

Unix / Linux - Utilisation des variables Shell

Dans ce chapitre, nous apprendrons à utiliser les variables Shell dans Unix. Une variable est une chaîne de caractères à laquelle nous attribuons une valeur. La valeur attribuée peut être un numéro, un texte, un nom de fichier, un appareil ou tout autre type de données.

Une variable n'est rien de plus qu'un pointeur vers les données réelles. Le shell vous permet de créer, d'attribuer et de supprimer des variables.

Noms variables

Le nom d'une variable ne peut contenir que des lettres (a à z ou A à Z), des nombres (0 à 9) ou le caractère de soulignement (_).

Par convention, les variables shell Unix auront leurs noms dans UPPERCASE.

Les exemples suivants sont des noms de variables valides –

```
_ALI  
TOKEN_A  
VAR_1  
VAR_2
```

Voici les exemples de noms de variables invalides –

```
2_VAR  
-VARIABLE  
VAR1-VAR2  
VAR_A!
```

La raison pour laquelle vous ne pouvez pas utiliser d'autres caractères tels que !, *, ou - est que ces caractères ont une signification particulière pour le shell.

Définition des variables

Les variables sont définies comme suit –

```
variable_name=variable_value
```

Par exemple –

```
NAME="Zara Ali"
```

L'exemple ci-dessus définit la variable NAME et lui attribue la valeur "Zara Ali. Les variables de ce type sont appelées **variables scalaires**. Une variable scalaire ne peut contenir qu'une seule valeur à la fois.

Shell vous permet de stocker la valeur souhaitée dans une variable. Par exemple –

```
VAR1="Zara Ali"  
VAR2=100
```

Accès aux valeurs

Pour accéder à la valeur stockée dans une variable, préfixez son nom avec le signe dollar (\$) –

Par exemple, le script suivant accédera à la valeur de la variable définie NAME et l'imprimera sur STDOUT –

[Démonstration en direct](#)

```
#!/bin/sh  
  
NAME="Zara Ali"  
echo $NAME
```

Le script ci-dessus produira la valeur suivante –

```
Zara Ali
```

Variables en lecture seule

Shell fournit un moyen de marquer les variables en lecture seule en utilisant la commande en lecture seule. Une fois qu'une variable est marquée en lecture seule, sa valeur ne peut pas être modifiée.

Par exemple, le script suivant génère une erreur en essayant de modifier la valeur de NAME –

[Démonstration en direct](#)

```
#!/bin/sh  
  
NAME="Zara Ali"  
readonly NAME  
NAME="Qadiri"
```

Le script ci-dessus générera le résultat suivant –

```
/bin/sh: NAME: This variable is read only.
```

Désinitier les variables

La désactivation ou la suppression d'une variable dirige le shell pour supprimer la variable de la liste des variables qu'elle suit. Une fois que vous avez désactivé une variable, vous ne pouvez pas accéder à la valeur stockée dans la variable.

Voici la syntaxe pour désétablir une variable définie à l'aide de la **déséquilibrer** commande –

```
unset variable_name
```

La commande ci-dessus désactive la valeur d'une variable définie. Voici un exemple simple qui montre comment fonctionne la commande –

```
#!/bin/sh
```

```
NAME="Zara Ali"
```

```
unset NAME
```

```
echo $NAME
```

L'exemple ci-dessus n'imprime rien. Vous ne pouvez pas utiliser la commande unset pour **déséquilibrer** variables marquées **en lecture seule**.

Types de variables

Lorsqu'un shell est en cours d'exécution, trois principaux types de variables sont présents –

- **Variables locales** – Une variable locale est une variable présente dans l'instance actuelle du shell. Il n'est pas disponible pour les programmes lancés par le shell. Ils sont définis à l'invite de commande.
- **Variables d'environnement** – Une variable d'environnement est disponible pour tout processus enfant du shell. Certains programmes ont besoin de variables d'environnement pour fonctionner correctement. Habituellement, un script shell définit uniquement les variables d'environnement nécessaires aux programmes qu'il exécute.
- **Variables shell** – Une variable shell est une variable spéciale définie par le shell et requise par le shell pour fonctionner correctement. Certaines de ces variables sont des variables d'environnement tandis que d'autres sont des variables locales.