

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"; #val može biti i neskalar (nesting)`**Petlje:****While:** `while (condition) {...}`**Foreach:** `foreach $varijabla (@polje) {...}`**For:** `for (initialization; condition; step) {...}`**Uvjeti:****If-else:** `if (condition) {...} else if {...} else {...}`**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

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
		{n, m}	Prethodni izraz barem n i najviše m puta
\\w	Klasa alfanumeričkih znakova i podvlaka	\\b	Granica riječi
\\d	Klasa znamenki	\\B	Ne-granica riječi
\\s	Klasa praznih znakova (\\f, \\t, \\n, \\r,)	^	Početak retka
\\W	[^\\w]	\$	Kraj retka
\\D	[^\\d]	/broj	Podudara sa broj-tim nađenim izrazom
\\S	[^\\s]		
?	Prethodni izraz najviše jednom		

matching	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
substitution	vraća da li je u nizu zamijenio regularni izraz sa zamjenom – opcija /g mijenja sve, a ne samo prvo
s/izraz/zamjena/opcije	#bool

transliteration	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:

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