

PB173 Perl

08 Úvodzovkovité operátory

Roman Lacko <xlacko1@fi.muni.cz>

Obsah

Úvodzovkovité operátory	1
Zhoda so vzorom	2
Nahradzovanie	8
Transliterácia	12
Externé príkazy	15

Úvodzovkové operátory

Okrem úvodzoviek (*quote operators*), ktoré sme videli minule, má Perl aj úvodzovkové operátory (*quote-like operators*).

```
/PATTERN/FLAGS
```

```
m/PATTERN/FLAGS
```

```
qr/PATTERN/
```

```
s/PATTERN/REPLACEMENT/FLAGS
```

```
y/SOURCE-SET/TARGET-SET/FLAGS
```

```
tr/SOURCE-SET/TARGET-SET/FLAGS
```

```
qx 'COMMAND'
```

Zhoda so vzorom



Tento operátor sme videli na cvičení o regulárnych výrazoch.

```
$value =~ /PATTERN/FLAGS  
$value =~ m/PATTERN/FLAGS
```

- V skalárnom kontexte vráti logickú hodnotu, či reťazec vyhovuje výrazu.
- V zoznamovom kontexte vráti hodnoty zachytené skupinami ().

Podobne ako pri ostatných *quote operators*, aj tu je možné namiesto `m//` použiť iné znaky bez efektu na chovanie, s výnimkou `' '`.

```
my $string = "b";  
"abc" =~ m"a$string";           # True, match on <ab>c  
"abc" =~ m'a$string';          # False
```

Zhoda so vzorom

Príznamy

FLAGS za koncom úvodzovkovitých operátorov mení ich chovanie. Tieto príznaky je možné kombinovať.

- **m** - chápe reťazec ako viac riadkov (mení chovanie **^** a **\$**)
- **s** - chápe reťazec ako jeden riadok (mení chovanie **.**)
- **n** - vypne zachytávanie hodnôt v skupinách (okrem pomenovaných)
- **i** - zhoda bez dôrazu na veľkosť písmen
- **x, xx** - povolí komentáre a medzery vo výraze
- **g** - test na viacnásobnú zhodu

m: Multiple lines

```
/PATTERN/m
```

- `^` a `$` znamenajú obvykle začiatok a koniec reťazca.
- `S /m` znamenajú začiatok a koniec *riadka*.

s: Single line

```
/PATTERN/s
```

- `.` obvykle akceptuje všetko okrem `\n`
- `S /s` bude `.` akceptovať všetko

x, xx: Extended patterns

```
/PATTERN/x  
/PATTERN/xx
```

- Povolí medzery a komentáre začínajúce # v regulárnom výraze.
- /xx navyše povolí medzery aj v [].
- Vypnúť efekt vo výraze je možné s konštrukciou \Q...\E

```
say "$&" if $string =~ m{  
    ^           # Start of string  
    \w         # Word character  
    \Q#\E     # Literal <#>  
}xx;
```

g: Global matching

```
/PATTERN/g
```

Tento príznak mení, čo test zhody vracia pri násobnom volaní:

- V skalárnom kontexte pri každom volaní pokračuje od poslednej zhody.

```
say "$&" while "abc" =~ /../;    # Prints <a> indefinitely
say "$&" while "abc" =~ /../g;  # Prints <a>, <b>, <c>
```

- V zoznamovom kontexte vráti zachytené hodnoty skupín, ak vzor neobsahuje skupiny, potom vráti všetky vyhovujúce podreťazce.

```
say for "abc" =~ /../g;          # Prints <a>, <b>, <c>
say for "abcd" =~ /(.) ./g;     # Prints <a>, <c>
```


qr operátor

Operátor `qr//` vráti regulárny výraz ako hodnotu, ktorú je možné uložiť do skalárnej premennej a použiť neskôr ako `m//`.

```
my $regex = qr/a.*x/;

if ($string =~ $regex) {
    # ...
}

say ref $regex;           # Regexp
```



Dereferencia vráti normalizovaný reťazec, ale môže sa meniť s verziou Perlu.

Nahrádzovanie

```
s/PATTERN/REPLACEMENT/FLAGS
```

- Vyhodnotí **/PATTERN/** ako **m/PATTERN/**,
vyhovujúcu časť nahradí výrazom **REPLACEMENT**.
- Používa sa s operátorom **=~** alebo **!~**.

```
my $str = "this or that";  
  
say $str if $str =~ s/or/and/;           # this and that
```

Nahradzovanie

```
$string =~ s/PATTERN/REPLACEMENT/FLAGS
```

- Vrátí počet vykonaných nahradení
 - Bez **/g** je to vždy 0 alebo 1.
 - S **/g** to môže byť viac.
- Návratovú hodnotu môže zmeniť **/r** príznak.

Nahradzovacia časť

Výraz **REPLACEMENT** môže obsahovať okrem reťazca aj interpolované premenné a špeciálne sekvencie.

```
my $string = "armadillo";  
say $str if $str =~ s/(.)\g1/$12/;      # armadil2o  
say $str if $str =~ s/^(.)/\u$1/;      # Armadil2o
```

Špeciálne sekvencie **\X** fungujú **pred interpolovaným reťazcom**:

- **\l**, **\u** zmenia veľkosť nasledujúceho znaku.
- **\L**, **\U** menia veľkosť písmen až do **\E**.
- **\F** zapne *fold-case* (viď **fc** s minulého semináru) až po **\E**.
- **\Q** vypne význam špeciálnych znakov až po **\E**.

Príznamy pre `s///`

Operátor `s///` podporuje všetky príznaky ako `m//`, a navyše:

- `r` - Vytvorí kópiu reťazca a tú po zmene vráti.

```
say "Cat" =~ s/t/lf/r; # Calf
```

- `e` - Vyhodnotí **REPLACEMENT** ako Perl výraz

```
say "<e,4>" =~ s/<([^\,]+),(\d+)>/$1 x $2/re; # eeee
```

- `ee` - Vyhodnotí **REPLACEMENT** ako reťazec, a potom ako výraz

```
say '<${ENV{USER}}>' =~ s/<(.*?)>/$1/re; # ${ENV{USER}}
say '<${ENV{USER}}>' =~ s/<(.*?)>/$1/ree; # pazuzu
```

Transliterácia

```
tr/SEARCH/REPLACE/FLAGS  
y/SEARCH/REPLACE/FLAGS
```

- `tr//` a `y//` sú synonymá.
- Nahradí znaky v **SEARCH** zodpovedajúcimi znakmi v **REPLACE**, alebo ich zmaže (podľa príznakov).
- Nepoužíva interpoláciu.

```
my $string = 'abba';  
say $string =~ tr/a-c/0-2/r;           # 0110  
say $string =~ tr/$ab/0-2/r;         # 1221
```

Transliterácia

Ak je zoznam **REPLACE** kratší než **SEARCH**, doplní sa posledným znakom alebo kópiu **SEARCH**:

```
$string =~ tr/a-e/01/;           # Same as <tr/abcde/01111/>  
$string =~ tr/abc//;           # Same as <tr/abc/abc/>
```

Chovanie v týchto prípadoch je možné zmeniť modifikátormi.

Modifikátory

- `/c` nahradí komplement **SEARCH**.
- `/d` nedoplní kratší **REPLACE**, ale znaky zmaže.
- `/r` upraví kópiu reťazca (ako pri **s///**).
- `/s` stlačí násobný výskyt nahradených znakov na jeden.

```
say "abc" =~ tr/ab/x/r;           # xxc
say "abc" =~ tr/ab/x/dr;         # xc
say "abc" =~ tr/ab/x/cr;         # abx

say "mississippi" =~ tr/ips/ha/sr; # mhahahah
say "mississippi" =~ tr[s][p]sr;  # mipippi
```


Externé príkazy

```
`COMMAND OR EXPRESSION`  
qx"COMMAND OR EXPRESSION"
```

Posledný úvodzovkový operátor, `qx""`, slúži na spustenie externého procesu.

- Ak reťazec neobsahuje špeciálne znaky shellu, spustí sa priamo, inak sa spustí ekvivalent `sh -c "COMMAND OR EXPRESSION"`.



V závislosti od platformy sa môže chovať rôzne (napr. na Windows).

- V skalárnom kontexte vráti výstup príkazu alebo `undef`.
- V zoznamovom kontexte vráti jednotlivé riadky výstupu.

Návratový kód procesu bude v premennej `$?`.

Externé príkazy

Viacriadkový príkaz je možné spustiť pomocou *here-document*:

```
my @files = <<~`ENDSCRIPT`;  
  for i in {1..10}; do  
    name="\$file\$i"  
    if [ -e "\$name" ]; then  
      echo "\$name"  
    fi  
  done  
ENDSCRIPT
```

Odbočka: `system()` a kamaráti

Videli sme už špeciálnu verziu `open`, ktorá sa správa ako `popen(3)`:

```
open my $pipe, '-|', "gpg --decrypt $secret"  
  or die "$secret: Cannot invoke gpg: $?";
```

Perl má na komunikáciu ďalšie mechanizmy známe z C a POSIX:

```
system PROGRAM[, LIST]  
  
fork  
exec PROGRAM[, LIST]  
wait  
waitpid PID, FLAGS
```



`perldoc -f FUNKCIA, IO::Pipe`