

# 1 Introduction

Les objectifs de la formation

Connaître les caractéristiques des principaux outils de scripting Unix/Linux

Savoir lire des scripts Unix/Linux écrits en Shell, Perl, Python Ruby ou AWK

Être capable d'écrire des scripts simples d'exploitation Unix/Linux

Comprendre comment choisir l'outil le plus adapté pour résoudre un problème particulier

Prez perso et demander le niveau des gens

## 2 Le Shell - les bases

### 2.1 Qu'est ce qu'un shell

Un shell Unix est une interface homme machine (IHM) en ligne de commande (CLI). Il fournit à la fois un langage de commandes interactives et un langage de scripting. Le shell traite des commandes ou scripts.

Il ne faut pas confondre un shell avec un terminal. Un terminal était initialement physiquement un écran et un clavier. Aujourd'hui lorsque l'on parle de terminal on parle d'émulateur de terminal, c'est une catégorie de logiciels permettant de fournir un GUI pour lancer *des* shells (bash, python, zsh, fish, powershell, ruby ...).

Émulateur de terminaux connus : alacritty, Windows Terminal, urxvt, GNOME Terminal, PuTTY.

La confusion est courante car sur Windows historiquement le nom du shell et de l'émulateur de terminal étaient les mêmes (cmd, powershell...), ce n'est plus le cas avec Windows 11 et le Windows Terminal.

### 2.2 Shell Unix

Abus de langage shell = unix shell / bash Principaux shells, principales différences, tout pareil oseb, focus sur Bash.

### 2.3 Détails d'OS pour utiliser shell et autres langages

stdin / stdout droits fichiers, +x, shebang return code,

### 2.4 Concepts/Binaires et builtins

Shell outil de manipulation de binaires linux + fournis des builtins Philo kiss, pleins de petits binaires linux avec une seule tache, trop complexe go python Tout est texte, absolument tout RTFM

## 2.5 Structures de contrôle

if then else fi while do done until do done for do done case esac  
test et [ et [[

## 2.6

Scripting : variables control, loops, test, case (and arg parse, diff - et -), functions,  
utilisation de binaires externes

Puissance de juste one commande + pipe + redirections

Les différents shell (ksh, bash, zsh, csh, ...)

Le Shell POSIX/ISO

L'écriture de script Shell

Activation des commandes POSIX/ISO

Les caractères spéciaux (jockers, échappements, redirection)

Les variables

Les structures de contrôle

## 3 Le langage Perl - les bases

Prez, utilité de nos jours, spécifité

Présentation de Perl

Les variables scalaires, les tableaux, les opérateurs

Les instructions de contrôle

Les tableaux associatifs (hash)

## 4 Le langage Python - les bases

Prez, importance de python, pourquoi (communauté), scope (ansible, blabla, 3600 trucs)

Présentation de Python

Variables et expressions

Les tableaux, les chaînes de caractères

Les instructions de contrôle

Les dictionnaires (hash)

## **5 Le langage Ruby - les bases**

- Présentation de Ruby
- Les variables
- Les chaînes de caractères
- Les structures de contrôle
- Les tableaux, les itérateurs - Les hash

## **6 Les expressions régulières (RegExp)**

- Importance de grep et sed RegExp en Shell (via grep et sed)
- RegExp en Perl
- RegExp en Python
- RegExp en Ruby

## **7 La modularité en Shell, Perl, Python et Ruby**

- parler de direnv, de requirements.txt, de venv python, de classes python (osef un peu) y a des trucs pour shell (bpkg) mais pas le but, illogique, shell = spécifique task
- Les fonctions
- Les paquetages
- L'approche objet
- Utilisation de bibliothèques externes

## **8 La programmation parallèle en Shell, Perl, Python et Ruby**

- différentes concurrency xargs pipes et parallel pour shell multithreading et async pour python

## **9 Résoudre des problèmes avec le Shell, Perl, Python et Ruby**

- Ecrire des scripts d'exploitation (activer une application, les signaux, ...)
- Manipuler des fichiers
- Faire des calculs
- Ecrire des CGI Web
- Accéder à des bases de données
- Manipuler des fichiers XML (parsing, validation, création)

Créer des applications réseaux TCP/IP

Ptit serveur web Génération de fichier dans un dossier service shell filewatcher qui trigger un truc xargs et des pipes pour multiprocessing perl ou grep pour regex awk pour un truc

9 - AWK : un sous-ensemble POSIX/ISO du langage Perl

10 - Conclusion

Quel outil pour quoi faire ?

**Subtitle**