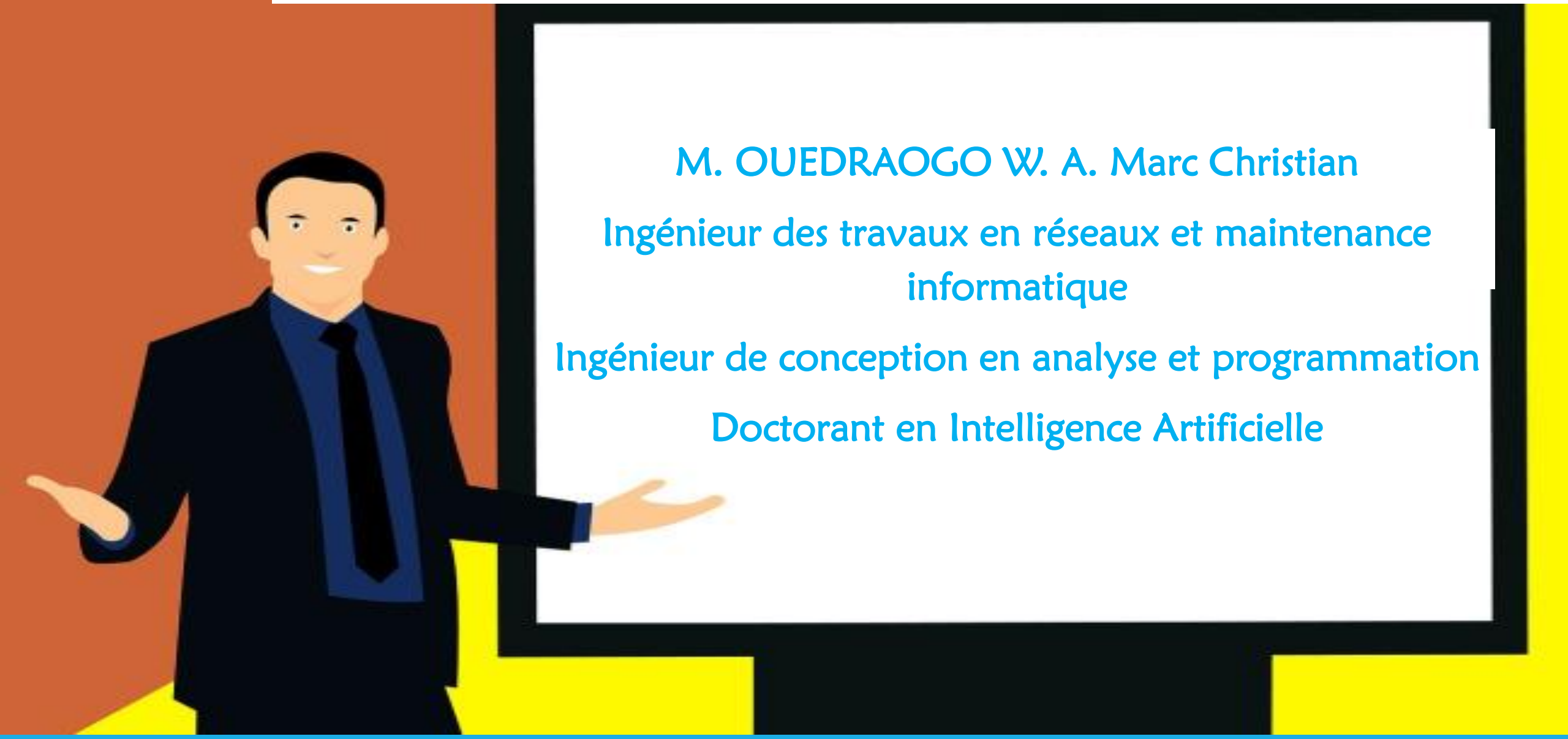
Administration des systèmes

Année académique :2021-2022

Présentation du formateur



M. OUEDRAOGO W. A. Marc Christian

Ingénieur des travaux en réseaux et maintenance informatique

Ingénieur de conception en analyse et programmation

Doctorant en Intelligence Artificielle

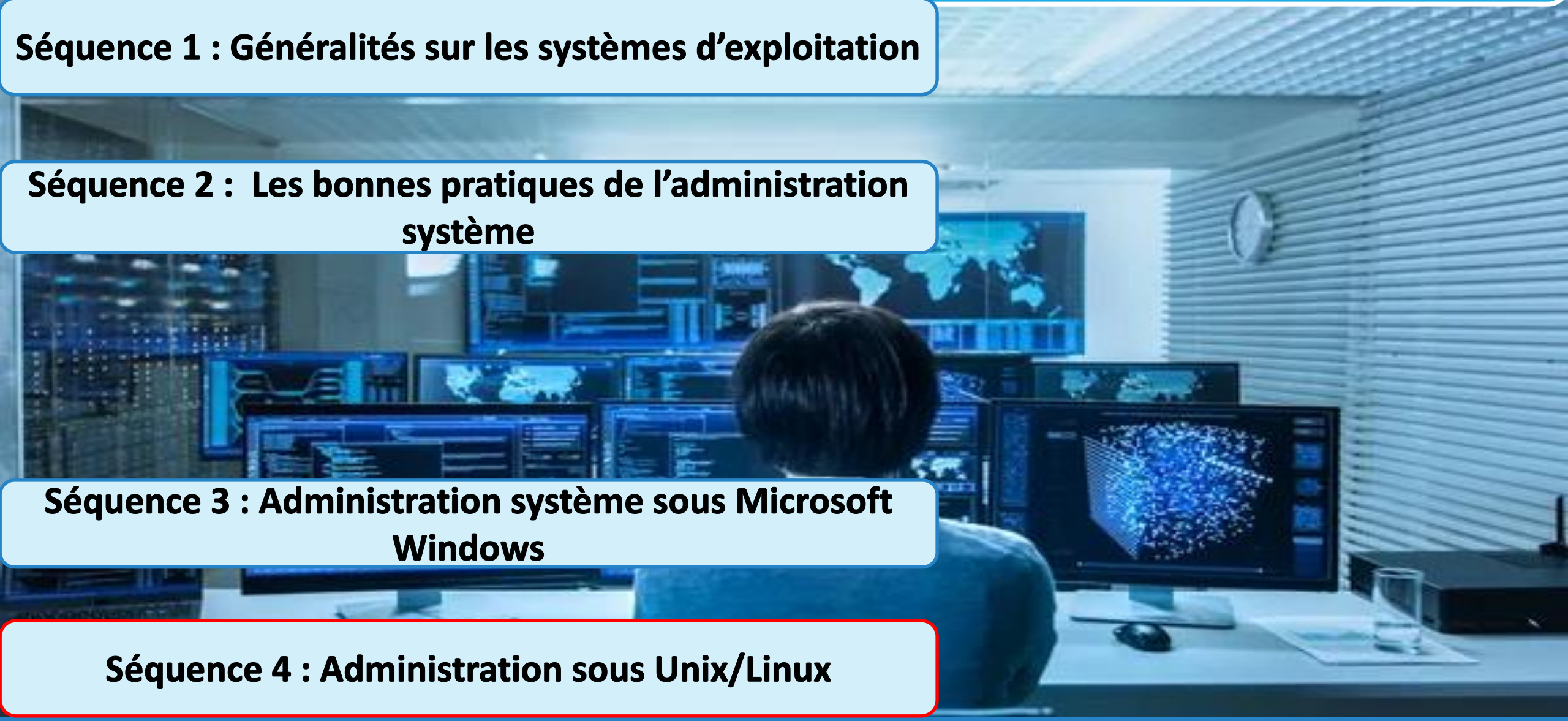
PLAN GENERAL DU COURS

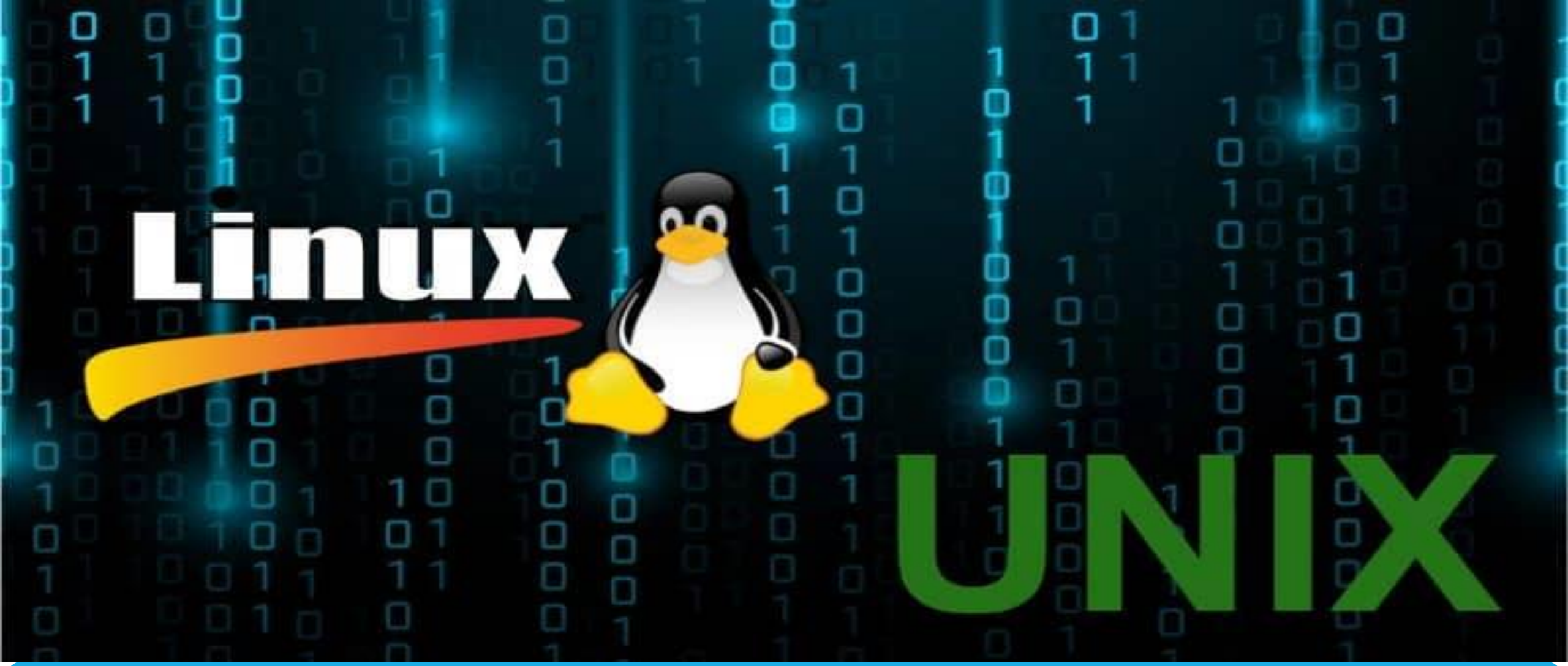
Séquence 1 : Généralités sur les systèmes d'exploitation

Séquence 2 : Les bonnes pratiques de l'administration système

Séquence 3 : Administration système sous Microsoft Windows

Séquence 4 : Administration sous Unix/Linux





Séquence 4 : Administration système sous Unix/Linux

OG4 : Appliquer l'administration sous Unix/Linux

- S'informer sur les systèmes d'exploitation Unix/Linux
- Utiliser le terminal (shell) Unix
- Utiliser les script shell (bash)

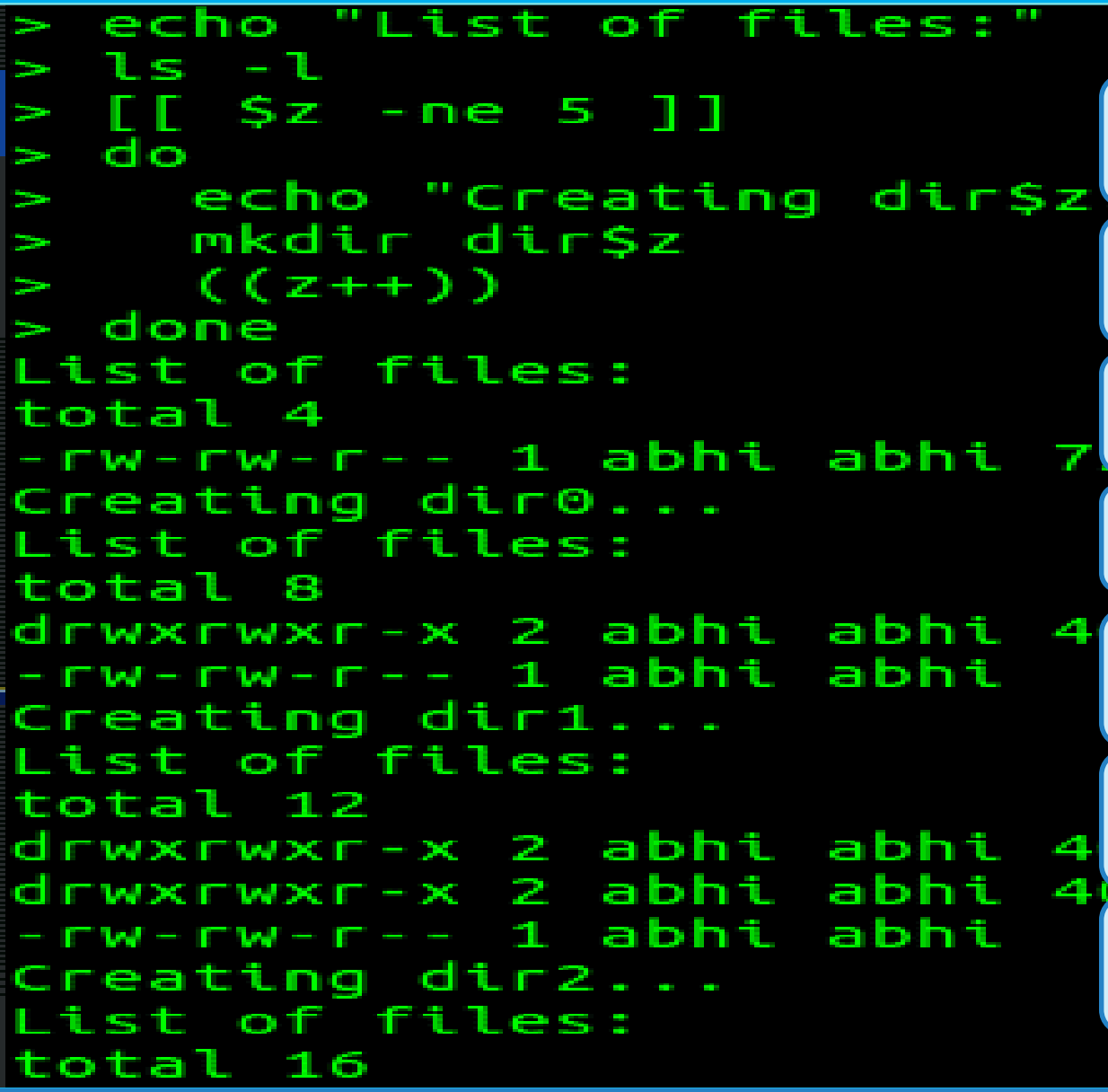
PLAN

Chapitre IX : Présentation de Unix/Linux

Chapitre X : Le terminal Linux

Chapitre XI : Introduction au scripting Linux

Chapitre XI : Introduction au scripting Linux (Bash)



Préambule

Bash est l'acronyme de "**Bourne-Again Shell**".

Il s'agit d'un interpréteur de ligne de commande par défaut pour les systèmes d'exploitation basés sur UNIX et Linux.

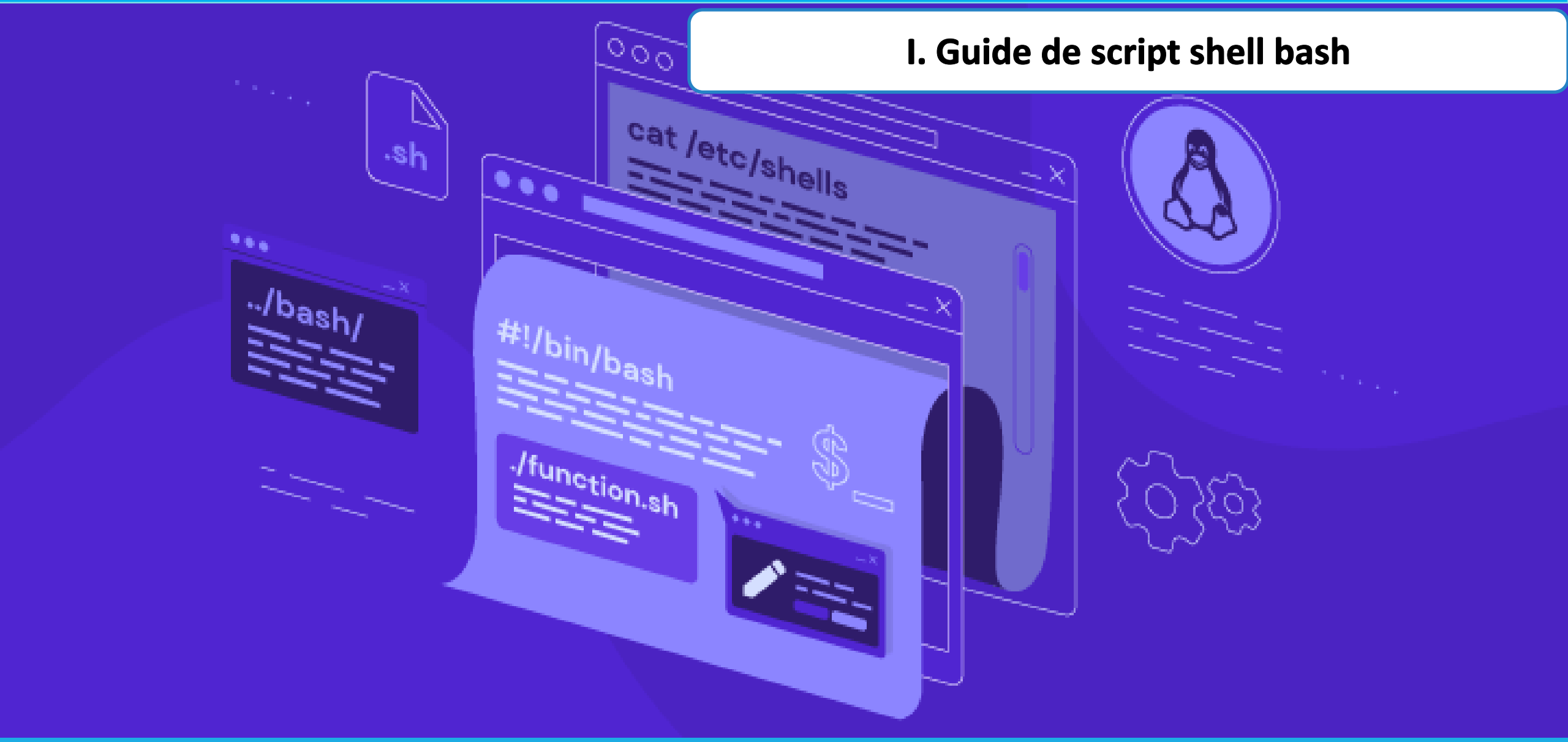
Dans les systèmes d'exploitation basés sur UNIX et Linux, une fenêtre de terminal est composée d'un shell et de Bash.

Apprendre bash peut être très utile pour effectuer une automatisation.

Bash est super puissant et possède de nombreuses fonctionnalités.

Chapitre XI : Introduction au scripting Linux (Bash)

I. Guide de script shell bash



I. Guide de script shell bash

Pour les scripts shell bash, nous devons utiliser le shell et l'éditeur de texte.

L'extension du fichier de script shell est **.sh**.

Pour créer un nouveau fichier de script, créez un fichier texte et enregistrez-le avec l'extension de fichier **.sh**.

Le **gedit** est l'éditeur de texte par défaut pour Ubuntu et de nombreux systèmes d'exploitation basés sur Linux.

Créons un nouveau fichier **myscript.sh**.

I. Guide de script shell bash

Pour exécuter le fichier de script bash, nous devons d'abord **modifier les autorisations du fichier et le rendre exécutable.**

Les autorisations sont modifiées par la commande **chmod +x.**

Pour rendre le fichier de script bash exécutable, vous devez ouvrir votre terminal et accéder au dossier ou au répertoire où le fichier est stocké, puis exécuter la commande suivante.

chmod +x filename

Exécutons maintenant :
chmod +x myscript.sh

Chapitre XI : Introduction au scripting Linux (Bash)

II. Exécution du fichier de script bash

II. Exécution du fichier de script bash

Le fichier de script bash peut être exécuté de deux manières :

- **bash nom_de_fichier**
- **./nom_de_fichier**

Pour exécuter le fichier bash, accédez au dossier ou au répertoire dans lequel le fichier bash est enregistré.

Dans le premier mode d'exécution, écrivez simplement bash, puis le nom du fichier sur le terminal et appuyez sur Entrée.

Le fichier de script bash sera exécuté.
Dans le second cas, écrivez « **./filename** » sur le terminal et appuyez sur Entrée.

Nous utiliserons la deuxième méthode pour exécuter le fichier bash tout au long de cette partie.

II. Exécution du fichier de script bash

La première ligne du script commence toujours par "#!" suivi du chemin vers l'interpréteur de script : **#!**<chemin-vers-interpreter>

En fait, l**a séquence de caractères "#!" est appelée Shebang**.

Elle sert à indiquer au système quel interpréteur de script il doit utiliser pour exécuter le code contenu dans le fichier.

Cette valeur ne correspond pas forcément à Bash, tout dépend du script.

Cela est un bon indicateur pour faire fonctionner un script correctement en l'exécutant à l'aide du bon binaire.

II. Exécution du fichier de script bash

Pour un script Bash, le shebang à inclure sera le chemin vers le binaire bash, à savoir :
#!/bin/bash

Dans le même style, pour exécuter un script en utilisant le binaire de Python, on utilisera :
#!/usr/bin/python

Toujours avec la même logique : sur la première ligne du script.

Par contre, si l'on crée un script et que l'on ne précise pas le Shebang dans l'en-tête, le système va l'exécuter avec l'interpréteur par défaut associé au Shell de l'utilisateur qui lance le script.

Chapitre XI : Introduction au scripting Linux (Bash)

III. Commande d'écho

III. Commande d'écho

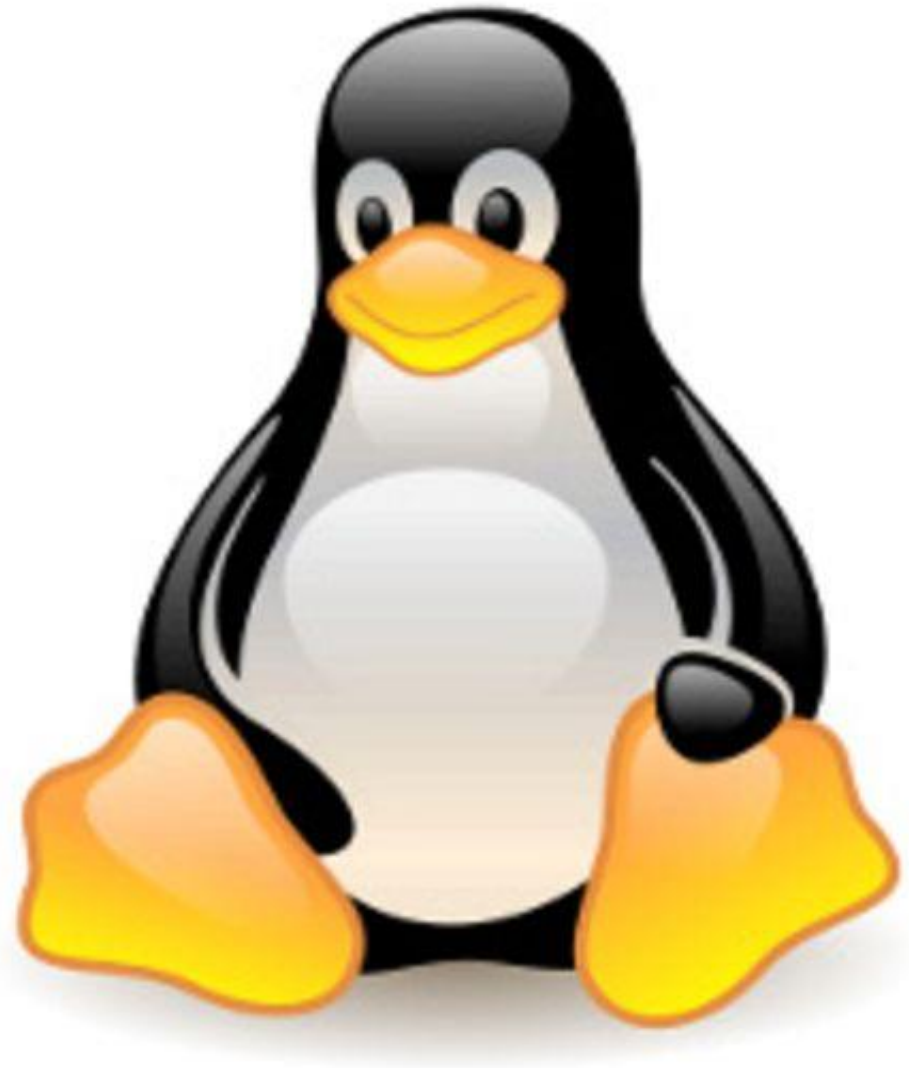
La commande echo est la commande la plus fondamentale et la plus basique des scripts bash.

Il est principalement utilisé pour imprimer le texte ou la sortie du terminal Linux. La commande echo imprimera le texte ou les données sur le terminal quoi que vous écriviez.

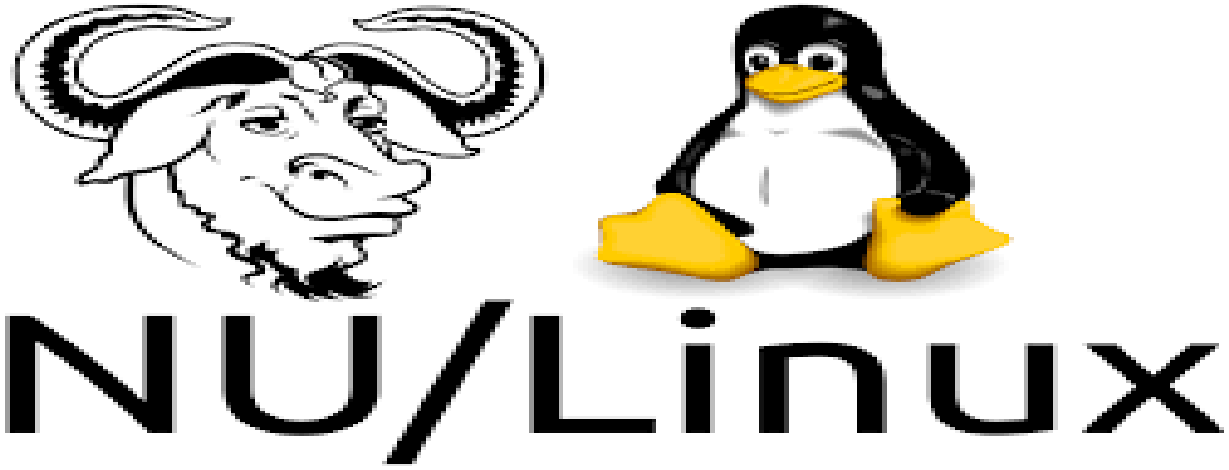
Ouvrons notre fichier **myscript.sh** et utilisons la commande **echo** pour imprimer du texte sur le terminal :

```
#!/bin/bash
echo "Hello World."
echo "We are executing the echo command"
```

Chapitre XI : Introduction au scripting Linux (Bash)



IV. Commentaires dans le fichier de script bash



IV. Commentaires dans le fichier de script bash

Les commentaires sont la partie importante des programmes informatiques.

Ce sont des lignes non exécutables. Les commentaires ne font qu'améliorer la lisibilité du code et aident à comprendre le but de notre code ou script.

Dans le fichier de script bash, nous pouvons ajouter les commentaires sur une seule ligne et les commentaires sur plusieurs lignes.

Les commentaires sur une seule ligne commencent par le symbole '**#**'.

Les commentaires multi-lignes commencent par l'apostrophe (**'**) et **:** est utilisé pour ajouter des lignes de commentaires. Voyons un exemple de commentaires dans les scripts bash.

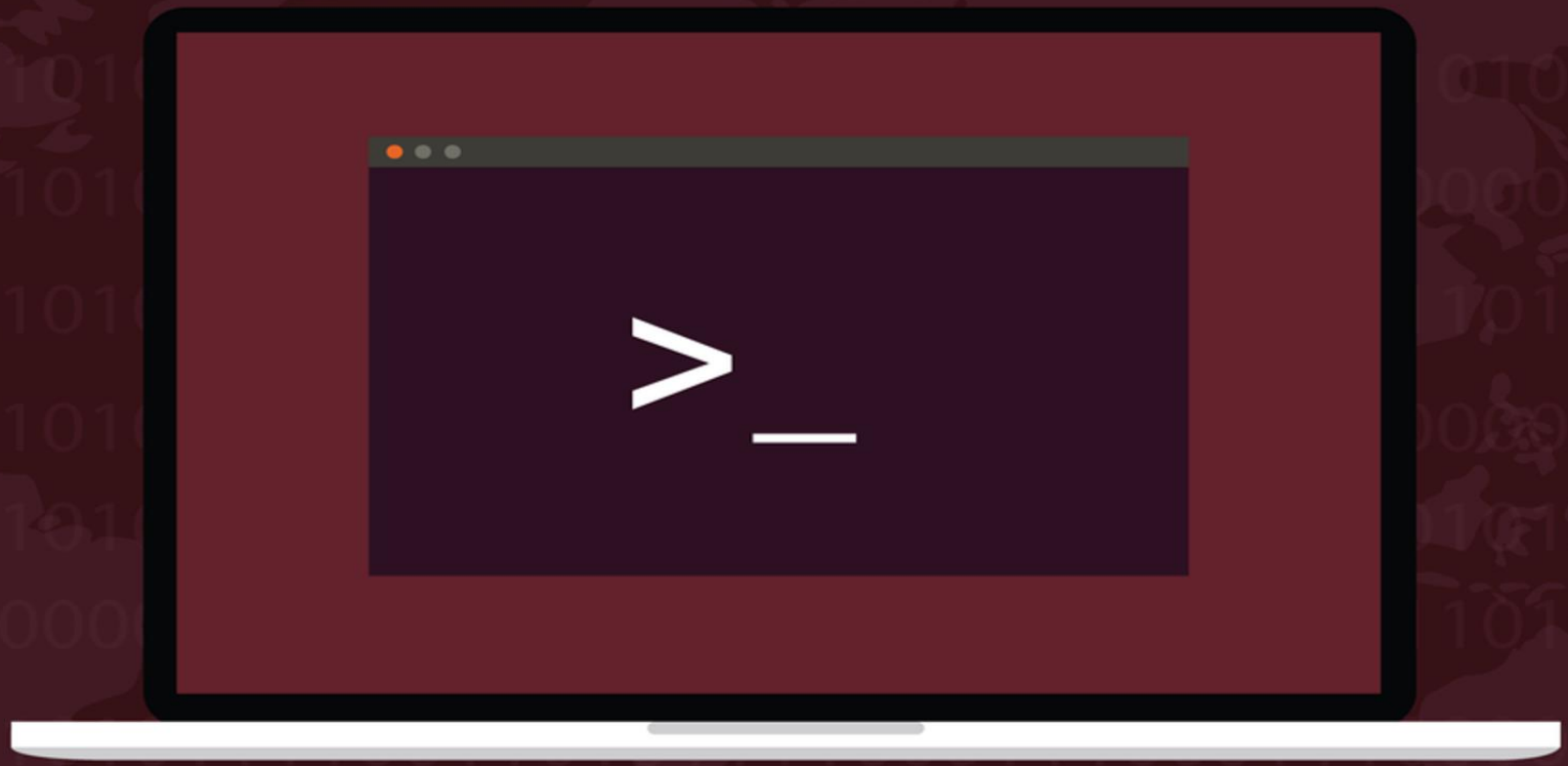
IV. Commentaires dans le fichier de script bash

```
#!/bin/bash
# Utilisation de la commande echo
echo "Hello World "
 '

Un commentaire multi lignes
En bash '
```

Chapitre XI : Introduction au scripting Linux (Bash)

V. Exemple d’utilsiation de commande dans un script bash : La commande ls



V. Exemple d’utilsiation de commande dans un script bash : La commande ls

Comme vous le savez bien, la commande **ls** répertorie les informations sur les fichiers et les répertoires dans un système de fichiers.

Avec diverses options, la commande ls peut être utilisée. Écrivons la commande ls dans notre fichier **myscript.sh** et exécutons-la.

La commande ls peut également être exécutée directement sur le terminal.

```
#!/bin/bash
# Utilisation de la commande ls
ls
```

Chapitre XI : Introduction au scripting Linux (Bash)

VI. Déclaration et utilisation des variables

VI. Déclaration et utilisation des variables

En rappel, les variables sont déclarées pour stocker les données ou certaines informations.

Les variables sont l'aspect important de tout langage de programmation.

Nous pouvons stocker des valeurs ou des informations dans des variables et plus tard nous pouvons les utiliser.

La déclaration et l'utilisation des variables sont très simples dans bash.

Les variables sont simplement déclarées en écrivant le nom de la variable.

Lors de l'accès ou de l'utilisation du nom de la variable, nous écrivons le symbole '**$**' avec la variable.

Déclarons la variable et utilisons-les dans notre fichier de script bash.

VI. Déclaration et utilisation des variables

```
#!/bin/bash
#Déclaration d'une variable VAR
VAR="Welcome to the bash scripting"
#Utilisation de la variable VAR
echo $VAR
#Déclaration d'une variable numérique num1
num1=10
# Déclaration d'une variable numérique num2
num2=20
#Calcul de la somme de num1 et num2 et stockage du résultat dans la variable num3
num3=$(($num1+$num2))
#Affichage du résultat de la somme
echo "La somme est : $num3"
```

Chapitre XI : Introduction au scripting Linux (Bash)

VII. Quelques variables d’environnements

VII. Quelques variables d’environnements

Si vous exécutez la commande **printenv** ou **env** sans aucun argument, une liste de toutes les variables d'environnement s'affichera.

Les commandes **printenv** et **env** impriment uniquement les variables d'environnement.

Si vous souhaitez obtenir une liste de toutes les variables, y compris l'environnement, le shell et les variables, ainsi que les fonctions du shell, vous pouvez utiliser la commande **set.**

VII. Quelques variables d’environnements

Voici quelques-unes des variables d'environnement les plus courantes :

$USER, $USERNAME : L'utilisateur actuellement connecté.
$HOME : Le répertoire personnel de l'utilisateur actuel.
$PWD : contient le chemin absolu vers le répertoire courant (permet de savoir où on est dans l'arborescence).
$OLDPWD : contient le chemin absolu vers le répertoire courant précédent (permet de savoir d'où on vient).
$EDITOR : L'éditeur de fichier par défaut à utiliser. C'est l'éditeur qui sera utilisé lorsque vous taperez editdans votre terminal.
$SHELL : Le chemin du shell de l'utilisateur actuel, tel que bash ou zsh.
$LOGNAME : Le nom de l'utilisateur actuel.
$PATH : Une liste de répertoires à rechercher lors de l'exécution des commandes. Lorsque vous exécutez une commande, le système recherche ces répertoires dans cet ordre et utilise le premier exécutable trouvé.
$LANG : Les paramètres régionaux actuels.
$TERM : L'émulation de terminal actuelle.
$MAIL : Emplacement où le courrier de l'utilisateur actuel est stocké.

CONCLUSION

MERCI DE VOTRE AIMABLE ATTENTION!!!