



Shell Stripping

Lien : <https://innov-maroc.com/formation/shell-stripping>

 DURÉE
5 jours (35h)

 RÉFÉRENCE
WLUM02

 CATÉGORIE
**Linux Administration Et
Préparation Aux
Certifications Lpic**

OBJECTIFS DE LA FORMATION

À l'issue de cette formation, vous serez capable de :

- ✓ Maîtriser l'automatisation des tâches répétitives et chronophages
- ✓ Concevoir des scripts `bash` robustes, sécurisés et portables pour des environnements de production
- ✓ Parser, transformer et analyser des fichiers de logs ou des données textuelles pour en extraire des informations utiles
- ✓ Gagner en autonomie et en efficacité dans la gestion quotidienne d'infrastructures Linux
- ✓ Développer des outils en ligne de commande pour simplifier les processus de déploiement et de maintenance
- ✓ Intégrer les bonnes pratiques de scripting dans une chaîne d'outils DevOps (CI/CD)

POUR QUI ?

- ✓ Administrateurs Systèmes Linux/Unix cherchant à automatiser leur travail
- ✓ Ingénieurs DevOps et SRE (Site Reliability Engineers) voulant renforcer leurs fondamentaux
- ✓ Développeurs Back-end qui interagissent quotidiennement avec des serveurs Linux
- ✓ Techniciens de support IT souhaitant évoluer vers des postes d'administration ou d'ingénierie
- ✓ Architectes Cloud ayant besoin de créer des scripts d'initialisation et de configuration d'instances

INNOV MAROC



Programme détaillé

1 / Fondamentaux de la ligne de commande et du Shell Bash

- Maîtriser l'environnement : navigation, variables d'environnement (\$PATH, \$HOME), historique et alias
- Gestion des entrées/sorties : redirection standard (>, >>), redirection d'erreur (2>), et pipes (|)
- L'art du quoting : comprendre la différence critique entre les guillemets simples (' '), doubles (" "), et l'absence de guillemets

2 / Structures de contrôle et logique de script

- Expressions conditionnelles : maîtriser `test`, les crochets `[ ]` et les doubles crochets `[ [ ] ]`
- Logique de branchement : construire des blocs `if-then-else-fi` et des structures `case` robustes
- Boucles et itérations : utiliser les boucles `for`, `while` et `until` pour traiter des listes de fichiers ou des lignes de texte

3 / Fonctions, portée des variables et gestion des arguments

- Créer des fonctions modulaires et réutilisables avec `function nom() { }`
- Gérer la portée des variables : distinction entre variables globales, locales (`local`) et environnementales (`export`)
- Parser les arguments de script : maîtriser les variables spéciales ` \$#`, ` \$@`, ` \$\*`, ` \$1` et l'utilisation de `getopts`

4 / La philosophie Unix en action : la puissance du chaînage

- Filtrage de texte avec `grep` : utiliser les expressions régulières basiques (BRE) et étendues (ERE) pour extraire des informations

- Transformation de flux avec ``sed`` : maîtriser la substitution (``s/old/new/g``) et la suppression de lignes (``d``)
- Combinaison ``find`` et ``xargs`` : appliquer des commandes sur des ensembles de fichiers trouvés de manière récursive et sécurisée

5 / Traitement de données structurées avec ``awk``

- Comprendre la structure d'un programme ``awk`` : les blocs ``BEGIN``, le bloc principal ``{}`` et ``END``
- Manipuler des champs et des enregistrements : utiliser ``$1``, ``$2``, ``$NF`` et la variable ``FS`` pour parser des fichiers CSV ou des logs
- Appliquer des logiques conditionnelles et des calculs directement dans ``awk`` pour générer des rapports

6 / Gestion avancée des processus et du système

- Contrôle des jobs en arrière-plan : utiliser ``&``, ``jobs``, ``fg``, ``bg`` et ``nohup``
- Signaux et terminaison propre : intercepter les signaux (SIGINT, SIGTERM) avec la commande ``trap`` pour nettoyer les fichiers temporaires
- Créer des subshells ``( )`` pour isoler des changements d'environnement sans affecter le script principal

7 / Interaction avec l'utilisateur et lecture de données

- Lire des entrées utilisateur de manière interactive et sécurisée avec ``read``
- Utiliser les "here documents" (``<<EOF``) pour intégrer des blocs de texte multi-lignes dans les scripts
- Implémenter des menus de sélection simples avec la commande ``select``

8 / Scripting robuste et "defensive programming"

- Écrire des scripts qui échouent rapidement : utiliser ``set -e`` et ``set -o pipefail`` pour stopper l'exécution en cas d'erreur
- Vérifier les prérequis : tester l'existence de fichiers, de répertoires, ou la disponibilité de commandes avant de les utiliser
- Intégrer l'analyse statique : utiliser ``ShellCheck`` pour détecter les erreurs communes et les mauvaises pratiques avant l'exécution

9 / Automatisation de tâches d'administration système

- Cas pratique 1 : développer un script de sauvegarde complet pour un répertoire, avec compression, horodatage et rotation des archives
- Cas pratique 2 : créer un script de surveillance qui vérifie l'espace disque, la charge CPU et envoie une alerte par email
- Cas pratique 3 : automatiser la gestion des logs, incluant la rotation, la compression et l'archivage

10 / Principes DevOps : Idempotence et scripts de déploiement

- Concevoir des scripts idempotents : s'assurer qu'un script exécuté plusieurs fois produit le même résultat qu'après une seule exécution
- Créer un script de déploiement simple : cloner un dépôt Git, installer des dépendances et redémarrer un service systemd
- Intégrer des variables de configuration depuis un fichier externe pour séparer le code de la configuration

🔗 Approche pédagogique

- ✓ Support Ecrit et Projection
- ✓ Exposés Interactifs, Podcasts et Vidéos
- ✓ Brainstorming et Jeux de Rôle
- ✓ Cas Pratiques et Labs inclus pour leur impact opérationnel
- ✓ Test de Validation des Acquis des Connaissances

📅 Prochaines dates programmées

📅 28 Sep. au 02 Oct. 2026

📍 Présentiel - Casablanca

📅 23 au 27 Nov. 2026

📍 Présentiel - Casablanca

📅 Autres dates possibles sur demande. Contactez-nous pour organiser une session intra-entreprise.

🔄 Réservation & Renseignements

📞 Téléphone : +212 522 247 210

✉ Email : contact@innov-maroc.com

🌐 Web : <https://www.innov-maroc.com>

INNOV MAROC