

# Écriture de scripts en Shell sous Unix/Linux certification AVIT® à distance en option

Cours Pratique de 3 jours - 21h  
Réf : SHL - Prix 2024 : 2 030€ HT

Le shell est à la fois un langage et un programme dont nous vous présentons ici le comportement détaillé afin de maîtriser sa programmation. Ce cours vous permettra d'acquérir une véritable autonomie dans l'écriture de scripts en shell dans des domaines d'applications concrets (surveillance, automatisation, installation logicielle, traitement des fichiers...).

## OBJECTIFS PÉDAGOGIQUES

À l'issue de la formation l'apprenant sera en mesure de :

Connaître les différentes instructions utilisables dans un script shell

Ecrire des scripts simples d'exploitation Unix/Linux

Effectuer le debugging d'un script shell

Enrichir un script shell avec des fonctions et sous-shells

Manipuler les fichiers avec les commandes grep, find, sed, awk

## TRAVAUX PRATIQUES

Différents exercices (simples et complexes) seront réalisés pour acquérir une véritable autonomie dans l'écriture de scripts en shell.

## LE PROGRAMME

dernière mise à jour : 03/2022

### 1) Présentation et rappels

- Les différents interpréteurs : Bourne Shell, Korn Shell, Bash, C Shell, Tcsh...
- Disponibilité des interpréteurs sur les divers systèmes Unix.
- Le point sur la normalisation (impacts sur l'écriture des scripts).
- Les apports GNU (gawk, gsed...).
- Différences Bourne Shell/Korn Shell/Bash.

### 2) Programmation par scripts

- Outils de développement.
- Mécanisme d'exécution des scripts.
- Règles de recherche des commandes.
- Principes d'exécution d'une commande (exec, pipeline, sous-shell, background...).
- Principes d'exécution d'un script.

### 3) Mécanismes de base

- Lecture et analyse de la ligne de commande.
- Expansion des accolades, développement du tilde, remplacement des paramètres.
- Substitution des commandes et évaluation arithmétique.
- Procédés d'échappement (banalisation).
- Les redirections (entrée et sortie standard, fichiers, tubes, document en ligne).

## PARTICIPANTS

Développeurs, techniciens et administrateurs Unix/Linux.

## PRÉREQUIS

Connaissances de base d'un système Linux/Unix et de la programmation. Expérience souhaitable de l'utilisation d'un de ces systèmes.

## COMPÉTENCES DU FORMATEUR

Les experts qui animent la formation sont des spécialistes des matières abordées. Ils ont été validés par nos équipes pédagogiques tant sur le plan des connaissances métiers que sur celui de la pédagogie, et ce pour chaque cours qu'ils enseignent. Ils ont au minimum cinq à dix années d'expérience dans leur domaine et occupent ou ont occupé des postes à responsabilité en entreprise.

## MODALITÉS D'ÉVALUATION

Le formateur évalue la progression pédagogique du participant tout au long de la formation au moyen de QCM, mises en situation, travaux pratiques...

Le participant complète également un test de positionnement en amont et en aval pour valider les compétences acquises.

## MOYENS PÉDAGOGIQUES ET TECHNIQUES

- Les moyens pédagogiques et les méthodes d'enseignement utilisés sont principalement : aides audiovisuelles, documentation et support de cours, exercices pratiques d'application et corrigés des exercices pour les stages pratiques, études de cas ou présentation de cas réels pour les séminaires de formation.
- À l'issue de chaque stage ou séminaire, ORSYS fournit aux participants un questionnaire d'évaluation du cours qui est ensuite analysé par nos équipes pédagogiques.
- Une feuille d'émargement par demi-journée de présence est fournie en fin de formation ainsi qu'une attestation de fin de formation si le stagiaire a bien assisté à la totalité de la session.

## MODALITÉS ET DÉLAIS D'ACCÈS

L'inscription doit être finalisée 24 heures avant le début de la formation.

## ACCESSIBILITÉ AUX PERSONNES HANDICAPÉES

Vous avez un besoin spécifique d'accessibilité ? Contactez Mme FOSSE, référente handicap, à l'adresse suivante psh-accueil@orsys.fr pour étudier au mieux votre demande et sa faisabilité.

#### 4) Fonctionnement en interactif

- Invocation du shell (options).
- Les différents fichiers de démarrage.
- Notions d'environnement (variables, alias, fonctions).
- Historique et rappel des commandes.
- Contrôle de jobs.
- La complémentation des noms.
- Terminaison du shell.

#### 5) Construction de shell-scripts portables (ksh/bash)

- Interface avec un shell-script.
- Structuration d'un shell-script. Notions de sous-shell.
- Appel d'un shell : les différentes méthodes. Options utilisables. Compléments sur l'environnement. Options et arguments.
- Préambule du shell-script : qui interprète le shell-script ? Commentaires.
- Paramètres de position (initialisation, sauvegarde, décalages).
- Variables locales. Variables globales.
- Déclaration et visibilité des fonctions.
- Sortie du shell-script. Fonction de sortie. Conventions utilisées. Valeur de retour.
- Enchaînement de shell-scripts.
- Structures de contrôle du shell : commandes simples, pipelines, listes de pipelines.
- Commandes composées, sous-shells et fonctions. Mécanismes de sélection et d'itération. Menus.
- Entrées/sorties. Interactions avec le système. Arguments en ligne de commande.
- Opérations de tests. Compléments sur les instructions. Redirection locale dans les fichiers de commande.

#### 6) Robustesse, debugging

- Robustesse d'un shell-script : vérifier l'initialisation des variables.
- Gestion avancée des arguments en ligne de commande (getopts).
- Tests sur le type des variables (expr). La commande eval.
- Gestion des signaux.
- Nommage et destruction des fichiers temporaires.
- Cas particulier d'exécution d'un shell-script par cron (la crontab).
- Debugging d'un shell-script : commandes de debugging.
- Signaux de trace. Journalisation.
- Journalisation.

#### 7) Extensions du Korn Shell et Bash

- Tableaux de variables.
- Notations spécifiques.
- Opérations arithmétiques.
- Les alias suivis.
- Notion de coprocessus.
- Les commandes internes spécifiques.

#### 8) Outils supplémentaires (grep, find, sed, awk)

- Outils d'assistance pour la création de scripts : grep et find.
- Expressions rationnelles : outil grep.
- Recherche et traitement de fichiers : outil find.
- Sed : principes et aspects des scripts. Commandes simples de recherche, remplacement, insertion.
- Manipulation de flux de texte avec sed. Aperçu des commandes avancées.
- Éléments généraux de programmation avec awk. Utilisation des variables et des fonctions.
- Exemples complets de scripts awk (statistiques système, calculs...).
- Présentation des fonctions intégrées awk : mathématique, traitement de chaîne, interaction avec le système...

# LES DATES

---

## CLASSE À DISTANCE

2024 : 24 juin, 23 sept., 20 nov.

## AIX-EN-PROVENCE

2024 : 23 sept., 20 nov.

## VALENCE

2024 : 23 sept., 20 nov.

## GRENOBLE

2024 : 23 sept., 20 nov.

## MONTPELLIER

2024 : 23 sept., 20 nov.

## ORLÉANS

2024 : 17 juin, 16 sept., 13 nov.

## RENNES

2024 : 23 sept., 20 nov.

## CAEN

2024 : 23 sept., 20 nov.

## MULHOUSE

2024 : 23 sept., 20 nov.

## LE MANS

2024 : 16 sept., 13 nov.

## STRASBOURG

2024 : 23 sept., 20 nov.

## TOURS

2024 : 16 sept., 13 nov.

## PARIS

2024 : 17 juin, 16 sept., 13 nov.

## ANGERS

2024 : 23 sept., 20 nov.

## CLERMONT-FERRAND

2024 : 23 sept., 20 nov.

## LILLE

2024 : 23 sept., 20 nov.

## NANCY

2024 : 23 sept., 20 nov.

## REIMS

2024 : 16 sept., 13 nov.

## PAU

2024 : 23 sept., 20 nov.

## MARSEILLE

2024 : 23 sept., 20 nov.

## AVIGNON

2024 : 23 sept., 20 nov.

## ROUEN

2024 : 16 sept., 13 nov.

## TOULON

2024 : 23 sept., 20 nov.

## LYON

2024 : 24 juin, 23 sept., 20 nov.

## BORDEAUX

2024 : 23 sept., 20 nov.

## DIJON

2024 : 23 sept., 20 nov.

## LIMOGES

2024 : 23 sept., 20 nov.

## NIORT

2024 : 23 sept., 20 nov.

## NANTES

2024 : 23 sept., 20 nov.

## BREST

2024 : 23 sept., 20 nov.

## METZ

2024 : 23 sept., 20 nov.

## SAINT-ÉTIENNE

2024 : 23 sept., 20 nov.

## SOPHIA-ANTIPOLIS

2024 : 23 sept., 20 nov.

## TOULOUSE

2024 : 23 sept., 20 nov.