



Introdução à linguagem Perl



Nelson Corrêa de Toledo Ferraz
<nferraz@gnubis.com.br>



Perl



Practical Extraction and Report Language



Larry Wall





Alguns Exemplos



```
print "Hello, world!\n";
```

```
perl -nle 'print if $_ eq reverse' /usr/dict/words
```

```
($a)=@ARGV;$a||=10;$b=40;$d=1;$e=($b-$a)*$f=($c/2);  
$g=0;$j=5;$e=1;$h=$f;foreach(0..$a){$i[$f]=(100/$c)*$_}while($j){@k=();if  
($h<$f){$h=$f;$j--;$e=-$e}elseif($h>$f){$h=$f+$a-1;$j--;$e=-$e}$h  
=1;@l=(' 'x$a;$l[$h-($f)]=&d;push @k,'*'x($f+1);push @k,@l;push @k,'*'x  
($c-$f+1);push @k,' ',&j;print join(' ',@k);if(rand(100)<=$i[$f]){&f--;}  
else{&f++;&g++;&y=&x='';vec($x,f%ENO,1)=1;if(select  
($y=&x,undef,undef,.1)){until($s=spread($x,&b,1)){}&e=-&e;}else  
{print"\n"}&h+=&e}print"&g\n";
```



Variáveis



- Escalares
- Arrays
- Hashes



Escalares



```
$nome = "Larry Wall";
```

```
$numero = 42;
```

```
print "O número preferido de $nome é $numero.";
```

Arrays



```
@nomes = ("Larry", "Guido", "Zeev", "Andy");
```

```
@numeros = (1, 2, 2.5, 2.75);
```

```
@x = ("a", 2, "c", 10);
```



Arrays



```
@nomes = (  
    "Larry",  
    "Guido",  
    "Zeev",  
    "Andy"  
);
```



Laço foreach



```
@nomes = ("Larry", "Guido", "Zeev", "Andy");  
  
foreach (@nomes) {  
    print;  
}
```



TIMTOWTDI



“There Is More Than One Way To Do It”



DWIM



“Do What I Mean”

Perfeição?





Hashes



```
%linguagem = (  
    "Larry" => "Perl",  
    "Guido" => "Python",  
    "Zeev and Andy" => "PHP"  
);
```

```
print $linguagem{"Larry"};
```

```
@list = keys (%linguagem);
```

Hashes



```
%linguagem = (  
    "Larry" => "Perl",  
    "Guido" => "Python",  
    "Zeev and Andy" => "PHP"  
);  
  
foreach (keys %linguagem) {  
    print "$_ criou $linguagem{$_}\n";  
}
```



Abrindo um arquivo-texto



```
if (open(FILE, "filename.txt") {  
    # ...  
} else {  
    print "Erro!";  
}
```

```
open (FILE, "filename.txt") or die "Erro!";  
#...
```



Lendo o conteúdo de um arquivo



```
open (FILE,"filename.txt") or die "Erro!";  
  
while ($linha = <FILE>) {  
    # ...  
}  
  
close FILE;
```

```
open (FILE,"filename.txt") or die "Erro!";  
  
@conteudo = <FILE>;  
  
close FILE;
```



Lendo o conteúdo de um arquivo



```
while (<>) {  
    # ...  
}
```



Expressões Regulares



Expressões Regulares



- Úteis para manipular textos
 - Localizar strings
 - Substituições



```
l(a  
le  
af  
fa  
ll  
s)  
one  
l  
  
iness
```

Exemplos



```
"dilbert" =~ /bert/; # bate!  
"dogbert" =~ /bert/; # bate!
```

```
"dilbert" =~ /^bert/; # não bate  
"dogbert" =~ /^bert/; # não bate
```

```
"dilbert" =~ /bert$/; # bate!  
"dogbert" =~ /bert$/; # bate!
```

```
if ($nome =~ /bert/) {  
    ...  
}
```

Mais exemplos



```
/dilbert/  
/(dil|dog)bert/  
  
/...bert/  
  
/.{3}bert/  
  
/.+bert/
```

Ainda mais exemplos



```
if ($clock =~ /(\d\d):(\d\d):(\d\d)/) {  
    $hora = $1;  
    $min  = $2;  
    $sec  = $3;  
}
```

```
If ($nome_completo =~ /(\w+) (\w+)/) {  
    $nome = $1;  
    $sobrenome = $2;  
}
```



next if...



```
while (<>) {  
    next if /^#/g;  
    ...  
}
```

Substituições



s/foo/bar/

```
package Portugues;  
  
use Filter::Simple;  
  
FILTER {  
  
    s/para cada /foreach /g;  
    s/escrava /print /g;  
    s/se /if /g;  
  
}
```

```
use Portugues;  
  
escrava "Olá, mundo!\n";
```



Boas Práticas



- use strict
- use warnings
- Comentários
- Maiúsculas e minúsculas
- Espaçamento vertical e horizontal
- Procure a forma mais legível



use strict



```
use strict;
```

```
$valor = 123;      # Erro
```

```
my $valor = 123;   # OK
```



use warnings



Alerta sobre possíveis erros



Comentários



Comente seu código!



Maiúsculas e minúsculas



```
$ALL_CAPS_HERE      # constantes  
$Some_Caps_Here     # variáveis globais  
$no_caps_here       # variáveis locais
```

Espaçamento



```
$IDX = $ST_MTIME;  
$IDX = $ST_ATIME   if $opt_u;  
$IDX = $ST_CTIME   if $opt_c;  
$IDX = $ST_SIZE     if $opt_s;  
  
mkdir $tmpdir, 0700 or die "can't mkdir $tmpdir: $!";  
chdir($tmpdir)      or die "can't chdir $tmpdir: $!";  
mkdir 'tmp', 0777 or die "can't mkdir $tmpdir/tmp: $!";
```



Procure a forma mais legível



```
die "Can't open $foo: $!" unless open(FOO,$foo); # ?  
open (FILE, $foo) or die "Can't open $foo: $!"; # OK
```

```
print "Starting analysis\n" if $verbose; # OK  
$verbose && print "Starting analysis\n"; # ?
```



Perguntas?