

## Pokretanje

**Zaglavlje:** `#!/usr/bin/perl`

**Promjena privilegija:** `chmod 775 name.pl`

**Pokretanje:** `./name.pl [Arguments]`

## Znakovni nizovi:

<code>\l</code>	sljedeći znak u malo slovo	<code>\n</code>	novi red
<code>\u</code>	sljedeći znak u veliko slovo	<code>\t</code>	tab
<code>\L</code>	sve do <code>\E</code> u mala slova	<code>\007</code>	oktalna ASCII vrijednost 7
<code>\U</code>	sve do <code>\E</code> u velika slova	<code>\x7f</code>	hexa ASCII vrijednost 7f
<code>\E</code>	kraj <code>\L</code> i <code>\U</code>		

## Skalari (\$)

### Liste i polja (@):

`@lista = (element1, element2, ...)`

`$lista[0] = vrijednost; #Prvi element liste`

`$lista[-1] = vrijednost; #Zadnji element liste`

`$#lista #Indeks zadnjeg elementa`

### Hash (%): - skup parova ključ-vrijednost

`%hash = {"key1" => "val1", "key2" => "val2", ...}`

`$hash{"key"} = "val";`

### Petlje:

**While:** `while (condition) {...}`

**Foreach:** `foreach $varijabla (@polje) {...}`

**For:** `for (initialization; condition; step) {...}`

### Uvjeti:

**If-else:** `if (condition) {...} else if {...} else {...}`

## Potprogrami:

**Definicija:** `sub naziv {...} #bilo gdje u programu`

**Poziv:** `&naziv (argument1, argument2, ...); #povratna vrijednost je rezultat zadnjeg izraza`

## Specijalne varijable:

**<ID>** čita liniju iz IDa (STDIN, STDOUT, STDERR, DATA, ARGV, ARGVOUT) ili sa ulaznih datoteka ili prompta ako je ID blank; piše u ID (ne smije biti blank)

**\$\_** default varijabla – osobine normalnog skalara

**@\_** privatna lista parametara potprograma – osobine normalne liste

**@ARGV** polje argumenata programa – osobine normalnog polja

**\$0** skalar u kojem je ime programa

**#!** poruka sustava o razlogu greške

**\$broj** vrijednost broj-tog podudarenog izraza u zadnjem izvedenom regularnom izrazu (broj > 0)

**Regularni izrazi:** - defaultno se rade nad `$_` ako nije definiran binding (`$varijabla =~ REGULARNI_IZRAZ`)

-nađen izraz unutar zagrada se memorira (do sljedećeg izraza)

.	Podudaranje sa svakim ne-'\n' znakom	*	Prethodni izraz 0 ili više puta
[ZNAKOVI]	Podudaranje sa jednim od ZNAKOVA	+	Prethodni izraz barem jednom
[^ZNAKOVI]	Podudaranje sa jednim znakom koji nije ZNAKOVI	{n}	Prethodni izraz točno n puta
		{n, }	Prethodni izraz n ili više puta
\w	Klasa alfanumeričkih znakova i podvlaka	{n, m}	Prethodni izraz barem n i najviše m puta
\d	Klasa znamenki	\b	Granica riječi
\s	Klasa praznih znakova (\f, \t, \n, \r, )	\B	Ne-granica riječi
\W	[^\w]	^	Početak retka
\D	[^\d]	\$	Kraj retka
\S	[^\s]	/broj	Podudara sa broj-tim nađenim izrazom
?	Prethodni izraz najviše jednom		

## Operatori prema prioritetima:

1	( ) [ ] { }
2	Pointer (->)
3	++ --
4	Potencije (**)
5	\ ! - + ~
6	Bind (=~) !=
7	* / % Ponavljanje niza (x)
8	Binarni ( ) Nadovezivanje (.)
9	<< >>
10	Brojevni (< <= > >= == != <=>=)
11	Znakovni (!t !e gt ge eq ne cmp)
12	&
13	^
14	And (&&)
15	Or (  )
16	Range (..) ...
17	Ternary (?:)
18	= += -= .= *= /= %= i sl.
19	, =>
20	Not
21	And
22	or
23	Xor

<b>matching</b>	vraća da li je u nizu pronašao regularni izraz – opcije /i case insensitive, /s '.' podudara i sa '\n', /x dopušta razbijanje regularnog izraza radi bolje preglednosti, /g vraća sve podudarene podnizove) su kombinabilne
m/izraz/opcije	#bool; promjenjiv graničnik; m se može izostaviti na početku
<b>substitution</b>	vraća da li je u nizu zamijenio regularni izraz sa zamjenom – opcija /g mijenja sve, a ne samo prvo
s/izraz/zamjena/opcije	#bool

<b>transliteration</b>	mijenja podudarne znakova iz liste_znakova sparenih znakom iz liste_zamjene znak po znak i vraća broj zamjena – opcije (/c komplementira listu znakova, /d briše podudarne znakove za koje nema zamjene, /s slijed znakova koji su zamijenjeni istim znakom reducira na jedan znak)
tr/lista_znakova/lista_zamjene/opcije	

### Funkcije:

<b>print</b>	ispisuje jedan ili više znakova nizanih zarezom ili spojenih ' '
print STREAM "tekst" . "spojni tekst", \$skalar, "@interpolirana_lista"; #default STREAM STDOUT	
<b>printf</b>	print sa podrškom za formate (%g autobroj, %d int, %o oktalni, %x hexa, %c char, %s string, %f float)
printf STREAM "tekst %format1 %format2 ...", vartipa1, vartipa2, ...;	
<b>chomp</b>	uklanja \n sa kraja stringa, vraća broj uklonjenih
chomp(\$string);	
<b>qw</b>	zapis liste stringova bez navodnika, nema interpolacije, izbor graničnika
qw/string1 string2 .../	
<b>push</b>	dodaje element na kraj polja
push (@polje, vrijednost);	
<b>pop</b>	skida i vraća element s kraja polja
\$vrijednost = pop (@polje);	
<b>shift</b>	skida i vraća prvi element polja
\$vrijednost = shift (@polje);	
<b>unshift</b>	dodaje element na početak polja
unshift (@polje, vrijednost);	
<b>reverse</b>	vraća obrnuto poredanu ulaznu listu ili hash kojem su zamijenjeni ključ i vrijednost (!!!)
@obrnuta = reverse (@lista); ili %obrnuti = reverse (%hash);	
<b>sort</b>	vraća ulaznu listu poredanu po ASCII vrijednostima
@sortirana = sort (@lista);	
<b>scalar</b>	forsira korištenje skalarnog oblika ulaza
... = scalar (@lista);	
<b>return</b>	instant izlaz iz potprograma
return vrijednost; #vrijednost je skalar ili lista ovisno o implementaciji	
<b>open</b>	otvara file ime u stream pod NAZIV u zadanom smjeru ("ime" ili "<ime" čitanje, ">ime" pisanje, ">>ime" dopisivanje), vraća uspjeh otvaranja
open NAZIV, "smjerime"; ili open NAZIV, "smjer", "ime"; #bool	
<b>close</b>	zatvara stream NAZIV
close NAZIV;	
<b>die</b>	prijevremeni izlaz iz programa uz ispis greške, naziva programa i linije na STDERR
die "Poruka o grešci \$!";	
<b>select</b>	mijenja default STREAM u NAZIV
select NAZIV;	
<b>keys</b>	vraća listu ključeva hash-a, skalarni poziv daje broj elemenata
@k = keys %hash;	
<b>values</b>	vraća listu vrijednosti hash-a, skalarni poziv daje broj elemenata
@v = values %hash;	
<b>each</b>	svakim pozivom vraća sljedeći par ključ-vrijednost
(\$key, \$value) = each %hash;	
<b>exists</b>	provjerava postojanost ključa \$key u hashu
exists \$hash{"\$key"}; #bool	
<b>delete</b>	briše par sa ključem \$key iz hash-a
delete \$hash{"\$key"}; #bool	
<b>split</b>	dijeli ulazni niz na separatorima i vraća listu podnizova
@podnizovi = split /separator/, \$niz; #promjenjivi separator, može biti i reguljni izraz	
<b>join</b>	povezuje ulaznu listu u jedan niz umetanjem spojnice (mora biti string) i vraća dobiveni niz
\$spojeni = join \$spojnica, @podnizovi;	