

Stage pratique de 3 jour(s)

Réf : PRL

## Participants

Informaticiens concernés par des procédures d'exploitation en environnements Unix, Linux ou Windows.

## Pré-requis

Connaissances de base de la programmation et d'un système d'exploitation (Windows ou Linux/Unix).

Prix 2018 : 1920€ HT

## Dates des sessions

### Paris

20 nov. 2017, 18 déc. 2017  
8 jan. 2018, 19 mar. 2018  
14 mai 2018, 9 juil. 2018  
17 sep. 2018

### Aix

26 mar. 2018, 11 juin 2018  
17 sep. 2018

### Bordeaux

11 déc. 2017, 12 mar. 2018  
18 juin 2018, 10 sep. 2018

### Bruxelles

19 mar. 2018, 18 juin 2018  
3 sep. 2018

### Geneve

19 mar. 2018, 18 juin 2018  
3 sep. 2018

### Grenoble

11 déc. 2017, 12 mar. 2018  
18 juin 2018, 10 sep. 2018

### Lille

26 mar. 2018, 11 juin 2018  
17 sep. 2018

### Luxembourg

19 mar. 2018, 18 juin 2018  
3 sep. 2018

### Lyon

11 déc. 2017, 12 mar. 2018  
18 juin 2018, 10 sep. 2018

### Montpellier

26 mar. 2018, 11 juin 2018  
17 sep. 2018

### Nantes

26 mar. 2018, 11 juin 2018  
17 sep. 2018

### Rennes

26 mar. 2018, 11 juin 2018  
17 sep. 2018

### Sophia-antipolis

26 mar. 2018, 11 juin 2018  
17 sep. 2018

### Strasbourg

26 mar. 2018, 11 juin 2018  
17 sep. 2018

### Toulouse

11 déc. 2017, 12 mar. 2018  
18 juin 2018, 10 sep. 2018

## Modalités d'évaluation

# Langage Perl

## scripts système Unix/Linux/Windows

**Best**

*Ce stage vous apprendra à programmer ou maintenir des procédures d'exploitation écrites en Perl. Il permettra de maîtriser le langage lui-même mais aussi de découvrir la "philosophie Perl" avec notamment le recours à un grand nombre de modules permettant d'élargir quasiment à l'infini ses possibilités.*

### OBJECTIFS PEDAGOGIQUES

Maîtriser la syntaxe du langage PERL  
Mettre en œuvre des expressions régulières  
Gérer des fichiers et des répertoires  
Interagir avec le système d'exploitation  
Exploiter les données d'une base de données relationnelle  
Ecrire des scripts Perl dans un contexte Web

#### 1) Présentation

#### 2) L'essentiel du langage

#### 3) Expressions régulières

#### 4) Fonctions, procédures, modules

#### 5) Gestion des fichiers et des répertoires

#### 6) Interactions avec le système d'exploitation

#### 7) Aspects avancés, bases de données, implémentation objet

#### 8) Scripts Perl dans un contexte Web, module CGI

## 1) Présentation

- Historique du langage.
- Perl, pour quel usage ?
- Caractéristiques du langage Perl.
- Les différentes distributions et versions.
- La documentation officielle.
- Installation et première exécution d'un programme Perl.
- Fichiers sources, extension et interprétation.
- Les outils pour utiliser Perl.

### Travaux pratiques

*Installation de l'interpréteur Perl.*

## 2) L'essentiel du langage

- Structure d'un programme Perl, en-tête de script.
- Conventions de codage et règles de nommage.
- Ajout de commentaires.
- Les chaînes de caractères, interprétation de codes.
- Les variables, nommage et typage. Portée des variables.
- Les scalaires.
- Les opérateurs : calcul, incrément, affectation, comparaison...
- Les tableaux, initialisation, affichage du contenu.
- Les tableaux associatifs (hash table), les fonctions keys et values.
- Les références mémoires.
- Les variables systèmes.
- La notion de blocs.
- Les structures de contrôle conditionnelles : if, if... else, if... elsif... else.
- Les structures de contrôle itératives : while, for, until, foreach.
- Les flux entrées/sorties standard (stdin, stdout, stderr).

### Travaux pratiques

*Manipulation du langage à travers l'écriture de différents programmes.*

## 3) Expressions régulières

- Les cas d'utilisation des expressions.
- Panorama des différents motifs : les méta-caractères, les quantificateurs, les caractères spéciaux.
- Le Pattern Matching.
- Le Pattern Substitute.
- Options de la ligne de commande pour l'interpréteur.
- Programme tiers : sed, awk...
- Perl utilisé comme filtre supplémentaire.

### Travaux pratiques

*Mise en œuvre d'expressions régulières.*

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.

## 4) Fonctions, procédures, modules

- Définition des fonctions. Différence entre fonctions et procédures.
- La déclaration d'une fonction, le mécanisme d'appel, le passage d'arguments et la récupération des résultats.
- Visibilité des variables, l'opérateur my.
- Utilisation des références.
- Les fonctions intégrées dans Perl : chomp, chop, length...
- Notion de "package" et de module.
- Modules de la distribution standard.
- Modules du CPAN, recensement, documentation, sites de ressources.
- Installation de modules.
- Utilisation d'un module "classique" et d'un module "Orienté Objet". Les fonctions require et use.
- Fabrication d'un module.
- Documenter les modules créés.

### Travaux pratiques

*Ecriture de fonctions. Intégration d'un module dans un développement. Création d'un module.*

## 5) Gestion des fichiers et des répertoires

- Notion de filehandle.
- Gestion des fichiers : ouverture, lecture, écriture, fermeture...
- Informations sur les fichiers (propriétaire, taille, droits...), la fonction stat.
- L'opérateur diamant.
- Gestion des répertoires : changement, parcours, création, suppression...
- Gestion des erreurs sur manipulation des fichiers et des répertoires.

### Travaux pratiques

*Lecture/écriture dans un fichier. Manipulation des répertoires.*

## 6) Interactions avec le système d'exploitation

- Accès aux informations du système d'exploitation.
- Les commandes agissant sur le système : mkdir, rmdir, chown, chroot, chmod, ...
- Gestion des processus.
- Création de processus fils : system, exec.
- Les quotes inverses.
- Interactions avec le système Windows (modules Win32 et Win32API).

### Travaux pratiques

*Création de processus fils.*

## 7) Aspects avancés, bases de données, implémentation objet

- Tris personnalisés.
- Traitements complémentaires sur les chaînes de caractères.
- Structures de données élaborées à l'aide des références anonymes.
- Accès aux bases de données à l'aide du module DBI.
- Etablissement et configuration d'une connexion, requêtage et récupération des résultats.
- Création de la structure d'une classe et instanciation : paquetage, sous-programme et référence.
- Constructeur et destructeur d'objets.
- Invocation des méthodes d'une classe.
- Le ramasse-miettes.

### Travaux pratiques

*Exploitation des données d'une base de données relationnelle.*

## 8) Scripts Perl dans un contexte Web, module CGI

- Protocole HTTP, serveur Web, schéma fonctionnel CGI.
- Le module CGI.
- Variables d'environnement, méthodes GET et POST.
- Génération des éléments HTML du formulaire avec Perl.
- Traitement des formulaires, exploitation des résultats.

### Travaux pratiques

*Ecriture d'un script CGI en Perl.*