

- **Instruction**

```
print "pomme";
```

- **Types de valeur**

```
chaîne de caractères : "on"
```

```
entier : 29
```

```
booléens : True/False
```

- **Caractères séparateurs**

```
espace : (barre espace)
```

```
tabulation : \t
```

```
saut de ligne : \n
```

- **Variable**

```
créer : my $var ;
```

```
assigner une valeur : $var = "je";
```

```
les deux en un : my $var = "je";
```

- **Fonction d'affichage**

```
print "ma variable vaut $var\n";
```

- **Commentaires**

```
## Sur une ligne précédée de deux #
```

```
my $var; # En fin de ligne un seul
```

- **Opérations sur les entiers**

```
addition 2+3
```

```
soustraction 3-1
```

```
multiplication 2*3
```

```
division 2/3
```

```
auto-incrémentation ++
```

```
auto-décrémentation --
```

- **Opération sur les chaînes**

```
concaténation $var."e"
```

utile seulement quand il n'y a aucun caractère séparateur entre un nom de variable et une chaîne que l'on veut ajouter.

```
Sinon : "$var\nPas besoin!";
```

- **Supprimer saut de ligne final**

```
chomp $var;
```

- **Ouvrir un fichier et le parcourir**

```
open(IN, "<:utf-8", "fichier.txt") or die "impossible  
d'ouvrir le fichier : $!";
```

```
while my $ligne = <IN>
```

```
{
```

```
instruction à suivre pour chaque ligne;
```

```
}
```

## COMPTER LES LIGNES D'UN FICHIER

```
## Avant tout, il me faut quelque chose pour stocker  
l'information : je crée une variable.
```

```
my $compt;
```

```
## Avant de compter quoi que ce soit, je dois m'assurer que je  
part de 0 : j'assigne la valeur 0 à la variable
```

```
$compt = 0;
```

```
## Maintenant, je peux ouvrir le fichier
```

```
open(IN, "<:utf8", "texte.txt") or die "Impossible d'ouvrir le  
fichier : $!";
```

```
## Ensuite je parcours le fichier, une ligne à la fois
```

```
while (my $ligne = <IN>)
```

```
{
```

```
## À chaque ligne parcourue, je dois compter 1 de plus  
$compt++; # J'incrémente la variable $compt
```

```
} # Je sors de la boucle : j'ai parcouru toutes les lignes
```

```
## J'affiche le contenu de ma variable pour savoir combien de  
lignes j'ai parcourues
```

```
print "Le fichier texte.txt contient $compt lignes\n";
```

- **Modifier la casse**

tout en minuscules : `lc $var;`

tout en majuscule : `uc $var;`

1ère lettre minuscule : `lcfirst $var;`

1ère lettre majuscule : `ucfirst $var;`

- **Mesurer la longueur**

\$longueur = `length $var;`

---

## FONCTIONS DE REMPLACEMENT

- **Translitération**

remplacement à l'échelle du caractère,

toutes les occurrences à la fois

`$var = ~ tr///;`

Un seul caractère par un autre :

`$var = ~ tr/a/e/;`

Plusieurs caractères par un seul :

`$var = ~ tr/AÀÂaâ/e/;`

Plusieurs caractères deux à deux :

`$var = ~ tr/abc/efg/;`

- **Substitution**

remplacement à l'échelle de la chaîne de caractères, par défaut seulement la première occurrence

`$var = ~ s///;`

Option pour toutes les occurrences :

`$var = ~ s//g;`

Option pour insensible à la casse :

`$var = ~ s//i;`

Substituer un chaîne par une chaîne

`$var = ~ s/lui/elle/;`

Substituer un motif par une chaîne

`$var = ~ s/[Ll].i/elle/;`

Substituer un motif par rien

`$var = ~ s/[Ll].i//;`

## EXPRESSIONS CONDITIONNELLES

- **SI (CONDITION) {ALORS...}**

`if ($var =~ /^[A-Z]/)`

`{`

`print "$var\n";`

`}`

- **SI (CONDITION) {ALORS...}**

**SINON SI (CONDITION) {ALORS...}**

**SINON {ALORS...}**

`if ($var =~ /^A/)`

`{`

`print "la ligne débute par A\n";`

`}`

`elsif ($var =~ /^B/)`

`{`

`print "la ligne débute par B\n";`

`}`

`else`

`{`

`print "la ligne ne débute ni par A ni`

`par B\n";`

`}`