Perl

Nastavení interpretru: `#!/bin/usr/perl` Komentáře: `# toto je komentář`

1. Parametry příkazové řádky

- `-v` – zobrazí verzi a základní info
- `-e` *`příkaz`* – spustí kód daný na příkazové řádce
- `-W` – vypisuje varování
- `-X` – potlačí varování
- `-p -e 's/A/B/i'` – náhrada za program `sed`
- `-n` – skript je uvnitř `while(<>){...}`
- `-F/`*`n`*`/` – nastaví vstupní oddělovač polí pro `-a`
- `-a` – autosplit do pole `@F`
  `perl -F: -ane 'print "@F[0]\n"' /etc/passwd` – výpis loginů
- `perldoc -f` *`funkce`* – dokumentace *funkce*

2. Nastavení vlastností interpreta

- `use utf8;` – podpora UTF-8 pro všechny řetěz. operace
- `use` *`verze;`* – vynutí chování perlu dané verze
- `use warning;` – varování
- `use diagnostic;` – ladící hlášky a další varování
- `use strict;` – vynucení striktního kódu
- `use integer;` – jen celočíselná aritmetika
- `use locale;` – operace dle lokálních zvyklostí
- `no` *`pragma;`* – odteď zakáže dané nastavení

3. Balíky a moduly

- `import modul [ seznam ]` – importuje funkce a proměnné
- `no modul [ seznam ]` – zakáže funkce a proměnné
- `require verze` – vyžaduje k běhu perl min. dané verze
- `require "soubor"` – vloží kód ze souboru (`.pm`)
- `unimport modul [ seznam ]` – zruší předchozí import

4. Řídící výrazy

- `$out=`date`; $out=qx(date);` – ulož výstup sys. utility `date`
- `open(...) or die "Error";` – pokud open 0 provede die
- `print ... if(...);` – výpis pokud platí podmínka
- *`výraz`* `? $true : $false` – ternární výraz
- `{` *`výraz1; výraz2;`* `}` – uzavření části kódu do *bloku*
- **if** (*výraz*) *blok* [**elsif** (*výraz*) *blok* ...] [**else** *blok*]
- **unless** (*výraz*) *blok* [**else** *blok*]
- **while** (*výraz*) *blok* [**continue** *blok*]
- **until** (*výraz*) *blok* [**continue** *blok*]
- **for** ([*výraz*] ; [*výraz*] ; [*výraz*]) *blok*
- **foreach** *proměná* (*seznam*) *blok* [**continue** *blok*]
- *blok* [**continue** *blok*]
- **goto** *label*
- **last** [*label*] – okamžité opustí smyčk, přeskočí **continue** blok
- **next** [*label*] – spustí **continue** blok a provede další iteraci
- **redo** [*label*] – restartuje smyčky bez vyhodnocení podmínky, přeskočí **continue** blok
- **do** *blok* **while** *výraz* ; **do** *blok* **until** *výraz* ;
- **given** (*výraz*) **when** (*výraz*) *blok* **default** *blok*

4.1. Funkce

- `sub funkce [ ( prototyp ) ] [ atributy ]` *`blok`* ;
  - atributy: `method`, `lvalue`
  - prototyp: `$` skalární, `@ %` seznamy, `&` reference funkce, `*` akceptuje jméno, konstantu, skalární výraz, typeglob, reference typeglob, `\$ \@ \% \* \& \[...]` argument musí začínat daným znakem; `_` – vyžaduje skalár, `$_`; `+` – vyžaduje pole, hash nebo referenci; `;` – další argumenty volitelné;
- `funkce(); &funkce(); &funkce; funkce` – volání
- `defined &funkce` – testuje zda je funkce definována
- `return [` *`výraz`* `]` – návrat z funkce
- `$_[0], $_[1], $_[2], $_[3],...` – přístup k argumentům
- `sub add($$) { $a=shift; $b=shift; return $a+$b;}`

5. Proměnné

- `[ my ] $name =` *`hodnota`* – uloží hodnotu do `$name`
- `defined` *`jmého`* – test existence (nejen) proměnné
- `$nic = undef` – přiřadí nedefinovanou hodnotu
- `$pointer = \$name` – vytvoří ukazatel na `$name`
- `$$pointer` – hodnota na adrese `$pointer`
- `use constant PI => 3.14159;` – definice konstanty
- Číselné formáty:
  - `2006` – desítkový int.;
  - `0775` – osmičkově;
  - `0xBABE` – hexa;
  - `0b101010` – dvojkově;
  - `3.14, 314e-2` – desetinná čísla;
  - `0b0101_1010, 123_456` – oddělovač zlepší čitelnost;
- `scalar` *`výraz`* – vynutí skalární význam výrazu
- `__LINE__`, `__FILE__`, `__PACKAGE__` – aktuální řádek, skript, balík

5.1. Speciální proměnné a pole

- `use English;` – pro alter. jméno za lomítkem
- `$_` / `$ARG` – základní arg. mnoha funkcí
- `@_` / `@ARG` – seznam arg. mnoha funkcí
- `$.` / `$NR` – číslo řádku čteného souboru
- `$,` / `$OFS` – výstupní oddělovač prvků pole
- `$/` / `$RS` – vstupní oddělovač, obvykle `\n`
- `$\` / `$ORS` – výstupní oddělovač pro print (prázdný)
- `$!` / `$ERRNO` – chyba posledního příkazu
- `$0` / `$PROGRAM_NAME` – jméno skriptu
- `$$` / `$PID` – číslo procesu běžícího skriptu
- `$?` / `$CHILD_ERROR` – návrat. hodnota `` `...` ``
- `$1`...`$9` – nalezené podvýrazy regexpu
- `@ARGV` – pole s parametry z příkazové řádky
- `$<` / `$UID`, `$>` / `$EUID`, `$(` / `$GID`, `$)` / `$EGID`
- `%ENV` – hash s parametry prostředí
- `%SIG` – obsahuje obsluhu signálů
- `$^O` / `$OSNAME` – jméno operačního systému

5.2. Pole

- `@pole = (hodnota1, hodnota2,...);` – definice pole
- `@cisla=(0 .. 10);` – zadání číselného rozsahu
- `@znaky=('A' .. 'Z', 'a' .. 'z');` – znakový rozsah
- `$pole[1]` – přístup k 2. prvku
- `@pole[3,4,5]; @pole[3..5]` – výřez pole
- `$pole[-1]` – přístup k poslednímu prvku
- `@pole = ([1, 2], [3, 4]);` – dvourozměrné pole 2×2
- `$pole[1][0]` – přístup k prvku „3“
- `@pole` – v skalárním kontextu vrací počet prvků
- `$#pole` – poslední index `@pole`
- `$p = \@pole;` – vytvoří ukazatel na `@pole`
- `$$p[1], $p->[1]` – přístup k 2. prvku
- `exists` *`prvek`* – pravda pokud prvek existuje
- `pop` *`@pole`* – vyjme poslední prvek
- `push` *`@pole,list`* – přidá nový prvek na konec
- `shift` *`@pole`* – vyjme první prvek
- `unshift` *`@pole`* – vloží nový prvek na začátek
- `sort` *`@pole`* – třídí prvky pole
- `reverse` *`@pole`* – otočí pořadí prvků
- `wantarray` – pravda pro pole, nepravda pro skaláry
- `join` *`znak,@pole`* – spojí prvky do řetězce a oddělí znakem
- `@prvky=grep( /.../, @pole);` – nové pole s prvky dle regexpu
- `$prvky=grep( /.../, @pole);` – počet vyhovující výrazu

5.3. Asociativní pole – hashe

- `%pole=( "klíč1" => "hodnota", "klíč2" => "hodnota", ...)`
- `%pole=( "klíč1", "hodnota", "klíč2", "hodnota",...)`
- `$pole{'klíč1'};` – přístup k položce
- `$pointer = \%hash;` – vytvoří ukazatel na `%hash`
- `$$p['key'], $p->['key']` – přístup k prvku
- `@pole['a','b']` – výřez
- `keys` *`@pole`* – vrací všechny klíče
- `values` *`@pole`* – vrací všechny hodnoty
- `each` *`@pole`* – vrací pár klíč-hodnota
- `while ((@key,$val)=(each %pole)){...}`
- `foreach $key (sort keys %pole){...}` – seřaď podle klíčů
- `delete` *`prvek`* – vymaže pár klíč-hodnota

5.4. Řetězce

- `$string = "Ahoj!\n"; $string = qq/Ahoj\n/;`
- `$string = 'Hello'; $string = q/Hello/;`
- `$str = "řetěz" x` *`n`* = zopakuje řetěz $n\times$
- `q#Hello# q{Hello} q[Hello] q(Hello)`
- `@den = qw( Po Út St Čt Pá So Ne );`
- `@den = ( 'Po', 'Út', 'St', 'Čt', 'Pá', 'So', 'Ne' );`
- Zadání znaku: `\53`, `\053`, `\o{53}` – osmičkově; `\cC` – řídící znak Ctrl-C; `\xff` – hexa; `\N{GREEK SMALL LETTER ALPHA}`, `\N{U+03B1}`, `\x{03b1}` – Unicode znak „α“
- Formátovací sekvence:
  - `\E` – ukončuje `\L`, `\Q`, `\U`;
  - `\l` – následující převede na malá písmena;
  - `\L` – následující na malé, až po `\E`;
  - `\u` – následující převede na velká písmena;
  - `\U` – následující na velká, až po `\E`;
  - `\Q` – zaeskejpuje speciální znaky, až po `\E`
- `chomp` – odstraní „`\n` “ z konce řetězce
- `chop` – odstraní poslední znak z řetězce
- `eval` – provede zadaný kód
- `index str, substr [, offset ]` – vrátí první pozici podřetězce, od offsetu, nenalezeno vrací -1
- `rindex str, substr [, offset ]` – vrátí poslední pozici
- `lc, uc` – převede řetězec na malá / velká písmena
- `lcfirst, ucfirst` – řetězec s prvním pís. malým / velkým
- `length` – vrátí počet znaků
- `quotemeta` – zaeskejpuje speciální znaky
- `substr expr, offset [, len [, newtext ]]` – podřetězec
- `$str = sprintf(...)` – formátovaný zápis řetězce
- `@pole = split(`*`oddělovač`*`, $str);` – rozdělení do pole

5.4.1. Regulární výrazy

- `"výraz"`, `qr/výrax/`, `/výraz/`, `m/výraz/`
- `s/výraz/náhrada/`*`m`* – nahrazení
- `tr/.../.../`*`m`*, `y/.../.../`*`m`* – náhrada znaků
- Modifikátory *m*:
  - `i` – ignoruje velikost písmen;
  - `m` – víceřádkový mód (`^` a `$` = `\n`);
  - `g` – hledá všechny výrazy;
  - `s` – jednořádkový mód, „.“ zahrnuje `\n`;
  - `x` – povolí bílé znaky a komentáře;
- `$&` / `$MATCH` – zachycený řetězec
- `$`` / `$PREMATCH` – řetězec před `$&`
- `$'` / `$POSTMATCH` – řetězec za `$&`
- `.` (tečka) – libovolný znak
- `[...]`/`[^...]` – množina/negovaná množina znaků
- `^`/`$` – začátek/konec řádku nebo řetězce
- `( )` – seskupení výrazu
- `|` – nebo, alternativní výrazy
- `\`*`n`* – *n*-tý zachycený podvýraz

Meta	POSIX	Unicode	Množina	Popis
\d	[:digit:]	\p{IsDigit}	[0-9]	Číslice
\D		\P{IsDigit}	[^0-9]	Cokoliv mimo číslici
\s	[:space:]	\p{IsSpace}	[\t\n\r\f]	Bílý znak
\S		\P{IsSpace}	[^ \t\n\r\f]	Cokoliv mimo bílého znaku
\w	[:word:]	\p{IsWord}	[a-zA-Z0-9_]	Znaky identifikátorů
\W		\P{IsWord}	[^a-zA-Z0-9_]	Cokoliv mimo znaků identifikátorů
\b				hranice slova, opakem je \B
	[:alnum:]	\p{PosixAlnum}	[A-Za-z0-9]	Alfanumerické znaky
	[:xdigit:]	\p{PosixXDigit}	[A-Fa-f0-9]	Hexadecimální čísla
	[:print:]	\p{PosixPrint}	[\x20-\x7E]	Tisknutelné znaky
	[:alpha:]	\p{PosixAlnum}	[A-Za-z]	Abecední znaky

- Kvantifikátory:

Stand.	Líný	Rozsah
*	*?	0 a víc
+	+?	1 a víc
?	??	0 nebo 1 položka
{n}	{n}?	přesně $n\times$
{n,}	{n,}?	minimálně $n\times$
{n,m}	{n,m}?	min. $n\times$ a max. $m\times$.

Příklady:
- `a*b` – aab abc bc
- `a?b` – ab abc bc
- `a+b` – aab abc bc
- `[els]{1,3}` – all in the darkness
- `Pe(t|p)a` – Pera Peta Pepa
- `".*"` – "Ahoj" "Hello"
- `"[^"]*"` – "Ahoj" "Hello"

6. Práce se soubory

- Předdefinované ovladače: `STDIN, STDOUT, STDERR, DATA, ARGV`
- `open(FH, "<název"); open(FH, "název");` – otevře pro čtení; `>` – zápis; `>>` – připojení; `+<` – nejdřív čtení, pak zápis; `+>` – nejdřív zápis, pak čtení; `"příkaz |", "| příkaz"` – čtení, zápis z roury; `"|-", "-|"` – roura do forknutého příkazu; `>&FH`, `<&FH` – duplikace FH pro čtení, zápis
- `open(FH, "soubor") or die "Nelze otevřít $!\n";`
- `print FH "Zápis do souboru\n";`
- `close(FH);` – zavře soubor
- `while(<FH>){ print; }` – vypíše soubor po řádcích
- `@lines = <FH>;` – načte soubor do pole
- `<*>`, `<*.c>`, `<${pattern}>` – soubory v adresáři
- `stat` *`soubor`* – vrací seznam informací: `0/dev, 1/ino, 2/mode, 3/nlink, 4/uid , 5/gid, 6/rdev, 7/size, 8/atime, 9/mtime, 10/ctime, 11/blksize, 12/blocks`
- Operátory pro testování souborů:
  - `-r -w -x` – pro efektivní, `-R -W -X` reálné UID/GID
  - `-e -z` – existuje/nulová velikost
  - `-s` – vrací nenulovou velikost
  - `-f -d` – soubor/adresář
  - `-l -S -p` – symlink/socket/FIFO
  - `-T -B` – textový/binární
  - `-u -g -k` – setuid/setgid/sticky bit

6.1. Zpracování vstupu po řádcích

7. Operátory

- Porovnávání – číselné / řetězcové:
  - `==` / `eq` – shoda
  - `!=` / `ne` – nerovná se
  - `>` / `gt` – větší než
  - `<` / `lt` – menší než
  - `>=` / `ge` – větší nebo rovno
  - `<=` / `le` – menší nebo rovno
  - `<=>` / `cmp` – vrací –1, 0 nebo 1 pro menší, rovno nebo větší
- Další typy operátorů:
  - pre/postfixové: `$i++`, `++$i`, `$i--`, `--$i`,
  - aritmetické: `+`, `-`; `**` umocňování, `*`, `/`, `%` zbytek po dělení;
  - logické: `!/neg`, `&&/and`, `||/or`;
  - binární: `~`, `&`, `|`, `^` xor; `<<`, `>>` posuvy;
  - řetězcové: `.`, `,` spojení; `x` opakování
  - přiřazení: `= **= += *= &= <<= &&= -= /= |= >>= ||= .= %= ^= x=;`

8. Aritmetické funkce

- `abs` – absolutní hodnota
- `atan2 x,y;` – arkustanges y / x v $(-\frac{\pi}{2}, \frac{\pi}{2})$
- `cos, sin` – vstup v radiánech
- `exp` – mocnina $e = 2{,}718281\ldots$
- `int` – celá část čísla
- `log` – přirozený logaritmus
- `rand [` *`n`* `]` – náhodné desetiné číslo mezi 0 a 1 (nebo *n*)
- `sqrt` – odmocnina
- `srand` – nastavení pro generátor náhodných čísel
- `time` – aktuální unixový čas

9. Konverzní funkce

- `chr` *`výraz`* – vrátí znak reprezentovaný hodnotou
- `hex` *`výraz`* – vrátí desítkově ze šestnáctkové
- `localtime`, `gmtime` – převod unixového času na lokální nebo UTC, vrací seznam s daty: `0/sec`, `1/min`, `2/hour`, `3/mday`, `4/mon` (0 = leden), `5/year` (od roku 1900), `6/wday` (0 = neděle), `7/yday` (0 = 1. leden), `8/isdst`
- `oct` *`výraz`* – vrátí desítkově z osmičkové
- `ord` *`výraz`* – ASCII hodnota prvního znaku výrazu
- `pack` *`template,list`* – strukturovaná konverze
- `unpack` *`template,výraz`* – strukturovaná konverze

10. Výstup na obrazovku

- `print` *`value1, value2, value3;`*
- Dlouhý výstup
  `print <<"EOF";`
  `Řetězec na`
  `víc řádků`
  `EOF`
- `printf("formát", data...);`
  - `%b` – dvojkové
  - `%d`, `%i`, `%ld`, `%D` – znaménkové, long
  - `%e`, `%E` – desetinná č. ve vědecké notaci
  - `%f`, `%F`, `%g`, `%G` – desetinná čísla
  - `%o`, `%lo`, `%O` – osmičkové, long
  - `%p` – adresa, `%s` – řetězec, `%c` – znak
  - `%u`, `%lu`, `%U` – neznaménkové, long
  - `%x`, `%X`, `%lx` – šestnáctkové
  - `%%` – vypiš procento
  - `%-` – zarovnání vlevo
  - `%#` – doplní prefix `0x`, `0b`, `0` podle č. typu
  - `%`n – šířka sloupce n
  - `%.`n – n desetinných míst
  - `%0` – doplní hodnotu o předchozí nuly
- `use Data::Dumper; print Dumper(`*`proměná`*`)` – debugovací výpis strukturované proměnné