Perl áttekintés

Szigeti Szabolcs
szigi@ik.bme.hu

A Perl nyelv

- **Practical Extraction and Report Language**
- **Pathologically Eclectic Rubbish Lister**
- **Larry Wall – 1987**
- **Célok:**
  - **Teljes programozási nyelv**
  - **Könnyen, gyorsan használható**
  - **Erős szövegfeldolgozási képesség**

A Perl nyelv

- **Procedurális nyelv**
- **Sok hasznos megoldás, különböző nyelvekből:**
  - **Shell**
  - **C**
  - **Lisp**
  - **Regexp**
  - **....**
- **Interpreter (majdnem)**
- **Automatikus típusok, memória menedzsment**

A Perl-ről

- **TMTOWTDI – There's more than one way to do it!**
- **Svájci-láncfűrész**
- **Senki nem ír nulláról Perl programot, hanem egy létezőt kalapál át!**
- **Lustaság, türelmetlenség, önhittség!**

Hello World!

```
#!/usr/bin/perl
my $napszak;
print "Mi van most?";
$napszak=<>;
chomp $napszak;
print "Jo ${napszak}t!\n";
```

Típusok

- **Skalárok**
  `$skalár_változó`
- **Tömbök**
  `@tömb_változó`
- **Asszociatív tömbök**
  `%assoc_változó`

Skalárok

- **stringek (‘, vagy ” között)**
  - **‘ – karaktersorozat, escape: **
  - **” – karaktersorozat, de interpolálás (változók, spec karakterek lehetnek)**
  - **Tetszőleges méretű**
- **numerikus típus**
  - **egészek**
  - **lebegőpontosok**

Skalárok (2)

- **string és numerikus között automatikus konverzió**

```
$num=5;
$szoveg='ize';
$valami="Az $szoveg az $num darab";
```

Tömbök

- **Lista: `(1, 2, 5, "lista", "eleme")`**
- **Lista: `qw/ez is egy lista/`**
- **Lista: `(5 .. 28), ('B' .. 'F')`**
- **Tömb is lista:**

```
@tömb=(1,6,8);
```

- **`$tömb[2],@tömb[2 .. 6], @tömb[2,5]`**
- **`$#tömb` – utolsó elem indexe**

Asszociatív tömbök

- **%assoc**
- **$assoc{'valami'}='masvalami';**
- **@idebele=%assoc – kulcs – érték párok**

Assoc példa

```
#!/usr/bin/perl

my (%leltar,$aru,$darab);
do {
        print "Arucikk:";
        $aru=<>;
        chomp $aru;
        print "Darab:";
        $darab=<>;
        chomp $darab;
        $leltar{$aru}+=$darab;
        } while ($darab);

while ( ($aru,$darab) = each(%leltar)) {
        print "$aru --- $darab db\n";
        };
```

Operátorok

- **C-hez hasonló**
- **+ - * / % ** ++ -- +=**
- **< <= > >= == != <=>**
- **lt le gt ge eq ne cmp**
- **String: . x**

Vezérlés

- **Blokkok: { …… }**
  - **bárhol lehet**
  - **lokális változó (my)**

```
if (expr) { …
} elsif (expr) { …
} else { … }
```

(lehet unless az if helyett)

```
while (expr) { … }
```

Vezérlés 2

```
do { … } while (expr);

for (init; expr; inc) { … }

foreach $valtozo (@lista) { … }
```

Nincs switch!

```
 SWITCH: {
              if (/^abc/) { $abc = 1; last SWITCH; }
              if (/^def/) { $def = 1; last SWITCH; }
              if (/^xyz/) { $xyz = 1; last SWITCH; }
              $nothing = 1;
          }
```

vagy:

```
SWITCH: {
              $abc = 1, last SWITCH  if /^abc/;
              $def = 1, last SWITCH  if /^def/;
              $xyz = 1, last SWITCH  if /^xyz/;
              $nothing = 1;
          }
```

Regexp

```
• if ($vizsgált_string =~ /^abc*(1|2)/) { …}

#!/usr/bin/perl

while (<>) {
        chomp;
        print "Az input: $_\n";

        /^a/ && do {print "A-val kezdodik\n";};
        /bb/ && do {print "van benne 2 b\n";};
        /x[1-9]$/ && do {print "a vege x-
  szam\n";};
        }
```

Szubrutinok

```
sub EzEgySzubrutin {
…
return "valami";}

print &EzEgySzubrutin;
```

@_ az argumentumok listája

Szubrutinok 2

```
sub ArgSub {
my ($arg1,$arg2,$arg3) = @_;
…

&ArgSub("na","mi","van");
```

Referencia

- **$skalref = \$valami**
- **$tombref= \@tomb**
- **$subref=\&sub**

- **$$skalref**
- **&$subref(1,2,3)**

Modul, package és namespace

- **Kész kódokat modulokba helyezhetünk:**
  - **`require Modulenév;` (futásidőben)**
  - **`use Modulnév;`**
- **A modul package-ekből áll**
- **Minden package-nek saját névtere van:**
  - **`$Package_név::változó`**
- **Egymásba ágyazott package-ek lehetnek**

Objektum orientáltság

- **Erős kötődés a package-ekhez**
  - **`Osztály->metódus(@args);`**
  - **`Osztály::metódus("Osztály", @args);`**

```
{package Alakzat;
   sub oldalak {
        my $osztaly=shift;
        print "a $osztaly ",$osztaly->noldal," oldalu\n"
   }
   sub konstr {
        my $osztaly=shift;
        my $nev=shift;
        bless \$nev, $osztaly;
         }
}
```

Objektum orientáltság

```
{package Negyzet;
  @ISA = qw(Alakzat);
  sub noldal { 4 };

}

my $alak = Negyzet->konstr("kicsinegyzet");
$alak->oldalak();
```