях. Да, что-то можете делать, но при этом будет горечь от потерянного времени, так как понимаете, что далеко не всё из инструментария грер умеете использовать.

Но это можно исправить.

Шаг за шагом с проверкой знаний

Вы помните, как иногда бывало в школе, или другой какой-нибудь учёбе — встречается сложный момент в объяснении, ты его не понимаешь, и всё, что дальше, абсолютно непонятно. Моя проблема в освоении грер была такая же. Я долго продирался через вещи, на которые потом смотрел с удивлением, чего я тут тогда понять не мог? Всё же так просто!

Я думаю, вы, если уже пытались въехать в грер, попадали на эти грабли. Что я предлагаю, чтобы больше не по-

лизовать лбом ручки этих грабель? Я предлагаю серию текстовых уроков по каждому аспекту использования GREP. В тексте мы обойдём все неувязки неверной локализации и трудности перевода, вы, конечно, будете знать, где хранятся все команды, но практиковаться будем, вводя их с клавиатуры.

Вот в соседней колонке интересная картинка, неплохо объясняющая, как человек усваивает информацию, проценты справа — это объём информации, оставшийся в памяти через две недели после события. Как видите, **2** это как если бы мы черпали бесплатную информацию из интернета. Но я хочу, чтобы вы были на уровнях **6** и **7**. Будете вы там или нет, зависит от вашего желания и настроения, можно подвести коня к водопою, но нельзя заставить его пить.

Вот такая идея, как с уровнями **1, 2** после каждого урока попадать на уровни **6** и **7**.

У каждого будет матрица усвоения уроков, примерно такая:

Уроки	П	О	С
Название			
Название			

П — понял урок, О — можешь повторить/объяснить, С — готов предложить свой пример.

Эту матрицу каждый заполняет сам. Очевидно, что в колонке **П** всегда должен быть плюс. Это ведь маленькие уроки, и всё должно быть понятно, поскольку объём



нового очень небольшой. Названия уроков достаточно информативные, и вы, глядя на неё, легко сможете найти то, что надо освежить в памяти. А варианты **О** и **С** — это ваш уровень активного знания конкретного урока. Да, это будет именно живая учёба в формате «понятие - объяснение - пример», а не бесполезная трата времени, чтобы «разобраться» с очередным аспектом grep-обработки.

В курсе обучения будет теоретических 17 занятий, подкреплённых задачкой, в котором сейчас 112 задач. Число большое, но это нельзя назвать недостатком. Тут такая метафора в голову приходит — ступеньки лестницы. Вы их замечаете, когда по ним вверх-вниз носитесь? Нет, потому что хоть их и много, но они невысокие. А будь они высотой с письменный стол, много бы мы с этажа на этаж набегали бы? И вы за

одну-две недели пройдёте все эти ступени. А упомянутая выше матрица накопления знаний будет показывать, на каких ступеньках вы скользите, спотыкаетесь. И тут я помогу вам полностью разобраться в этих проблемных вопросах.

И ещё важный, на мой взгляд, момент, который тут реализован: как только в процессе обучения будет набран опыт, достаточный для решения какой-то задачи обработки текста, это сразу будет показано на конкретном примере. Мне не нравится идея сперва всё растолковать, а потом решать прикладные задачи. Невозможно сперва выучить все правила и слова иностранного языка, а потом начать учиться на нём говорить. И нельзя научиться плавать, стоя на берегу. Точно так же и в ггер — обязательна практика на конкретных примерах. И она у вас будет, более сотни задач с решениями, если вы возьмёте на себя труд понять их, то скоро будете мыслить категориями эффективных замен, вопрос будет только, какой способ выбрать.

Важно только идти от урока к уроку, а не знакомиться с ними выборочно, ориентируясь на своё ощущение, что какой-то вопрос вы якобы знаете, и на него не надо тратить время. Смысл этого обучения в практике, и к каждой новой задаче надо подходить с опытом, полученным в предыдущих занятиях.

Учить или не учить?

К сожалению, сейчас характерной чертой стало чувство ложной компетентности — когда доступ к информации приравнивается к владению знаниями. «Можно всё найти в интернете», это главный аргумент, чтобы оставить всё как есть. Да, теорию в каком-то объёме можно черпнуть из сети. Но только теорию, и хоть без теории практика невозможна, но и без практики теория бесполезна.

И это должна быть основательная практика. Мало кому удаётся поглотить яблоко целиком. Его приходится раскусывать и разжёвывать. Усвоение требует расщепления.

Если вы уверены, что всё знаете в этой теме, то не тратьте время на сообщение мне об этом.

Эта информация для тех, кто не хочет, чтобы всё оставалось как есть сейчас. Я готов помочь вам в этом.

А знание ггер — это великолепная подмога в вёрстке, такой кайф понимать варианты просмотров, различия в однострочном и многострочном режимах, смотреть на задачу и сразу видеть запрос, решающий её. Да, учёба не дешёвая, но она окупится удовольствием от работы.

На следующих страницах описание, что есть в каждом из 17 занятий, и какие дополнительные материалы вы ещё получите.

Михаил Иванюшин
dotextok@gmail.com

Краткое описание каждого занятия

Занятие 1

GREP — поиск по образцу

Самый простой способ начать изучать GREP — это попробовать повторить примеры его использования. Но сначала давайте поймём, в чем различие между GREP и обычным текстовым поиском. Именно это сравнение и покажет, в чём заключается мощь GREP.

Занятие 2

Поиск/Замена. Метасимволы

Начало знакомства с метасимволами. Тут рассказ не обо всех, дальше по ходу дела будут появляться новые. После этого занятия вы уже сможете объяснить, почему метасимвол `\d` сноску не находит, а `\D` находит. Ну конечно, если сами сперва это попробуете. Не просто читать, а сразу пробовать, чтобы руки помнили, это очень важно в такой учёбе.

Занятие 3 Повторение операций

Универсальность инструмента `grep` не только в большой палитре вариантов шаблонов поиска, но и в наличии вариантов повторения поиска в тексте. Это те самые квантификаторы, что уже упоминались в первых занятиях, тут будет вся информация о них. Также названы операторы, не попавшие в меню команд.

Занятие 4 Кавычки

Это индизайн сам определяет, какая кавычка — открывающая или закрывающая, а в `grep` это надо делать нам, оформив правильно запрос.

Занятие 5

Наборы символов в формате Posix

Posix — это вариант группировки знаков по сходству начертания или применения. В стандартном наборе знаков поиска приведены не все имеющиеся наборы. Кроме того, эти наборы можно объединять и использовать вместе с другими шаблонами поиска.

Занятие 6 Исключение из поиска

Для наборов тоже можно определить противоположный диапазон действия. По-английски такой набор называется *Negated Character Class*, и когда это переводят как негативный класс, то это хоронит весь смысл идеи — определить пространство знаков, противоположное тем, что собраны в этом наборе.

Это тоже инструмент уменьшения области выборки, как и ленивый поиск, и вариант с использованием набора исключенных из поиска знаков работает заметно быстрее, поскольку ленивый оператор, чтобы сделать самую короткую выборку, должен перебрать все возможные. А на это уходит время. При использовании набора исключенных из рассмотрения знаков таких метаний по тексту нет.

Занятие 7 Кодировка Unicode

Это несомненное достоинство, что при работе с *grep*-запросами можно использовать варианты кодировки сходных по использованию знаков или отдельных знаков. Формат записи знака: `\p` затем в фигурных скобках название или сокращение. Буквы сокращения не всегда совпадают с первыми буквами полного названия. При работе с русским языком надо знать, что в стандартном наборе открывающих кавычек русской открывающей кавычки — `99` в нижнем регистре — нет.

Занятие 8

Адресация в запросе поиска

Это то, что в английской документации называется *back reference*, и переводится на русский язык как «обратная ссылка». Что такое и как работают операторы `\1`, `\2` и пр., об этом тут подробно, с примерами растолковано.

Занятие 9

Результаты поиска, не включаемые в выделение (просмотр вперед и назад)

Пожалуй, самые используемые методы поиска, когда определяется не только то, что надо найти, но и текст, который должен быть перед или после искомого текста. Особенность и исключительная полезность этого метода в том, что, во-первых, тексты до и после искомого текста, если они найдены, в выборку не включаются, во-вторых, условие для текстов до и после может быть изменено с точностью донаоборот: текст выделяется, если тексты до после отсутствуют.

Объяснено, почему появился и как работает вариант положительного просмотра назад `\K`.

Занятие 10 GREP-стили

grep-стиль — это символьный стиль, применённый к тексту при выполнении запроса, определённого в установках абзацного стиля на вкладке Стилль GREP. Эти команды grep-запросов не содержат поиска с учётом форматирования, т.е. нельзя найти, например, только курсивные слова.

Добавление нового стиля в список этих стилей, приоритет выполнения, временное отключение, и другие вопросы рассмотрены в этом занятии.

Занятие 11

Без маркировки подвыражения

Для идентификации в поле замены используемого в поле поиска подвыражения предусмотрено десять записей, \$0 для всего выражения, и \$1–\$9 для подвыражений. Но для решения задачи в поле поиска может потребоваться больше девяти шаблонов поиска, и если в поле записи попадут не больше девяти, эти, так скажем, вспомогательные шаблоны можно не маркировать. В этом занятии показано, как это делается.

Занятие 12

Режимы однострочный (Single-line) и многострочный (Multiline)

Да, эти сходные по написанию слова пишутся по-английски именно так: одно слово с дефисом, второе в одно слово.

При знакомстве с этой опцией может возникнуть непонимание. Первое предположение наличия двух режимов — однострочный и многострочный — используется или первый или второй. Но они используются одновременно, и каждый из режимов или включён или выключен, у них разные области действия.

Занятие 13

Горизонтальные и вертикальные пробелы (Horizontal & Vertical Spaces)

До версии CS6 метасимвол \s искал все шпации, табуляции, переводы и разрывы строк. Начиная с CS6, появилось ещё два метасимвола для поиска этих знаков в тексте. Для поиска шпаций и табуляций можно использовать \h, для поиска переводов строки и разрывов строки предусмотрен метасимвол \v.

Занятие 14 Рекурсия

У меня не было пока необходимости использовать рекурсию в каких-то рабочих проектах. Поэтому имеющиеся запросы и материалы — это на данный момент своего рода спортивный интерес.

И в этом занятии интересен опыт применения скрипта DoQueryList для поиска и окрашивания цветом разных вариантов перевёртышей.

Занятие 15 Об использовании \$0

Просто сделать такой grep-запрос, чтобы он находил границу между словами. Но найденная граница это не буква, не знак, и по этой причине её автоматически заменить на перевод строки нельзя. Лишь можно в ручном режиме идти от границе к границе и оператором `\r$0` вставлять переводы строки.

А можно ли переделать запрос поиска межсловной границы, чтобы он искал букву, и рядом с буквой ставил перевод строки?

Можно. Вот об этом данное занятие.

Занятие 16 DoQueryList

Эта программа появилась в сети в 2012 году. Работает со статьёй или выделенным текстом. Текстовые и grep-запросы могут выполняться как из системных

папок, так и из папок пользователя. Пятиязычный интерфейс. В этом занятии три примера применения этого скрипта.

Это очень надёжный помощник в подготовке текстов, когда одной очередностью запросов надо обрабатывать отдельные фрагменты статьи или целиком разные материалы.

Занятие 17

Скрытые проблемы в работе со сносками

Этот материал предполагалось разместить в одном из занятий для демонстрации различия обычных и конечных сносок. Но по ходу подготовки курса данная тема развернулась так неожиданно, что я решил сделать по этому вопросу отдельное занятие. Такой путеводитель по граблям.

Это, конечно, не столько учебная, сколько справочная информация о косяках в реализации работы со сносками. Но коль скоро эти недоделки относятся именно к grep, пусть это тоже будет отдельным занятием.

Хотя такой материал вполне можно разместить в дополнительных материалах, т.к. он не учебный.

Дополнительные материалы

В процессе постижения возможностей грег-обработки я пересмотрел массу ресурсов. Что-то показалось особенно интересным, и я для себя перевёл по быстрому, с использованием автоматического переводчика и последующим вычитыванием, три материала:

Reducing (? ...) Syntax Confusion

<http://www.rexegg.com/regex-disambiguation.html>

Перевод называется *Избавление от синтаксической путаницы.pdf*. Так сложилось, что пара символов (? чаще всего встречается в грег-операторах. Они похожи, но выполняют разные задачи, и мы поневоле теряемся в этом почти похожем разнообразии.

(?:...) и (?=...), (?<=...) и (?>...), (?1)... и (?1), (?<=...) и (?=...), (?<!...) и (?!...), (? : ...), (?>...)

Этот материал объясняет различия между ними. Я взялся за этот перевод достаточно давно, и возможно, моих тех знаний не хватило для абсолютно точного перевода всех описаний. Но что есть, то есть.

Mastering Lookahead and Lookbehind

<http://www.rexegg.com/regex-lookarounds.html>

Эта страница сайта в материале *О просмотре до и после [Lookahead & Lookbehind].pdf*

В этой работе досконально объясняется, как сделать запрос, проверяющий пароль.

Пароль должен соответствовать четырем условиям:

1. Пароль должен содержать от 6 до 10 символов \w
2. Он должен содержать хотя бы один строчный символ [a-z]
3. Он должен содержать не менее трех символов верхнего регистра [A-Z]
4. Он должен содержать хотя бы одну цифру \d

Regular Expression Recursion

<https://www.regular-expressions.info/recurse.html>

Recursive Regular Expressions

<http://www.rexegg.com/regex-recursion.html>

Эти две статьи собраны в файле *Рекурсия в GREP.pdf*. Чем больше я погружался в эту тему, тем меньше понимал. И я для себя решил, что если буду просто использовать проверенные грег-запросы с использованием рекурсии, то этого для меня достаточно.

В папке non-joiner & zero-width joiner материалы о различии символа non-joiner и символа нулевой ширины.

В папке Problems with Using the Endnotes marker ~U серия скрин-шотов обсуждения на форуме Treasures of GREP проблемы обработки сносок. Из этого обсуждения появилось занятие 17.

В папке Marc Autret статья Аутрета с его запросом поиска слов-перевёртышей.

Приобретение

Если что-то в ггер непонятно, надо спрашивать. Лучший ответ — это увидеть решение того, что непонятно. Видео (не меня на экране, а выполнения задачи средствами GREP) стоит тысячи слов — покажи мне, и я пойму. Но вы можете быть так уверены в своих силах и имеющемся знании, что вам достаточно будет, может быть, обмена несколькими письмами.

Эти два варианта возможного сотрудничества и определяют стоимость данного курса.

1 вариант

- 17 занятий, 112 задач, дополнительные материалы
- только письменные ответы на вопросы

2000 руб.

2 вариант

- 17 занятий, 112 задач, дополнительные материалы
- ответы на вопросы в формате текстом и/или видео (в зависимости от сути вопроса)

5000 руб.

Возврата денег не будет, т.к. в ознакомительных материалах достаточно информации, чтобы принять верное решение.

До Нового года действует скидка:

1 вариант **1500** рублей

2 вариант **4000** рублей.

Если для вас эта тема новая, то скорее всего вы даже в таком разложенном на детали курсе потеряете и темп усвоения, и кураж, если понадеетесь на то, что сможете пройти без видеопомощи первый вариант. В освоении нового важно именно увидеть, как это действует.

Решите, что для вас лучше — после двух-трёх недель учёбы знать ггер так же хорошо, как вы сейчас знаете оформление абзацного стиля, или купить первый вариант, и он станет очередными мегабайтами в одной из папок похороненных возможностей?

