

Regular Expressions ve grep, awk, sed ile Kullanımı

Koray OKSAY
koray.oksay@linux.org.tr

29 Mart 2014



http://en.wikipedia.org/wiki/Regular_expression

http://en.wikipedia.org/wiki/Deterministic_finite_automaton

http://en.wikipedia.org/wiki/Nondeterministic_Finite_Automaton

REGULAR EXPRESSIONS (DÜZENLİ İFADELER)



Düzenli İfadeler (RegEx)

Bir pozisyon ile eşleşen meta karakterler:

`^` Satır başı

`$` Satır sonu

`\<` Kelime başı

`\>` Kelime sonu



Düzenli İfadeler (RegEx)

Tek bir karakter ile eşleşen meta karakterler:

`.` Herhangi bir karakter

`[]` Karakter listesinden bir karakter

`[^]` Listedekiler haricinde bir karakter

`\ch` Özel karakteri anlamıyla kullanır



Düzenli İfadeler (RegEx)

Niceleme meta karakterleri:

- ? Kendisinden önceki karakter 0 ya da 1 defa tekrarlayabilir
- * Kendisinden önceki karakter 0 ya da satır sonuna kadar tekrarlayabilir
- + Kendisinden önceki karakter 1 ya da satır sonuna kadar tekrarlayabilir



Düzenli İfadeler (RegEx)

Niceleme meta karakterleri:

{n, m} Kendisinden önceki karakter en az n, en çok m defa tekrarlanabilir

{n} Kendisinden önceki karakter n defa tekrarlamalıdır

{n, } Kendisinden önceki karakter en az n defa tekrarlayabilir



Düzenli İfadeler (RegEx)

Diğerler meta karakterler:

| Ayırdığı ifadelerden bir tanesinin eşleşmesini sağlar

- ()
- "|" için sınırları belirler
 - Niceleme karakterleri için gruplama yapar
 - Backreference için tutma işlemi yapar

\1, \2, ... Geri başvuruları kullanmak için kullanılır



POSIX Eklentileri

POSIX	Non-standard	Perl/Tcl	Vim/sed	ASCII	Description
[:alnum:]				[A-Za-z0-9]	Alfanümerik karakterler
	[:word:]	\w	\w	[A-Za-z0-9_]	Alfanümerik karakterler ve “_”
		\W	\W	[^A-Za-z0-9_]	Alfanümerik ve “_” harici karakterler
[:alpha:]			\a	[A-Za-z]	Alfabetik karakterler (harfler)
[:blank:]			\s	[\t]	Boşluk ve “tab” karakteri
		\b	\< \>	(?<=\W) (?=\w) (?<=\w) (?=\W)	Kelime sınırları
[:digit:]		\d	\d	[0-9]	Rakamlar
		\D	\D	[^0-9]	Rakam olmayan karakterler
[:graph:]				[\x21-\x7E]	Görünür karakterler
[:lower:]			\l	[a-z]	Küçük harfler
[:print:]			\p	[\x20-\x7E]	Görünür karakterler ve boşluk
[:punct:]				[[!\"#\$%&'()*+,-./:;<=>?@\^_`{ }~ -]	Noktalama işaretleri
[:space:]		\s	_s	[\t\r\n\v\f]	Boşluk (whitespace) karakterleri
		\S		[^ \t\r\n\v\f]	Boşluk olmayan (non-whitespace)
[:upper:]			\u	[A-Z]	Büyük harfler
[:xdigit:]			\x	[A-Fa-f0-9]	Onaltılık sistemde (hexadecimal) rakamlar



Uygulamaların Karşılaştırılması

Meta Karakter	grep	egrep (grep -E)	sed	awk	gawk
<code>^ \$</code>	✓	✓	✓	✓	✓
<code>\< \></code>	✓	✓	✓		✓
<code>[] [^] .</code>	✓	✓	✓	✓	✓
<code>*</code>	✓	✓	✓	✓	✓
<code>?</code>		✓	✓	✓	✓
<code>+</code>		✓		✓	✓
<code>{n,m} {n} {n,}</code>		✓	✓		✓
<code> </code>		✓		✓	✓
<code>(...) \1 \2...</code>		✓	✓		
<code>(ab)* (ab) +</code>		✓		✓	✓
<code>&</code>			✓	✓	✓



Bash: Globbing (Dosya Adı Değiştirme)

Bash (ve diğerleri) için aşağıdaki meta karakterler vardır, bunlar Regular Expressions (RegEx) ile kariştirilmamalıdır:

*	0 ya da daha fazla karakter
?	Tam olarak 1 karakter
[abc]	Liste içerisinde bir karakter
[!abc]	Liste haricinde bir karakter
{a,ab}	Listelendir ifadelerden bir tanesi
[a-z]	Dizi içerisinde bir karakter
[!a-z]	Dizi haricinde bir karakter
\	Özel karakterin gerçek anlamıyla kullanımı



Yardımcı Araçlar

Regex Tester – Mozilla Firefox

Firefox ▾ Regex Tester

regexpal.com

regexpal 0.1.4 – a JavaScript regular expression tester

Case insensitive (i) \$ match at line breaks (m) Dot matches all (s; via XReqExp)

Options Quick Reference

:1? [0-9] / [1-3] ? [0-9] / [0-9] [0-9] :

Steve Blenheim:238-923-7366:95 Latham Lane, Easton, PA 83755:11/12/56:20300
Betty Boop:245-836-8357:635 Cutesy Lane, Hollywood, CA 91464:6/23/23:14500
Igor Chevsky:385-375-8395:3567 Populus Place, Caldwell, NJ 23875:6/18/68:23400
Norma Corder:397-857-2735:74 Pine Street, Dearborn, MI 23874:3/28/45:245700
Jennifer Cowan:548-834-2348:583 Laurel Ave., Kingsville, TX 83745:10/1/35:58900
Jon DeLoach:408-253-3122:123 Park St., San Jose, CA 04086:7/25/53:85100
Karen Evich:284-758-2857:23 Edgecliff Place, Lincoln, NB 92743:7/25/53:85100
Karen Evich:284-758-2867:23 Edgecliff Place, Lincoln, NB 92743:11/3/35:58200
Karen Evich:284-758-2867:23 Edgecliff Place, Lincoln, NB 92743:11/3/35:58200
Fred Fardbarkle:674-843-1385:20 Parak Lane, Duluth, MN 23850:4/12/23:780900
Fred Fardbarkle:674-843-1385:20 Parak Lane, Duluth, MN 23850:4/12/23:780900
Lori Gortz:327-832-5728:3465 Mirlo Street, Peabody, MA 34756:10/2/65:35200
Paco Gutierrez:835-365-1284:454 Easy Street, Decatur, IL 75732:2/28/53:123500
Ephram Hardy:293-259-5395:235 Carlton Lane, Joliet, IL 73858:8/12/20:56700
James Ikeda:834-938-8376:23445 Aster Ave., Allentown, NJ 83745:12/1/38:45000
Barbara Kertz:385-573-8326:832 Ponce Drive, Gary, IN 83756:12/1/46:268500
Lesley Kirstin:408-456-1234:4 Harvard Square, Boston, MA 02133:4/22/62:52600
William Kopf:846-836-2837:6937 Ware Road, Milton, PA 93756:9/21/46:43500
Sir Lancelot:837-835-8257:474 Camelot Boulevard, Bath, WY 28356:5/13/69:24500
Jesse Neal:408-233-8971:45 Rose Terrace, San Francisco, CA 92303:2/3/36:25000
Zippy Pinhead:834-823-8319:2356 Bizarro Ave., Farmount, IL 84357:1/1/67:89500
Arthur Putie:923-835-8745:23 Wimp Lane, Kensington, DL 38758:8/31/69:126000
Popeye Sailor:156-454-3322:945 Bluto Street, Anywhere, USA 29358:3/19/35:22350
Jose Santiago:385-898-8357:38 Fife Way, Abilene, TX 39673:1/5/58:95600
Tommy Savage:408-724-0140:1222 Oxbow Court, Sunnyvale, CA 94087:5/19/66:34200
Yukio Takeshida:387-827-1095:13 Uno Lane, Ashville, NC 23556:7/1/29:57000
Vinh Tranh:438-910-7449:8235 Maple Street, Wilmington, VM 29085:9/23/63:68900

Need more power? Get [RegexBuddy](#), a powerful regex tester that inspired Regexpal.
Round out your mastery over regex with [RegexMagic](#) and [PowerGREP](#).

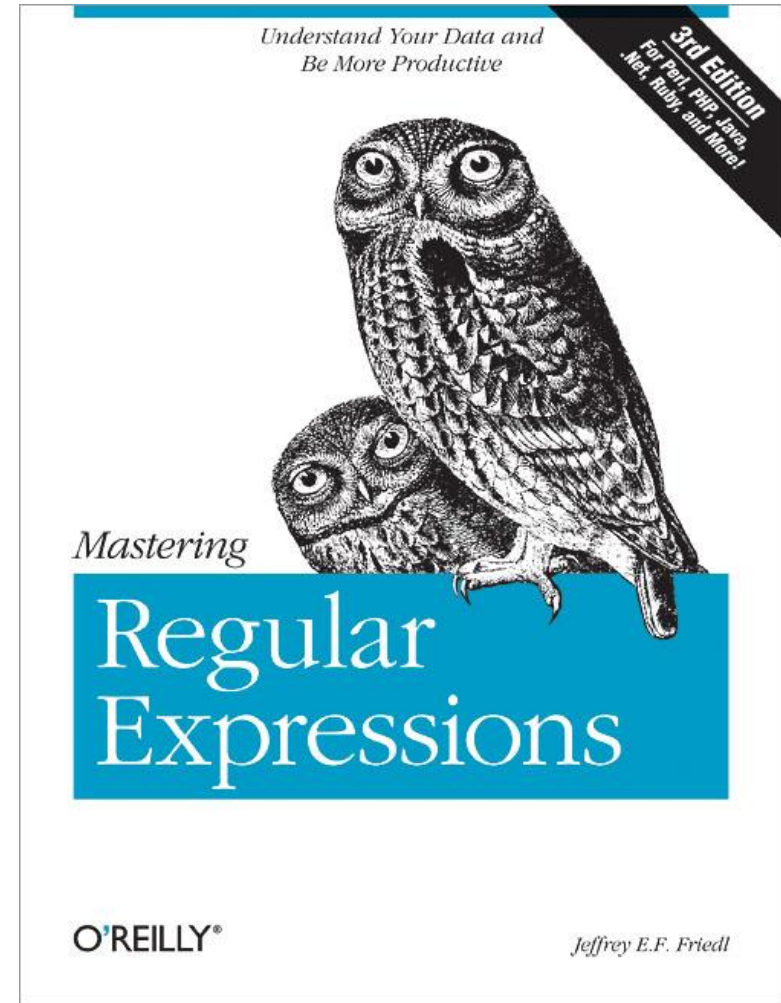
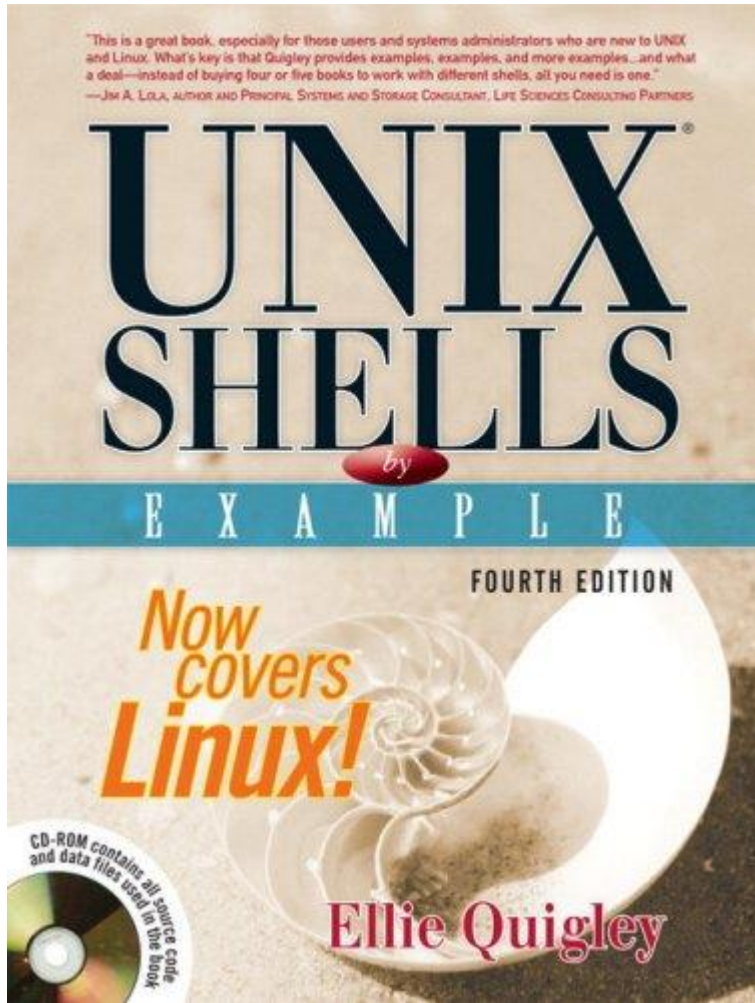
[Permalink](#) — © 2012 Steven Levithan. Powered by XReqExp and Regexp Colorizer.

Any character except newline.
A period (and so on for *, \{, \}, etc.)
The start of the string.
The end of the string.
A digit, word character [A-Za-z0-9_], or whitespace.
Anything except a digit, word character, or whitespace.
Character a, b, or c.
a through z.
Any character except a, b, or c.
Either aa or bb.
Zero or one of the preceding element.
Zero or more of the preceding element.
One or more of the preceding element.
Exactly n of the preceding element.
n or more of the preceding element.
Between m and n of the preceding element.
Same as above, but as few as possible.
Capture expr for use with \1, etc.
Non-capturing group.
Followed by expr.
Not followed by expr.

Near-complete reference



Kaynak Önerileri



koray.oksay@linux.org.tr

www.koksay.net

@korayoksay

TEŞEKKÜRLER

