

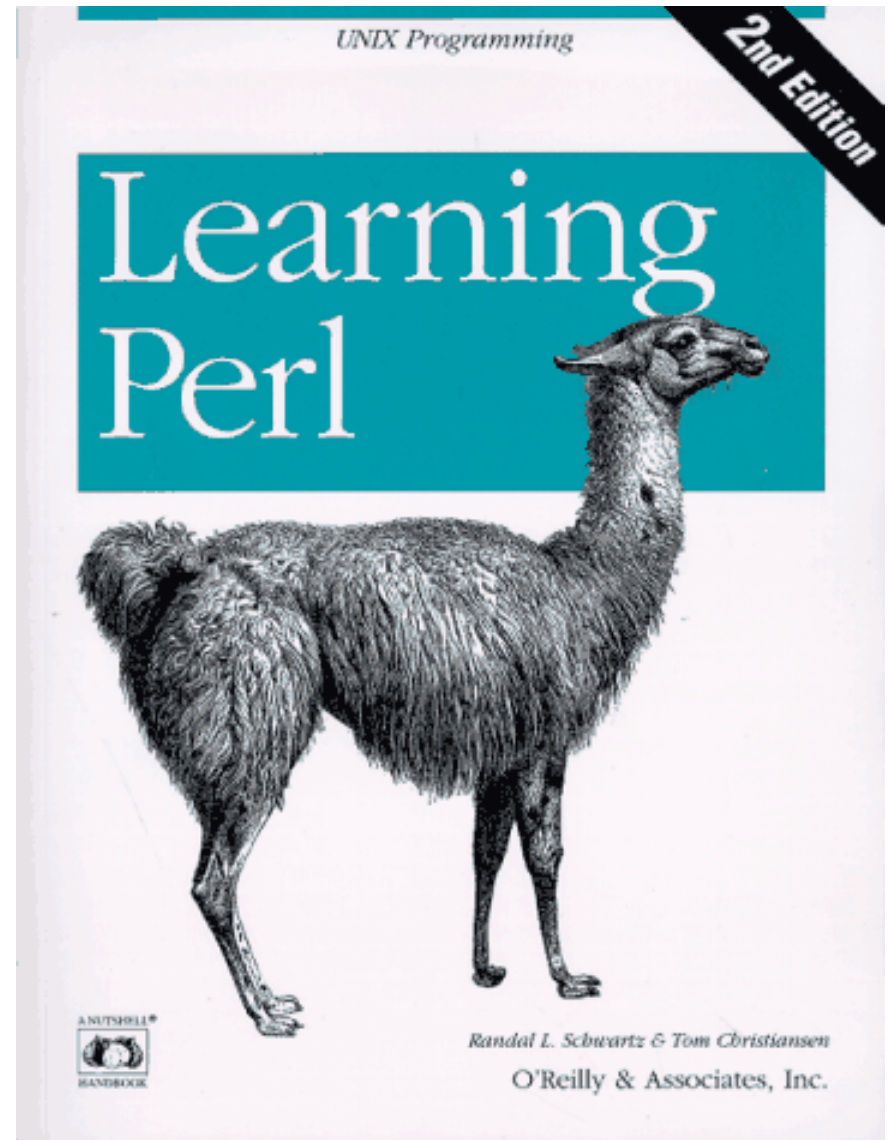
Η γλώσσα PERL

Παντελής Μπάγκος

Παν/μιο Θεσσαλίας
2025

Η γλώσσα PERL

- Practical Extracting and Report Language
- Larry Wall
- Τι είναι, TMTOWTDI
- Ποιος ο σκοπός της —→ Διαχείριση αρχείων, γρήγορη καμπύλη μάθησης
- Διαθεσιμότητα, πλατφόρμες —→ Σχεδόν όλες
- Ένα απλό πρόγραμμα



Practical Programming Through Computer Science

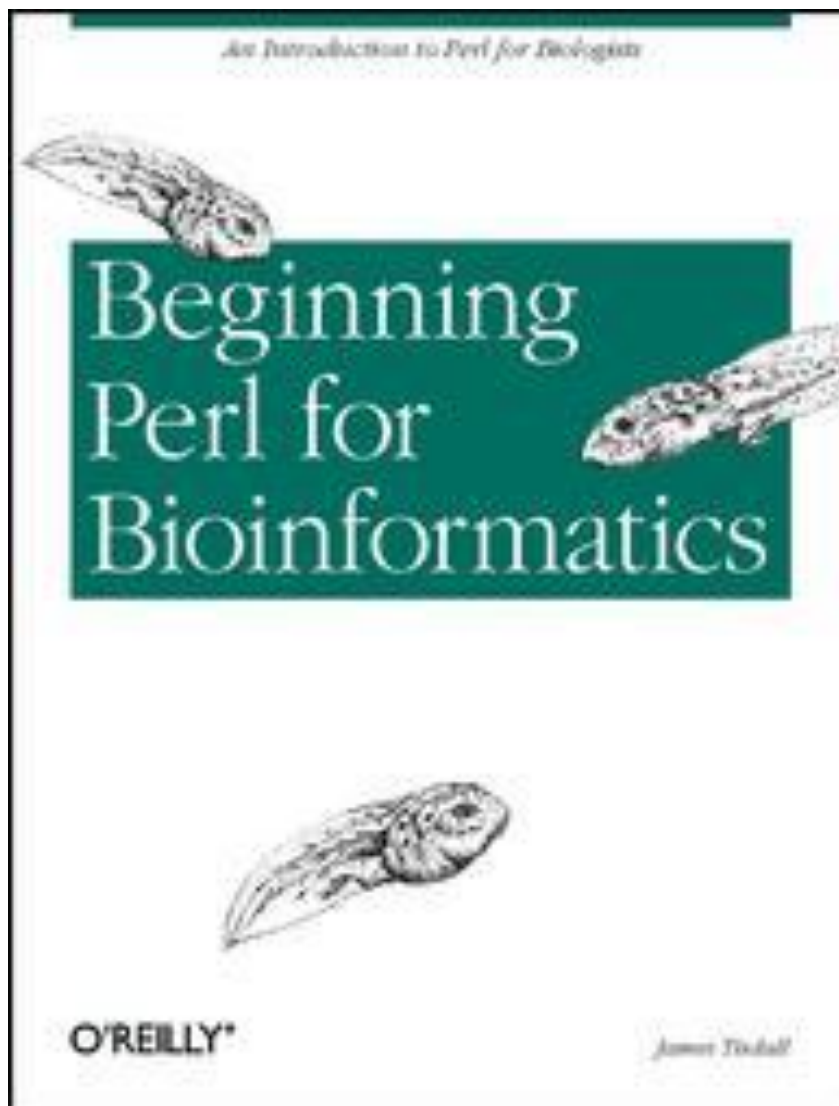
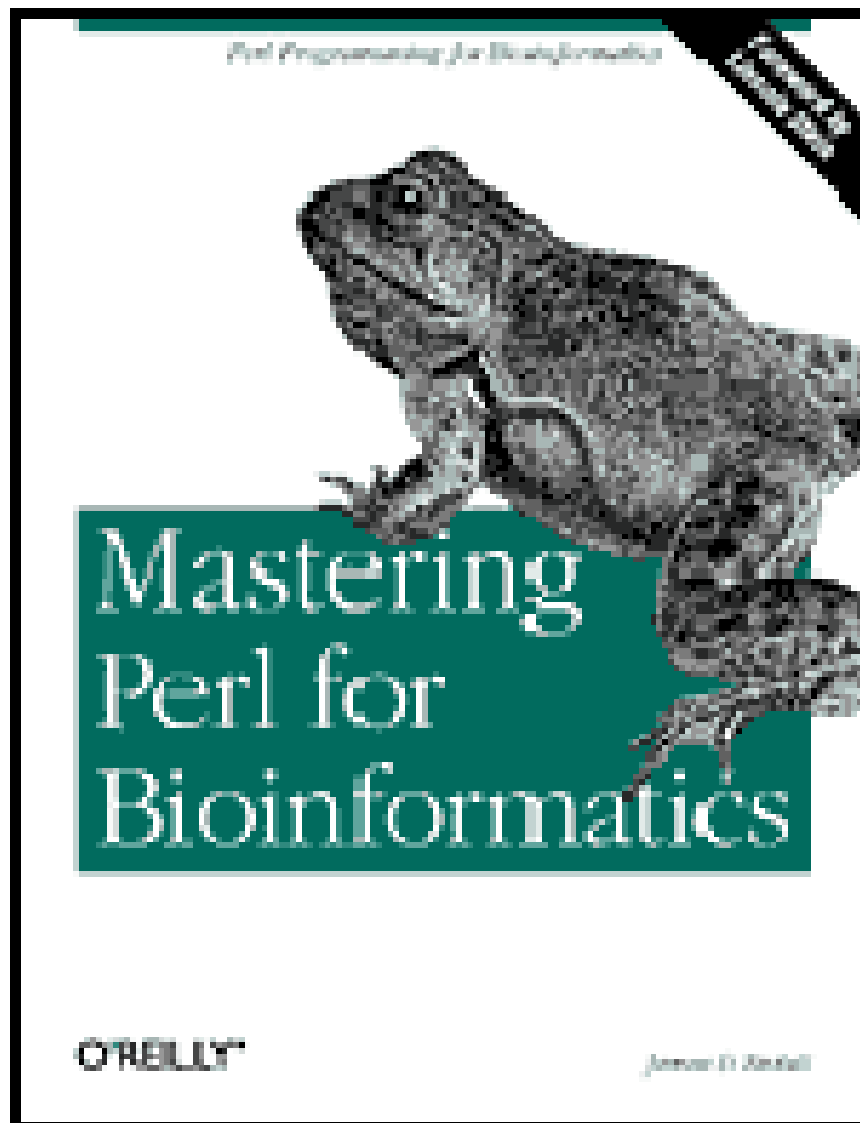
Mastering

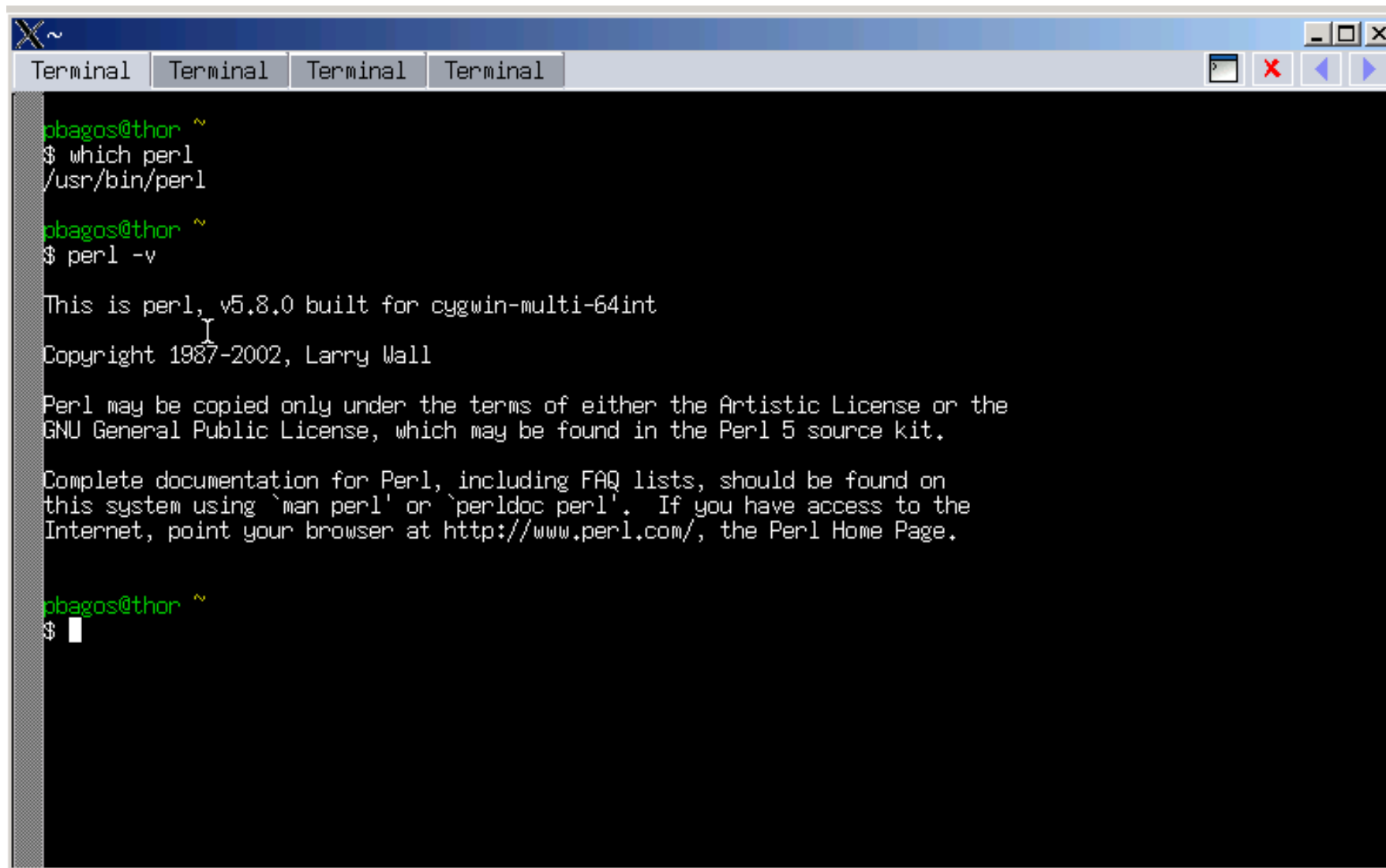
Algorithms with Perl



O'REILLY®

Jon Orwant, Jarkko Hietaniemi & John Macdonald



A screenshot of a terminal window with a blue title bar and four tabs labeled 'Terminal'. The terminal has a black background with green text for the prompt and white text for the output. The user 'pbagos@thor' has executed 'which perl' and 'perl -v'. The output shows the path '/usr/bin/perl' and the Perl version 'v5.8.0' built for 'cygwin-multi-64int'. It also displays the copyright '1987-2002, Larry Wall' and licensing information regarding the Artistic License and GNU GPL. Documentation instructions are provided, pointing to 'man perl' or 'perldoc perl' for local help and 'http://www.perl.com/' for the Perl Home Page. The prompt '\$' is followed by a cursor.

```
pbagos@thor ~  
$ which perl  
/usr/bin/perl  
  
pbagos@thor ~  
$ perl -v  
  
This is perl, v5.8.0 built for cygwin-multi-64int  
Copyright 1987-2002, Larry Wall  
  
Perl may be copied only under the terms of either the Artistic License or the  
GNU General Public License, which may be found in the Perl 5 source kit.  
  
Complete documentation for Perl, including FAQ lists, should be found on  
this system using `man perl' or `perldoc perl'. If you have access to the  
Internet, point your browser at http://www.perl.com/, the Perl Home Page.  
  
pbagos@thor ~  
$
```

One-liners

- `perl -ne`
`'BEGIN{$/= ">"; $" = "; } ($d, $_) = / (. * ?) \`
`n (. + ?) > ? $ / s ; push @ { $h { lc ( ) } } , $d`
`if $_ ; END { for (keys %h) { print "> @ { $h { $_ }`
`} $_ " } } ' filename`
- `perl -ne`
`'BEGIN{$/= ">"} if (/ ^ \s* ( \S+ ) / ) { open (F`
`, "> $1" ) || warn "$1 write`
`failed: $! \n" ; chomp ; print F "> ", $_ } '`

Hello World

```
#!/usr/bin/perl  
# To proto mas programma  
print "Hello World\n";
```

Επιλογές

- `perl program.pl`
- `./program.pl`
- `perl -e '...perl program...'`

Greetings

```
#!/usr/bin/perl
# To deytero mas programma
print "What is your name\?\n";
$name = <STDIN>;
chomp $name;
print "Hello $name\n";
```

Scalars και Variables

- Το ποιο απλό είδος δεδομένων
- Μπορεί να είναι αριθμός (number) ή String
- Numbers (προσοχή στο 0 στην αρχή):
 - 5
 - -245
 - 1.25
 - 7.25e45
 - -12e44
 - -1.2e-25

Strings

- Single quoted (`' '`)
- Double quoted (`" "`)
 - Escape (`\t`, `\n`, `\r`, `\U`, etc)
 - Interpolation

Παραδείγματα

- (`` ``)

printf, sprintf

```
$x=3.7009
```

```
print $x;
```

```
printf ("% .3f\n", $x);
```

```
$y=sprintf ("% .3f", $x);
```

```
print "$y\n";
```

Operators

- Numbers

`+, -, *, /, **, %, <, >, ==, !=`

- Strings

`., x, eq, ne, lt, gt, le, ge`

Προσοχή στην προτεραιότητα

Τελεστές (Operators) όπως και στις υπόλοιπες γλώσσες, ορίζουμε τα σύμβολα ή τις δεσμευμένες εκφράσεις μια γλώσσας που πραγματοποιούν πράξεις ή συγκρίσεις μεταξύ μεταβλητών. Οι βασικοί τελεστές που χρησιμοποιούνται για αριθμητικές πράξεις, φαίνονται στον [Πίνακα 12.1](#) ενώ οι τελεστές που χρησιμοποιούνται για τις πράξεις και τις συγκρίσεις συμβολοσειρών (string), δίνονται στον [Πίνακα 12.2](#).

Αριθμητικοί Τελεστές	Περιγραφή
+	Πρόσθεση
-	Αφαίρεση
*	Πολλαπλασιασμός
/	Διαίρεση
%	Υπόλοιπο Διαίρεσης
**	Δύναμη

Πίνακας 12.1: Αριθμητικοί τελεστές της Perl

Αριθμητικοί Τελεστές	Αλφαριθμητικοί Τελεστές	Σύγκριση
+	.	πρόσθεση
*	x	πολλαπλασιασμός
==	eq	Ισο
!=	ne	Άνισο
>	gt	Μεγαλύτερο από
<	lt	Μικρότερο από
>=	ge	Μεγαλύτερο ή ίσο
<=	le	Μικρότερο ή ίσο

Πίνακας 12.2: Αριθμητικοί τελεστές πράξεων και σύγκρισης και οι αντίστοιχοι τελεστές συμβολοσειρών στην Perl. Ο πολλαπλασιασμός στην περίπτωση των συμβολοσειρών αφορά τον πολλαπλασιασμό αριθμού με συμβολοσειρά.

Variables

- `$variable_name=String or number or expression`
- Ειδικές μεταβλητές (`$_`, `$/`, `$[`, `$1`, κλπ)
- `$number=5;`
- `$name="George";`
- `$exp=3*$number+($number+1);`
- `$a+=5;`
- `$b*=3;`
- `$c .= " ";`
- `++$a; ή $a++;`
- `$a--;`

substr, index

- `$name="Takis";`
- `$x=substr($name, 0, 1);`
- `$y=substr($name, 0, 1, "L");`
- `print "$name\t$x\t$y\n";`
- `$position=index($name, "k");`
- `$position=rindex($name, "k");`

chomp, chop

- `chomp ($x)`
- `chop ($x)`

Interpolation

```
$test='bla bla bla';  
print "this is a $test\n";  
print `this is a $test\n`;  
print `this is a \ $test\n`;  
print "this is a ${test}bla  
  bla\n";
```

Upper Case, Lower Case

```
$bigname="\Uname";
```

```
$name="name"; $bigname="\U$name";
```

```
$capname="\u$name"
```

```
uc, lc, ucfirst, lcfirst
```

chomp	Κόβει το τελικό \n από την μεταβλητή	
chop	Κόβει τον τελευταίο χαρακτήρα από τη μεταβλητή	
substr	\$x = substr(\$name, 0, 1, "L");	Αντικαθιστά στην ακολουθία \$name 1 χαρακτήρα ξεκινώντας από τη θέση την 0 και αποθηκεύει την νέα στην μεταβλητή \$x Η μεταβλητή \$name δεν αλλάζει.
	\$x = substr(\$name, 0, 1);	Αντιγράφει ακολουθία μήκους 1 με αρχική θέση την 0 από την ακολουθία \$name στην μεταβλητή \$x .
index	Index(\$name, "k");	Δίνει την θέση του k στην ακολουθία \$name
rindex	Επιστρέφει ό,τι και η index μετρώντας όμως από το τέλος της ακολουθίας.	

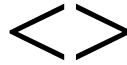
Πίνακας 12.3: Συναρτήσεις διαχείρισης συμβολοσειρών στην Perl

STDIN

```
$a=<STDIN>;
```

```
chomp ($a) ;
```

```
chomp ($a=<STDIN>) ;
```



```
while (<>)  
{  
    print $_;  
}
```

Εκτελέστε:

```
Perl program.pl file
```

Input-Output

- `open FILEHANDLE, "filename";`
- `open IN, "/etc/passwd";`
- `$x=<IN>;`
- `print $x;`
- `close IN;`
- `open OUT, ">(>) tempfile";`
- `print OUT "bla bla bla\n";`

@ARGV

- `perl program.pl file1 file2 ...`
- `file1: $ARGV[0]`
- `file2: $ARGV[1]`
- ...