

# Formulaire Script

## ❖ Les chaînes de caractères

### 1. Généralités

- ✓ Les chaînes littérales : ''
- ✓ Les chaînes substituées : « »
- ✓ Les chaînes exécutées : ``
- ✓ Borner la variable : \${mvariable}

### 2. Opérations sur les chaînes

- ✓ Basename : enlève le chemin on l'extension (garde le nom du fichier)
- ✓ Dirname : garde le chemin et enlève le nom du fichier

Expression	Résultat
<code>\${variable#modèle}</code>	Suppression de la plus courte sous-chaîne à gauche
<code>\${variable##modèle}</code>	Suppression de la plus longue sous-chaîne à gauche
<code>\${variable%modèle}</code>	Suppression de la plus courte sous-chaîne à droite
<code>\${variable%%modèle}</code>	Suppression de la plus longue sous-chaîne à droite
<code>\${variable :ind}</code>	Supprime les caractères de 0 à ind d'un chaîne
<code>\${variable :ind :nb}</code>	Extrait nb caractères à partir de ind d'une chaîne
<code>\${#variable}</code>	Donne la longueur de la chaîne considérée
<code>\${variable/modele/}</code>	Supprime la première occurrence de modele dans variable
<code>\${variable//modele/}</code>	Supprime toutes les occurrences de modele dans variable
<code>\${variable/modele1/modele2}</code>	Remplace la première occurrence de modele1 par modele2 dans variable
<code>\${variable//modele1/modele2}</code>	Supprime toutes les occurrences de modele1 par modele2 dans variable

❖ Tests ( on note [expression] )

Operation générale	
<b>!</b>	Negation
<b>Expression1 -a Expression2</b>	Conjonction
<b>Expression1 -o Expression2</b>	Disjonction
Opérations sur les chaînes	
<b>s1 = s2</b>	Egalité
<b>s1 != s2</b>	Différence
<b>-n s1</b>	Chaîne vide
<b>-z s1</b>	Chaîne non vie
Opérations sur les nombres	
<b>n1 -eq n2</b>	=
<b>n1 -ne n2</b>	≠
<b>n1 -lt n2</b>	<
<b>n1 -le n2</b>	≤
<b>n1 -gt n2</b>	>
<b>n1 -ge n2</b>	≥
Tests sur les fichiers	
<b>-e fic1</b>	Existence du fichier
<b>-s fic1</b>	Existence du fichier non vide
<b>-f fic1</b>	Existence du fichier ordinaire
<b>-d rep1</b>	Existence du répertoire
<b>-r fic1</b>	Existence du droit de lecture
<b>-w fic1</b>	Existence du droit d'écriture
<b>-x fic1</b>	Existence du droit d'exécution

## ❖ Les structures de contrôle

```
if cond
then
...
elif
then
...
else
...
fi
```

---

```
case VAR in
cas 1)
...
... ;;
cas 2)
...
... ;;
*)
... ;;
esac
```

---

```
for VAR in Liste_Valeur
do
```

```
...
...
done
```

---

```
while cond
do
```

```
...
...
done
```

---

```
while read var
do
```

```
...
...
done < fic
```

Expression arithmétique : `$((...))`  
Récupération des arguments : `$0,$1 ...`  
Nombre de paramètre : `$#`  
Décalage des paramètres : `shift`

## ❖ Rappel qqes commandes unix

### → Sort

La commande **sort** permet de trier les données d'un fichier. Les options sont les suivantes :

- **-n** le tri se fait sur un nombre
- **-tx** Le caractère **x** est le séparateur de champ
- **-r** le tri se fait dans l'ordre décroissant
- **-u** supprime les lignes doublons
- **-d** le tri se fait sur des caractères alphanumériques
- **-f** pas de différence entre les minuscules et les majuscules

```
sort nom-fic : par défaut tri alphabétique du fichier
sort -t : +1 -2 nom-fic : tri alpha. du fichier selon la deuxième colonne
                    (du second +1 au troisième -2)
sort -n -t : +3 -4 +0 nom-fic : tri numérique (-n) du fichier selon la
                    quatrième colonne
```

```
sort -t : +3.2 +0 nom-fic : tri alpha. du fichier sur le troisième
caractère (2) de la quatrième colonne (3). En cas d'égalité, on fait un
second tri sur la première colonne (+0).
```

### → Wc

```
wc -l nom-fichier : compte le nombre de lignes
wc -w nom-fichier : compte le nombre de mots
wc -c nom-fichier : compte le nombre de caractères
```

### → Head

```
head +10 nom-fichier : toutes les lignes du fichier de la 10 ème au début
head -10 nom-fichier : les 10 premières lignes (mm effet que ci dessus)
```

### → Tail

```
tail +10 nom-fichier : toutes les lignes du fichier de la 10 ème jusqu'à la fin
tail -10 nom-fichier : les 10 dernières lignes
```

### → Cut

La commande **cut** permet d'extraire certains champs d'un fichier. Les options sont les suivantes :

- **-c** extrait suivant le nombre de caractères
- **-f** extrait suivant le nombre de champs
- **-dx** Le caractère **x** est le séparateur de champ

```
cut -c-10 nom-fic : extrait les 10 premiers caractères de chaque ligne
cut -c2-5 nom-fic : extrait les 2ème au 5ème caractères de chaque ligne
cut -c25- nom-fic : extrait du 25ème caractère à la fin de chaque ligne
cut -d: -f1,4 nom-fic : extrait les blocs let4, les : etant les séparateurs
de bloc
cut -d : -f3- nom-fic : extrait du 3ème blocs jusqu'à la fin
```