



Perl Programming

Regular Expression

Dionysios Kandylas
Univeristy of Thessaly

Κανονικές Παραστάσεις

Μια κανονική παράσταση (regular expression) είναι ένα μοτίβο – ένα πρότυπο- το οποίο εξετάζεται αν ταιριάζει με ένα αλφαριθμητικό.

Ισχυρή, ευέλικτη και αποτελεσματική επεξεργασία κειμένου. Οι Κανονικές παραστάσεις είναι σαν μια μικρή γλώσσα προγραμματισμού.

Κανονικές Παραστάσεις

- Τελεστής αντιστοιχίας
- Τελεστής αντικατάστασης
- Τελεστής μετάφρασης

Τελεστής αντιστοιχίας

- τελεστής αντιστοιχίας **m//**
- βασική φόρμα **m/PATTERN/**;
- **=~** αντιστοιχία μεταξύ μεταβλητής

```
#!/usr/bin/perl
use strict;
use warnings;

my $myString = "Hello Everyone";
if($myString =~m/one/){
    print "match.";
}

exit(0);
```

Results:- match.

Τελεστής αντιστοιχίας

Αντικατάσταση **m** με //

```
#!/usr/bin/perl
use strict;
use warnings;

my $myString = "Hello Everyone";
if($myString =~ /one/){
    print "match.";
}

exit(0);
```

Results:- match.

Τροποποιήσεις Τελεστή αντιστοιχίας

Modifier	Meaning
g	Match globally, i.e., find all occurrences.
i	Do case-insensitive pattern matching.
m	Treat string as multiple lines.
o	Evaluates the expression only once. Use this modifier when the pattern is a variable running in the loop and may be changed during running.
s	Treat string as single line.
x	Allows you to use white space in the expression for clarity.

Τροποποιήσεις Τελεστή αντιστοιχίας

```
#!/usr/bin/perl
use strict;
use warnings;

my $myString = "Hello Everyone";
foreach my $myMatch ($myString =~/e/ig){
    print"match.\n";
}

exit(0);
```

Results:-

```
match.
match.
match.
match.
```

Τελεστής αντικατάστασης

- τελεστής αντικατάστασης **s///**
- βασική φόρμα **s/PATTERN/REPLACEMENT/;**

```
#!/usr/bin/perl
use strict;
use warnings;

my $myString = "Hello Everyone";
my $myCount = $myString =~ s/Hello/Hi/;
print "$myString \n";
print "$myCount \n";

exit(0);
```

Results:-

Hi Everyone

1

Τροποποιήσεις Τελεστή αντικατάστασης

Modifier	Meaning
g	Match globally, i.e., find all occurrences.
i	Do case-insensitive pattern matching.
m	Treat string as multiple lines.
o	Evaluates the expression only once. Use this modifier when the pattern is a variable running in the loop and may be changed during running.
s	Treat string as single line.
x	Allows you to use white space in the expression for clarity.
e	Evaluates the replacement as if it were a Perl statement, and uses its return value as the replacement text

Τελεστής Μετάφρασης

- Τελεστής μετάφρασης **tr///**

```
#!/usr/bin/perl
use strict;
use warnings;

my $myString = "aaabbbccc";
my $myCount = $myString =~tr/abc/edf/;
print "$myString \n";
print "$myCount \n";

exit(0);
```

Results:-

eedddfff

9

Τροποποιήσεις Τελεστή Μετάφρασης

Modifier	Meaning
c	Complement <i>SEARCHLIST</i> .
d	Delete found but unreplaced characters.
s	Squash duplicate replaced characters

Τροποποιήσεις Τελεστή Μετάφρασης

```
#!/usr/bin/perl
use strict;
use warnings;

my $myString = "H^el^l^o Ever^yone";
my $myCount = $myString =~tr/^//d;
print "$myString \n";
print "$myCount \n";

exit(0);
```

Results:-

Hello Everyone

4

Μετα-χαρακτήρες

Μεταχαρακτήρας	Περιγραφή
.	Οποιοσδήποτε χαρακτήρας εκτός από την αλλαγή γραμμής
^	Αρχή μιας γραμμής
\$	Τέλος μιας γραμμής
\w	Οποιοσδήποτε χαρακτήρας λέξης
\W	Οποιοσδήποτε χαρακτήρας εκτός από χαρακτήρα λέξης
\s	Χαρακτήρας διαστήματος
\S	Οποιοσδήποτε χαρακτήρας εκτός από χαρακτήρα διαστήματος
\d	Οποιοδήποτε ψηφίο
\D	Οποιοσδήποτε χαρακτήρας εκτός από ψηφίο

Κλάσης χαρακτήρων

Code	Matches
\d	A digit, same as [0-9]
\D	A nondigit, same as [^0-9]
\w	A word character (alphanumeric), same as [a-zA-Z_0-9]
\W	A non-word character, [^a-zA-Z_0-9]
\s	A whitespace character, same as [\t\n\r\f]
\S	A non-whitespace character, [^ \t\n\r\f]
\C	Match a character (byte)
\pP	Match P-named (Unicode) property
\PP	Match non-P
\X	Match extended unicode sequence

Code	Matches
\l	Lowercase until next character
\u	Uppercase until next character
\L	Lowercase until \E
\U	Uppercase until \E
\Q	Disable pattern metacharacters until \E
\E	End case modification

Ποσοδείκτες

Ποσοδείκτης	Περιγραφή
?	0 ή 1 εμφάνιση (είναι σημαντικός, γιατί αναγκάζει το πρότυπο να μην είναι «άπληστο»)
+	1 ή περισσότερες εμφανίσεις
.	0 ή περισσότερες εμφανίσεις
{n,m}	Μεταξύ n και m εμφανίσεων
{n, }	Τουλάχιστον n εμφανίσεις
{ ,m}	Όχι περισσότερες από m εμφανίσεις

Άσκηση 1:

Να βρεθεί πόσες φορές εμφανίζεται το start codon ATG σε μία αλληλουχία DNA

ATGCCATGAGATGTT

Άσκηση 1:

```
#!/usr/bin/perl
use strict;
use warnings;

my $mySeq = "ATGCCATGAGATGTT";
my $myCount = () = $mySeq =~m/ATG/g;
print "$mySeq \n";
print "$myCount \n";

exit(0);
```

Άσκηση 2:

Να αντικαταστήσετε την ακολουθία DNA σε RNA

ATGCCATGAGATGTT

Άσκηση 2:

```
#!/usr/bin/perl
use strict;
use warnings;

my $mydna = "ATGCCATGAGATGTT";
$mydna =~ s/T/U/g;
print "RNA: $mydna\n";

exit(0);
```

Άσκηση 3:

Να βρεθεί το ID από ένα αρχείο fasta

```
>seq1234 description of seq
```

Άσκηση 3:

```
#!/usr/bin/perl
use strict;
use warnings;

my $myfasta = ">seq1234 description of seq";
if ($myfasta =~ /^>(\S+)/) {
    print "ID: $1\n";
}

exit(0);
```