

Guide de Référence Pratique de Perl

Jean-Michel Richer (2008)

Caractères

`\n` passage à la ligne (LF)
`\r` retour chariot (CR)
`\t` tabulation (HT)
`\f` form feed (FF)
`\b` backspace (BS)
`\a` alerte (BEL)
`\e` escape (ESC)
`\u` prochain caractère en majuscule
`\U` prochains caractères en majuscule
`\l` prochain caractère en minuscule
`\L` prochains caractères en minuscule
`\Q` ajoute `\` devant les prochains car.
`\E` fin de `\U \L` ou `\Q`

Opérateurs de comparaison

`== eq` égalité
`!= ne` différence
`> gt` supérieur
`>= ge` supérieur ou égal
`< lt` inférieur
`<= le` inférieur ou égal

Opérateurs logiques

`&& and` et logique
`|| or` ou logique
`xor` ou exclusif
`! not` négation

Perl et Unix

première ligne:
`#!/usr/bin/perl`
donner les droits en exécution
`chmod +x fichier.pl`

Liens hypertextes

www.perl.org
cpan.perl.org

Opérateurs arithmétiques

`+` addition
`-` soustraction
`*` multiplication
`/` division
`%` modulo (reste div. entière)
`**` puissance
`++` incrémentation
`--` décrémentatation
`>>` décalage à droite
`<<` décalage à gauche

Variables scalaires

introduites par `$`
`$entier = 1;`
`$reel = 12.34E+5`
`$chaine = "le chat";`

tableaux

introduits par `@`
`@tab1 = (1,2,3,4);`
`@tab2 = (1..1000);`
`@tab3 = ('le', 'chat', 'mange', 'la', 'souris');`
`@tab4 = qw/le chat mange la souris/;`
on manipule un élément grâce à `$` et `[]`
`print $tab1[0];`

Mapping (tableau associatif)

introduits par `%`
`%map1 = (1 => 'jean', 2 => 'paul');`
`%map2 = ('anglais' => 14, 'français'=>15);`
on manipule un élément grâce à `$` et `{}`
`print $map1{0};`

Opérateurs des mappings

`keys` liste des clés
`values` liste des valeurs
`each` énumération (clé,valeur) avec `while`

sprintf

`%d` entier
`%f` réel
`%s` chaîne
`sprintf("%04d %.2f %20s", $entier, $reel, $chaine);`

Opérateurs des chaînes de car.

`.` concaténation
`x` duplication
`+` addition (si nombres)
`chop` supprime le dernier caractère
`chomp` supprime `\n`
`length` longueur de la chaîne
`reverse` inverse les caractères
`substr` sous-chaîne
`lc` convertir en minuscule
`uc` convertir en majuscule

Opérateurs des tableaux

`scalar` nombre d'éléments
`push` ajouter à la fin
`unshift` ajouter au début
`pop` supprimer le dernier
`shift` supprimer le premier
`split` convertir une chaîne en tableau
`join` convertir un tableau en chaîne
`sort` trier



Département Informatique
UFR Sciences
2 Boulevard Lavoisier
49045 Angers Cedex 01

Structures de contrôle

```
if (condition1) {
  instructions1;
} elsif (condition2) {
  instructions2;
} else {
  instructions3;
}
```

```
for (init;cond;incr) {
  instructions;
}
```

```
while (condition) {
  instructions;
}
```

```
foreach $var (liste) {
  instructions;
}
```

Expressions régulières (ER)

Recherche

m/cherche/modif

Substitution

s/cherche/remplace/modif

Remplacement

tr/set1/set2/modif

ER - Modificateurs

/i pas de différence maj/minuscule
/s . peut représenter \n
/m ^ et \$ entre \n
/x ignore les espaces
/o compile une seule fois
/g globalement
/cg continue la recherche si pas trouvé

Répétiteurs

* 0 ou plusieurs fois
+ 1 ou plusieurs fois
? 0 ou 1 fois
{n} exactement n fois
{n,} au moins n fois
{n,m} au moins n au plus m fois

Métacaractères

. n'importe quel caractère sauf \n
| alternative
() regroupement
[] classe de caractères
^ début de chaîne ou de ligne
\$ fin de chaîne ou de ligne

Classes de caractères

\d [0-9] chiffre
\D [^0-9] non chiffre
\s [\t\n\r\f] blanc
\S [^ \t\n\r\f] non blanc
\w [0-9a-zA-Z] alphanumérique
\W [^0-9a-zA-Z] non alphanumérique

Exemples

Afficher les entiers de 1 à 1000

```
foreach $i (1..1000) {
  print "$i\n";
}
```

Somme des entiers impairs de 1 à 1000

```
$somme=0;
foreach $i (1..1000) {
  if (($i % 2) != 0) {
    $somme = $somme + $i;
  }
}
ou
for ($i=1;$i<=1000;$i=$i+2) { ...
}
```

Exemples

Moyenne des notes

```
%notes=(francais=>12, anglais => 14,
  maths => 17);
$somme=0;
while (($cle,$note)=each(%notes)) {
  $somme += $note;
}
$moyenne=$somme / scalar keys %notes;
```

Ouverture et lecture d'un fichier

```
$nom="file.txt";
open(FICHIER,"<$nom") or
  die("impossible");
```

```
while ($ligne=<FICHIER>) { ... }
ou
@lignes=<FICHIER>;
```

Recherche des nombres premiers

```
Sub est_premier {
  my ($n)=@_;
  $est_premier=1;
  for (my $i=2;$i<$n;++$i) {
    if (($n % $i)==0) {
      $est_premier=0;
    }
  }
  return $est_premier;
}
```

@premiers=();

```
foreach $k (1..999) {
  if (est_premier($k)) {
    print $n est
  }
}
print join(", ",@premiers);
```