Unix / Linux - Mécanismes de citation Shell

Dans ce chapitre, nous discuterons en détail des mécanismes de citation Shell. Nous commencerons par discuter des métacaractères.

Les métacaractères

Unix Shell fournit divers métacaractères qui ont une signification particulière lors de leur utilisation dans n'importe quel script Shell et provoque la terminaison d'un mot, sauf indication contraire.

Par exemple, **?** correspond à un seul caractère lors de la liste des fichiers dans un répertoire et un ***** correspond à plus d'un personnage. Voici une liste de la plupart des caractères spéciaux du shell (également appelés métacaractères) −

```
* ? [ ] ' " \ $ ; & ( ) | ^ < > new-line space tab
```

Un caractère peut être cité (c'est-à-dire, fait pour se défendre) en le précédant avec un ****.

Exemple

L'exemple suivant montre comment imprimer un ***** ou un **?** −

Démo en direct

```
#!/bin/sh

echo Hello; Word
```

Lors de l'exécution, vous recevrez le résultat suivant −

```
Hello
./test.sh: line 2: Word: command not found

shell returned 127
```

Essayons maintenant d'utiliser un caractère cité −

Démo en direct

```
#!/bin/sh

echo Hello\; Word
```

Lors de l'exécution, vous recevrez le résultat suivant −

```
Hello; Word
```

Le **$** le signe est l'un des métacaractères, il doit donc être cité pour éviter une manipulation spéciale par la coque −

Démo en direct

```
#!/bin/sh

echo "I have \$1200"
```

Lors de l'exécution, vous recevrez le résultat suivant −

```
I have $1200
```

Le tableau suivant répertorie les quatre formes de cotation −

Sr.No.	Citation et description
1	**Citation unique** Tous les personnages spéciaux entre ces citations perdent leur signification particulière.
2	**Double devis** La plupart des caractères spéciaux entre ces citations perdent leur signification particulière avec ces exceptions − • $ • ` • \$ • \' • \" • \\
3	**Backslash** Tout caractère suivant immédiatement la barre oblique inverse perd sa signification particulière.
4	**Citation de retour** Tout ce qui se trouve entre les guillemets arrière serait traité comme une commande et serait exécuté.

Les citations uniques

Considérez une commande echo qui contient de nombreux caractères de shell spéciaux −

```
echo <-$1500.**>; (update?) [y|n]
```

Mettre une barre oblique inverse devant chaque personnage spécial est fastidieux et rend la ligne difficile à lire −

```
echo \<-\$1500.\*\*\>\; \(update\?\) \[y\|n\]
```

Il existe un moyen facile de citer un grand groupe de personnages. Mettez un seul devis (') au début et à la fin de la chaîne −

```
echo '<-$1500.**>; (update?) [y|n]'
```

Les caractères dans des guillemets simples sont cités comme si une barre oblique inverse était devant chaque personnage. Avec cela, la commande echo s'affiche correctement.

Si un seul devis apparaît dans une chaîne à produire, vous ne devez pas placer la chaîne entière dans des guillemets simples, vous devez plutôt précéder celle en utilisant une barre oblique inverse (\ ) comme suit −

```
echo 'It\'s Shell Programming
```

Les doubles citations

Essayez d'exécuter le script shell suivant. Ce script shell utilise un seul devis −

Démo en direct

```
VAR=ZARA
echo '$VAR owes <-$1500.**>; [ as of (`date +%m/%d`) ]'
```

Lors de l'exécution, vous recevrez le résultat suivant −

```
$VAR owes <-$1500.**>; [ as of (`date +%m/%d`) ]
```

Ce n'est pas ce qui devait être affiché. Il est évident que les guillemets simples empêchent la substitution de variables. Si vous souhaitez remplacer des valeurs variables et faire fonctionner des virgules inversées comme prévu, vous devrez alors mettre vos commandes entre guillemets comme suit −

Démo en direct

```
VAR=ZARA
echo "$VAR owes <-\$1500.**>; [ as of (`date +%m/%d`) ]"
```

Lors de l'exécution, vous recevrez le résultat suivant −

```
ZARA owes <-$1500.**>; [ as of (07/02) ]
```

Les guillemets doubles enlèvent la signification particulière de tous les caractères, sauf les − suivants

- **$** pour la substitution de paramètres
- Backquotes pour la substitution de commande
- **\$** pour permettre des signes littéraux en dollars
- **\`** pour permettre des guillemets littéraux

- **\"** pour activer les guillemets doubles intégrés
- **** pour activer les barres obliques inverses intégrées
- Tous les autres **** les caractères sont littéraux (pas spéciaux)

Les caractères dans des guillemets simples sont cités comme si une barre oblique inverse était devant chaque personnage. Cela aide la commande echo à s'afficher correctement.

Si un seul devis apparaît dans une chaîne à produire, vous ne devez pas placer la chaîne entière dans des guillemets simples, vous devez plutôt précéder celle en utilisant une barre oblique inverse (\ ) comme suit −

```
echo 'It\'s Shell Programming'
```

Les guillemets

Mettre n'importe quelle commande Shell entre les deux **guillemets** exécute la commande.

Syntaxe

Voici la syntaxe simple pour mettre n'importe quel shell **commande** entre les guillemets −

```
var=`command`
```

Exemple

Le **date** la commande est exécutée dans l'exemple suivant et le résultat produit est stocké dans la variable DATA.

Démo en direct

```
DATE=`date`

echo "Current Date: $DATE"
```

Lors de l'exécution, vous recevrez le résultat suivant −

```
Current Date: Thu Jul  2 05:28:45 MST 2009
```