

PERL REGULAR EXPRESSION & DBM

컨텐츠검색플랫폼팀
조영일

REGULAR EXPRESSION

- 정규표현식 (Regular Expression, RE)
 - Language
 - For pattern matching & substitution
 - Used in egrep, Perl, Python, .NET, ...
 - PCRE (Perl-Compatible-Regular-Expression)
- 구성
 - Operator
 - Literal
 - Metacharacter
 - Character Class
 - Quantifier
 - Modifier

OPERATORS (1)

- Match(search)
 - `match($string, $pattern)`
 - `m/pattern/`
 - `/pattern/`
 - `qr/pattern/`
- 예제
 - It was many and many a year ago,
 - In a kingdom by the sea,
 - That a maiden there lived whom you may know
 - By the name of Annabel Lee
 - `/Annabel Lee/`
 - `/ea/`
 - `/[Bb]y/`

OPERATORS (2)

- Substitute (replace)
 - `subst($string, $pattern, $replacement)`
 - `s/pattern1/pattern2/`
- 예제
 - `s/boy/girl/`
 - `s/the/a/g`
 - `s/violin|piano|guitar/instrument/g`

LITERAL (1)

- Literal

- 일반 문자, 숫자

- A-Z, a-z, 0-9

- 특수기호

- Perl 문법 상, 연산자나 타입기호, 문법요소로 사용됨

- @, \$, %, ^, +, -, *, /, ^, ~, =, ` , ‘ , “ , | , ; , #

- RE에서는 메타문자로서의 역할을 수행함

- [], !, ^, \$, (), +, -, *, /, \w, .., |, ?, #

- 그러므로 특수기호 본래의 의미대로 사용하려면 \를 앞에 붙여서 “탈출 (escape)”시켜야 함

- Ex) \<, \w, \w\w, \w/, \w[, \w], \w(, \w), \w?, \w*, \w+, \w^, \w\$, \w!, \w{, \w},

LITERAL (2)

- 패턴 매칭의 예
 - /Frodo/;
 - if (/zzz/) {
 - print;
 - }
 - while (<>) {
 - if (/ab*c/) {
 - print \$_;
 - }
 - }

METACHARACTER

Code	의미
\	다음에 위치하는 문자를 metacharacter로 만듦
.	임의의 한 문자
^	문자열 또는 행의 처음 위치
\$	문자열 또는 행의 마지막 위치
*	앞에 위치한 패턴요소가 0번 이상 반복됨
+	앞에 위치한 패턴요소가 1번 이상 반복됨
?	앞에 위치한 패턴요소가 0 또는 1번 나타남
{n, m}	앞에 위치한 패턴요소가 n번부터 m번까지 반복됨
[abcdef]	[] 안에 나열된 문자 중의 한 문자
[^abcdef]	[] 안에 나열된 문자가 아닌 한 문자
(abcdef)	()로 둘러싸인 문자열을 하나의 그룹으로 묶음
abc def	앞의 패턴이 나타나지 않으면 뒤의 패턴이 나타남

ASSERTION

- 예제

- Though nurtured like the sailing moon
- In beauty's murderous brood
- She walked awhile and blushed awhile
- And on my pathway stood
- Until I thought her body bore
- A heart of flesh and blood

- /th./

- /[^]A/

- /awhile\$/

- /sto*d/

- /b.o+d/

QUANTIFIER

- 예제

- I will arise and go now, and go to Innisfree,
- And a small cabin build there,
- of clay and wattle made:
- Nine bean-rows will I have there,
- a hive for the honey-bee,
- And live alone in the bee-loud glade.

- /.ows? /

- /[hl]ive/

- /[^aeiou] [aeiou]{1,2}/

- /(arise|build|go|have|live)/

CHARACTER CLASS (1)

Code	의미
\a	Alert
\n	newline - 다음 행
\r	Carriage return - 현재 행 처음으로 돌아감
\t	Tab
\f	Formfeed
\e	Escape
\d	숫자, [0-9], \D는 숫자가 아닌 것
\w	글자, [a-zA-Z_0-9], \W는 글자가 아닌 것
\s	공백, [\t\n\r\f], \S는 공백이 아닌 것
\b	단어의 경계, \B는 단어의 경계가 아닌 것
\1, ... \9	패턴 안에서 ()로 둘러싸인 그룹을 가리키는 backreference

CHARACTER CLASS (2)

- 예제

- fooley 555-1234 2400/1200/300 B
- foot 555-6699 1200/300 B
- macfoo 555-6480 1200/300 A
- sabafoo 555-2127 1200/300 C

- `/B\Wn/`

- `/oo\Wt555/`

- `/(Ww+)/`

- `/(Wd+)\W/(Wd+)/`

- `/(Ww+foo)\Wb/s/(foo)/\W1bar/`

MODIFIER (1)

Code	의미
i	대소문자 구분없이 패턴 매칭
m	문자열을 여러 행으로 간주함 ^와 \$가 \n에 대해 일치됨
s	문자열을 한 행으로 간주함 ^와 \$는 \n를 무시하지만 .은 \n에 대해 일치됨
x	공백과 주석을 패턴에 포함시킴
o	패턴을 컴파일해서 재사용함
g	대상 문자열에 대해 패턴 매칭을 반복함

MODIFIER (2)

- 예제
 - Tyger! Tyger! burning bright
 - In the forests of the night,
 - What immortal hand or eye
 - Could frame thy fearful symmetry?
- /what/i
- /Tyger/g
- while (<>) {
 - print if /somepattern/o; # deprecated, default modifier
- }

패턴 매칭의 결과 사용법

- 예제

- `print "This line matches... \n" if $line =~ m/flower/;`
- `if ('hot cross buns' =~ /cross/) { print "$`, $&, $' \n"; }`
- `if ('Bilbo Baggins's birthday is September 22' =~ /(.*?)'s birthday is (.*)/) {`
 - `print "Person: $1, Date: $2 \n";`
- `}`
- `if (($key, $value) = /(w+): (.*)/) { ... }`
- `while ($line =~ /(sun|moon|star)/g) { print $1; }`
- `while (<FILE>) { s/good/nice/g; }`

확장 문법

- (? =PATTERN): positive lookahead
- (? !PATTERN): negative lookahead

- (? <=PATTERN): positive lookbehind
- (? <!PATTERN): negative lookbehind

- (? :PATTERN): no backreferencing

- (? {CODE}): match-time evaluation

DBM HASH IN PERL

- DBM (Database Management)
 - 해시(hash)처럼 DB 파일을 사용할 수 있음
 - Berkeley DB (DB_File)
 - NDBM (NDBM_File)
 - GDBM (GDBM_File)

DBM 사용예 (1)

- Berkeley DB 사용

- use DB_File;
- tie %name_age_map, "DB_File", "dbmfilename";
- %name_age_map = ("Park" => 32, "Choi" => 12);
- \$name_age_map{"Kim"} = 46;
- \$name_age_map{"Lee"} = 58;
- foreach \$name (sort keys %name_age_map) {
 - print "\$name: \$name_age_map{\$name}\n";
- }
- untie %name_age_map;

DBM 사용예 (2)

- GDBM 사용
 - use GDBM_File;
 - tie %name_age_map, "GDBM_File",
"dbmfilename", &GDBM_WRCREAT, 0644;