

Stage pratique de 3 jour(s)  
Réf : SHL

## Participants

Développeur, Techniciens et administrateurs Unix/Linux.

## Pré-requis

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

Prix 2017 : 1880€ HT

## Dates des sessions

### Paris

6 juin 2017, 10 juil. 2017  
28 août. 2017, 25 sep. 2017  
23 oct. 2017, 20 nov. 2017  
18 déc. 2017

### Aix

12 juin 2017, 4 sep. 2017  
18 déc. 2017

### Bordeaux

19 juin 2017, 11 sep. 2017  
11 déc. 2017

### Bruxelles

19 juin 2017, 9 oct. 2017  
11 déc. 2017

### Geneve

19 juin 2017, 9 oct. 2017  
11 déc. 2017

### Grenoble

19 juin 2017, 11 sep. 2017  
11 déc. 2017

### Lille

12 juin 2017, 4 sep. 2017  
18 déc. 2017

### Luxembourg

19 juin 2017, 9 oct. 2017  
11 déc. 2017

### Lyon

19 juin 2017, 11 sep. 2017  
11 déc. 2017

### Montpellier

12 juin 2017, 4 sep. 2017  
18 déc. 2017

### Nantes

12 juin 2017, 4 sep. 2017  
18 déc. 2017

### Rennes

12 juin 2017, 4 sep. 2017  
18 déc. 2017

### Sophia-antipolis

12 juin 2017, 4 sep. 2017  
18 déc. 2017

### Strasbourg

12 juin 2017, 4 sep. 2017  
18 déc. 2017

### Toulouse

19 juin 2017, 11 sep. 2017  
11 déc. 2017

## Modalités d'évaluation

# Ecriture de scripts en shell sous Unix/Linux

**Best**

*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 PEDAGOGIQUES

Connaître les différentes instructions utilisable 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

### 1) Présentation et rappels

### 2) Programmation par scripts

### 3) Mécanismes de base

### 4) Fonctionnement en interactif

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

### 6) Robustesse, Debugging

### 7) Extensions du Korn Shell et Bash

### 8) Outils supplémentaires (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.*

## 1) Présentation et rappels

- Principes.
- 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).

## 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.

L'évaluation des acquis se fait tout au long de la session au travers des multiples exercices à réaliser (50 à 70% du temps).

### 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.

### 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.
- A 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.

- 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.
- Eléments 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...