



WEBFORCE
BE THE CHANGE



TRAVAUX PRATIQUES - FILIÈRE INFRASTRUCTURE DIGITALE

M106 - Automatiser les tâches d'administration



63 heures



SOMMAIRE



1. DÉCOUVRIR LES CONCEPTS DE BASE DE LA PROGRAMMATION

- Découvrir la programmation structurée
- Découvrir la programmation orientée objet
 - Utiliser les conditions et les boucles

2. DÉVELOPPER DES PROGRAMMES

- Concevoir des programmes
- Créer un script pour faciliter les opérations de gestion

3. APPLIQUER L'ADMINISTRATION SYSTÈME

- Connaître les commandes de base d'administration
 - Administrer les ordinateurs à distance

4. CRÉER DES PROGRAMMES POUR LES TÂCHES D'ADMINISTRATION

- Automatiser les tâches redondantes
- Optimiser l'exécution des tâches d'administration

5. CRÉER DES FICHIERS LOGS

- Comprendre la persistance des données
 - Manipuler les fichiers logs
- Tester le fonctionnement des scripts

MODALITÉS PÉDAGOGIQUES



WEBFORCE
BE THE CHANGE



1

LE GUIDE DE SOUTIEN
Il contient le résumé théorique et le manuel des travaux pratiques



2

LA VERSION PDF
Une version PDF est mise en ligne sur l'espace apprenant et formateur de la plateforme WebForce Life



3

DES CONTENUS TÉLÉCHARGEABLES
Les fiches de résumés ou des exercices sont téléchargeables sur WebForce Life



4

DU CONTENU INTERACTIF
Vous disposez de contenus interactifs sous forme d'exercices et de cours à utiliser sur WebForce Life



5

DES RESSOURCES EN LIGNES
Les ressources sont consultables en synchrone et en asynchrone pour s'adapter au rythme de l'apprentissage



WEBFORCE
BE THE CHANGE



PARTIE 1

Découvrir les concepts de base de la programmation

Dans ce module, vous allez :

- Découvrir la programmation structurée
- Découvrir la programmation orientée objet
- Utiliser les conditions et les boucles



25 heures

ACTIVITÉ n° 1

Écriture d'algorithmes

Compétences visées :

- Les objets algorithmiques
- Structure d'un algorithme

Recommandations clés :

- Lire attentivement l'énoncé et les questions
- Se référer au cours



12 heures



CONSIGNES

1. Pour le formateur :

- Laisser à l'apprenant l'occasion de comprendre seul l'énoncé.
- S'assurer de la bonne compréhension du contexte avant de lui laisser le temps de réfléchir et répondre.
- Discuter les réponses des apprenants avant de donner la solution.

2. Pour l'apprenant :

- Lire et bien comprendre la question.
- Essayer de trouver par vous-même une réponse à cette question et la noter.
- Parcourir les réponses proposées.
- Pour chaque réponse : comparez-la à votre réponse et cochez-la si elle lui correspond ou bien est compatible.

3. Conditions de réalisation :

- Individuel ou par groupes (2 ou 3 maximum).
- Support de résumé théorique accompagnant.
- Stylo et feuille de papier.

4. Critères de réussite :

- Le stagiaire est-il capable de :
 - Identifier les objets algorithmiques ?
 - Déterminer la structure d'un algorithme ?
- Réponses correctes pour au moins 70 % des questions.



Activité 1 : écriture d'algorithmes

Exercices



Exercice 1 :

Écrire un algorithme qui permet de lire :

- Le montant hors taxes d'un article : MontantHT (un réel)
- Le taux de TVA : TauxTVA (un réel)
- Le nombre d'articles : NbArticles (un entier)

Puis, le montant total toutes taxes comprises TotalTTC sachant que : **TotalTTC = Nbarticles * MontantHT * (1 + TauxTVA)**

Activité 1 : écriture d'algorithmes

Exercices



Exercice 2 :

- Écrire un algorithme qui saisit un nombre pair et qui détermine combien de fois ce nombre est divisible par 2.
- Exemple : le nombre 8 est divisible 3 fois par 2 (ne pas contrôler la parité du nombre).

Activité 1 : écriture d'algorithmes

Exercices



Exercice 3 :

- Traiter l'activité 2 en effectuant le contrôle sur la parité du nombre et en utilisant la boucle **Répéter jusqu'à**.

Activité 1 : écriture d'algorithmes

Corrections



Correction exercice 1 :

Algorithme Application_1

VAR

MontantHT,TauxTVA,TotalTTC:réel

Nbarticles:entier

Début

Ecrire("Donner le nombre d'articles")

Lire(Nbarticles)

Ecrire("Donner le montant hors taxe")

Lire(montantHT)

Ecrire("Donner le taux de TVA")

Lire(TauxTVA)

TotalTTC ← Nbarticles * MontantHT * (1 + TauxTVA)

Ecrire("Le montant total toutes taxes comprises est ", TotalTTC)

FIN

Activité 1 : écriture d'algorithmes

Corrections



Correction exercice 2 :

Algorithme DIV2

VAR

NB, N : entier

Début

Ecrire("Introduire un nombre pair")

Lire(NB)

$N \leftarrow 0$

Tant que (NB Mod 2=0) **Faire**

$NB \leftarrow NB \text{ Div} 2$

$N \leftarrow N + 1$

Fin Tant que

Ecrire ("Le nombre que vous avez introduit est divisible par 2", N, "fois")

FIN

Activité 1 : écriture d'algorithmes

Corrections



Correction exercice 3 :

Algorithme DIV2

VAR

NB, N : entier

Début

Répéter

Ecrire("Introduire un nombre pair")

Lire(NB)

jusqu'à (NB Mod 2=0)

$N \leftarrow 0$

Répéter

$NB \leftarrow NB \text{ Div} 2$

$N \leftarrow N + 1$

Jusqu'à (NB Mod 2 <>0)

Ecrire ("Le nombre que vous avez introduit est divisible par 2", N, "fois")

FIN

ACTIVITÉ n° 2

Opérateurs, le traitement itératif et le traitement conditionnel

Compétences visées :

- Programmer un traitement itératif
- Programmer un traitement conditionnel
- Manipuler les opérateurs

Recommandations clés :

- Lire attentivement l'énoncé et les questions
- Se référer au cours



13 heures



CONSIGNES

1. Pour le formateur :

- Laisser à l'apprenant l'occasion de comprendre seul l'énoncé.
- S'assurer de la bonne compréhension du contexte avant de lui laisser le temps de réfléchir et répondre.
- Discuter les réponses des apprenants avant de donner la solution.

2. Pour l'apprenant :

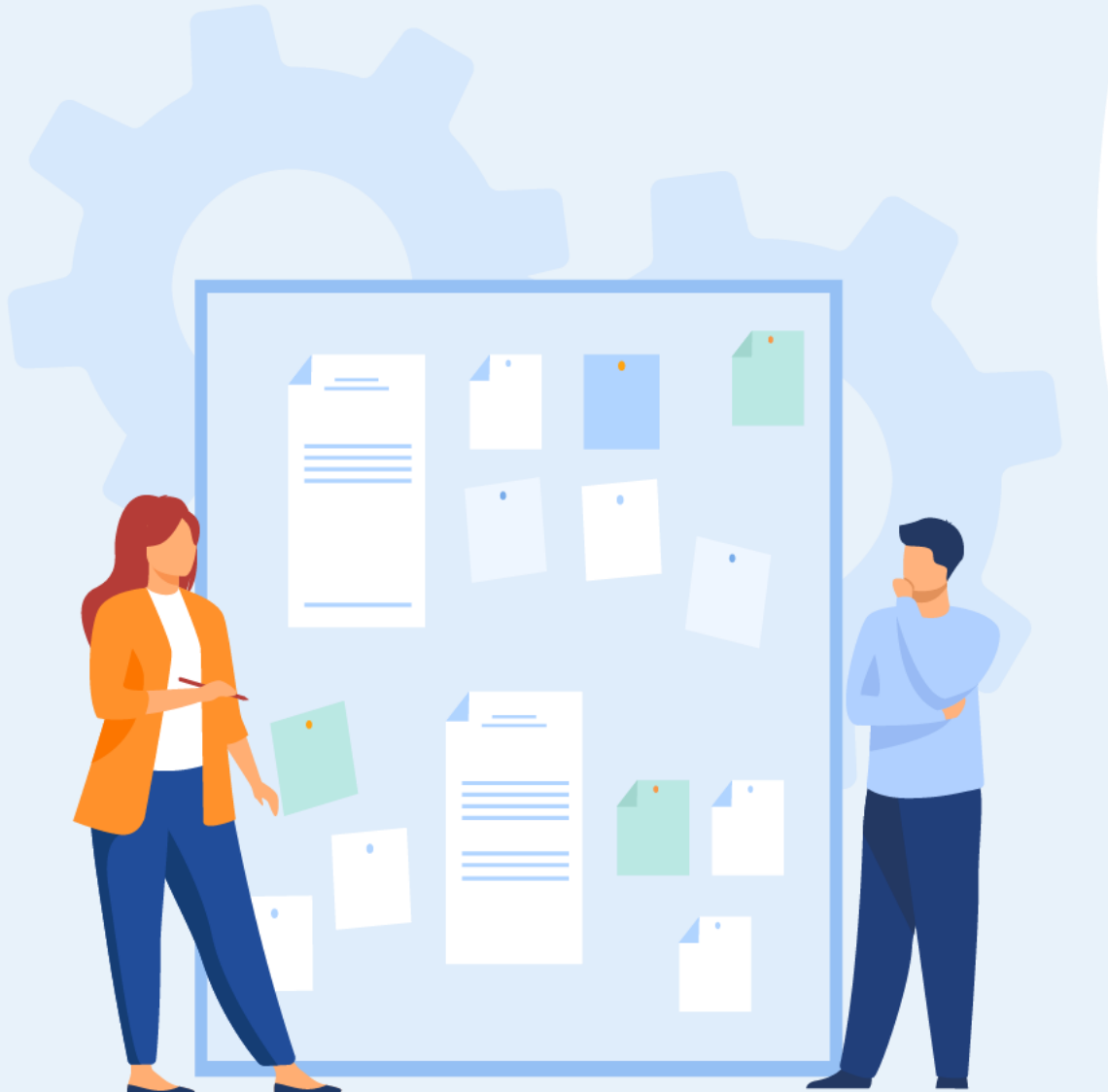
- Lire et bien comprendre la question.
- Essayer de trouver par vous-même une réponse à cette question et la noter.
- Parcourir les réponses proposées.
- Pour chaque réponse : comparez-la à votre réponse et cochez-la si elle lui correspond ou bien est compatible.

3. Conditions de réalisation :

- Individuel ou par groupes (2 ou 3 maximum).
- Support de résumé théorique accompagnant.
- Stylo et feuille de papier.

4. Critères de réussite :

- Le stagiaire est-il capable de :
 - Programmer un traitement itératif ?
 - Programmer un traitement conditionnel ?
- Réponses correctes pour au moins 70 % des questions.



Activité 2 : opérateurs, le traitement itératif et le traitement conditionnel

Exercices



Exercice 1 :

- Soit n un entier supérieur à
- Écrire un script Python qui détermine le seul entier p , positif ou nul tel que $2^p \leq n < 2^{p+1}$.

Activité 2 : opérateurs, le traitement itératif et le traitement conditionnel

Exercices



Exercice 2 :

- Écrire un script Python qui permet de convertir un entier positif a représenté dans une base b ($2 \leq b \leq 8$) en base 10 et inversement.
- Le programme affiche le message "BRAVO" si le résultat de la deuxième conversion nous ramène au nombre initial a .

Activité 2 : opérateurs, le traitement itératif et le traitement conditionnel

Exercices



Exercice 3 :

- Soit une suite d'entiers donnée, les moyennes mobiles sont les moyennes calculées à partir de chaque trois nombres consécutifs.

Exemple :

La suite des entiers 4 3 6 7 8

Les moyennes mobiles sont 4.33 5.33 7

1. Écrire l'algorithme Moy_Mobile qui permet d'introduire une suite de n entiers et d'afficher l'ensemble des moyennes mobiles. L'algorithme est censé calculer au moins une moyenne mobile.



NB

- L'utilisateur indique l'arrêt de la saisie des valeurs en entrant la valeur -1

2. Traduire l'algorithme Moy_Mobile en Python.

Activité 2 : opérateurs, le traitement itératif et le traitement conditionnel

Corrections



Correction exercice 1 :

```
n=int(input("donner un entier n"))
p=0
while 2**(p+1)<=n:
    p+=1
print("valeur de p",p)
```

Activité 2 : opérateurs, le traitement itératif et le traitement conditionnel

Corrections



Correction exercice 2 :

```
#-----  
#-----Exercice 2-----  
#-----  
a=int(input("donner l'entier à convertir"))  
#contrôle à la saisie de la base 2<=b<=8  
b=1  
while b<2 or b>8:  
b=int(input("donner la base b"))  
#-----on suppose que a est bien représenté en base b -----  
#-----Conversion de a en base 10 -----  
n=a  
p=0  
res=0  
while(n!=0):  
    r=n%10  
res=res+(r*(b**p))  
    p=p+1  
n=n//10
```

```
print("le résultat de la première conversion est",res)  
# -----Conversion du résultat en base b -----  
-----  
n=res  
p=0  
res=0  
print('n1= ',n)  
while(n!=0):  
    r=n%b  
res=res+(r*(10**p))  
    p=p+1  
n=n//b  
print("le résultat de la deuxième conversion est",res)  
if a==res:  
    print('BRAVO')  
else:  
    print('NO')
```

Activité 2 : opérateurs, le traitement itératif et le traitement conditionnel

Corrections



Correction exercice 3 :

```
# -----on ne travaille pas avec n : on ne connaît pas le nombre d'entiers à saisir-----  
# -----Le script calcule au moins une moyenne -----  
  
a=-1  
while a==-1:  
    a=int(input('donner a'))  
    b=-1  
while b==-1:  
    b=int(input('donner b'))  
    c=-1  
while c==-1:  
    c=int(input('donner c'))  
while c!=-1:  
    moy=(a+b+c)/3  
    print(moy)  
    a=b  
    b=c  
    c=int(input('donner c'))
```



PARTIE 2

Développer des programmes

Dans ce module, vous allez :

- Concevoir des programmes
- Créer un script pour faciliter les opérations de gestion



12 heures



ACTIVITÉ n° 1

Traduction des algorithmes en scripts

Compétences visées :

- Traduire un algorithme en un script Python

Recommandations clés :

- Lire attentivement l'énoncé et les questions
- Se référer au cours



02 heures

CONSIGNES

1. Pour le formateur :

- Laisser à l'apprenant l'occasion de comprendre seul l'énoncé.
- S'assurer de la bonne compréhension du contexte avant de lui laisser le temps de réfléchir et répondre.
- Discuter les réponses des apprenants avant de donner la solution.

2. Pour l'apprenant :

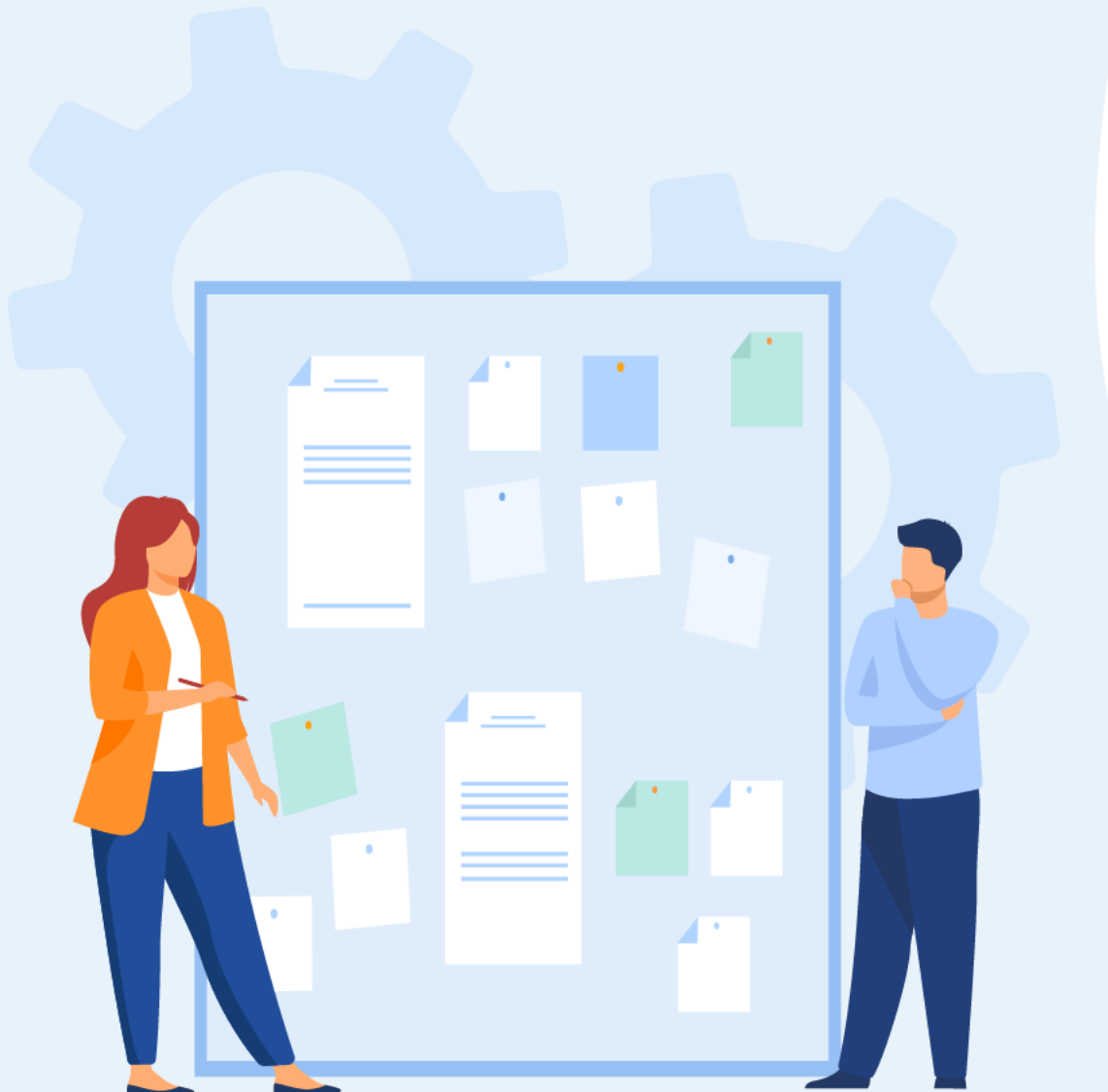
- Lire et bien comprendre la question.
- Essayer de trouver par vous-même une réponse à cette question et la noter.
- Parcourir les réponses proposées.
- Pour chaque réponse : comparez-la à votre réponse et cochez-la si elle lui correspond ou bien est compatible.

3. Conditions de réalisation :

- Individuel ou par groupes (2 ou 3 maximum).
- Support de résumé théorique accompagnant.
- Stylo et feuille de papier.

4. Critères de réussite :

- Le stagiaire est-il capable de :
 - Traduire un algorithme en un script ?
- Réponses correctes pour au moins 70 % des questions.



Activité 1 : traduction des algorithmes en scripts

Exercice



Exercice : traduire les blocs suivants en Python

- **Bloc n° 1**

```
v ← expression  
cl.saisir(x)  
e.afficher("Entrez n: ")  
cl.saisir(n)  
e.afficher("Total : ", t)
```

- **Bloc n° 2**

```
d : Durée // définition  
d1: Durée // définition  
Accès aux champs de d :  
d.h ← 12  
d.m ← 25  
d.s ← 40  
d1 ← Durée(15, 34, 12)
```

- **Bloc n° 3**

```
Point: type agrégat  
x, y : réels  
fagrégat  
Utilisation :  
s1 : le Point (1, 12.2)  
s2 : Point
```

- **Bloc n° 4**

```
t : tableau sur [0..NMAX-1] de type éléments  
nbje : tableau sur [0..11] d'entiers =  
[31,28,31,30,31,30,31,31,30,31,30,31]  
chiffres : tableau sur [0..9] de caractères  
= ['0','1', '2', '3', '4', '5', '6', '7', '8', '9']
```


Activité 1 : traduction des algorithmes en scripts

Exercice



Exercice : traduire les blocs suivants en Python (suite)

- Bloc n° 5

fonction nba(x : chaine) → entier ≥ 0

// nba(x) renvoie le nombre de A contenus dans la chaine x

lexique de nba

na : entier ≥ 0 // na = nombre de a (pp)

i : entier ≥ 0 // indice de parcours de x

algorithme de F

i = 0 ; na ← 0

tantque i < longueur(x) faire

si nième(x,i) = 'a' ou nième(x,i) = 'A' alors na ← na + 1 fsi

i ← i + 1

ftq

renvoyer (na)

Activité 1 : traduction des algorithmes en scripts

Correction



Correction exercice :

- Bloc n° 1

```
#affectation :  
v = expression  
  
# saisie au clavier:  
x = input() # x est une chaine  
x = int(input()) # saisie convertie  
n = int(input("Entrez n: "))  
  
#affichage :  
print("Total :", t)  
  
# affichage sans retour à la ligne :  
print("Total :", t, end="")
```

- Bloc n° 2

```
d = Duree() # définition  
d1 = Duree() # définition  
  
# Accès aux champs de d :  
d.h = 12  
d.m = 25  
d.s = 40  
d1 = Duree(15, 34, 12)
```

- Bloc n° 3

```
class Point(object):  
    # Représentation du type Point  
    def __init__(self, x=0.0, y=0.0)  
        self.x = float(x) # abscisse  
        self.y = float(y) # ordonnée  
  
    # Utilisation  
s1 = Point(1 ,12.2)  
s2 = Point()
```

Activité 1 : traduction des algorithmes en scripts

Correction



Correction exercice (suite) :

- Bloc n° 4

```
#En Python il n'y a pas de tableau en natif, il n'y a que des listes
# Les éléments d'une liste sont indicés à partir de 0
# Pour créer un tableau de n éléments, il faut donc construire un liste de n
éléments:
t = [0] * NMAX
# Pour les tableaux prédéfinis c'est très simple :
nbjm = [31,28,31,30,31,30,31,31,30,31,30,31]
chiffres = ['0','1','2','3','4','5','6','7','8','9']
```

- Bloc n° 5

```
def nba(x):
    # nba(x) renvoie le nombre de 'a' contenus dans la chaîne x
    # lexique de nba
    na = int() # Nombre de a de la pp de x
    i = int() # intermédiaire : indice de parcours de x
    # algorithme de nba
    i = 0
    na = 0
    while i < len(x):
        if (x[i] == 'a') or (x[i] == 'A'):
            na = na + 1
        #fsi
        i = i + 1
    #ftq
    return na
```



ACTIVITÉ n° 2

Les commandes PowerShell

Compétences visées :

- Tester les commandes PowerShell

Recommandations clés :

- Lire attentivement l'énoncé et les questions
- Se référer au cours



02 heures

CONSIGNES

1. Pour le formateur :

- Laisser à l'apprenant l'occasion de comprendre seul l'énoncé.
- S'assurer de la bonne compréhension du contexte avant de lui laisser le temps de réfléchir et répondre.
- Discuter les réponses des apprenants avant de donner la solution.

2. Pour l'apprenant :

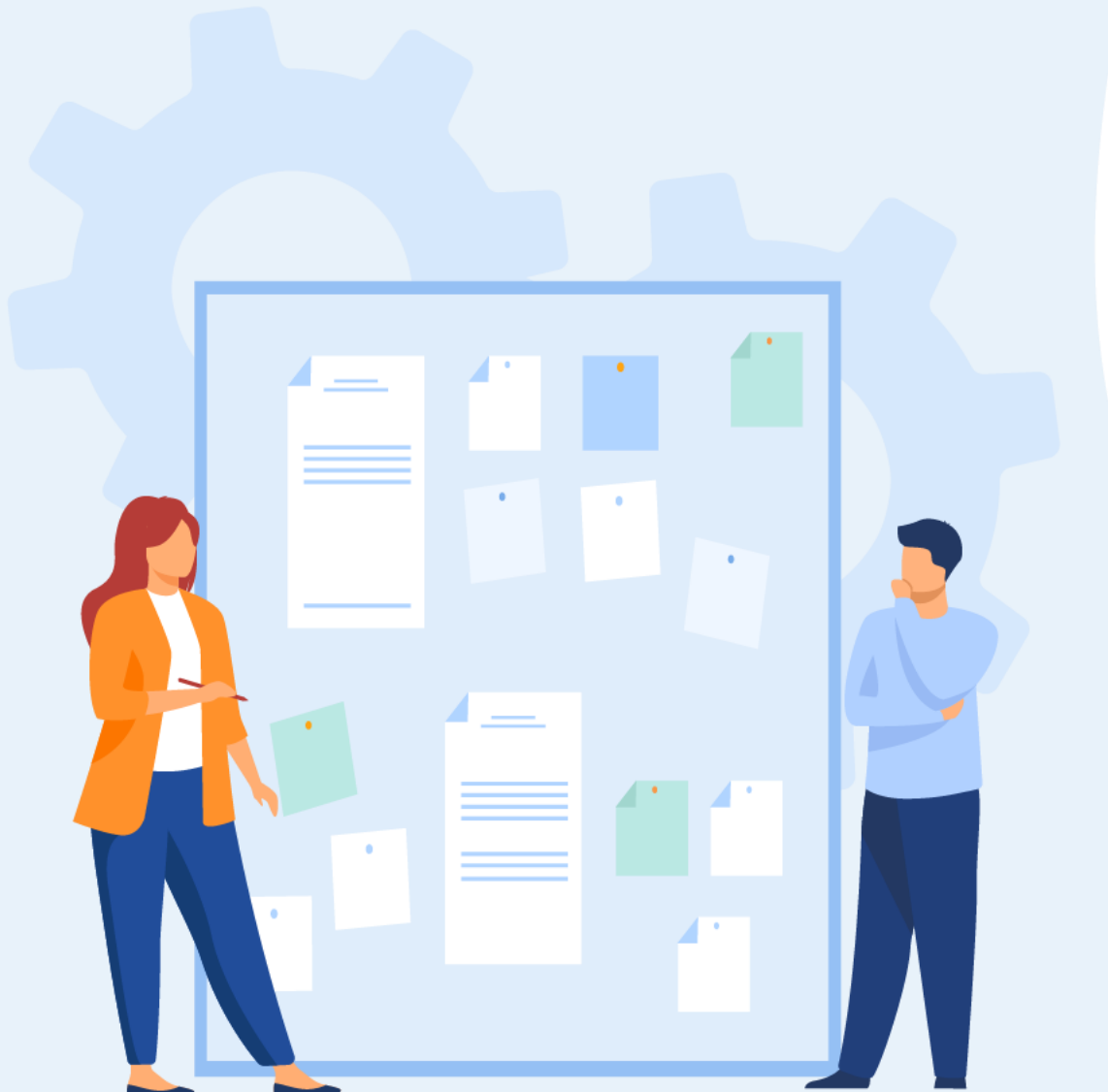
- Lire et bien comprendre la question.
- Essayer de trouver par vous-même une réponse à cette question et la noter
- Parcourir les réponses proposées.
- Pour chaque réponse : comparez-la à votre réponse et cochez-la si elle lui correspond ou bien est compatible.

3. Conditions de réalisation :

- Individuel ou par groupes (2 ou 3 maximum).
- Support de résumé théorique accompagnant.
- Stylo et feuille de papier.

4. Critères de réussite :

- Le stagiaire est-il capable de :
 - Utiliser les commandes PowerShell ?
- Réponses correctes pour au moins 70 % des questions.



Activité 2 : les commandes PowerShell

Exercices



Exercices :

1. Afficher toutes les commandes disponibles pour une utilisation dans votre session en cours.
2. Trouver le contenu de l'aide pour une commande particulière.
3. Contrôler le niveau de sécurité entourant les scripts PowerShell.
4. Quels services sont installés sur le système ?
5. Extraire des données que vous pouvez utiliser dans un rapport ou envoyer à quelqu'un d'autre.
6. Paser les événements logs de votre machine.
7. Effacer les entrées de votre historique de commandes.

Activité 2 : les commandes PowerShell

Corrections



Correction exercice 1 :

- **Get-Command** est une commande facile à utiliser qui affiche toutes les commandes disponibles pour une utilisation dans votre session en cours. Tapez simplement cette commande :

```
get-command
```

- La sortie ressemblera à ceci :

CommandType	Name	Definition
-----	-----	-----
Cmdlet	Add-Content	Add-Content [-Path] <String[...]
Cmdlet	Add-History	Add-History [[-InputObject] ...]
Cmdlet	Add-Member	Add-Member [-MemberType]

Activité 2 : les commandes PowerShell

Corrections



Correction exercice 2 :

- La commande Get-Help est essentielle pour toute personne utilisant PowerShell, offrant un accès rapide aux informations dont vous avez besoin pour exécuter et travailler avec toutes les commandes disponibles.
- Si vous voulez des exemples, par exemple, vous entreriez ce qui suit :

```
Get-Help [[-Name] <String>] [-Path <String>] [-Category <String[]>] [-Component <String[]>]  
>]  
[-Functionality <String[]>] [-Role <String[]>] [-Examples] [<CommonParameters>]
```


Activité 2 : les commandes PowerShell

Corrections



Correction exercice 3 :

- Microsoft désactive les scripts par défaut pour empêcher l'exécution de scripts malveillants dans l'environnement PowerShell. Cependant, les développeurs souhaitent pouvoir écrire et exécuter des scripts. La commande Set ExecutionPolicy vous permet donc de contrôler le niveau de sécurité entourant les scripts PowerShell. Vous pouvez définir l'un des quatre niveaux de sécurité :
 1. **Restreint** : il s'agit du niveau de sécurité par défaut qui empêche l'exécution des scripts PowerShell. Dans ce niveau de sécurité, vous ne pouvez entrer des commandes que de manière interactive.
 2. **Tous signés** : ce niveau de sécurité permet aux scripts de s'exécuter uniquement s'ils sont signés par un éditeur digne de confiance.
 3. **Signé à distance** : dans ce niveau de sécurité, tous les scripts PowerShell créés localement sont autorisés à s'exécuter. Les scripts créés à distance ne sont autorisés à s'exécuter que s'ils ont été signés par un éditeur réputé.
 4. **Sans restriction** : comme son nom l'indique, le niveau de sécurité sans restriction permet à tous les scripts de s'exécuter en supprimant toutes les restrictions de la politique d'exécution.
- De même, si vous travaillez dans un environnement inconnu, vous pouvez facilement savoir quelle est la politique d'exécution actuelle à l'aide de cette commande :

```
Get-ExecutionPolicy
```

Activité 2 : les commandes PowerShell

Corrections



Correction exercice 4 :

- Il est utile de savoir quels services sont installés sur le système.
- Vous pouvez facilement accéder à ces informations avec la commande suivante :

```
Get-Service
```

- La sortie ressemblera à ce qui suit :

```
Status Name DisplayName
-----
Running AdobeActiveFile... Adobe Active File Monitor V4
Stopped Alerter Alerter
Running ALG Application Layer Gateway Service
Stopped AppMgmt Application Management
Running ASChannel Local Communication Channel
```

Activité 2 : les commandes PowerShell

Corrections



Correction exercice 5 :

- Si vous avez besoin d'extraire des données que vous pouvez utiliser dans un rapport ou envoyer à quelqu'un d'autre, le ConvertTo-HTML est un moyen simple de le faire. Pour l'utiliser, dirigez la sortie d'une autre commande vers la commande ConvertTo-HTML et utilisez le commutateur -Property pour spécifier les propriétés de sortie souhaitées dans le fichier HTML. Vous devrez également fournir un nom de fichier.
- Par exemple, le code suivant crée une page HTML qui répertorie les alias PowerShell dans la console actuelle :

```
PS C:\> get-alias | convertto-html > aliases.htm  
PS C:\> invoke-item aliases.htm
```

Activité 2 : les commandes PowerShell

Corrections



Correction exercice 6 :

Vous pouvez utiliser PowerShell pour analyser les journaux d'événements de votre ordinateur à l'aide de l'applet de commande Get-EventLog. Plusieurs paramètres sont disponibles. Utilisez le commutateur -Log suivi du nom du fichier log pour afficher un log spécifique. Vous utiliserez la commande suivante, par exemple, pour afficher le log des applications :

```
Get-EventLog -Log "Application"
```

Correction exercice 7 :

Que faire si vous souhaitez effacer les entrées de votre historique de commandes ?

Utilisez la cmdlet Clear-History. Vous pouvez également l'utiliser pour supprimer uniquement des commandes spécifiques. Par exemple, la commande suivante supprimerait les commandes qui incluent "help" ou se terminent par "command".

```
PS C:\> Clear-History -Command *help*, *command
```

Si vous souhaitez ajouter des entrées à une session, utilisez :

```
Add-History
```



ACTIVITÉ n° 3

Les commandes Bash

Compétences visées :

- Exécuter les commandes Bash

Recommandations clés :

- Lire attentivement l'énoncé et les questions
- Se référer au cours



04 heures

CONSIGNES

1. Pour le formateur :

- Laisser à l'apprenant l'occasion de comprendre seul l'énoncé.
- S'assurer de la bonne compréhension du contexte avant de lui laisser le temps de réfléchir et répondre.
- Discuter les réponses des apprenants avant de donner la solution.

2. Pour l'apprenant :

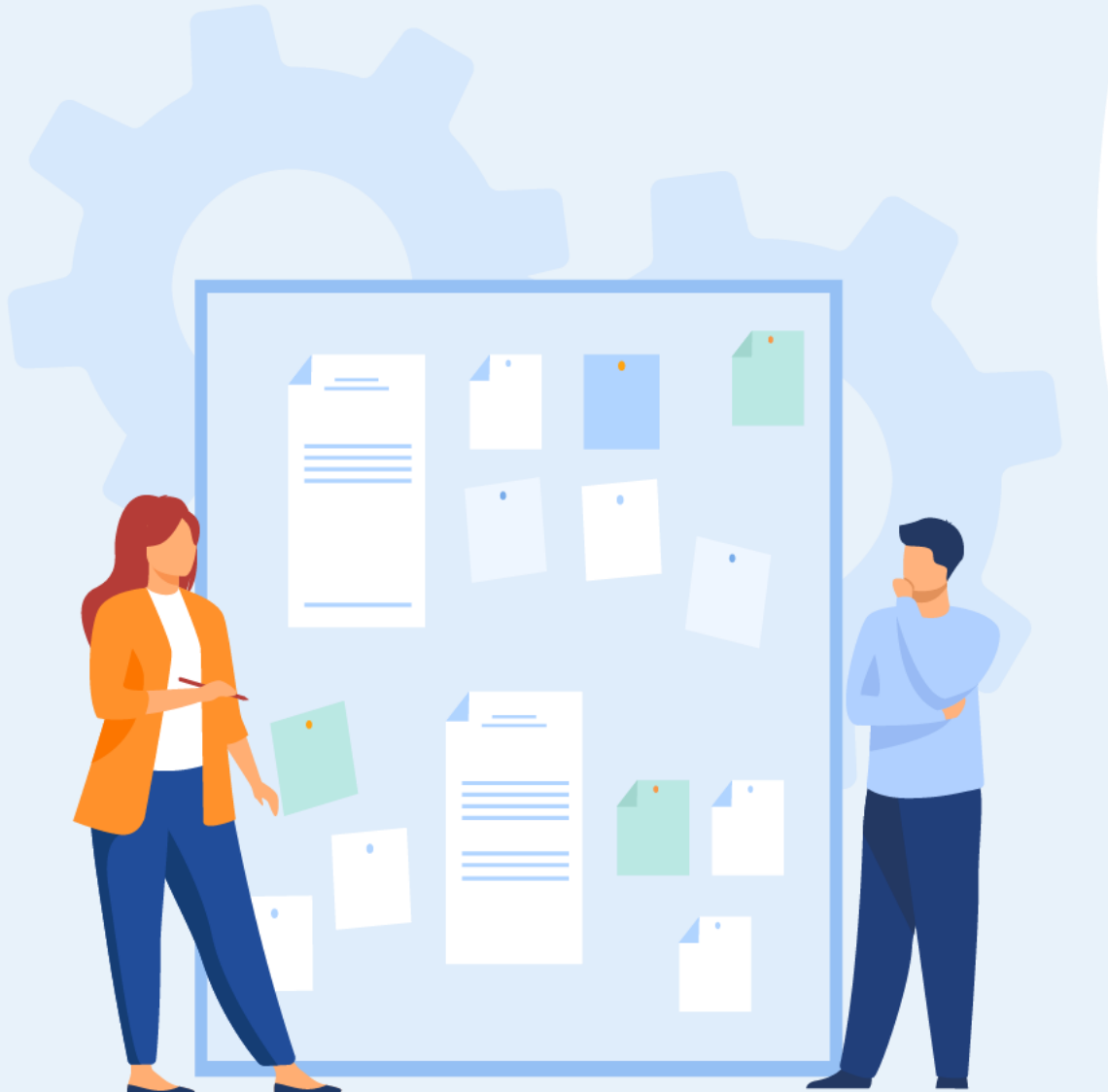
- Lire et bien comprendre la question.
- Essayer de trouver par vous-même une réponse à cette question et la noter.
- Parcourir les réponses proposées.
- Pour chaque réponse : comparez-la à votre réponse et cochez-la si elle lui correspond ou bien est compatible.

3. Conditions de réalisation :

- Individuel ou par groupes (2 ou 3 maximum).
- Support de résumé théorique accompagnant.
- Stylo et feuille de papier.

4. Critères de réussite :

- Le stagiaire est-il capable de :
 - Utiliser les commandes Bash ?
- Réponses correctes pour au moins 70 % des questions.



Activité 3 : les commandes Bash

Exercices



Exercice 1 : créer et exécuter le premier programme Bash

1. Vous pouvez exécuter un script Bash à partir du terminal ou en exécutant n'importe quel fichier Bash. Exécutez la commande suivante depuis le terminal pour exécuter une instruction Bash très simple. La sortie de la commande sera "Hello World".

```
$ echo "Hello world"
```

2. Ouvrez n'importe quel éditeur pour créer un fichier Bash. Ici, l'éditeur nano est utilisé pour créer le fichier et le nom du fichier est défini sur "First.sh".
3. Ajoutez le script Bash suivant au fichier et enregistrez le fichier.

```
#!/bin/bash  
echo "Hello world"
```

4. Exécutez le fichier Bash de deux manières. Une façon consiste à utiliser la commande Bash et une autre consiste à définir l'autorisation d'exécution sur le fichier Bash et à exécuter le fichier.

Activité 3 : les commandes Bash

Exercices



Exercice 2 : utilisation de la commande echo

1. Vous pouvez utiliser la commande echo avec diverses options. Certaines options utiles sont mentionnées dans l'exemple suivant. Lorsque vous utilisez la commande echo sans aucune option, une nouvelle ligne est ajoutée par défaut. L'option "-n" est utilisée pour imprimer n'importe quel texte sans nouvelle ligne et l'option "-e" est utilisée pour supprimer les caractères de barre oblique inverse de la sortie. Créez un nouveau fichier Bash avec un nom, "echo_example.sh" et ajoutez le script suivant.

```
#!/bin/bash
echo "Printing text with newline"
echo -n "Printing text without newline"
echo -e "\nRemoving \t backslash \t characters\n"
```

Exécutez le fichier avec la commande Bash.

Exercice 3 : utilisation du commentaire

1. Le symbole "#" est utilisé pour ajouter un commentaire sur une seule ligne dans le script Bash. Créez un nouveau fichier nommé "comment_example.sh" et ajoutez le script suivant avec un commentaire sur une seule ligne.

```
#!/bin/bash
# Add two numeric value
((sum=25+35))
#Print the result
echo $sum
```

Exécutez le fichier avec la commande Bash.

Activité 3 : les commandes Bash

Exercices



Exercice 4 : utilisation de la boucle while

1. Créez un fichier Bash avec le nom "while_example.sh" pour connaître l'utilisation de la boucle while. Dans l'exemple, la boucle while sera itérée 5 fois. La valeur de la variable count augmentera de 1 à chaque étape. Lorsque la valeur de la variable count sera 5, la boucle while se terminera.

```
#!/bin/bash
valid=true
count=1
while [ $valid ]
do
echo $count
if [ $count -eq 5 ];
then
break
fi
((count++))
done
```

Exécutez le fichier avec la commande Bash.

Activité 3 : les commandes Bash

Exercices



Exercice 5 : obtenir l'entrée de l'utilisateur

1. La commande "read" est utilisée pour prendre l'entrée de l'utilisateur dans Bash. Créez un fichier nommé "user_input.sh" et ajoutez le script suivant pour prendre l'entrée de l'utilisateur. Ici, une valeur de chaîne sera extraite de l'utilisateur et affichera la valeur en combinant une autre valeur de chaîne.

```
#!/bin/bash
echo "Enter Your Name"
read name
echo "Welcome $name to LinuxHint"
```

Exécutez le fichier avec la commande Bash.

Activité 3 : les commandes Bash

Corrections



Correction exercice 1 : créer et exécuter le premier programme BASH

1. Vous pouvez exécuter un script Bash à partir du terminal ou en exécutant n'importe quel fichier Bash. Exécutez la commande suivante depuis le terminal pour exécuter une instruction Bash très simple. La sortie de la commande sera "Hello World".

```
ubuntu@ubuntu-VirtualBox:~$ echo "Hello World"
Hello World
ubuntu@ubuntu-VirtualBox:~$
```

2. Ouvrez n'importe quel éditeur pour créer un fichier Bash. Ici, l'éditeur nano est utilisé pour créer le fichier et le nom du fichier est défini sur "First.sh".

```
$ nano First.sh
```

3. Ajoutez le script Bash suivant au fichier et enregistrez le fichier.

```
GNU nano 2.8.6 File: First.sh Modified
#!/bin/bash
echo "Hello World"

^G Get Help  ^O Write Out ^W Where Is  ^K Cut Text  ^J Justify   ^C Cur Pos
^X Exit      ^R Read File ^\ Replace   ^U Uncut Text ^T To Linter ^_ Go To Line
```

Activité 3 : les commandes Bash

Corrections



Correction exercice 1 : créer et exécuter le premier programme BASH (suite)

4.

```
$ bash First.sh
```

Ou

```
$ chmod a+x First.sh  
$ ./First.sh
```

```
ubuntu@ubuntu-VirtualBox:~$ bash First.sh  
Hello World  
ubuntu@ubuntu-VirtualBox:~$ chmod a+x First.sh  
ubuntu@ubuntu-VirtualBox:~$ ./First.sh  
Hello World  
ubuntu@ubuntu-VirtualBox:~$
```

Activité 3 : les commandes Bash

Corrections



Correction exercice 2 : utilisation de la commande echo

```
ubuntu@ubuntu-VirtualBox:~/code$ bash echo_example.sh
Printing text with newline
Printing text without newline
Removing      backslash      characters

ubuntu@ubuntu-VirtualBox:~/code$
```

Correction exercice 3 : utilisation du commentaire

```
ubuntu@ubuntu-VirtualBox:~/code$ bash comment_example.sh
60
ubuntu@ubuntu-VirtualBox:~/code$ █
```

Activité 3 : les commandes Bash

Corrections



Correction exercice 4 : utilisation de la boucle while

```
$ bash while_example.sh
```

```
ubuntu@ubuntu-VirtualBox:~/code$ bash while_example.sh
1
2
3
4
5
ubuntu@ubuntu-VirtualBox:~/code$
```

Correction exercice 5 : obtenir l'entrée de l'utilisateur

```
$ bash user_input.sh
```

```
ubuntu@ubuntu-VirtualBox:~/code$ bash user_input.sh
Enter Your Name
Fahmida
Welcome Fahmida to LinuxHint
ubuntu@ubuntu-VirtualBox:~/code$
```



ACTIVITÉ n° 4

Modules spécifiques pour les opérations de gestion

Compétences visées :

- Initialiser et créer un ensemble
- Réaliser des opérations sur les listes

Recommandations clés :

- Lire attentivement l'énoncé et les questions
- Se référer au cours
- Se mettre dans le contexte de la problématique posée dans l'activité



04 heures

CONSIGNES

1. Pour le formateur :

- Laisser à l'apprenant l'occasion de comprendre seul l'énoncé.
- S'assurer de la bonne compréhension du contexte avant de lui laisser le temps de réfléchir et répondre.
- Discuter les réponses des apprenants avant de donner la solution.

2. Pour l'apprenant :

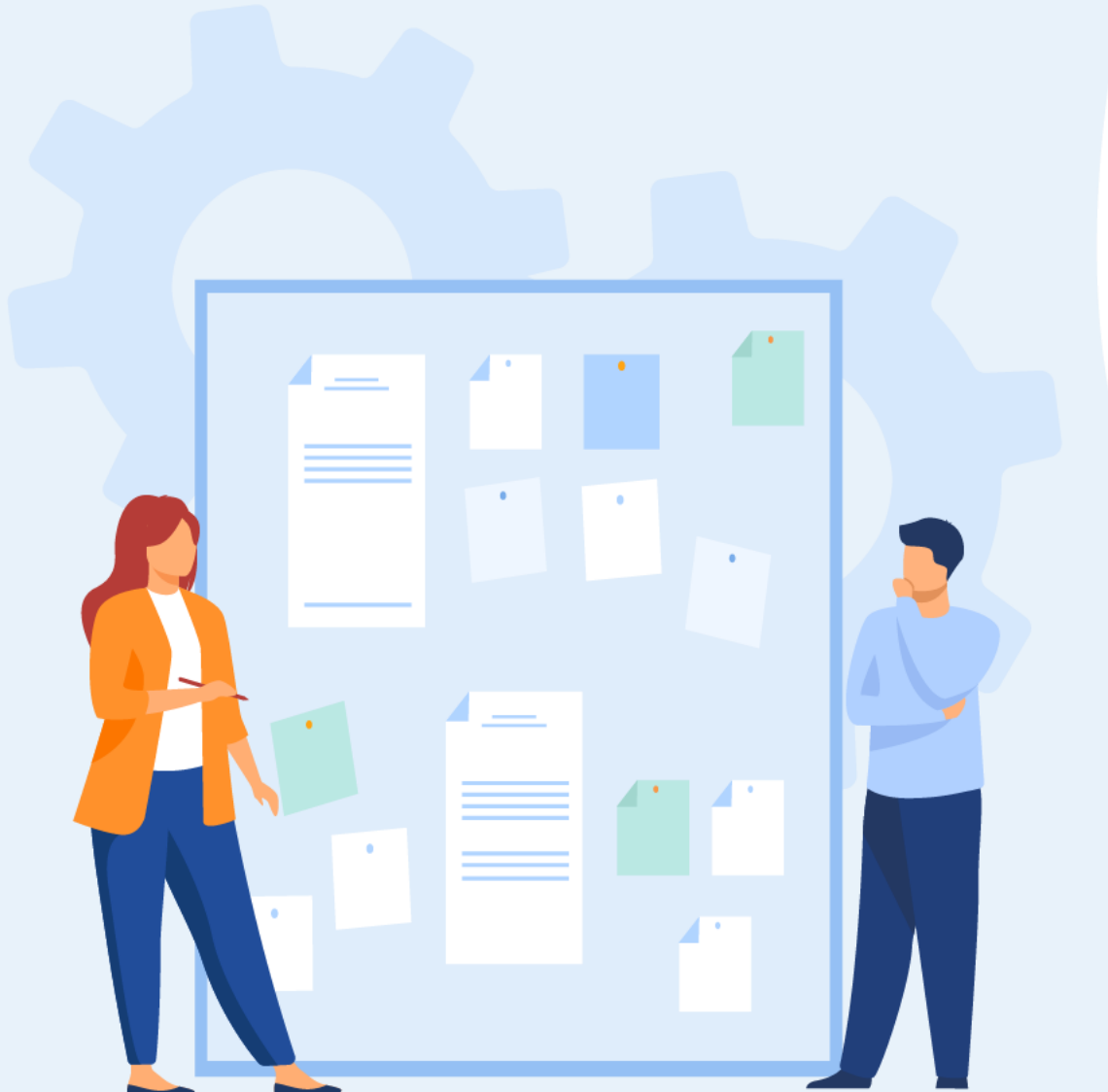
- Lire et bien comprendre la question.
- Essayer de trouver par vous-même une réponse à cette question et la noter.
- Parcourir les réponses proposées.
- Pour chaque réponse : comparez-la à votre réponse et cochez-la si elle lui correspond ou bien est compatible.

3. Conditions de réalisation :

- Individuel ou par groupes (2 ou 3 maximum).
- Support de résumé théorique accompagnant.
- Stylo et feuille de papier.

4. Critères de réussite :

- Le stagiaire est-il capable de :
 - Manipuler les ensembles et les listes ?
- Réponses correctes pour au moins 70 % des questions.



Activité 4 : modules spécifiques pour les opérations de gestion

Exercices



Exercice 1 : les listes

1. Écrire un script qui détermine si une chaîne de caractères donnée est un palindrome (c'est-à-dire une chaîne qui peut se lire indifféremment dans les deux sens), comme par exemple "SOS", "radar", "elle".
2. Écrire un script qui accepte une chaîne et crée un palindrome en ajoutant à la fin une copie inversée de cette chaîne.

Exercice 2 : les listes

1. Soit la liste $L=[0,1,2,3,4,5,6,7,8,9,10]$, créer les listes suivantes par compréhension à partir de L :

a) $L1=[0,2,4,6,8,10,12,14,16,18,20]$

b) $L2=[[0,0],[1,1],[2,2],[3,3],[4,4],[5,5],[6,6],[7,7],[8,8],[9,9],[10,10]]$

c) $L3=[0,0,1,1,2,2,3,3,4,4,5,5,6,6,7,7,8,8,9,9,10,10]$

d) $L4=[0,1,2,3,4,5,6,7,8,9,10, 0,1,2,3,4,5,6,7,8,9,10]$

e) $L5=[1,2,2,3,3,3,4,4,4,4,5,5,5,5,6,6,6,6,6,7,7,7,7,7,7,8,8,8,8,8,8,9,9,9,9,9,9,9,10,10,10,10,10,10,10,10,10,10]$

2. Soient les listes suivantes :

- $L1= [31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31]$

- $L2 = ['Janvier', 'Février', 'Mars', 'Avril', 'Mai', 'Juin', 'Juillet', 'Août', 'Septembre', 'Octobre', 'Novembre', 'Décembre']$

3. Soient les listes suivantes :

- $L1= [31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31]$

- $L2 = ['Janvier', 'Février', 'Mars', 'Avril', 'Mai', 'Juin', 'Juillet', 'Août', 'Septembre', 'Octobre', 'Novembre', 'Décembre']$

Créer la liste $L3$ **par compréhension** à partir de $L1$ et $L2$ telle que :

$L3= [('Janvier',31),('Février',28),('Mars',31),...]$.

Activité 4 : modules spécifiques pour les opérations de gestion

Exercices



Exercice 3 : les listes

- Écrire un script Python qui cherche le premier et le deuxième maximum présents dans une liste L **en faisant un seul parcours** de la liste.

Exemple :

L= [32, 5, 12, 8, 3, 75, 2, 15]. Le programme affichera : "le premier maximum est 75, le deuxième est 32"

Activité 4 : modules spécifiques pour les opérations de gestion

Corrections



Correction exercice 1 : les listes

----- Solution 1 avec while -----

```
palindrome=True
i=0
while i<=(n//2)-1 and palindrome:
    if c[i]!=c[n-i-1]:
        palindrome=False
    else:
        i=i+1
if palindrome:
    print('palindrome')
else:
    print("non")
```

-----Solution 2 avec for -----

```
palindrome=True
for i in range(n//2):
    if c[i]!=c[n-i-1]:
        palindrome=False
        break
if palindrome:
    print('palindrome')
else:
    print("non")
```

Activité 4 : modules spécifiques pour les opérations de gestion

Corrections



Correction exercice 1 : les listes (suite)

```
# -----Solution 3 : recursive-----  
def palindrome_rec(c):  
    if len(c)==1:  
        return True  
    elif len(c)==2:  
        return c[0]==c[len(c)-1]  
    else:  
        return (c[0]==c[len(c)-1] and palindrome_rec(c[1:len(c)-1]))
```

Activité 4 : modules spécifiques pour les opérations de gestion

Corrections



Correction exercice 2 : les listes

```
L=[i for i in range(11)]
print("L=",L)
#[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
```

```
L1=[i*2 for i in L]
print("L1=",L1)
#[0, 2, 4, 6, 8, 10, 12, 14, 16, 18, 20]
```

```
L2=[[i,i] for i in L]
print("L2=",L2)
#[[0, 0], [1, 1], [2, 2], [3, 3], [4, 4], [5, 5], [6, 6], [7, 7], [8, 8], [9, 9], [10, 10]]
```

```
L3=[i for i in L for j in range(2)]
print("L3=",L3)
#[0, 0, 1, 1, 2, 2, 3, 3, 4, 4, 5, 5, 6, 6, 7, 7, 8, 8, 9, 9, 10, 10]
```

```
L4=[i for j in range(2) for i in L]
print("L4=",L4)
#[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
```

```
L5=[i for i in L for j in range(i)]
print("L5=",L5)
#[1, 2, 2, 3, 3, 3, 4, 4, 4, 4, 5, 5, 5, 5, 5, 6, 6, 6, 6, 6, 6, 7, 7, 7, 7, 7, 7, 7, 8, 8, 8, 8,
8, 8, 8, 8, 9, 9, 9, 9, 9, 9, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10]
```

Activité 4 : modules spécifiques pour les opérations de gestion

Corrections



Correction exercice 3 : les listes

```
#-----Exercice 3 : deux premiers max d'une liste -----  
#-----  
L=[1,20,5,18,10,13,4]  
m1=L[0]  
m2=L[1]  
if m1<m2:  
    m1,m2=m2,m1  
for i in range(2,len(L)):  
    if L[i]>m1:  
        m2=m1  
        m1=L[i]  
    else:  
        if L[i]>m2:  
            m2=L[i]  
print(m1,m2):
```

Activité 4 : modules spécifiques pour les opérations de gestion

Exercices



Exercice 1 : les dictionnaires

- Écrire un script qui lit une chaîne de caractères et construit un dictionnaire contenant la fréquence de tous les caractères de la chaîne saisie.

Exercice 2 : les dictionnaires

- Soit un dictionnaire Classe permettant de mémoriser les noms des étudiants d'une classe, leur âge et leur moyenne. Le nom des étudiants servira de clé du dictionnaire et les valeurs seront constituées des tuples de la forme (âge, moyenne).

1. Donner un script Python qui permet de remplir le dictionnaire Classe avec les informations de n étudiants.
2. Donner le script Python qui permet de consulter les informations relatives à un étudiant donné.

Le résultat de la consultation devra être une ligne de texte de la forme suivante :

Nom : Ben Moussa Ahmed – Âge : 19 – Moyenne : 12,1



NB

- Si le nom n'existe pas, le programme devra afficher : "Étudiant non reconnu".

3. Donner le script Python qui permet de déterminer le nombre d'étudiants dont l'âge ne dépasse pas 20 ans et ayant une moyenne supérieure ou égale à 10.

Exercice 3 : les dictionnaires

Soit le le dictionnaire Livres_BD de la forme suivante :

clef	valeur
Titre_Livre<str>	Tuple(<Auteur>,<nb_exemplaire>)

1. Créer et remplir le dictionnaire Livres_BD.
2. Afficher l'auteur d'un livre donné ainsi que le nombre d'exemplaires.
3. Afficher l'ensemble des auteurs de la base.
4. Afficher l'ensemble des livres empruntables.
5. Étant donné le nom d'un auteur, afficher l'ensemble des titres des livres écrits par cet auteur.

Activité 4 : modules spécifiques pour les opérations de gestion

Corrections



Correction exercice 1 : les dictionnaires

```
ch=input("saisir une chaine de carcateres")
```

```
#----- Solution 1-----
```

```
D1=dict()
```

```
for i in ch:
```

```
nb=ch.count(i)
```

```
D1[i]=nb
```

```
print("Solution 1- Le dictionnaire frequence",D1)
```

```
#----- Solution 2 -----
```

```
D2=dict()
```

```
for i in ch:
```

```
if i not in D2:
```

```
nb=ch.count(i)
```

```
D2[i]=nb
```

```
print("Solution 2- Le dictionnaire frequence",D2)
```

```
#-----Solution 3 par comprehension -----
```

```
D3={i:ch.count(i) for i in ch}
```

```
print("Solution 3- Le dictionnaire frequence",D3)
```

Activité 4 : modules spécifiques pour les opérations de gestion

Corrections



Correction exercice 2 : les dictionnaires

```
#-----Remplissage du dictionnaire Classe avec n étudiants-----
n=int(input("saisir n"))
classe=dict()
for i in range(n):
    nom_prenom=input('donner le nom et le prenom')
    age=int(input("donner l'age"))
    moy=float(input('donner la moyenne'))
    classe[nom_prenom]=(age,moy)
print("le dictionnaire classe")
for i in classe.items():
    print(i)
```

```
# -----Question 2 recherche par clé-----
np=input('donner le nom et le prenom à chercher')
if np in classe:
    #if np in classe.keys():
        print('age',classe[np][0])
        print('moyenne',classe[np][1])
    else:
        print('non existant')
#Question 3
nb=0
for i in classe.values():
    if i[0]<=20 and i[1]>=10:
        nb=nb+1
print('le nb verifiant les criteres',nb)
```



WEBFORCE
BE THE CHANGE



PARTIE 3

Appliquer l'administration système

Dans ce module, vous allez :

- Connaître les commandes de base d'administration
- Administrer les ordinateurs à distance



13 heures



ACTIVITÉ n° 1

Commande d'administration Windows

Compétences visées :

- Tests de commandes d'administration Windows

Recommandations clés :

- Lire attentivement l'énoncé et les questions
- Se référer au cours



02 heures

CONSIGNES

1. Pour le formateur :

- Laisser à l'apprenant l'occasion de comprendre seul l'énoncé.
- S'assurer de la bonne compréhension du contexte avant de lui laisser le temps de réfléchir et répondre.
- Discuter les réponses des apprenants avant de donner la solution.

2. Pour l'apprenant :

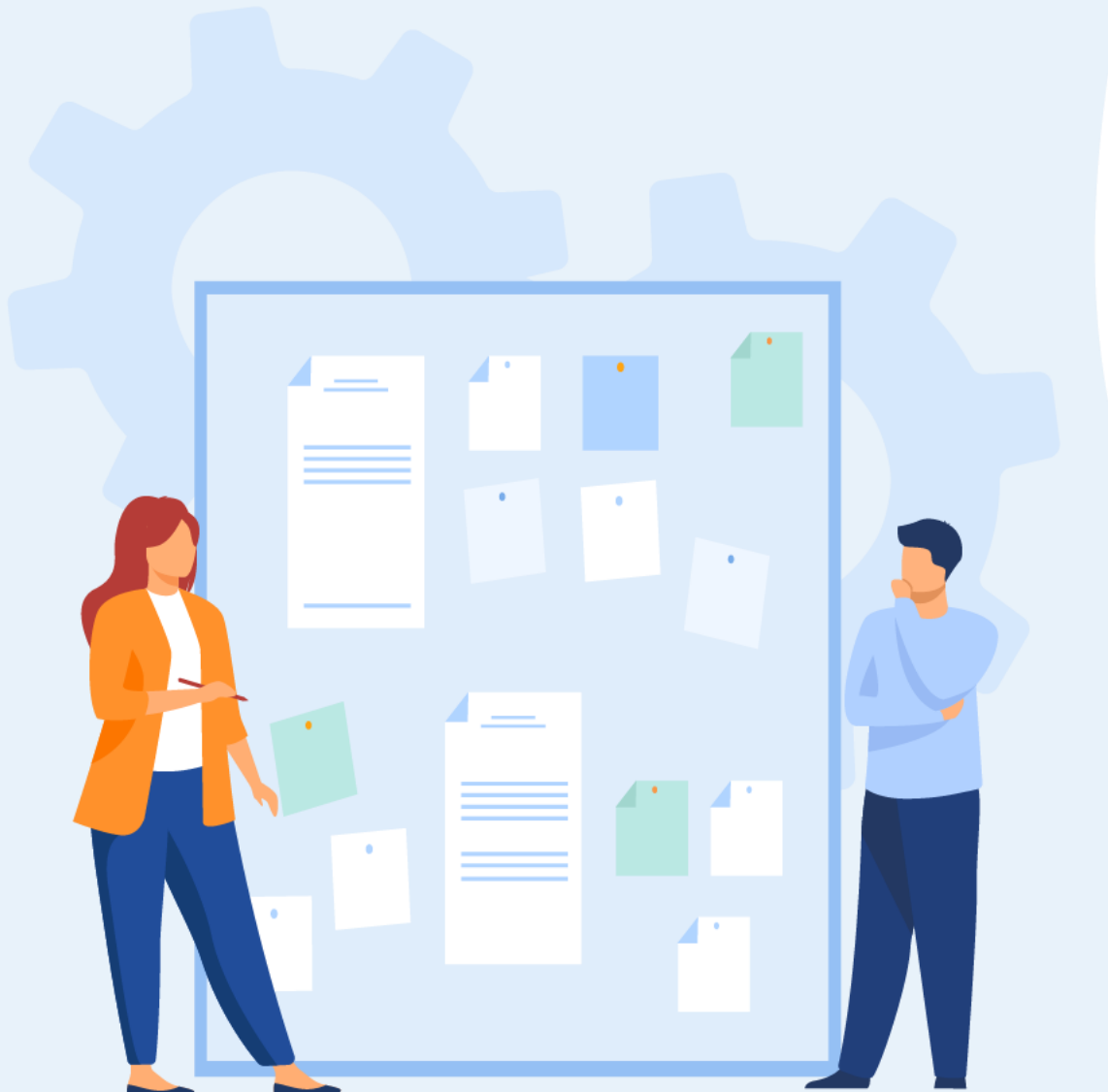
- Lire et bien comprendre la question.
- Essayer de trouver par vous-même une réponse à cette question et la noter.
- Parcourir les réponses proposées.
- Pour chaque réponse : comparez-la à votre réponse et cochez-la si elle lui correspond ou bien est compatible.

3. Conditions de réalisation :

- Individuel ou par groupes (2 ou 3 maximum).
- Support de résumé théorique accompagnant.
- Stylo et feuille de papier.

4. Critères de réussite :

- Le stagiaire est-il capable de :
 - Utiliser les commandes d'administration Windows ?
- Réponses correctes pour au moins 70 % des questions.



Le framework Microsoft PowerShell (PS) développé pour la gestion des systèmes et l'automatisation des tâches administratives va bien au-delà du simple remplacement du vénérable shell de ligne de commande. PS offre bien plus en s'intégrant avec des services et des applications pour fournir une gestion globale des clients et des serveurs.

1. Accéder au module Active Directory

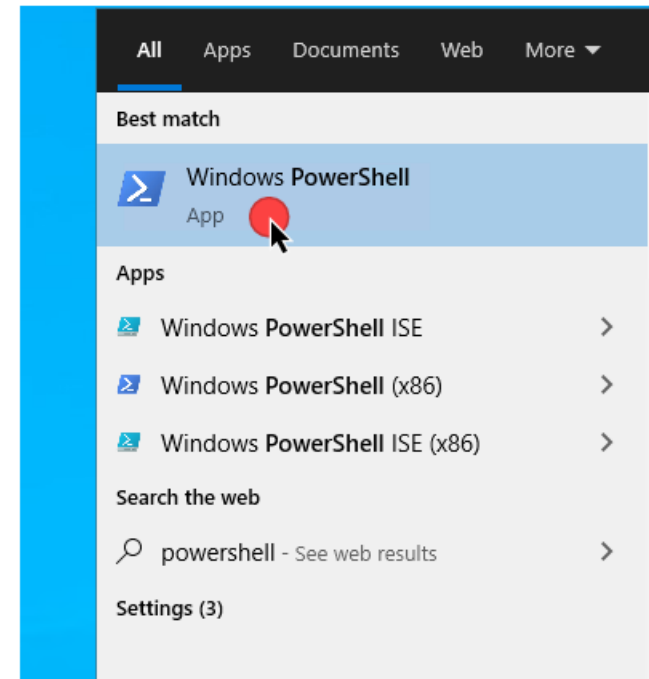
Le module Active Directory pour PowerShell consolide les groupes de cmdlets utilisés pour administrer Active Directory sur site. Ces commandes permettent de gérer les domaines, ainsi que les utilisateurs, les groupes et les objets. Commencez par importer le module Active Directory.



Remarque

- Vous devez effectuer cette opération à chaque exécution de PS avant que les applets de commande spécifiques à AD ne soient disponibles.

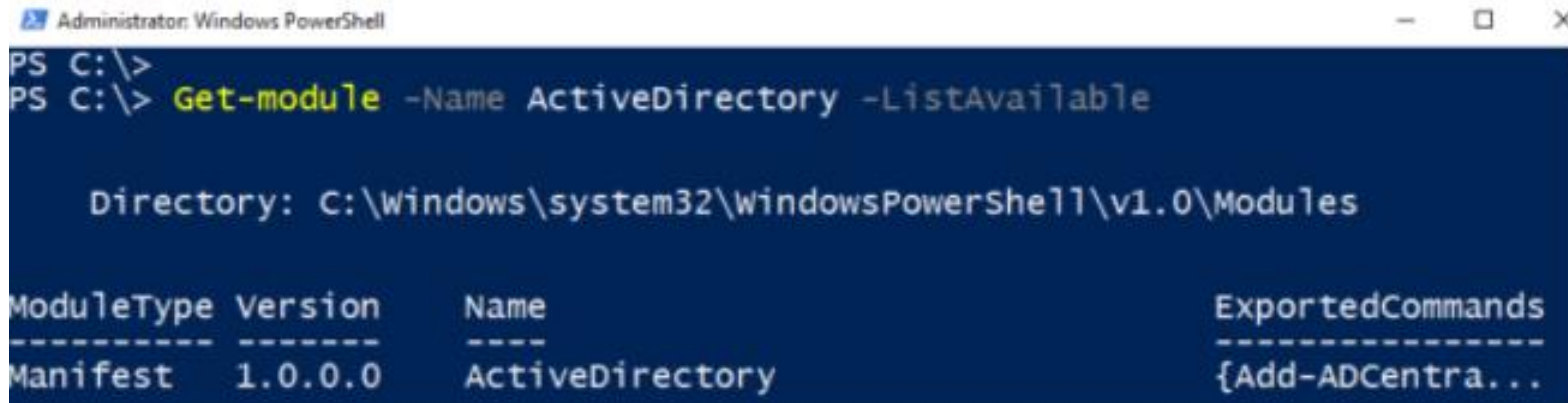
- Les étapes suivantes s'appliquent à Windows 10 et à toutes les versions de Windows Server. Pour importer le module Active Directory pour PowerShell :
 - a) Cliquez sur le bouton Démarrer et recherchez "powershell". Sélectionnez **Windows PowerShell** dans les résultats de recherche.



Activité 1 : commande d'administration Windows

- b) Pour vérifier que le module est disponible, exécutez la commande [Get-Module](#) en spécifiant -Name (ActiveDirectory) et le paramètre -ListAvailable pour rechercher tous les modules installés sur le système.

Get-Module -Name ActiveDirectory -ListAvailable



```
Administrator: Windows PowerShell
PS C:\>
PS C:\> Get-module -Name ActiveDirectory -ListAvailable

Directory: C:\windows\system32\windowspowershell\v1.0\Modules

ModuleType Version      Name                               ExportedCommands
-----
Manifest    1.0.0.0      ActiveDirectory                    {Add-ADCentra...
```

- c) Importez le module à l'aide de la cmdlet Import-Module, en spécifiant encore une fois le nom du module (ActiveDirectory) à l'aide du paramètre -Name.

Import-Module -Name ActiveDirectory

2. Créer un objet ordinateur

- Pour créer un nouvel objet ordinateur dans une unité d'organisation spécifiée, entrez la cmdlet :

```
New-ADComputer -Name "ComputerName" -SamAccountName  
"ComputerName" -Path "OU=Computers,DC=Domain,DC=com"
```

- Spécifier le nom de l'objet pour les paramètres **-Name** et **-SamAccountName**. N'oubliez pas d'inclure le nom distinctif (DN) entre guillemets après le paramètre **-Path** qui désigne l'endroit où l'objet doit être créé.
- Ceci est particulièrement utile lors de la gestion de plusieurs sites et/ou domaines, en particulier lors de la scénarisation des modifications apportées à AD, telles que les créations de nouveaux comptes, qui peuvent être liées à de nouvelles embauches lorsqu'elles sont intégrées par les ressources humaines.

3. Créer de nouveaux groupes de sécurité

Saisissez la cmdlet suivante :

```
New-ADGroup -Name "Security Group Name" -SamAccountName  
"SecurityGroupName" -GroupCategory Security -GroupScope Global -  
DisplayName "Security Group Name" -Path "CN=Groups,DC=Domain,DC=com"  
-Description "Brief description of the what security group is used for"
```

- Cette applet de commande facilite la création de tels groupes, ce qui réduit considérablement les frais administratifs. La gestion des comptes et des ordinateurs est beaucoup plus facile lorsque vous appliquez des paramètres de gestion basés sur les rôles aux groupes de sécurité plutôt que d'apporter des modifications à chaque compte et objet individuellement.

4. Créez un nouveau compte utilisateur

- Saisissez la cmdlet suivante :

```
New-ADUser -Name "User Account Name" -SamAccountName "UserAccountName" -AccountPassword (ConvertTo-SecureString "password" -AsPlainText -Force) -DisplayName "User Name" -Enabled $True -GivenName "FirstName" -Path "CN=Users,,DC=Domain,DC=com" -Server "controller.domain.com" -Surname "LastName" -UserPrincipalName username@domain.com
```

- La génération de nouveaux comptes d'utilisateurs peut devenir une tâche fastidieuse, compte tenu de tous les nombreux paramètres et champs de texte présents. Et bien que tous ne soient pas obligatoires, beaucoup sont certainement utiles lorsqu'ils sont renseignés, comme l'e-mail et le titre pour n'en nommer que quelques-uns.

5. Ajouter/supprimer des utilisateurs ou des objets informatiques vers/depuis des groupes

- Saisissez la cmdlet suivante :

```
Add-ADGroupMember SecurityGroupName -Members Username01 -Server "controller.domain.com" Remove-ADGroupMember SecurityGroupName -Members Username01 -Server "controller.domain.com"
```

Une fois les comptes d'utilisateurs et les objets informatiques créés, les administrateurs peuvent choisir de les ajouter aux groupes de sécurité existants pour une plus grande facilité de gestion. En exécutant cette applet de commande, le paramètre **-Members** répertoriant les comptes à ajouter deviendra membre du groupe de sécurité. Lors de l'ajout de plusieurs comptes, ceux-ci peuvent être séparés par une virgule sur une seule ligne. De plus, le paramètre **-Server**, bien que facultatif, spécifiera le contrôleur de domaine auquel se connecter pour traiter les modifications. Pour supprimer des membres d'un groupe, utilisez l'applet de **commande Remove-ADGroupMember** qui suit une syntaxe similaire.

6. Obtenir le mot de passe stocké localement à partir d'un objet informatique

- Saisissez la cmdlet suivante :

```
Get-AdmPwdPassword -NomOrdinateur "ordinateur.domaine.net"
```

- Pour les organisations qui ont mis en œuvre Local Administrator Password Solution (LAPS) de Microsoft, il existe une cmdlet qui peut récupérer le mot de passe stocké dans l'objet ordinateur dans l'Active Directory et l'afficher en une seule étape facile à récupérer plutôt que de parcourir la base de données pour chaque objet pour rechercher l'attribut ms-mcs-AdmPwd.

7. Joindre un ordinateur à un domaine

- Saisissez la cmdlet suivante :

```
Add-Computer -DomainName "domain.com" -Credential Domain\Username -Restart -Force
```

- Joindre un ordinateur au domaine est l'une des tâches les plus courantes que les professionnels de l'informatique effectuent (des dizaines de milliers de fois au cours de leur carrière). Et bien que la tâche ne prenne que quelques minutes par machine, elle peut être exécutée à partir de PS pour rationaliser et avec quelques ajustements, l'ensemble du processus peut même être entièrement automatisé. N'oubliez pas que l'applet de commande, comme détaillé ci-dessus, demandera à l'administrateur son mot de passe pour terminer la jonction de domaine avant de redémarrer l'appareil.

8. Activer/désactiver des utilisateurs, des ordinateurs ou des comptes de service

- Saisissez la cmdlet suivante :

Enable-ADAccount -Identity "ComputerName"

Disable-ADAccount -Identity "Username"

- Ce que fait cette cmdlet est assez évident. Elle active (ou désactive) un compte d'utilisateur, un objet informatique ou un compte de service géré par AD pour permettre (ou empêcher) l'utilisateur ou le compte d'ordinateur d'être authentifié avec ou sur le réseau.

9. Déverrouiller les comptes d'utilisateurs

- Saisissez la cmdlet suivante :

Unlock-ADAccount -Identity "Username"

- Il est souvent noté par l'analyse statistique que le service informatique passe un temps considérable à résoudre les appels d'assistance liés au déverrouillage des comptes des utilisateurs et à la réinitialisation de leurs mots de passe. Ce dernier est couvert ci-dessus avec la cmdlet Set-ADUser. Cela s'applique essentiellement aux comptes dans lesquels les utilisateurs ont dépassé leurs compteurs de tentatives de mot de passe et ont effectivement verrouillé leur compte pour qu'il ne soit pas utilisé. Une fois déverrouillée, la fonctionnalité AD (y compris l'authentification du compte) est à nouveau disponible.

10. Localisez les comptes d'ordinateurs ou d'utilisateurs désactivés

- Saisissez la cmdlet suivante :

`Search-ADAccount -AccountDisabled | FT Name,ObjectClass`

- La cmdlet Search-ADAccount peut être utilisée pour plus que la simple recherche de comptes désactivés. De nombreux paramètres peuvent être invoqués pour rechercher efficacement AD et rapporter les résultats correspondants. Cependant, pour mettre en évidence sa fonctionnalité, nous avons décidé d'utiliser les comptes désactivés car il s'agit d'une cmdlet importante et utile à connaître.

11. Réparer une relation de confiance rompue entre un client et le domaine

- Saisissez la cmdlet suivante :

`Test-ComputerSecureChannel-Server "controller.domain.com«`

- Un canal rompu, comme PS s'y réfère, se produit lorsque l'ordinateur client a une relation de confiance rompue avec le domaine. Il cessera d'authentifier les connexions au domaine puisque le client et le domaine n'ont plus la configuration du canal, agissant comme un lien entre les deux. L'applet de commande ci-dessus teste cette relation par rapport au contrôleur de domaine pour vérifier que le lien existe toujours. Sinon, l'ajout du paramètre -Repair réinitialise ce lien entre les deux, rétablissant le canal.



Conclusion

- Il ne s'agit en aucun cas d'une liste exhaustive de toutes les cmdlets disponibles pour les systèmes ou les admin de domaine, mais plutôt d'une liste des cmdlets couramment utilisées quotidiennement pour gérer les comptes administrateurs, modifier les objets et obtenir des informations pertinentes utilisées pour résoudre les problèmes qui affectent la productivité.

ACTIVITÉ n° 2

Commande d'administration Linux

Compétences visées :

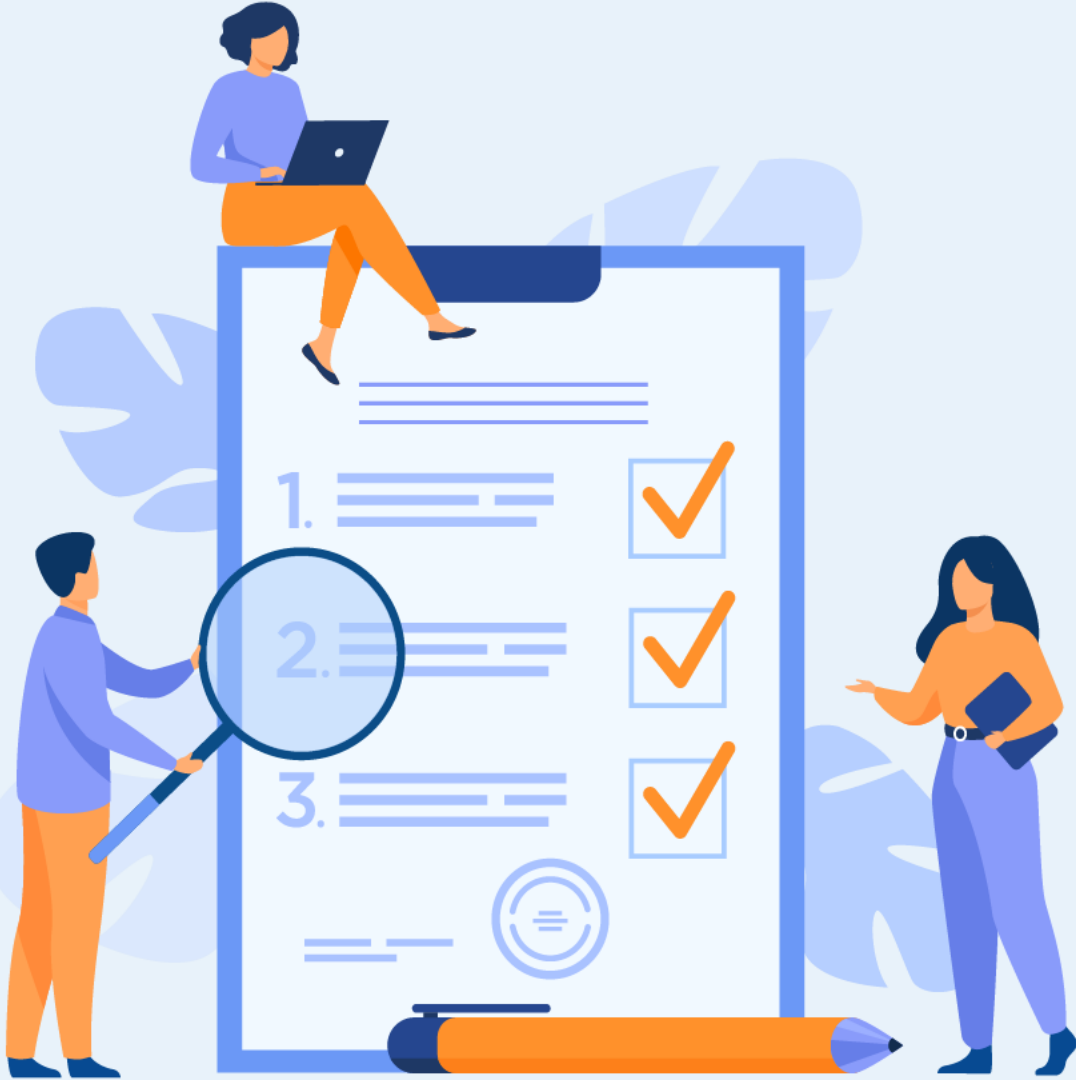
- Tests de commandes d'administration Linux

Recommandations clés :

- Lire attentivement l'énoncé et les questions
- Se référer au cours



02 heures



CONSIGNES

1. Pour le formateur :

- Laisser à l'apprenant l'occasion de comprendre seul l'énoncé.
- S'assurer de la bonne compréhension du contexte avant de lui laisser le temps de réfléchir et répondre.
- Discuter les réponses des apprenants avant de donner la solution.

2. Pour l'apprenant :

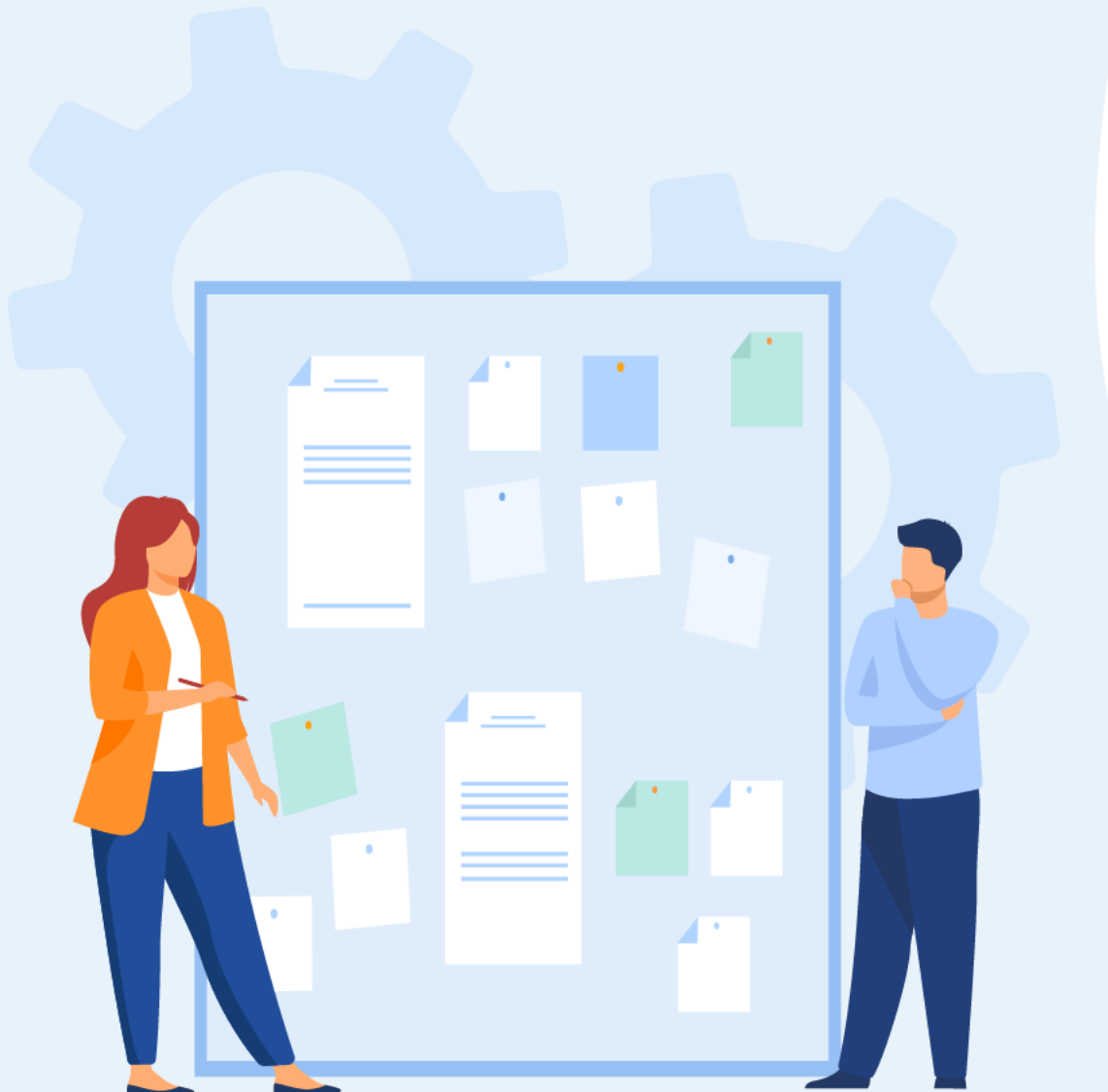
- Lire et bien comprendre la question.
- Essayer de trouver par vous-même une réponse à cette question et la noter.
- Parcourir les réponses proposées.
- Pour chaque réponse : comparez-la à votre réponse et cochez-la si elle lui correspond ou bien est compatible.

3. Conditions de réalisation :

- Individuel ou par groupes (2 ou 3 maximum).
- Support de résumé théorique accompagnant.
- Stylo et feuille de papier.

4. Critères de réussite :

- Le stagiaire est-il capable de :
 - Utiliser les commandes d'administration Linux ?
- Réponses correctes pour au moins 70 % des questions.



Activité 2 : commande d'administration Linux

Prise en main



- Le shell est un programme qui sert d'interface entre l'utilisateur et le noyau, dans un terminal en mode texte.
- Un **terminal** est un programme qui gère l'affichage de texte et la saisie, et lance un shell.

Par exemple : **xterm**, **gnome-terminal**. Raccourcis clavier : Ctrl+Alt+T

- Une console = terminal physique (écran + clavier)
- Consoles virtuelles : Ctrl+Alt+F1 .. F6
- Retour au mode graphique : Ctrl+Alt+F7 .. F12 (displays X11)
- Un shell peut aussi lire des lignes depuis un fichier (un script shell).

Fonctionnement habituel d'un shell = le "mode interactif" :

- Affiche une invite, appelée "prompt" (par exemple. \$).
- Attend que l'utilisateur saisisse une ligne + [Entrée].
- Analyse la syntaxe (symboles, mots-clés...).
- Exécute les commandes correspondantes.
- Attend leur terminaison, puis réaffiche le prompt.

Le terminal Gnome sur Ubuntu

- Pour ouvrir un terminal, pressez simultanément les 3 touches Ctrl+Alt+T.
- Dans la plupart des logiciels avec une interface graphique, les copier-coller se font avec Ctrl+C (copier), Ctrl+V (coller), Ctrl+X (couper).
- Dans un terminal, il faut presser en plus Shift ; il faut donc faire Ctrl+Shift+C (copier), Ctrl+Shift+V (coller), Ctrl+Shift+X (couper).
- Vous pouvez rajouter un nouvel onglet dans un terminal avec Ctrl+Shift+T, et une nouvelle fenêtre avec Ctrl+Shift+N.
- Dans le terminal, vous pouvez rappeler les commandes précédentes avec les flèches Up et Down, compléter une commande en appuyant sur la touche Tab, faire un recherche dans les commandes précédentes avec Ctrl+R, vider le terminal avec Ctrl+L.

Activité 2 : commande d'administration Linux

Prise en main



Éditeur vi

- Les activités d'administration système se font très souvent dans un terminal distant, sans mode graphique. Il existe un éditeur de texte qui fonctionne dans un terminal et qui est disponible sur tous les systèmes Unix : vi (prononcer "vie ail").
- Il y a 2 modes :
 - **Le mode « commande »** : vous y êtes par défaut. Les touches provoquent des actions.
par exemple :
 - x supprime un caractère
 - i passe en mode insertion
 - :q Enter quitte vi
 - **Le mode « édition »** : ce que vous tapez est inséré dans le texte ; pressez Esc (la touche Échap) pour revenir au mode commande.

La version de vi que nous allons utiliser est en réalité vim (vi iMproved) et nous allons la découvrir à l'aide de son tutoriel vimtutor.

- Installer vim et vimtutor sur Ubuntu en tapant : **\$ sudo apt install vim vimtutor**
- En cas d'erreur d'installation (paquet vimtutor non trouvé), essayez :
 - \$ sudo apt update**
 - \$ sudo apt install vim**
- Ouvrez un terminal et tapez : **\$ vimtutor**

Activité 2 : commande d'administration Linux

Systèmes de fichiers



- Sous Linux, les mémoires de masse apparaissent dans le répertoire /dev où sont rangés tous les périphériques (devices) :
- Floppy disk : /dev/fd0, /dev/fd1 1^{er} lecteur détecté, 2^e, ...
- Disques durs IDE :
 - /dev/hda, /dev/hdb, ... 1^{er} disque détecté, 2^e, ...
 - /dev/hda1 1^{re} partition de hda
- Disques durs SATA, SCSI, clés USB :
 - /dev/sda, /dev/sdb, ... 1^{er} disque détecté, 2^e, ...
 - /dev/sda1 1^{re} partition de sda
- Disques optiques : /dev/sr0, /dev/sr1 1^{er} disque détecté, 2^e, ...
- Disques NVMe (un type de SSD sur bus PCI Express) :
 - /dev/nvme0n1 1^{er} block device
 - /dev/nvme0n1p1 1^{re} partition
- Cartes mémoires MMC et SD :
 - /dev/mmcblk0 1^{er} device
 - /dev/mmcblk0p1 1^{re} partition

Outils pour manipuler les partitions

- Listing étendu : ls -l /dev

affiche tous les périphériques de masse, mélangés à plein d'autres choses.

- lsblk : list block devices

affiche les disques, partitions, points de montage.

```
$ lsblk
```

```
NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
```

```
sda 8:0 0 20G 0 disk
```

```
└─sda1 8:1 0 512M 0 part /boot/efi
```

```
└─sda2 8:2 0 1K 0 part
```

```
└─sda5 8:5 0 19,5G 0 part /
```

```
sr0 11:0 1 58,2M 0 rom /media/thiel/VBox_GAs_6.1.16
```

Activité 2 : commande d'administration Linux

Systèmes de fichiers



1. Afficher les disques, partitions, secteurs :

```
$ sudo fdisk -l
```

➡ fdisk : gère les disques en MBR et en GPT ; ne modifie pas le disque tant qu'on n'a pas demandé d'écrire la nouvelle table de partitions.

2. Manipuler le disque sda en mode interactif :

```
$ sudo fdisk /dev/sda
```

3. Créer un système de fichiers sur une partition :

- Le système de fichiers est la structure qui permet de stocker des fichiers. Le formatage est l'opération qui crée le système de fichiers sur une partition de disque. Linux supporte beaucoup de système de fichiers comme ext4, JFS, ReiserFS, NTFS et FAT. Ext4 est le système de fichiers par défaut de Linux. NTFS et FAT sont des systèmes de fichiers de Microsoft et donc à destination de Windows.

```
$ sudo mkfs.ntfs -h
```

-h: Affiche l'aide de la commande mkfs

Activité 2 : commande d'administration Linux

Gestion des utilisateurs et des groupes



1. Ajouter un utilisateur au système

```
sudo adduser nom_utilisateur
```

2. Ajouter un groupe au système

```
sudo groupadd nom_groupe
```

3. Supprimer un utilisateur

```
sudo deluser nom_utilisateur
```

4. Supprimer un groupe

```
sudo groupdel nom_groupe
```

5. Ajouter un utilisateur à un groupe

```
sudo adduser nom_utilisateur nom_groupe
```

ACTIVITÉ n° 3

Manipulation de commandes d'administration réseaux

Compétences visées :

- Tester les commandes d'administration réseaux

Recommandations clés :

- Lire attentivement l'énoncé et les questions
- Se référer au cours



02 heures



CONSIGNES

1. Pour le formateur :

- Laisser à l'apprenant l'occasion de comprendre seul l'énoncé.
- S'assurer de la bonne compréhension du contexte avant de lui laisser le temps de réfléchir et répondre.
- Discuter les réponses des apprenants avant de donner la solution.

2. Pour l'apprenant :

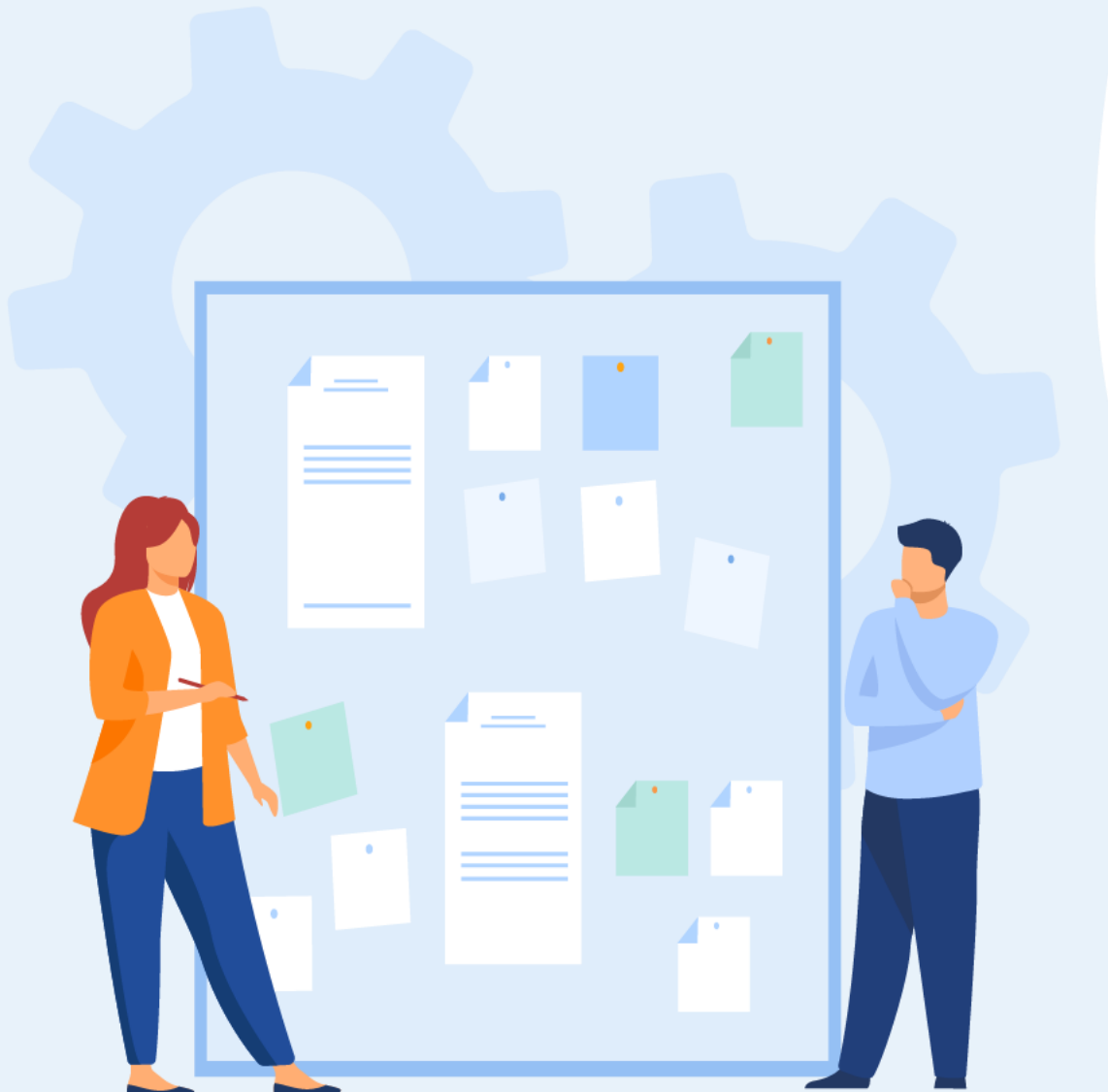
- Lire et bien comprendre la question.
- Essayer de trouver par vous-même une réponse à cette question et la noter.
- Parcourir les réponses proposées.
- Pour chaque réponse : comparez-la à votre réponse et cochez-la si elle lui correspond ou bien est compatible.

3. Conditions de réalisation :

- Individuel ou par groupes (2 ou 3 maximum).
- Support de résumé théorique accompagnant.
- Stylo et feuille de papier.

4. Critères de réussite :

- Le stagiaire est-il capable de :
 - Utiliser les commandes d'administration réseaux ?
- Réponses correctes pour au moins 70 % des questions.



Activité 3 : manipulation de commandes d'administration réseaux



1. Afficher le nom d'hôte d'un système Linux

```
$ hostname
```

2. Vérifier la connectivité entre deux systèmes (par exemple la connectivité entre mon réseau local avec l'adresse IP de 192.168.2.103)

```
$ ping 192.168.2.103
```

- La commande ping continue d'envoyer le ICMP ping du paquet jusqu'à ce que vous l'interrompiez en appuyant sur Ctrl + C sur le clavier. Cependant, vous pouvez limiter les paquets envoyés en utilisant le -c option.
- Nous envoyons 5 paquets de demandes d'écho, et une fois terminé, la commande ping s'arrête.

```
$ ping 192.168.2.103 -c 5
```

- vous pouvez également envoyer une requête ping au nom de domaine d'un hôte ou d'un serveur. Par exemple, vous pouvez envoyer un ping à Google comme indiqué.

```
$ ping google.com
```

3. Afficher l'itinéraire emprunté par un paquet ping ICMP de votre appareil vers l'hôte ou le serveur de destination (par exemple Google).

```
$ traceroute google.com
```

4. Afficher la configuration IP de la machine

```
$ ifconfig
```

5. Fournir les informations sur le DNS

```
$ nslookup ubuntu.com
```

6. Fournir les informations sur les connexions réseau d'une machine

Pour afficher les interfaces réseau connectées à votre PC, exécutez : `$ netstat -i`

ACTIVITÉ n° 4

Administration à distance Windows

Compétences visées :

- Réaliser des tests de fonctionnement de commandes d'administration à distance Windows

Recommandations clés :

- Lire attentivement l'énoncé et les questions
- Se référer au cours



02 heures



CONSIGNES

1. Pour le formateur :

- Laisser à l'apprenant l'occasion de comprendre seul l'énoncé.
- S'assurer de la bonne compréhension du contexte avant de lui laisser le temps de réfléchir et répondre.
- Discuter les réponses des apprenants avant de donner la solution.

2. Pour l'apprenant :

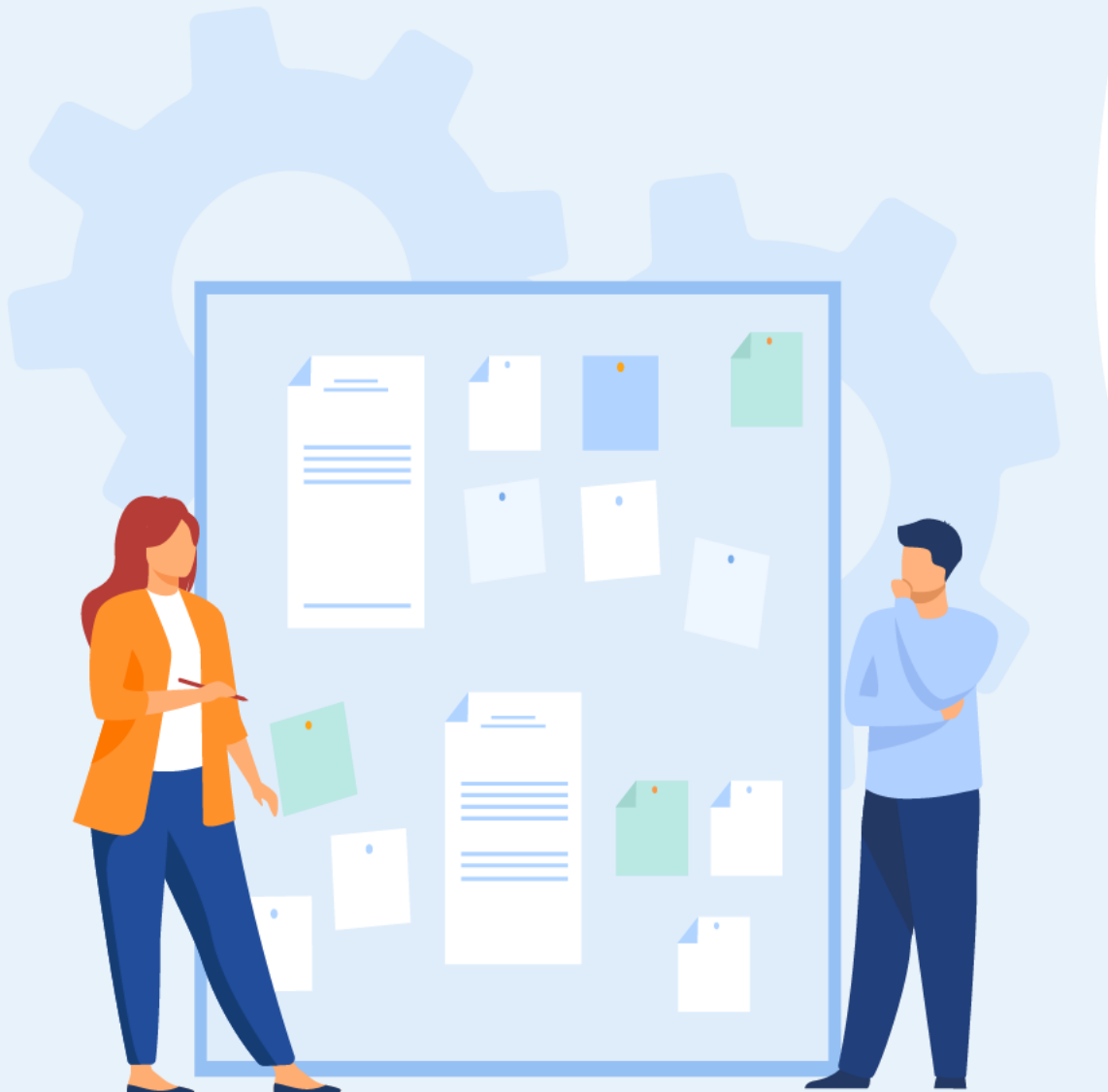
- Lire et bien comprendre la question.
- Essayer de trouver par vous-même une réponse à cette question et la noter.
- Parcourir les réponses proposées.
- Pour chaque réponse : comparez-la à votre réponse et cochez-la si elle lui correspond ou bien est compatible.

3. Conditions de réalisation :

- Individuel ou par groupes (2 ou 3 maximum).
- Support de résumé théorique accompagnant.
- Stylo et feuille de papier.

4. Critères de réussite :

- Le stagiaire est-il capable de :
 - Utiliser les commandes d'administration à distance Windows ?
- Réponses correctes pour au moins 70 % des questions.



- La communication à distance Windows PowerShell :
 - Utilise le protocole WS-Management.
 - Permet d'exécuter n'importe quelle commande Windows PowerShell sur un ou plusieurs ordinateurs distants.
 - Établit des connexions persistantes.
 - Démarre des sessions interactives.
 - Exécute des scripts sur ordinateurs distants.

1. Démarrer une session interactive avec un seul ordinateur distant (ordinateur distant Server01)

Enter-PSSession Server01

Pour démarrer une session interactive avec un seul ordinateur distant, utilisez l'applet de commande [Enter-PSSession](#).

2. Terminer la session interactive

Exit-PSSession

3. Exécuter une commande à distance : par exemple, exécuter une commande [Get-UICulture](#) sur les ordinateurs distants Server01 et Server02

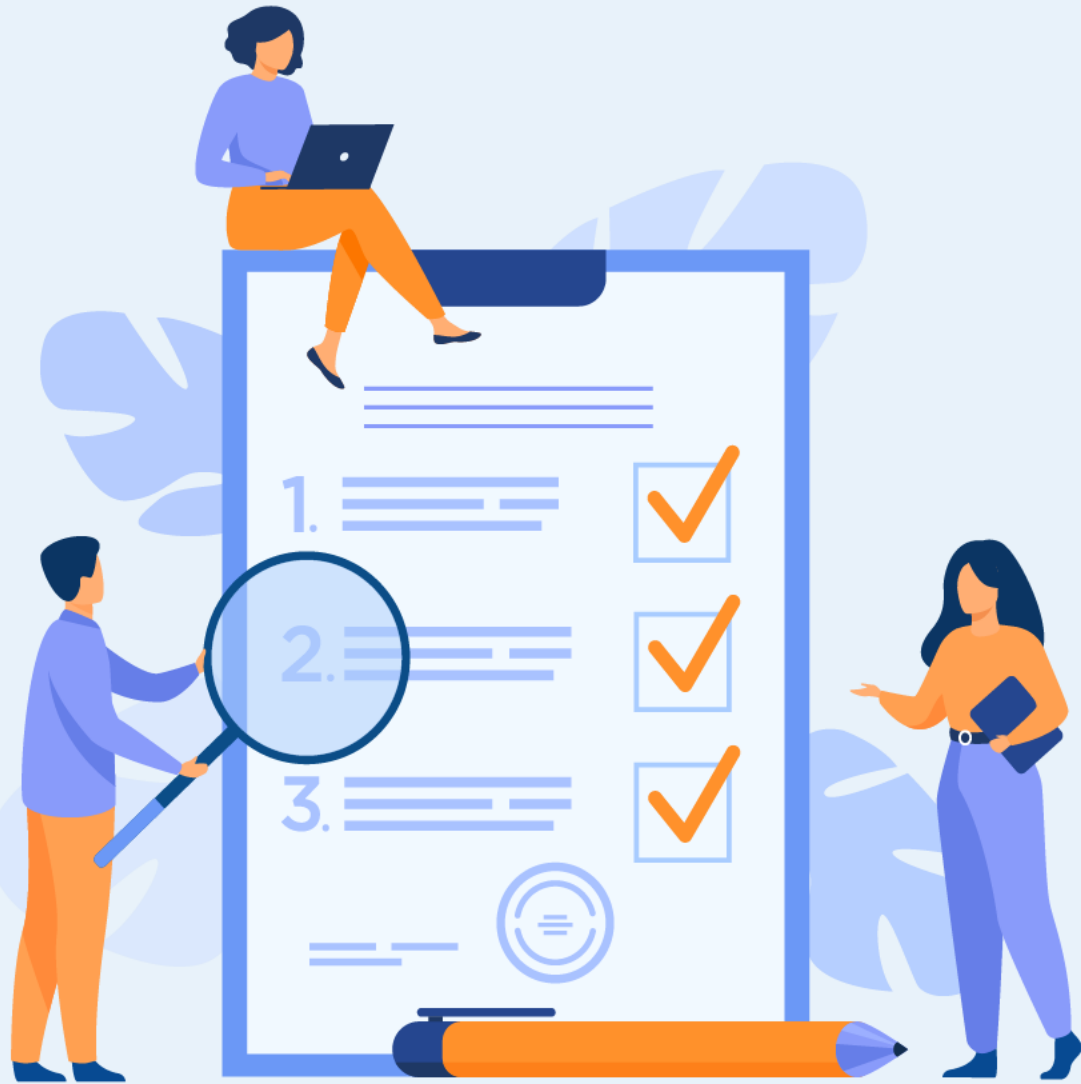
Invoke-Command -ComputerName Server01, Server02 -ScriptBlock {Get-UICulture}

4. Exécuter le script DiskCollect.ps1 sur les ordinateurs distants Server01 et Server02.

Invoke-Command -ComputerName Server01, Server02 -FilePath c:\Scripts\DiskCollect.ps1

5. Créer une session persistante sur Server01 et Server02

\$s = New-PSSession -ComputerName Server01, Server02



ACTIVITÉ n° 5

Administration à distance Linux

Compétences visées :

- Réaliser des tests de fonctionnement de commandes d'administration à distance Linux
- Connaître les connexions ssh et bureau à distance

Recommandations clés :

- Lire attentivement l'énoncé et les questions
- Se référer au cours



05 heures



WEBFORCE
BE THE CHANGE

CONSIGNES

1. Pour le formateur :

- Laisser à l'apprenant l'occasion de comprendre seul l'énoncé.
- S'assurer de la bonne compréhension du contexte avant de lui laisser le temps de réfléchir et répondre.
- Discuter les réponses des apprenants avant de donner la solution.

2. Pour l'apprenant :

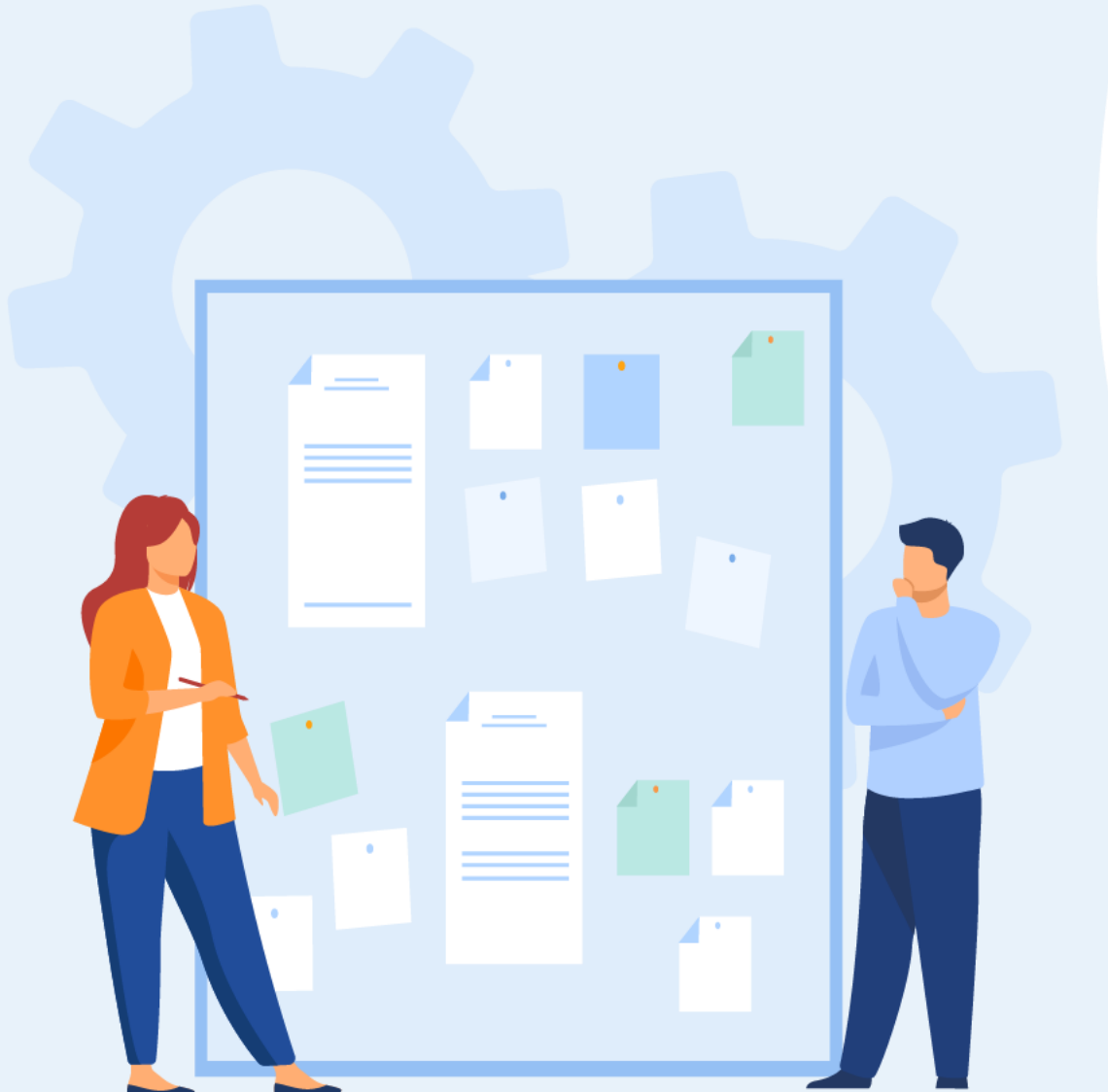
- Lire et bien comprendre la question.
- Essayer de trouver par vous-même une réponse à cette question et la noter.
- Parcourir les réponses proposées.
- Pour chaque réponse : comparez-la à votre réponse et cochez-la si elle lui correspond ou bien est compatible.

3. Conditions de réalisation :

- Individuel ou par groupes (2 ou 3 maximum).
- Support de résumé théorique accompagnant.
- Stylo et feuille de papier.

4. Critères de réussite :

- Le stagiaire est-il capable de :
 - Utiliser les commandes d'administration à distance Linux ?
- Réponses correctes pour au moins 70 % des questions.



Activité 5 : administration à distance Linux

Exercices



Il y a plusieurs façons d'administrer à distance un serveur Linux. Le plus courant est d'utiliser SSH.

Installation de OpenSSH

- Installer OpenSSH :

```
sudo apt-get install openssh-server
```

À la fin de l'installation, vérifier qu'il n'y a pas eu de messages d'erreur dans les informations qui se sont affichées à l'écran.

Configuration de OpenSSH

- Ouvrir le fichier de configuration de "openssh" dans l'éditeur de texte Vi :

```
sudo vi /etc/ssh/sshd_config
```

Pour quitter l'éditeur de texte Vi, appuyer sur la touche "Echap", puis ":", puis "q" et "!".

- Ouvrir la page du manuel de configuration du serveur ssh :

```
man sshd_config
```

Pour quitter la page de manuel, appuyer sur la touche "q".

Connexion au serveur SSH

Pour se connecter depuis un poste Windows au serveur SSH, il faut utiliser un "client SSH".

Activité 5 : administration à distance Linux

Exercices



Exemples de clients SSH :

Putty (client telnet / SSH) : <http://www.putty.org/>

WinSCP (transfert de fichiers - interface graphique) : <http://winscp.net/eng/docs/lang:fr>

JuiceSSH sur Android : <https://play.google.com/store/apps/>

Outil d'administration Webmin :

Webmin est un outil d'administration basé sur une interface Web pour les systèmes Unix/Linux. Il est très puissant et également très simple à utiliser. À travers une connexion sécurisée, il peut être utilisé pour administrer de façon sûre la plupart des services réseau (Apache, Postfix, Sendmail, FTP, MySQL, PostgreSQL, Samba, SSH, Bind, etc.) depuis n'importe quel poste sur le réseau.

Pour effectuer ce TP nous utilisons trois machines virtuelles :

- Une machine serveur sous Ubuntu 14.04.5-desktop-i386 (Adresse IP : 134.59.136.1)
- Deux machines clientes, Client 1 sous Ubuntu 14.04 (Adresse IP : 134.59.136.2) et Client 2 sous Windows 7 (Adresse IP : 134.59.136.3)

Exercices :

1. Tester les connexions entre les machines virtuelles (Ping).
2. Installer les paquets nécessaires à l'interface d'administration WebMin sur la machine serveur (Ubuntu 14.04.5). Puis tester la connexion à WebMin en utilisant l'URL <https://ubuntu:10000/> ou <https://localhost:10000/>.

Activité 5 : administration à distance Linux

Exercices



3. Installer les paquets nécessaires de serveur Bind sur la machine serveur (Ubuntu 14.04.5), puis procéder à la configuration de ce dernier en utilisant l'outil WebMin selon les paramètres suivants (commande d'installation : `apt-get install bind9`) :

1. Zone principale

Nom de domaine	Zarzara.edu
Fichier d'enregistrement	Automatique
Serveur primaire	Umc.zarzara.edu
Adresse électronique	admin@zarzara.edu
Adresse	134.59.136.1

2. Enregistrement adresses (serveur, client 1 et client 2)

Nom	umc. zarzara.edu (serveur) client-1. zarzara.edu (Client 1) client-2. zarzara.edu (Client 2)
Adresse	134.59.136.1 (serveur) 134.59.136.2 (Client 1) 134.59.136.3 (Client 2)
Mettre à jour l'adresse inverse	Oui

3. De la même manière, créer la zone inverse

Activité 5 : administration à distance Linux

Exercices



4. Installer les paquets nécessaires au serveur OpenSSH sur la machine serveur (Ubuntu 14.04.5), puis procéder à la configuration de ce dernier en utilisant l'outil WebMin (commande d'installation : `apt-get install ssh`).

Contrôle d'accès (autorisé)	Serveur et client 1
Contrôle d'accès (refusé)	Client 2
Port (par défaut)	22
Connexion	Par mot de passe

Activité 5 : administration à distance Linux

Exercices



5. Installer les paquets nécessaires au serveur web (Apache) sur la machine serveur (Ubuntu 14.04.5), puis procéder à la configuration de ce dernier en utilisant l'outil WebMin (commande d'installation : `apt-get install apache2`).
- Créer et configurer un "serveur virtuel" ou hôte virtuel avec les paramètres suivants :

Document racine	<code>/var/www/</code>
Port	Par défaut
ptions du serveur virtuel	
E-mail du responsable	Votre @ électronique
Nom d'hôte du serveur	Automatique
Page par défaut	<code>index.htm</code>

Activité 5 : administration à distance Linux

Exercices



6. Installer les paquets nécessaires au serveur MySQL server sur la machine serveur (Ubuntu 14.04.5), puis procéder à la configuration de ce dernier en utilisant l'outil WebMin.
- Créer une nouvelle base de données avec les options suivantes :

Nom de la base	fstech
Set caracteres	Utf8 (UTF-8-Unicode)
Table 1	Client 1

- Les champs

Field name	Data type	Type width	key?	Auto-increment?	Allow nulls?	Unsigned?	Default value
num	int		<input checked="" type="checkbox"/> Yes	<input checked="" type="checkbox"/> Yes	<input type="checkbox"/> Yes	<input type="checkbox"/> Yes	
Designation	char	50	<input type="checkbox"/> Yes	<input type="checkbox"/> Yes	<input checked="" type="checkbox"/> Yes	<input type="checkbox"/> Yes	
adresse	char	15	<input type="checkbox"/> Yes	<input type="checkbox"/> Yes	<input checked="" type="checkbox"/> Yes	<input type="checkbox"/> Yes	
			<input type="checkbox"/> Yes	<input type="checkbox"/> Yes	<input checked="" type="checkbox"/> Yes	<input type="checkbox"/> Yes	

Activité 5 : administration à distance Linux

Exercices



7. Installer les paquets nécessaires au serveur Samba server sur la machine serveur (Ubuntu 14.04.5), puis procéder à la configuration de ce dernier en utilisant l'outil WebMin.

Partager le dossier /home avec les machines clientes avec le nom de partage : mon-dossier et avec les permissions : 777.

8. Installer les paquets nécessaires au serveur Telnet sur la machine serveur (Ubuntu 14.04.5), puis procéder à la configuration de ce dernier en utilisant l'outil WebMin.

- Quel est l'intérêt de SSH par rapport à Telnet ?
- Tester la console avec : Telnet 127.0.0.1.
- Établir une connexion Telnet à la machine Client 1.

Activité 5 : administration à distance Linux

Corrections



Correction exercice 1 :

Étape 1 : configuration du réseau

- Configuration de l'adresse IP statique et de la passerelle sous Ubuntu.

```
root@ubuntu:~# ifconfig eth0 134.59.136.7
root@ubuntu:~# route add default gw 134.59.136.2
root@ubuntu:~# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0c:29:3b:ba:60
          inet addr:134.59.136.7  Bcast:134.59.255.255  Mask:255.255.0.0
          inet6 addr: fe80::20c:29ff:fe3b:ba60/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:4045 errors:0 dropped:0 overruns:0 frame:0
          TX packets:3813 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:4724293 (4.7 MB)  TX bytes:293026 (293.0 KB)
          Interrupt:19 Base address:0x2000
```

Activité 5 : administration à distance Linux

Corrections



Étape 2 : installation de l'outil WebMin

1. Installer les paquets nécessaires à WebMin en tapant les commandes suivantes dans le terminal :

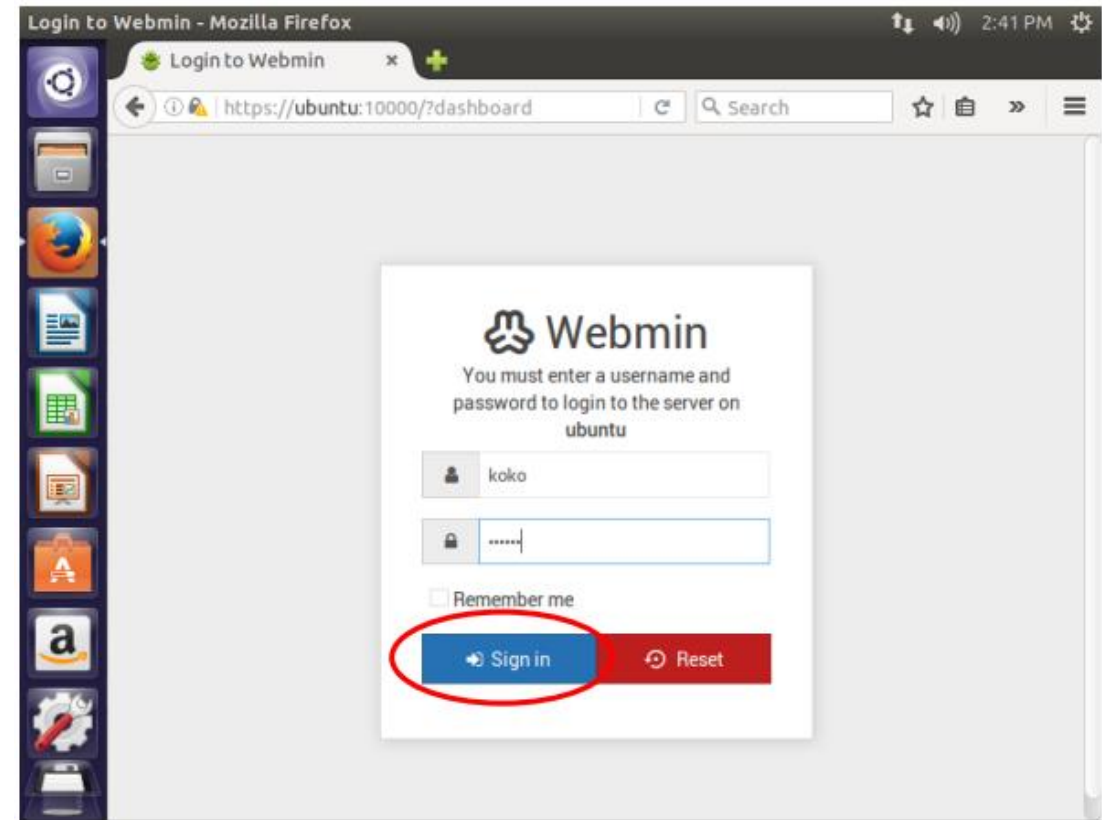
```
sudo sh -c 'echo "deb http://download.webmin.com/download/repository sarge contrib" > /etc/apt/sources.list.d/webmin.list'
```

```
wget -qO - http://www.webmin.com/jcameron-key.asc |sudo apt-key add -
```

```
sudo apt-get update
```

```
sudo apt-get install webmin
```

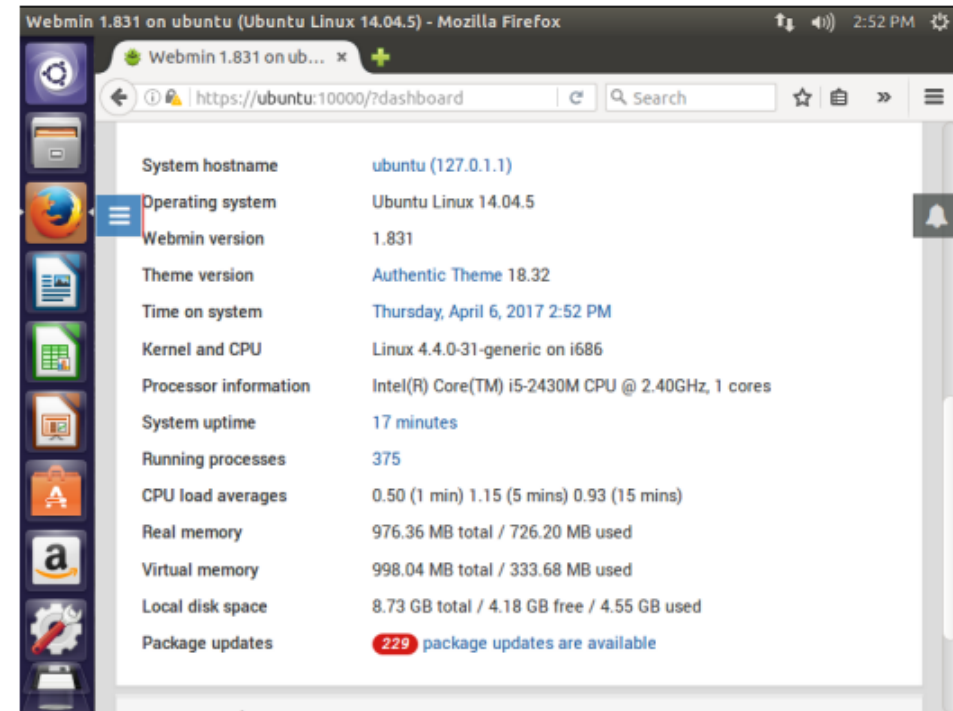
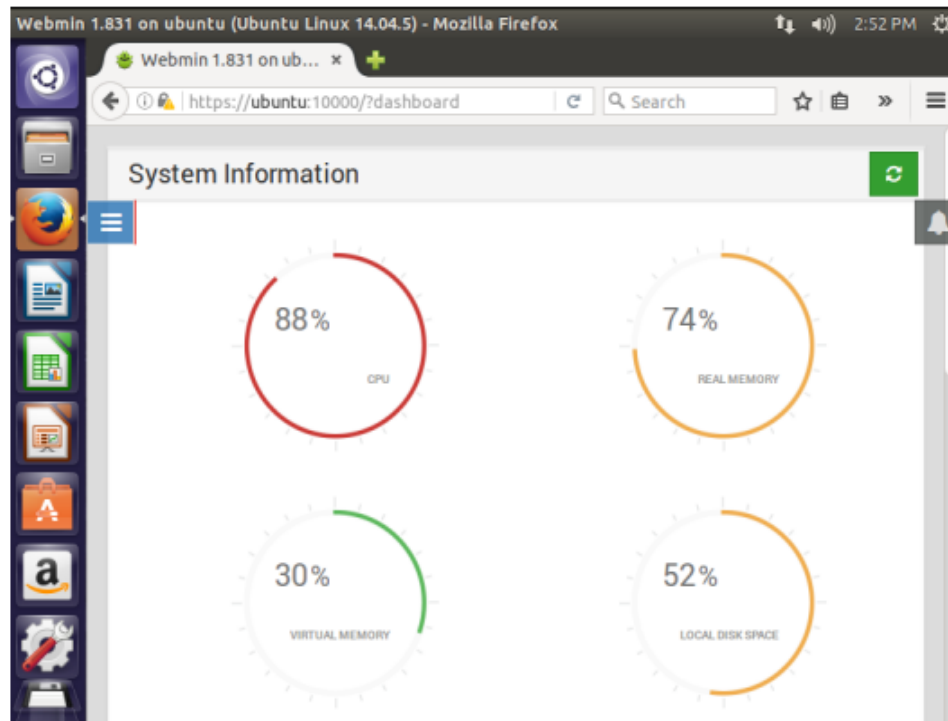
2. Pour se connecter à Webmin : taper l'URL suivante: `https://ubuntu:10000/` et se loguer avec le compte root créé "Ex : koko" et le mot de passe de la session Ex : « 123456 ».



Activité 5 : administration à distance Linux

Corrections

3. La figure suivante est la page permettant l'accès à Webmin après authentification (page d'accueil de WebMin).



Activité 5 : administration à distance Linux

Corrections



Étape 3 : installation des services réseau « Bind », « Apache », « SQL server », « Samba » et « Open ssh ».

1. Installation de services de noms DNS (BIND9 ou Berkley Internet Naming Daemon) : il faut passer en mode root (sudo -i), puis taper le mot de passe du compte root, puis lancer l'installation avec la commande (apt-get install bind9).

```
root@ubuntu: ~  
koko@ubuntu:~$ sudo -i 1  
[sudo] password for koko:  
root@ubuntu:~# apt-get install bind9 2  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
The following extra packages will be installed:  
  bind9-host bind9utils dnsutils libbind9-90 libdns100 libisc95 libisccc90  
  libisccfg90 liblwres90  
Suggested packages:  
  bind9-doc rblcheck  
The following NEW packages will be installed:  
  bind9 bind9utils  
The following packages will be upgraded:  
  bind9-host dnsutils libbind9-90 libdns100 libisc95 libisccc90 libisccfg90  
  liblwres90  
8 upgraded, 2 newly installed, 0 to remove and 230 not upgraded.  
Need to get 1,436 kB of archives.  
After this operation, 1,588 kB of additional disk space will be used.  
Do you want to continue? [Y/n] y  
Get:1 http://us.archive.ubuntu.com/ubuntu/ trusty-updates/main bind9-host i386 1  
:9.9.5.dfsg-3ubuntu0.13 [45.1 kB]  
3% [1 bind9-host 42.2 kB/45.1 kB 94%]
```


Activité 5 : administration à distance Linux

Corrections



À la fin de l'installation, le service bind9 démarre automatiquement.

```
#
* Starting domain name service... bind9 [ OK ]
Processing triggers for libc-bin (2.19-0ubuntu6.9) ...
Processing triggers for ureadahead (0.100.0-16) ...
Processing triggers for ufw (0.34~rc-0ubuntu2) ...
root@ubuntu:~#
```

2. Installation de service web Apache : Apache est un serveur http libre, c'est un des serveurs http les plus utilisés sur internet avec plus de 60 % des sites d'internet. Un serveur http est un serveur hébergeant un ou plusieurs sites Web qui sont accessibles par des navigateurs internet. Le protocole permettant l'échange de pages html est le protocole http, d'où le nom de serveur http. Ce protocole utilise généralement le port 80. Pour lancer l'installation d'un serveur Apache, on écrit la commande suivante :

```
koko@ubuntu:~$ sudo -i
[sudo] password for koko:
root@ubuntu:~# apt-get install apache2
Reading package lists... Done
Building dependency tree
Reading state information... Done
```


Activité 5 : administration à distance Linux

Corrections



3. Installation de serveur SSH : Secure Shell (SSH) est à la fois un programme informatique et un protocole de communication sécurisée. Il permet de se connecter à une machine distante avec une liaison sécurisée. Les données sont cryptées entre machines. Il permet d'exécuter des commandes sur un serveur distant. Pour lancer l'installation d'un serveur SSH, on écrit la commande suivante :

```
root@ubuntu:~# apt-get install ssh
Reading package lists... Done
Building dependency tree
Reading state information... Done
```

```
root@ubuntu:~# status ssh
ssh start/running, process 1121
```

4. Installation de serveur Samba : Samba est une application qui permet d'utiliser, sous Linux, le protocole SMB (Session Message Block) également appelé le protocole NetBIOS ou LanManager. Le protocole SMB est utilisé par Microsoft Windows pour partager des disques et des imprimantes. En utilisant les outils Samba d'Andrew Tridgel, les systèmes Unix (Linux inclus) peuvent également partager des disques et des imprimantes avec des hôtes Windows. Pour lancer l'installation d'un serveur Samba, on écrit la commande suivante :

```
root@ubuntu:~# apt-get install -y -f samba
Reading package lists... Done
Building dependency tree
Reading state information... Done
```

```
root@ubuntu:~# service samba status
nmbd start/running, process 1763
smbd start/running, process 720
```

Activité 5 : administration à distance Linux

Corrections



5. Installation de serveur SQL : le SQL server est un SGBD (Système de Gestion de Base de Données) qui possède toutes les caractéristiques pour pouvoir accompagner l'utilisateur dans la manipulation, le contrôle, le tri, la mise à jour, et bien d'autres actions encore, de bases de données grâce au langage SQL (Structured Query Language). Pour lancer l'installation d'un serveur SQL, on écrit la commande suivante :

```
root@ubuntu:~# apt-get install mysql-server
Reading package lists... Done
Building dependency tree
Reading state information... Done
```

```
root@ubuntu:~# service mysql status
mysql start/running, process 1279
root@ubuntu:~#
```

6. Installation de serveur Telnet : (Terminal Network). Le protocole Telnet est un protocole standard d'internet permettant de relier un client (système composé d'un affichage et d'un clavier) à un interpréteur de commande (côté serveur). Il permet d'ouvrir une connexion avec un serveur à distance comme si on le consultait sur place. Telnet est un protocole non sécurisé, les informations transitant n'étant pas cryptées contrairement à SSH. Pour lancer l'installation d'un serveur Telnet, on écrit la commande suivante :

```
koko@ubuntu:~$ sudo -i
[sudo] password for koko:
root@ubuntu:~# apt-get install xinetd telnetd
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  telnetd xinetd
0 upgraded, 2 newly installed, 0 to remove and 223 not upgraded.
Need to get 140 kB of archives.
After this operation, 438 kB of additional disk space will be used.
Get:1 http://us.archive.ubuntu.com/ubuntu/trusty/main xinetd i386 1:2.3.15-3ubu
ntu1 [102 kB]
24% [1 xinetd 33.0 kB/102 kB 32%] 3,644 B/s 29s
```

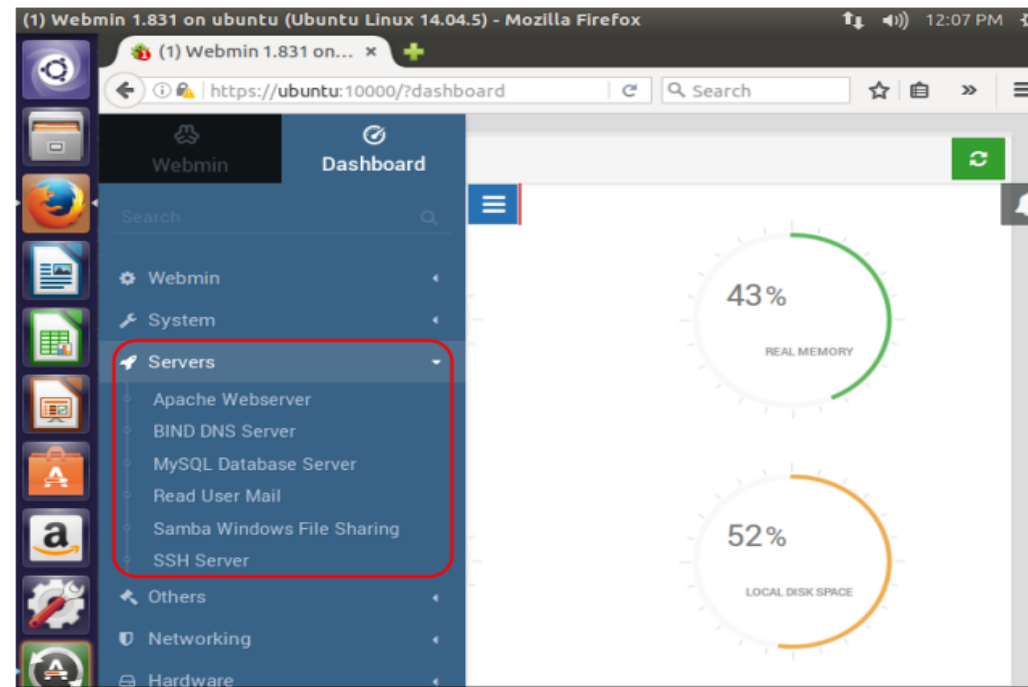
Activité 5 : administration à distance Linux

Corrections

- Pour quitter le mode root, tapez la commande exit :

```
root@ubuntu:~# exit  
logout  
koko@ubuntu:~$
```

- À la fin de l'installation, les serveurs s'affichent automatiquement dans la liste des serveurs dans l'onglet servers de Webmin. Nous pouvons donc procéder à la configuration de ces derniers via l'interface de Webmin et à partir de n'importe quelle machine du réseau.



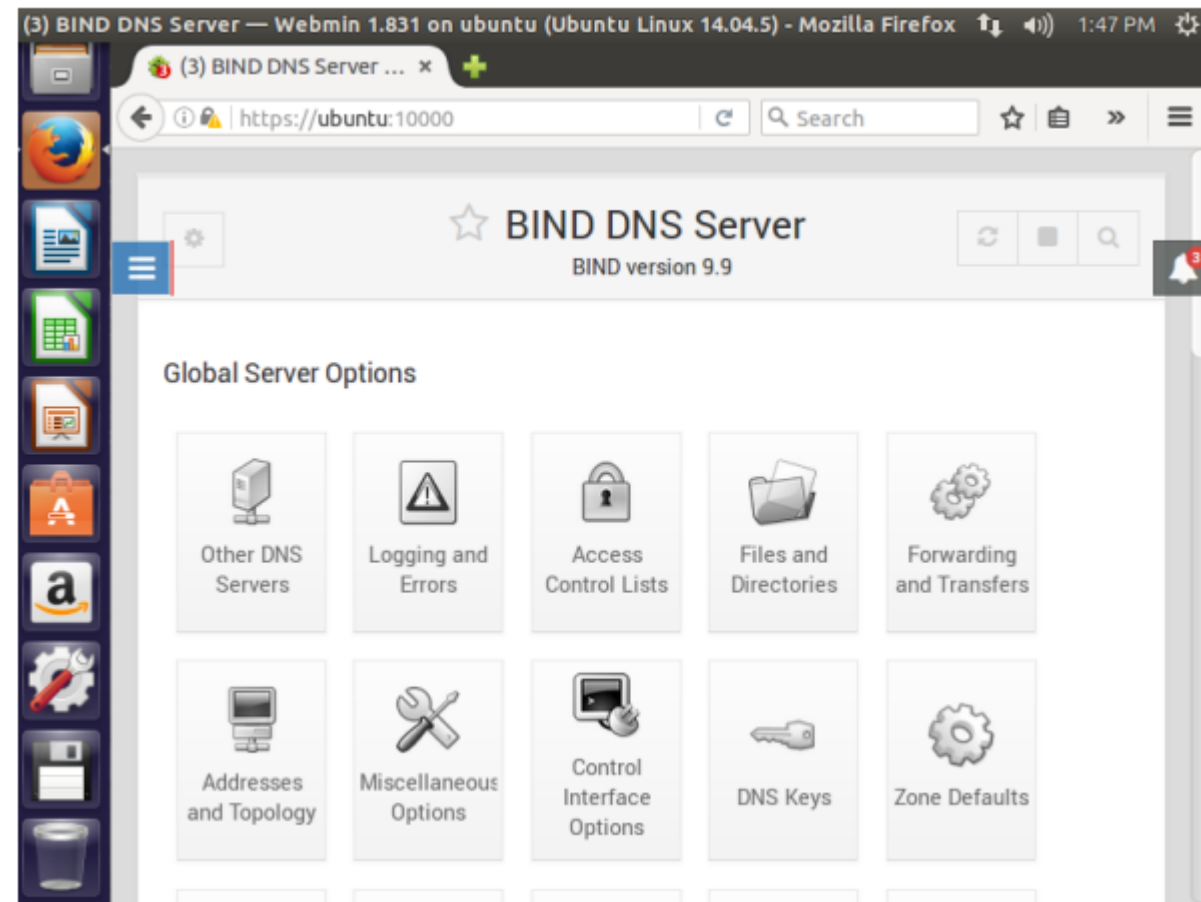
Activité 5 : administration à distance Linux

Corrections



Étape 4 : configuration des serveurs via Webmin

1. Configuration du serveur Bind via l'interface Webmin.



Activité 5 : administration à distance Linux

Corrections



- Création d'une nouvelle zone

Existing DNS Zones

Select all Invert selection **Create master zone** Create slave zone

Create stub zone Create forward zone Create delegation zone

Create zones from batch file

Create Master Zone

New master zone options

Zone type **1** Forward (Names to Addresses) Reverse (Addresses to)

Domain name / Network **2**

Records file **3** Automatic

Master server **4** **5** Add NS record

Email address **6**

Use zone template? Yes No IP address for template records

Add reverses for template addresses? Yes No

Refresh time seconds Transfer retry time

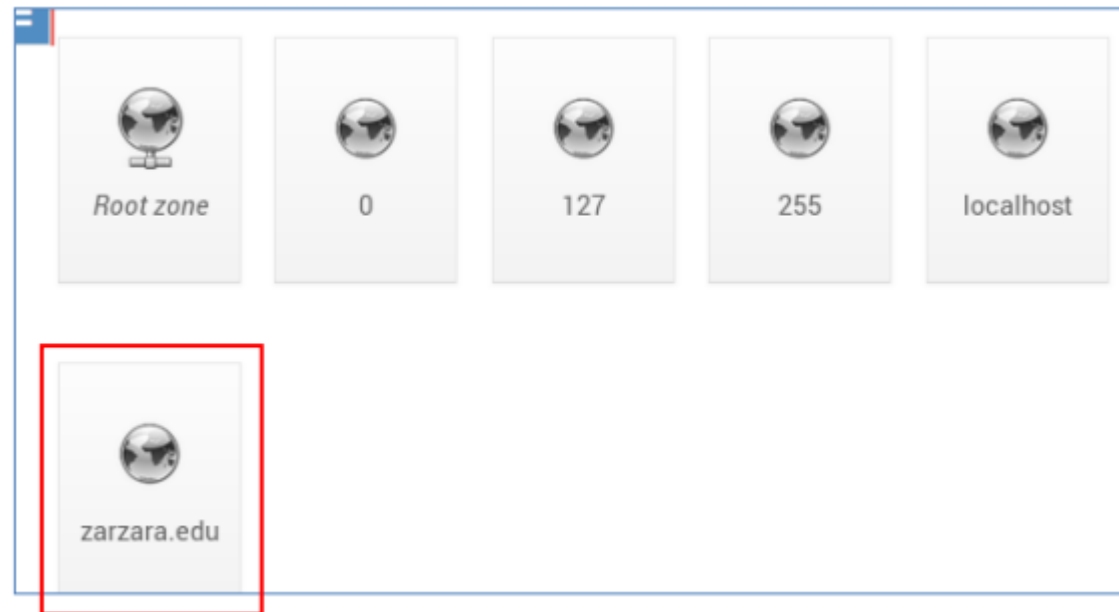
Expiry time seconds Negative cache time

7

Activité 5 : administration à distance Linux

Corrections

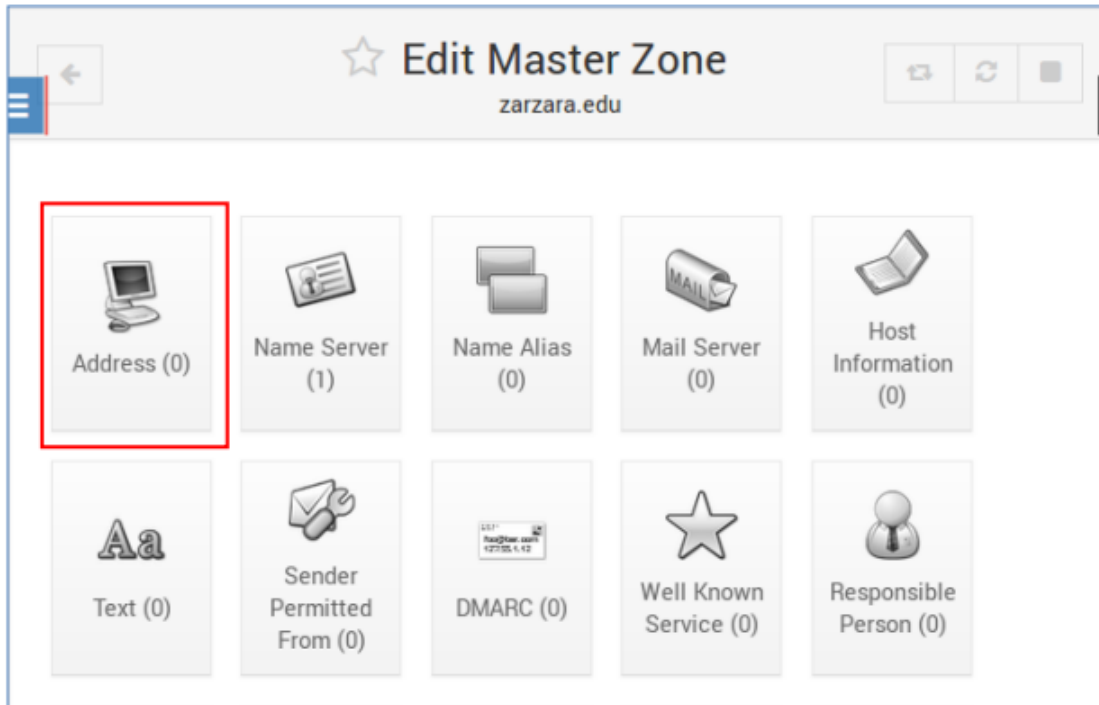
- Laissez tout le reste par défaut, puis cliquez sur l'icône de la nouvelle zone.



Activité 5 : administration à distance Linux

Corrections

- Nous pouvons à présent ajouter des adresses de machines



Edit Master Zone
zarzara.edu

Address (0)

Name Server (1)

Name Alias (0)

Mail Server (0)

Host Information (0)

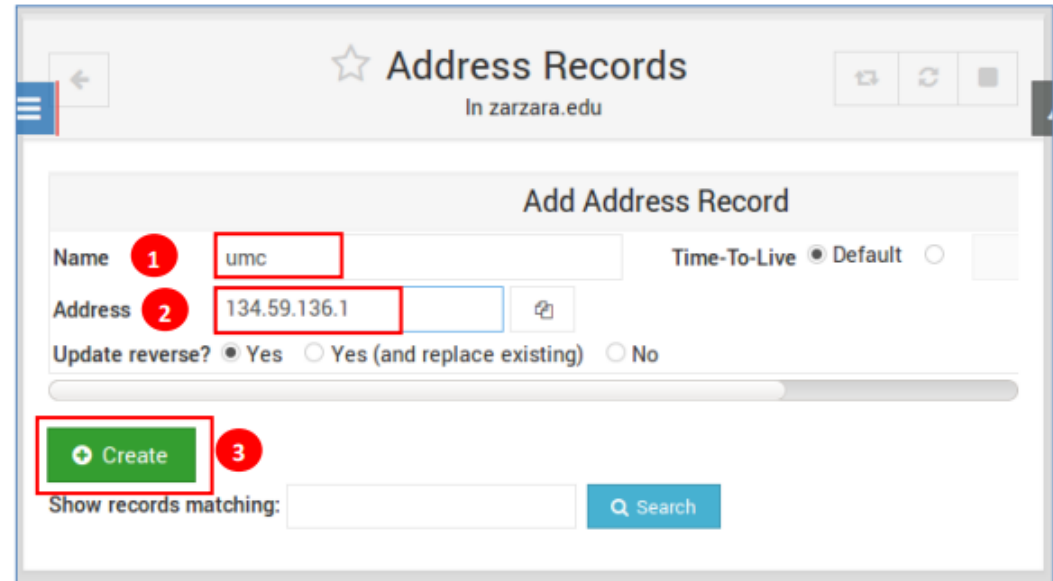
Text (0)

Sender Permitted From (0)

DMARC (0)

Well Known Service (0)

Responsible Person (0)



Address Records
In zarzara.edu

Add Address Record

Name **1** umc Time-To-Live Default

Address **2** 134.59.136.1

Update reverse? Yes Yes (and replace existing) No

3

Show records matching:

Activité 5 : administration à distance Linux

Corrections



- La liste des enregistrements créés dans la zone zarzara.edu :

	Name	Type	TTL	Values
<input type="checkbox"/>	zarzara.edu	NS	Default	ubuntu.
<input type="checkbox"/>	umc.zarzara.edu	A	Default	134.59.136.1
<input type="checkbox"/>	client-1.zarzara.edu	A	Default	134.59.136.2
<input type="checkbox"/>	client-2.zarzara.edu	A	Default	134.59.136.3

Select all Invert selection

Delete Selected

Activité 5 : administration à distance Linux

Corrections



- Nous pouvons tester le fonctionnement du serveur Bind avec la commande « nslookup ». Remarque : il faut démarrer le serveur avant de lancer le test.

```
root@ubuntu:~# service bind9 start
* Starting domain name service... bind9 [ OK ]
root@ubuntu:~#
```

```
root@ubuntu:~# /usr/sbin/named
root@ubuntu:~# /usr/sbin/named
root@ubuntu:~# nslookup
> server localhost
Default server: localhost
Address: 127.0.0.1#53
>
```

```
> umc.zarzara.edu
Server:          localhost
Address:         127.0.0.1#53

Name:   umc.zarzara.edu
Address: 134.59.136.1
```

Activité 5 : administration à distance Linux

Corrections



- Pour changer l'adresse IP du serveur DNS, il faut aller vers le fichier /etc/resolv.conf o Dans Webmin, aller dans Others/File Manager, o Rechercher le fichier resolv.conf dans le répertoire etc.

File Name	Permissions	Owner	Group	Size	Modified
resolv.conf	0777	root	root	000 bytes	2017/04/04 - 14:37:32

- Avant de modifier le fichier il faut changer les droits sur ce dernier

```
root@ubuntu:~# chmod 664 /run/resolvconf/resolv.conf
```

```
resolv.conf x
# Dynamic resolv.conf(5) file for glibc resolver(3) generated by resolvconf(8)
#     DO NOT EDIT THIS FILE BY HAND -- YOUR CHANGES WILL BE OVERWRITTEN
nameserver 134.59.136.7
search zarzara.edu
```

Activité 5 : administration à distance Linux

Corrections



- Test de fonctionnement de serveur DNS

```
root@ubuntu:~# nslookup
> ubuntu
Server:      134.59.136.7
Address:    134.59.136.7#53

Name:   ubuntu.zarzara.edu
Address: 134.59.136.7
>
```

```
> ubuntu.zarzara.edu
Server:      134.59.136.7
Address:    134.59.136.7#53

Name:   ubuntu.zarzara.edu
Address: 134.59.136.7
>
```

```
> client.zarzara.edu
Server:      134.59.136.7
Address:    134.59.136.7#53

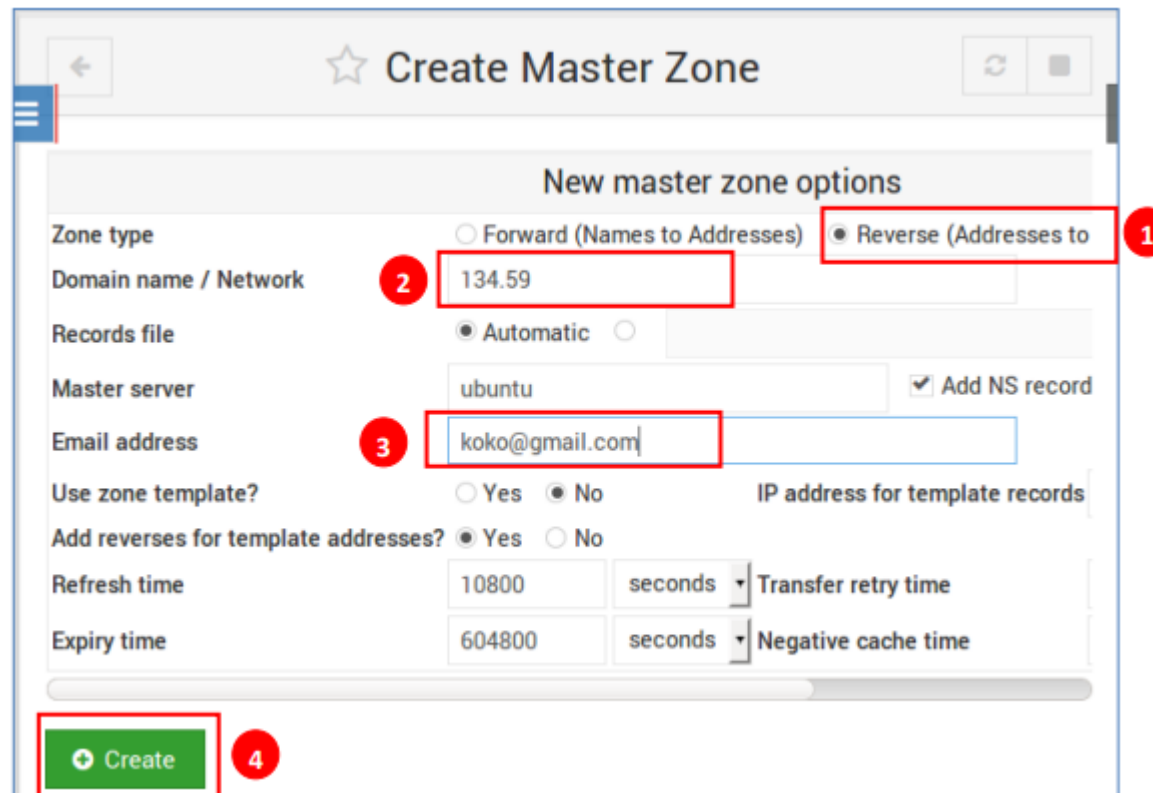
Name:   client.zarzara.edu
Address: 134.59.136.4
>
```

```
root@ubuntu:~# ping ubuntu.zarzara.edu
PING ubuntu.zarzara.edu (134.59.136.7) 56(84) bytes of data.
64 bytes from 134.59.136.7: icmp_seq=1 ttl=64 time=0.025 ms
64 bytes from 134.59.136.7: icmp_seq=2 ttl=64 time=0.074 ms
64 bytes from 134.59.136.7: icmp_seq=3 ttl=64 time=0.101 ms
64 bytes from 134.59.136.7: icmp_seq=4 ttl=64 time=0.112 ms
```

Activité 5 : administration à distance Linux

Corrections

- De la même manière, créer la zone inverse : Servers -> BIND DNS Server → Create master zone (création d'une zone primaire) :



☆ Create Master Zone

New master zone options

Zone type Forward (Names to Addresses) Reverse (Addresses to) 1

Domain name / Network 2 134.59

Records file Automatic

Master server ubuntu Add NS record

Email address 3 koko@gmail.com

Use zone template? Yes No IP address for template records

Add reverses for template addresses? Yes No

Refresh time 10800 seconds Transfer retry time

Expiry time 604800 seconds Negative cache time

4 Create

Activité 5 : administration à distance Linux

Corrections



- Aller ensuite dans adresse inverse et créer les adresses inverses comme suit :

Reverse Address Records
In 134.59

Add Reverse Address Record

Address **1** Time-To-Live Default

Hostname **2**

Update forward? Yes No

3

Show records matching:

Activité 5 : administration à distance Linux

Corrections



	Address	TTL	Hostname
<input type="checkbox"/>	134.59.136.7	Default	ubuntu.
<input type="checkbox"/>	134.59.136.4	Default	Client.

Select all Invert selection

The screenshot shows a DNS zone editor interface with a grid of zone buttons. Each button contains a globe icon and a label. The '134.59' zone is highlighted with a red border.

 Root zone	 0	 127	 134.59	 255
 localhost	 zarzara.edu			

Activité 5 : administration à distance Linux

Corrections



```
C:\Windows\system32\cmd.exe
Microsoft Windows [version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation. Tous droits réservés.
C:\Users\keltoum>ping ubuntu.zarzara.edu

Envoi d'une requête 'ping' sur ubuntu.zarzara.edu [134.59.136.7] avec 32 octets
de données :
Réponse de 134.59.136.7 : octets=32 temps=1 ms TTL=64
Réponse de 134.59.136.7 : octets=32 temps=1 ms TTL=64
Réponse de 134.59.136.7 : octets=32 temps<1ms TTL=64
Réponse de 134.59.136.7 : octets=32 temps<1ms TTL=64

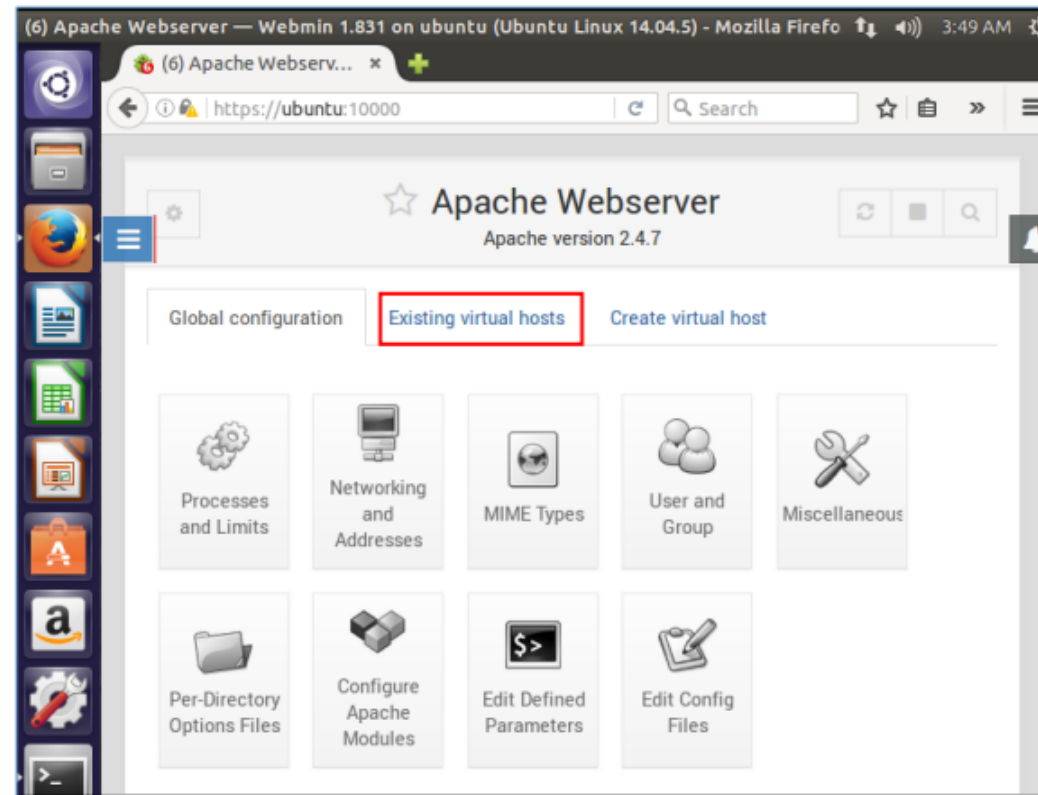
Statistiques Ping pour 134.59.136.7:
    Paquets : envoyés = 4, reçus = 4, perdus = 0 (perte 0%),
Durée approximative des boucles en millisecondes :
    Minimum = 0ms, Maximum = 1ms, Moyenne = 0ms

C:\Users\keltoum>_
```

Activité 5 : administration à distance Linux

Corrections

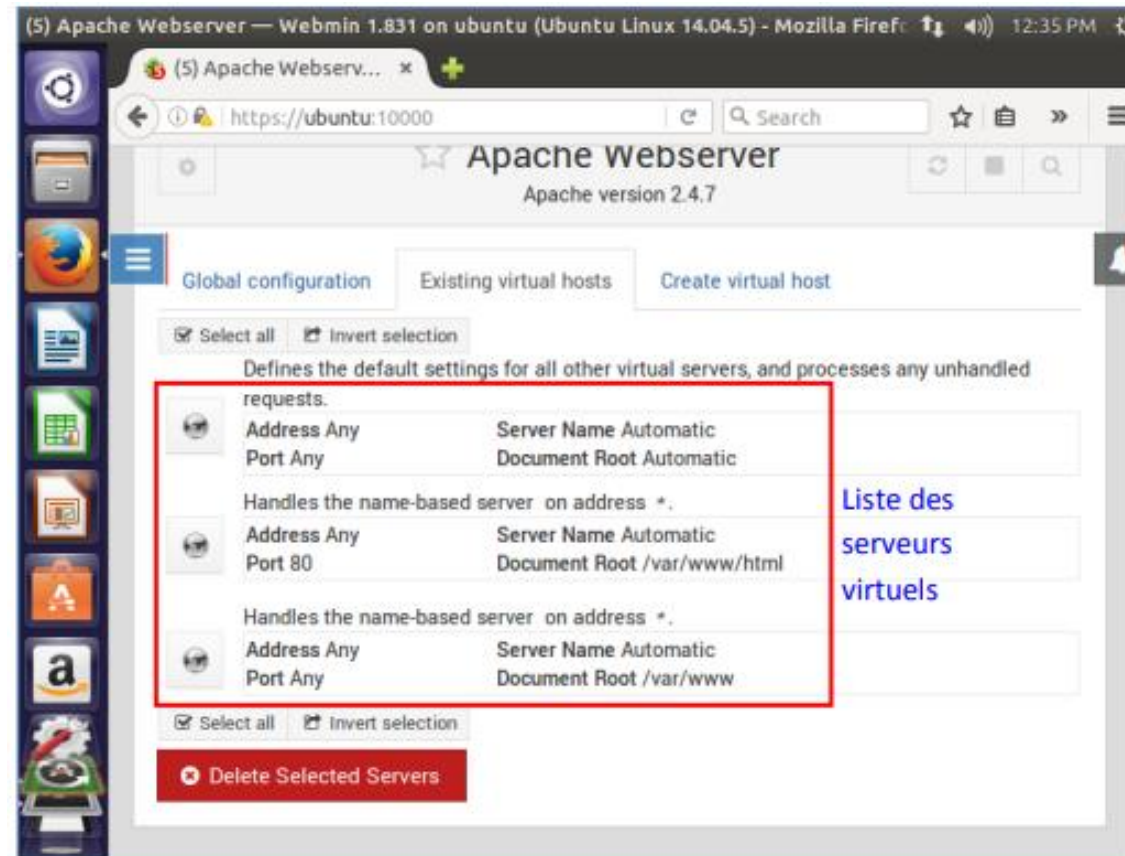
2. Configuration du serveur web Apache sous Webmin.
 - Ouvrir la fenêtre principale de configuration à partir de Webmin, celle-ci se présente comme suit :



Activité 5 : administration à distance Linux

Corrections

- Pour voir la liste des serveurs virtuels, clic sur l'onglet "Existing virtual hosts". C'est à partir de cette fenêtre que vous pouvez créer votre serveur virtuel hébergeant votre site.



The screenshot shows the Apache Webserver Webmin interface. The browser title is "(5) Apache Webserver — Webmin 1.831 on ubuntu (Ubuntu Linux 14.04.5) - Mozilla FireF...". The address bar shows "https://ubuntu:10000". The page title is "Apache webserver" and the version is "Apache version 2.4.7".

The interface has three tabs: "Global configuration", "Existing virtual hosts" (selected), and "Create virtual host". Below the tabs, there are "Select all" and "Invert selection" buttons. The main content area shows a table of virtual hosts:

Defines the default settings for all other virtual servers, and processes any unhandled requests.	
Address Any	Server Name Automatic
Port Any	Document Root Automatic
Handles the name-based server on address *.	
Address Any	Server Name Automatic
Port 80	Document Root /var/www/html
Handles the name-based server on address *.	
Address Any	Server Name Automatic
Port Any	Document Root /var/www

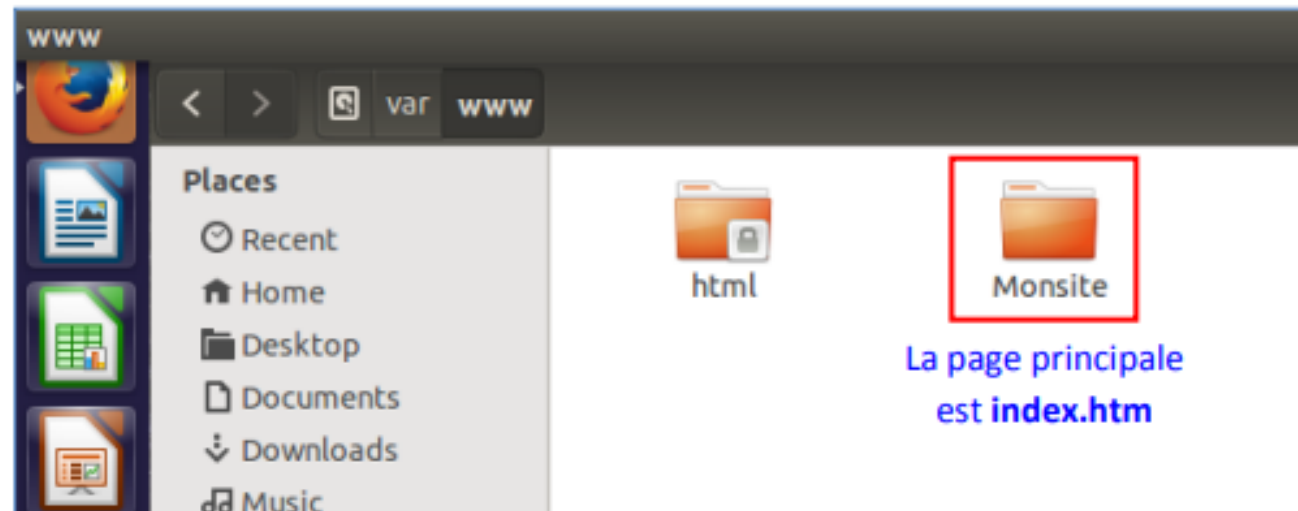
Below the table, there are "Select all" and "Invert selection" buttons, and a red "Delete Selected Servers" button. A blue label "Liste des serveurs virtuels" points to the table.

Activité 5 : administration à distance Linux

Corrections

- Création d'un serveur virtuel
 - La création et la configuration d'un "serveur virtuel" va vous permettre de configurer Apache afin d'y faire fonctionner un site Web et de façon optimale.
 - Mais avant vous devez copier le dossier de votre site dans le répertoire (var/www). Pour ce faire, il faut modifier les permissions sur ce dossier de la manière suivante :

```
root@ubuntu:~# chmod 777 /var/www/  
root@ubuntu:~#
```



Activité 5 : administration à distance Linux

Corrections



Global configuration Existing virtual hosts **Create virtual host** 1

2

Handle connections to address Those not handled by another virtual host Specific address .. 134.59.134.7

Listen on address (if needed)

Port 3 Default Any 80

Document Root 4

Allow access to this directory

Server Name Automatic

Add virtual server to file Standard *httpd.conf* file New file under virtual servers directory

Copy directives from

5

Handles the name-based server *ubuntu* on address *.

Address Any	Server Name ubuntu
Port 80	Document Root /var/www/Monsite

Activité 5 : administration à distance Linux

Corrections



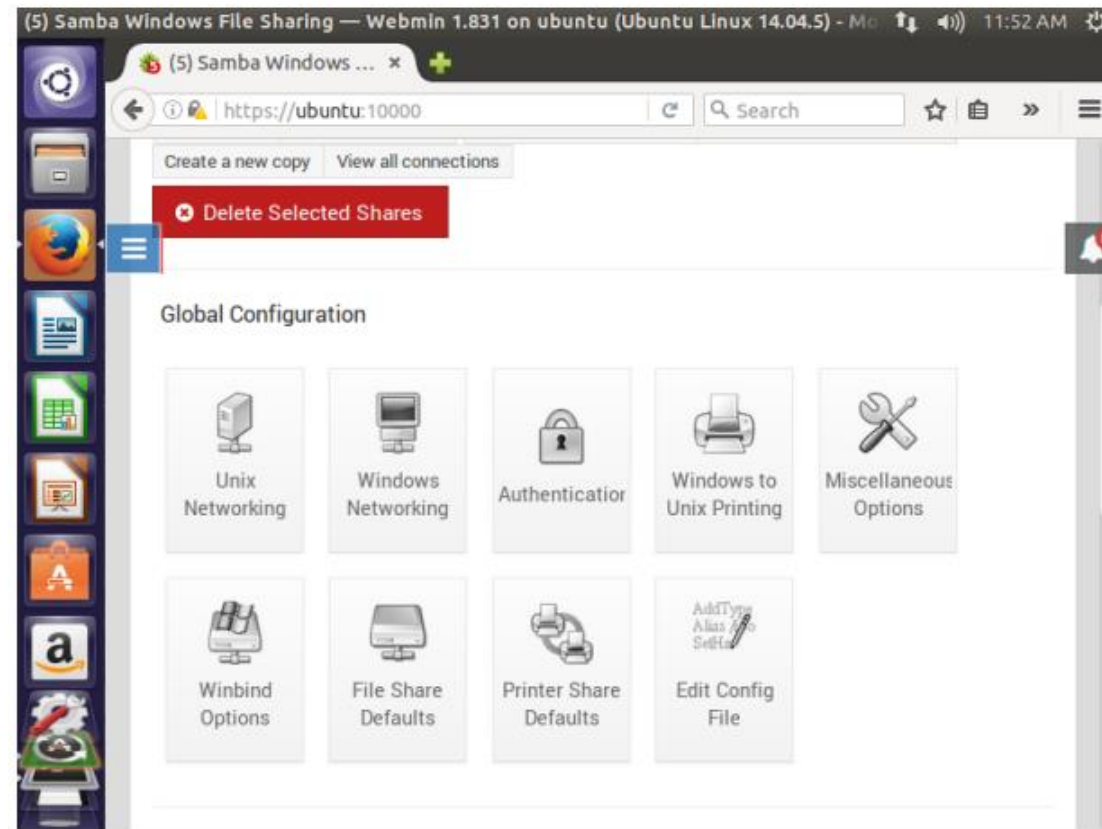
- Tester le fonctionnement de votre site via le navigateur Web.



Activité 5 : administration à distance Linux

Corrections

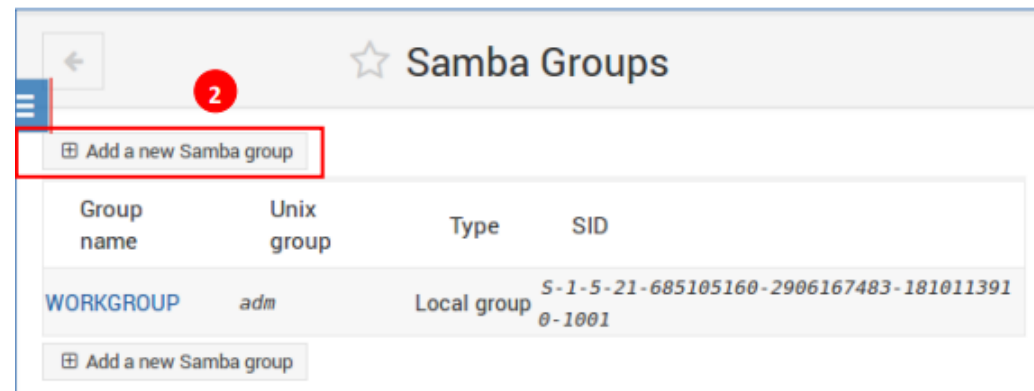
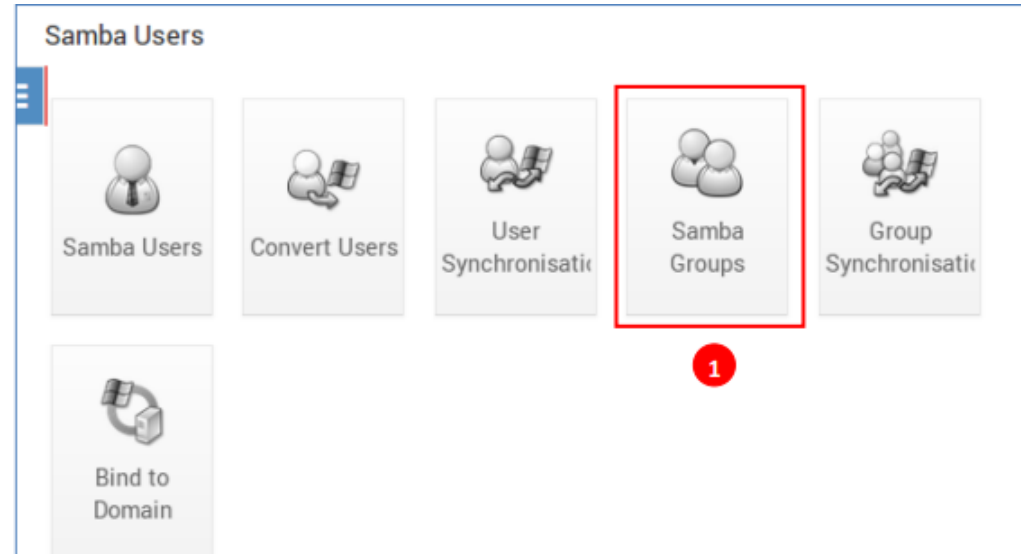
Étape 4 : configuration du serveur Samba



Activité 5 : administration à distance Linux

Corrections

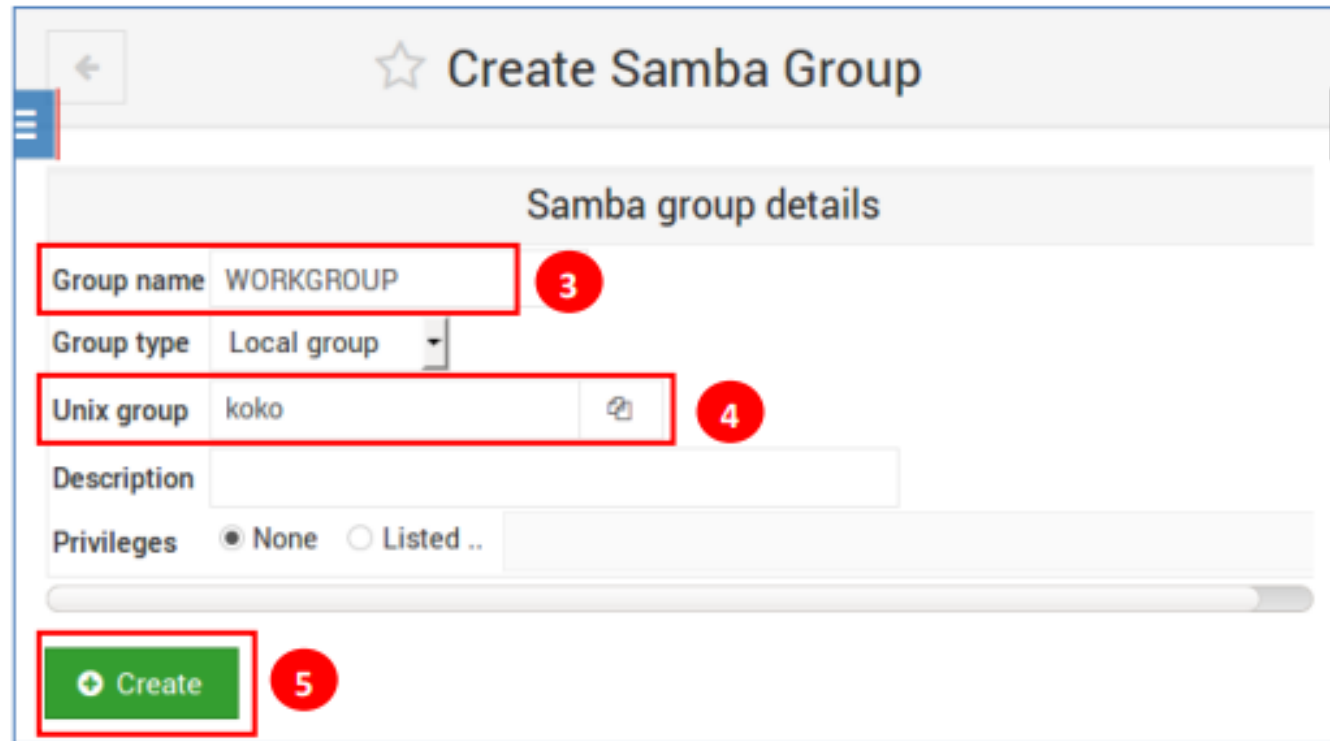
- Ajouter un nouveau groupe "WORKGROUP".



Activité 5 : administration à distance Linux

Corrections

- Ajouter un nouveau groupe "WORKGROUP".



← ☆ Create Samba Group

Samba group details

Group name WORKGROUP 3

Group type Local group

Unix group koko 4

Description

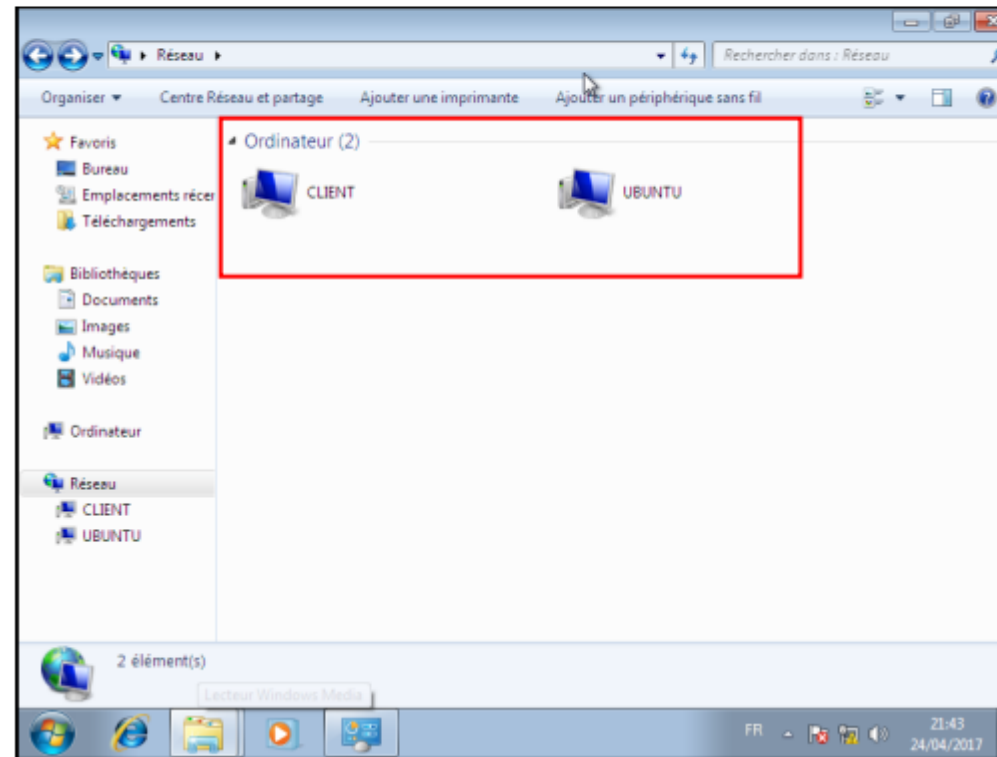
Privileges None Listed ..

➕ Create 5

Activité 5 : administration à distance Linux Corrections

- Le nouveau groupe a été créé.

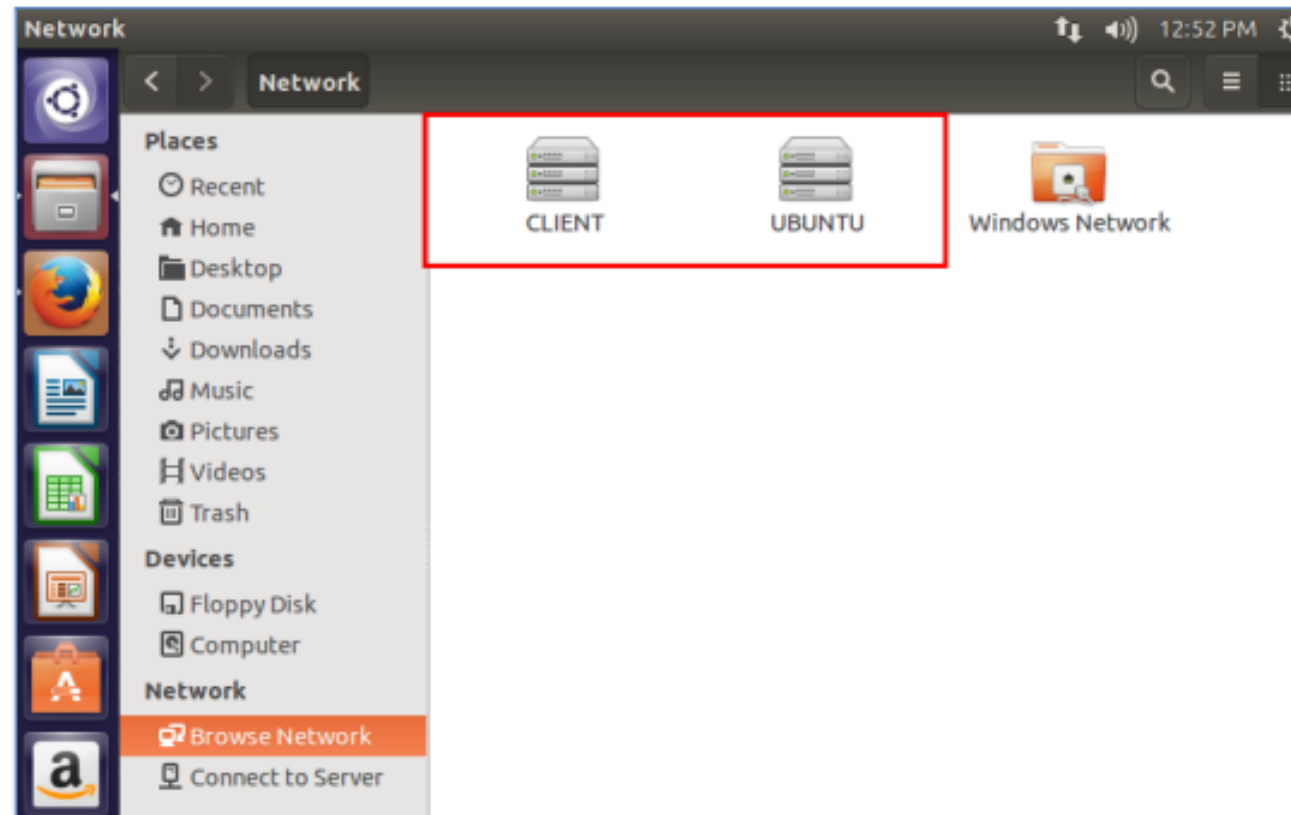
Group name	Unix group	Type	SID
WORKGROUP	adm	Local group	S-1-5-21-685105160-2906167483-1810113910-1001



Activité 5 : administration à distance Linux

Corrections

- Le nouveau groupe a été créé.

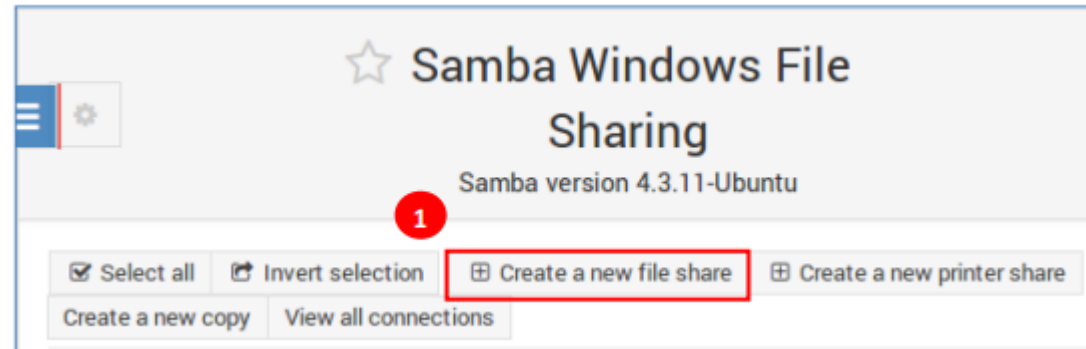


Activité 5 : administration à distance Linux

Corrections



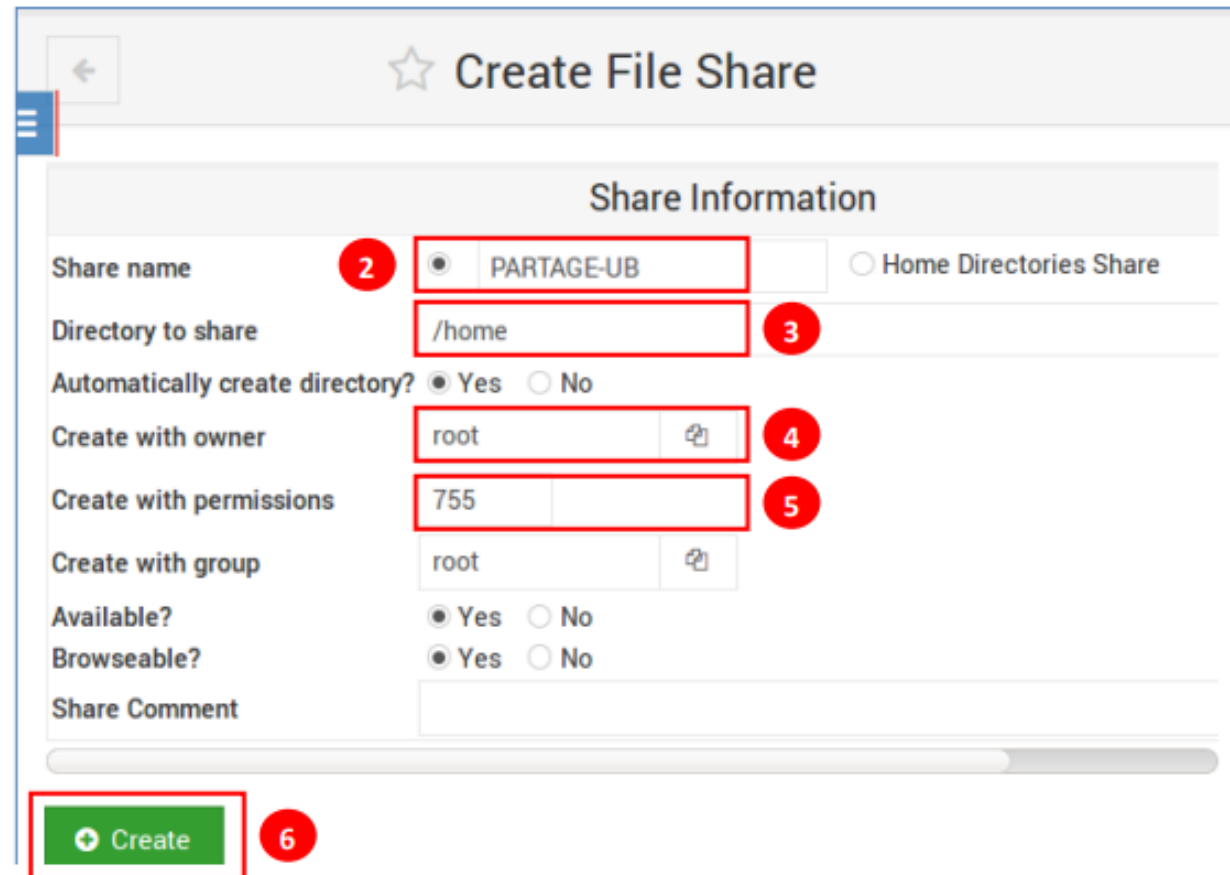
- Partager un fichier avec des machines Windows.



Activité 5 : administration à distance Linux

Corrections

- Partager un fichier avec des machines Windows.



☆ Create File Share

Share Information

Share name PARTAGE-UB Home Directories Share

Directory to share /home

Automatically create directory? Yes No

Create with owner root

Create with permissions 755

Create with group root

Available? Yes No

Browseable? Yes No

Share Comment

Create

Activité 5 : administration à distance Linux

Corrections



- Partager un fichier avec des machines Windows.

	Share Name	Path	Security
<input type="checkbox"/>	printers	All Printers	Printable to all known users
<input type="checkbox"/>	print\$	/var/lib/samba/printers	Read only to all known users
<input type="checkbox"/>	PARTAGE-UB	/home	Read only to all known users

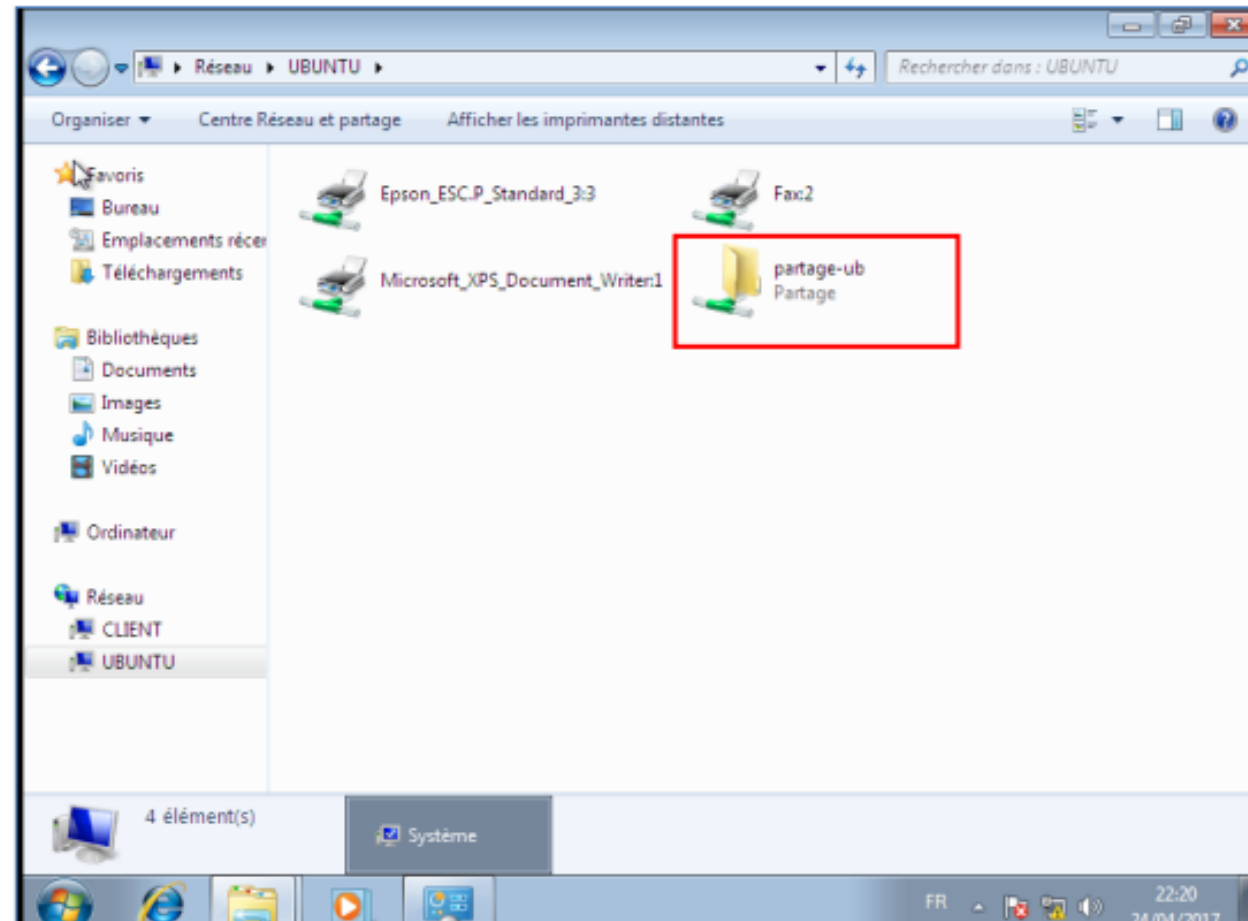
Select all Invert selection Create a new file share Create a new printer share

Create a new copy View all connections

Activité 5 : administration à distance Linux

Corrections

- Partager un fichier avec des machines Windows.

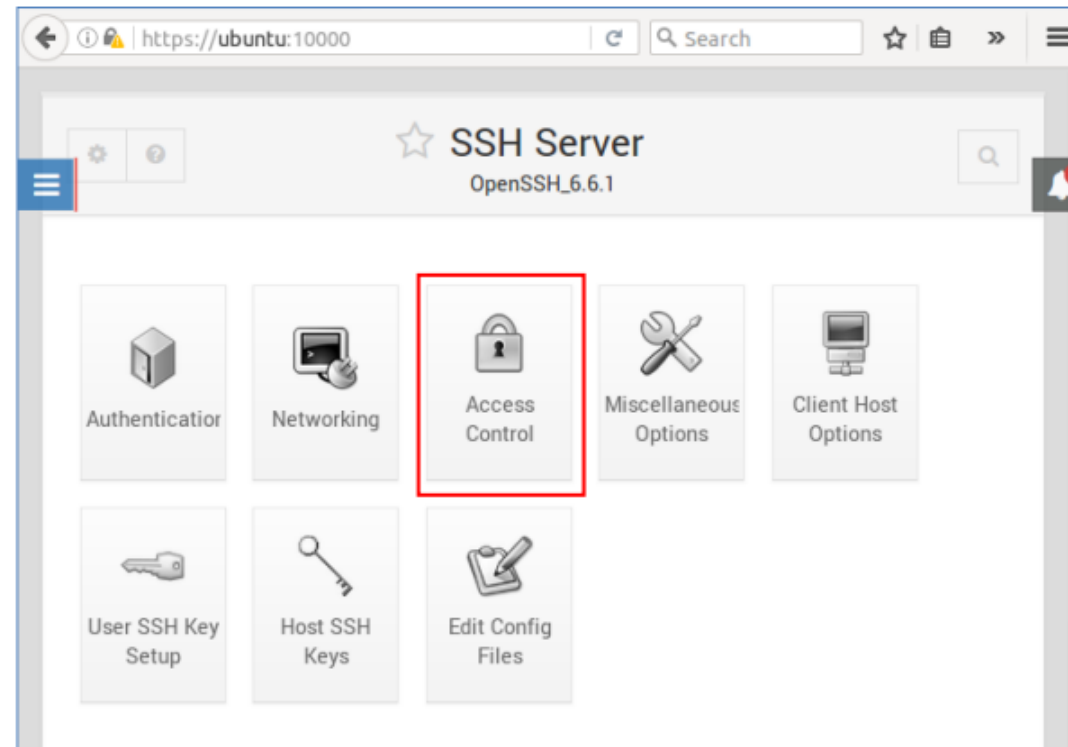


Activité 5 : administration à distance Linux

Corrections

3. Configuration du serveur SSH sous Webmin.

Après l'installation des paquets OpenSSH sur la machine serveur (Ubuntu 9.10), se connecter à Webmin et aller dans Servers-> SSH server. Vous aurez l'interface suivante :

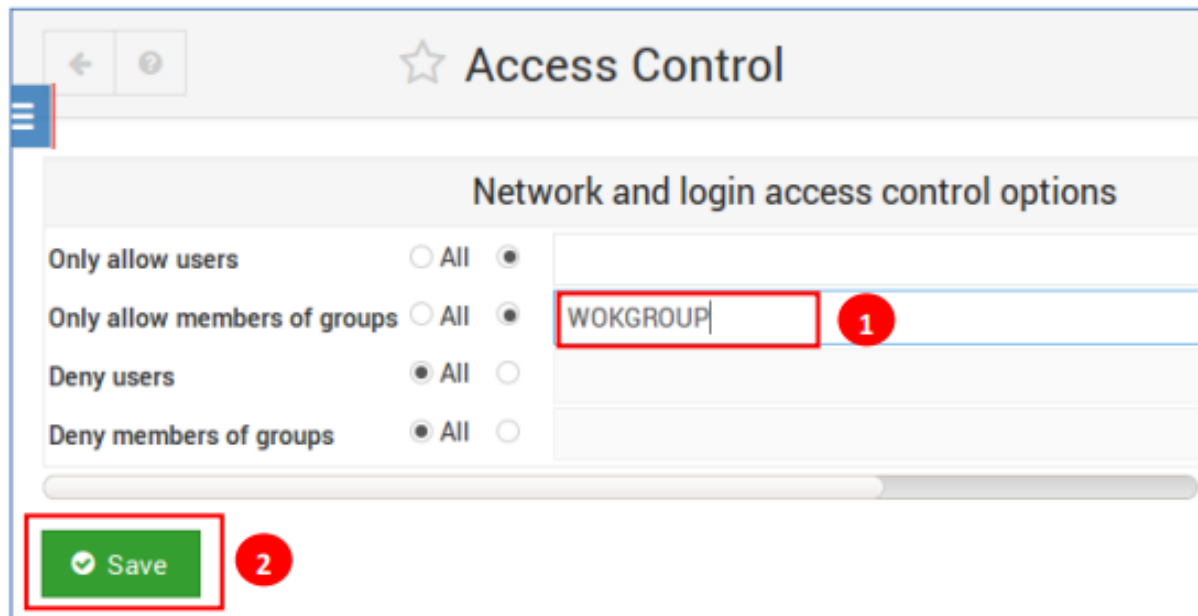


Activité 5 : administration à distance Linux

Corrections

- Restriction de l'accès au serveur SSH.

Sur la page principale du module, cliquez sur l'icône "contrôle d'accès" pour mettre en place une liste d'utilisateurs autorisés ou bien refusés.



☆ Access Control

Network and login access control options

Only allow users All

Only allow members of groups All WOKGROUP 1

Deny users All

Deny members of groups All

Save 2

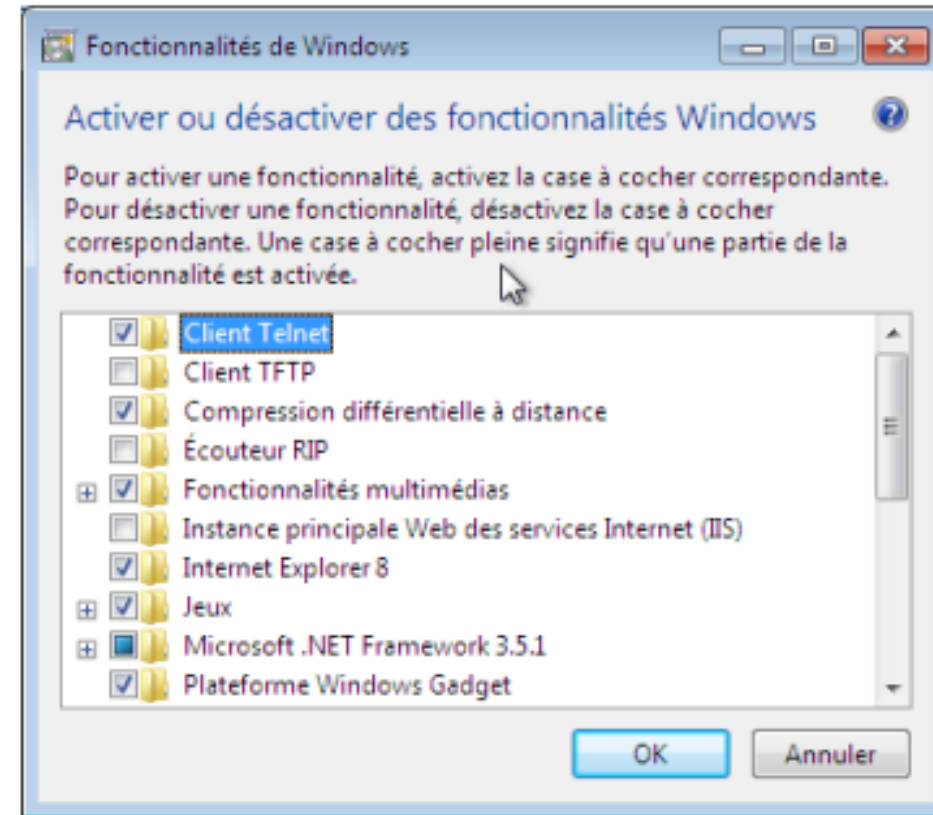
Activité 5 : administration à distance Linux

Corrections

- Commandes Telnet

Les commandes Telnet permettent de communiquer avec un ordinateur distant qui utilise le protocole Telnet.

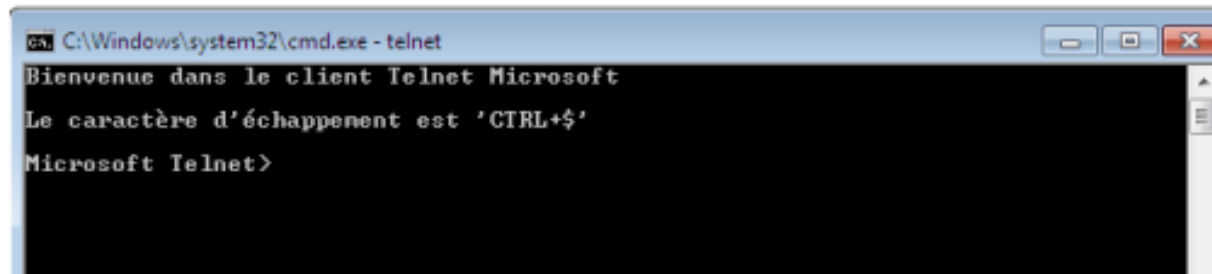
- Sous Windows, Telnet n'est pas installé par défaut (pour des raisons de sécurité). Pour l'activer, suivez la procédure ci-dessous :
 - Cliquez sur Démarrer et ensuite sur Panneau de configuration.
 - Dans la rubrique Programmes, cliquez sur Désinstaller un programme.
 - Cliquez sur le menu de gauche "Activer ou désactiver des fonctionnalités Windows". Après quelques secondes, la fenêtre ci-contre s'affiche.
 - Cochez à gauche la case Client Telnet et pour finir cliquez sur OK.



Activité 5 : administration à distance Linux

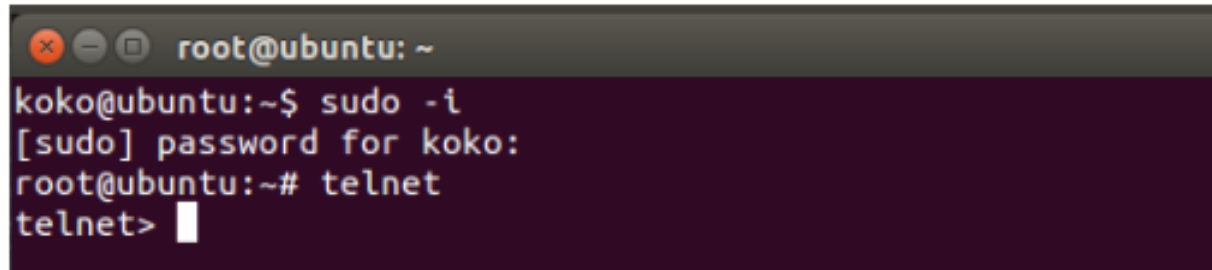
Corrections

- Puis, allez dans Démarrer > Exécuter, et tapez simplement la commande : telnet.



```
C:\Windows\system32\cmd.exe - telnet
Bienvenue dans le client Telnet Microsoft
Le caractère d'échappement est 'CTRL+$'
Microsoft Telnet>
```

- Pour les utilisateurs de Linux ou Mac OS, c'est quasiment pareil. Il suffit d'ouvrir un terminal et d'entrer la commande : telnet.



```
root@ubuntu: ~
koko@ubuntu:~$ sudo -i
[sudo] password for koko:
root@ubuntu:~# telnet
telnet> |
```

Activité 5 : administration à distance Linux

Corrections



- Les commandes standard de Telnet sont les suivantes :

Commande	Description
?	Affiche l'aide
close	Termine la session Telnet
display	Affiche à l'écran les paramètres de la connexion (type de terminal, port)
environ	Permet de définir les variables d'environnement du système d'exploitation
logout	Permet de se déconnecter
mode	Bascule entre les modes de transfert ASCII (transfert d'un fichier)
open	Permet de lancer une autre connexion à partir de la connexion en cours
quit	Quitte l'application Telnet
set	Modifie les paramètres de la connexion
unset	Charge les paramètres de connexion par défaut



WEBFORCE
BE THE CHANGE



PARTIE 4

Créer des programmes pour les tâches d'administration

Dans ce module, vous allez :

- Automatiser les tâches redondantes
- Optimiser l'exécution des tâches d'administration



07 heures



ACTIVITÉ n° 1

Automatisation de tâches redondantes

Compétences visées :

- Automatiser les tâches redondantes

Recommandations clés :

- Lire attentivement l'énoncé et les questions
- Se référer au cours



02 heures

CONSIGNES

1. Pour le formateur :

- Laisser à l'apprenant l'occasion de comprendre seul l'énoncé.
- S'assurer de la bonne compréhension du contexte avant de lui laisser le temps de réfléchir et répondre.
- Discuter les réponses des apprenants avant de donner la solution.

2. Pour l'apprenant :

- Lire et bien comprendre la question.
- Essayer de trouver par vous-même une réponse à cette question et la noter.
- Parcourir les réponses proposées.
- Pour chaque réponse : comparez-la à votre réponse et cochez-la si elle lui correspond ou bien est compatible.

3. Conditions de réalisation :

- Individuel ou par groupes (2 ou 3 maximum).
- Support de résumé théorique accompagnant.
- Stylo et feuille de papier.

4. Critères de réussite :

- Le stagiaire est-il capable de :
 - Automatiser les tâches redondantes ?
- Réponses correctes pour au moins 70 % des questions.



Activité 1 : automatisation de tâches redondantes



Quelques cas d'utilisation d'automatisation très populaires :

- **Service d'assistance automatisé** : diverses entreprises ou banques proposent des services d'assistance où une personne pose des questions par e-mail, SMS, Slack, etc. et reçoit immédiatement la réponse.
- **Remplissage automatique du formulaire** : cette tâche ennuyeuse consiste à remplir des formulaires en ligne et à réinitialiser les mots de passe oubliés, etc. Elle peut être facilement gérée par des scripts automatisés.
- **Le courrier automatique se déclenche quotidiennement** : cela aide quand vous devez envoyer une pièce jointe quotidiennement. Un script automatisé programmé quotidiennement gèrera facilement cette tâche.
- **Mises à jour rapides de la météo** : vous pouvez également vérifier la météo facilement et rapidement, mais si elle vous parvient automatiquement, c'est plus pratique. Donc, c'est aussi un exemple d'automatisation où vous recevez automatiquement des notifications de mises à jour météo quotidiennes.

Activité 1 : automatisation de tâches redondantes



- Ce ne sont là que quelques exemples d'automatisation de choses ennuyeuses qui rendent finalement votre vie un peu plus confortable. Mais dans cet article, nous allons automatiser les applications Windows.
- Donc, pour automatiser l'application Windows, nous allons utiliser Python. Python offre une grande lisibilité et une syntaxe facile à apprendre. En Python, nous allons utiliser le module **pywinauto** dont le seul but est d'automatiser les applications Windows. Alors, commençons.
- Commande pour installer pywinauto : **pip install pywinauto**
- Nous allons voir quelques exemples dans cet article. Tout d'abord, nous commencerons par l'automatisation du bloc-notes qui est assez basique. Après cela, nous automatiserons l'application Excel. Dans Excel, nous essaierons de lire les données des cellules et d'écrire des données dans les cellules. Et nous examinerons également d'autres exemples à venir. C'est ce que nous allons faire avec Python et pywinauto.
- Une fois que nous avons installé pywinauto, la première chose à faire est de déterminer quel backend d'accessibilité de pywinauto utiliser pour notre application. Il existe essentiellement deux back-ends pris en charge dans Windows.

Activité 1 : automatisation de tâches redondantes

Exercice



Exercice :

1. Automatiser le traitement d'un fichier Notepad
2. Automatiser l'application Excel

Activité 1 : automatisation de tâches redondantes

Correction



1. Automatiser Notepad avec pywinauto :

En regardant cet exemple, nous apprendrons les bases de l'automatisation à l'aide de pywinauto. Nous examinerons certains concepts tels que comment démarrer une application, comment la maximiser, comment écrire du texte dans Notepad.

- **Importation des fichiers nécessaires :**

```
from pywinauto import application
```

- Maintenant, démarrez notre application Notepad en utilisant la méthode de démarrage de l'application que nous avons importée ci-dessus.

```
app = application.Application()  
app.start("Notepad.exe")
```

- Nous devons extraire la fenêtre supérieure de notre application Notepad, il existe donc plusieurs façons de le faire.
- Nous pouvons obtenir une fenêtre supérieure comme suit :

app.windows()[0] : elle renvoie la liste de toutes les fenêtres et dans cette liste, la première fenêtre est ce dont nous avons besoin, alors capturez-la simplement à partir d'ici.

```
[<hwndwrapper.DialogWrapper - 'Untitled - Notepad', Notepad, 19282528>, <hwndwrapper.DialogWrapper - 'M', MSCTFIME UI, 28247444>, <hwndwrappe  
r.DialogWrapper - 'Default IME', IME, 16198120>]
```

Activité 1 : automatisation de tâches redondantes

Correction



- C'est toute la fenêtre que nous obtenons en imprimant le `app.windows()` et sur l'index 0 est la fenêtre dont nous avons besoin.
- `app.top_window()` : il ne vous renverra que la fenêtre supérieure de toutes les applications ouvertes et c'est notre fenêtre de bloc-notes dont nous avons besoin.
- `app['Untitled - Notepad']` : c'est une autre façon d'obtenir la fenêtre dont nous avons besoin. "Untitled - Notepad" est le nom de la fenêtre qui s'ouvre lors de l'exécution de la méthode `app.start()`.
- Avec cela, notre Notepad s'ouvrira. Maintenant, nous allons essayer d'écrire quelque chose dans la zone de texte. Donc, pour cela, utilisez simplement une méthode `type_keys()`. Tout ce que vous voulez taper dans la zone de texte, transmettez-le simplement comme une chaîne dans la méthode `type_keys()`.

```
text_message = "Hi Aliens! Text coming from pywinauto!"  
dlg.type_keys(text_message, with_spaces=True)
```

Activité 1 : automatisation de tâches redondantes

Correction



Output :

```
*Untitled - Notepad
File Edit Format View Help
Hi Aliens! Text coming from pywinauto!
```

Ln 1, Col 39 100% Windows (CRLF) UTF-8

Activité 1 : automatisation de tâches redondantes

Correction



- La plupart du temps, la fenêtre du bloc-notes s'ouvre comme un petit écran, donc si nous voulons la maximiser, nous pouvons simplement utiliser la méthode `is_maximized()`. Il renverra "False" si la fenêtre n'est pas dans un état entièrement maximisé.

```
if dlg.is_maximized() == False:  
    dlg.maximize()
```

Activité 1 : automatisation de tâches redondantes

Correction



2. Automatiser Excel

- Passons à l'étape suivante et travaillons à l'automatisation d'Excel. Cela va être intéressant car nous verrons comment vous pouvez lire des données à partir d'Excel, comment vous pouvez envoyer des données dans une feuille Excel. Commençons par ouvrir notre fichier Excel.

```
program_path = r"C:Program Files (x86)Microsoft Office\root\Office16\EXCEL.EXE"  
file_path = r"C:\Users\Desktop\sample.xlsx"
```

- Tout d'abord, démarrez notre application. Dans le cas d'Excel, c'est assez différent car nous devons ouvrir notre fichier xlsx via excel.exe, nous pouvons donc le faire comme ceci :

```
app = application.Application(backend="uia").start(r'{} {}'.format(program_path, file_path))
```

- Cela démarre notre application. Maintenant, nous devons sélectionner la fenêtre de sample.xlsx car nous devons travailler avec cela, donc comme indiqué ci-dessus, nous allons procéder de la façon suivante :

```
print(app.windows())
```

- Sortie :

```
[,,,]
```

Activité 1 : automatisation de tâches redondantes

Correction



- Ainsi, nous pouvons soit sélectionner la fenêtre d'index 0, soit obtenir notre fenêtre en donnant directement le nom.

```
dlg = app['sample.xlsx - Excel']
```

- Maintenant, disons que nous voulons savoir quels sont tous les attributs ou éléments de contrôle présents dans la fenêtre de l'application ouverte. Ensuite, nous pouvons utiliser la méthode `print_control_identifiers()`. Tous les noms possibles sont affichés sous forme de liste Python pour chaque contrôle sous forme d'arborescence.
- Pywinauto prend en charge les éléments de contrôle d'interface graphique les plus communs, notamment : bouton case à cocher, bouton radio, GroupBox, boîte combo, modifier, entête, zone de liste, affichage liste, menu popup, barre d'état, barre d'outils, info-bulles, arborescence, haut bas, ...

```
dlg.print_control_identifiers()
```

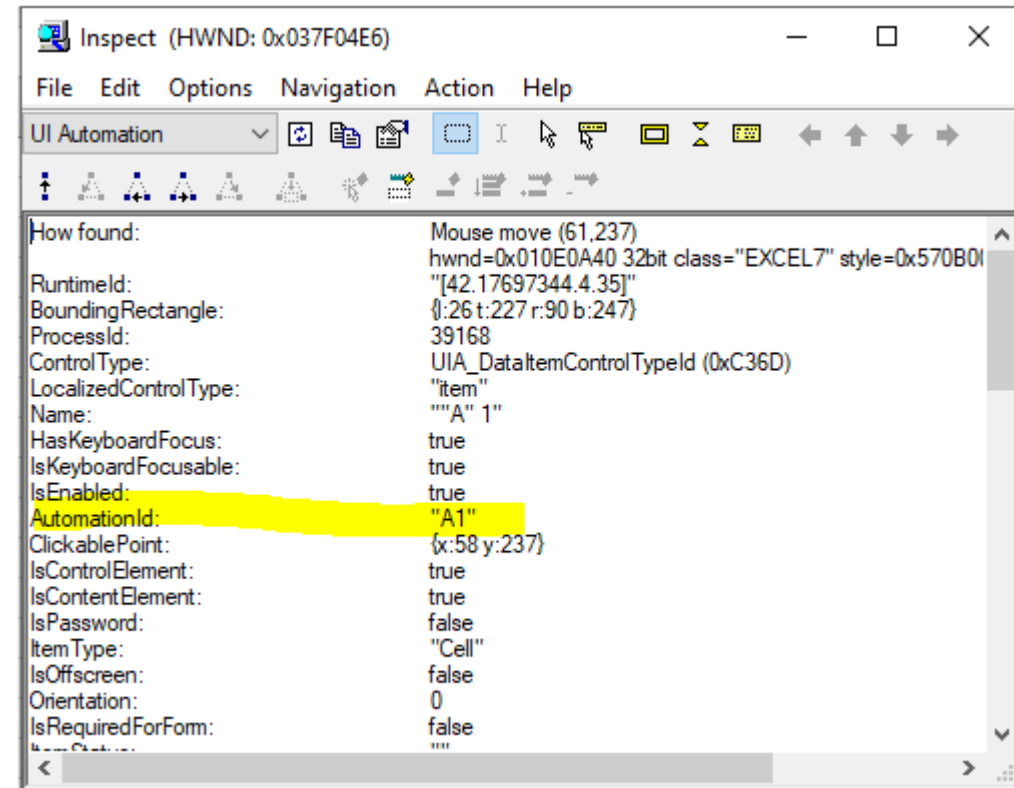
- Avant de poursuivre la lecture ou l'écriture de données dans une feuille Excel, je veux vous donner une brève introduction à l'outil d'inspection de l'interface utilisateur. Dans ce TP, je couvrirai Inspect.exe mais il existe également d'autres outils comme UISpy.exe, etc.

Activité 1 : automatisation de tâches redondantes

Correction

Inspect.exe

- C'est un excellent outil créé par Windows. Vous pouvez le télécharger à partir du lien hypertexte ci-dessus de inspect.exe. Une fois que vous avez ouvert cet outil, passez en mode d'automatisation MS UI, ce dont nous avons besoin.
- Inspect.exe est une application qui donnera les identifiants par lesquels vous pouvez extraire les identifiants de n'importe quel élément de l'interface utilisateur. Ainsi, dans l'outil inspect.exe, nous recherchons simplement l'ID d'automatisation (**Automation ID**) et nous pouvons ainsi accéder à n'importe quel élément de la fenêtre de l'application.
- Par exemple, si vous voulez l'ID d'automatisation de la cellule A1 dans la feuille Excel, cliquez simplement avec votre souris sur cette cellule et vous obtiendrez l'ID d'automatisation.



Activité 1 : automatisation de tâches redondantes

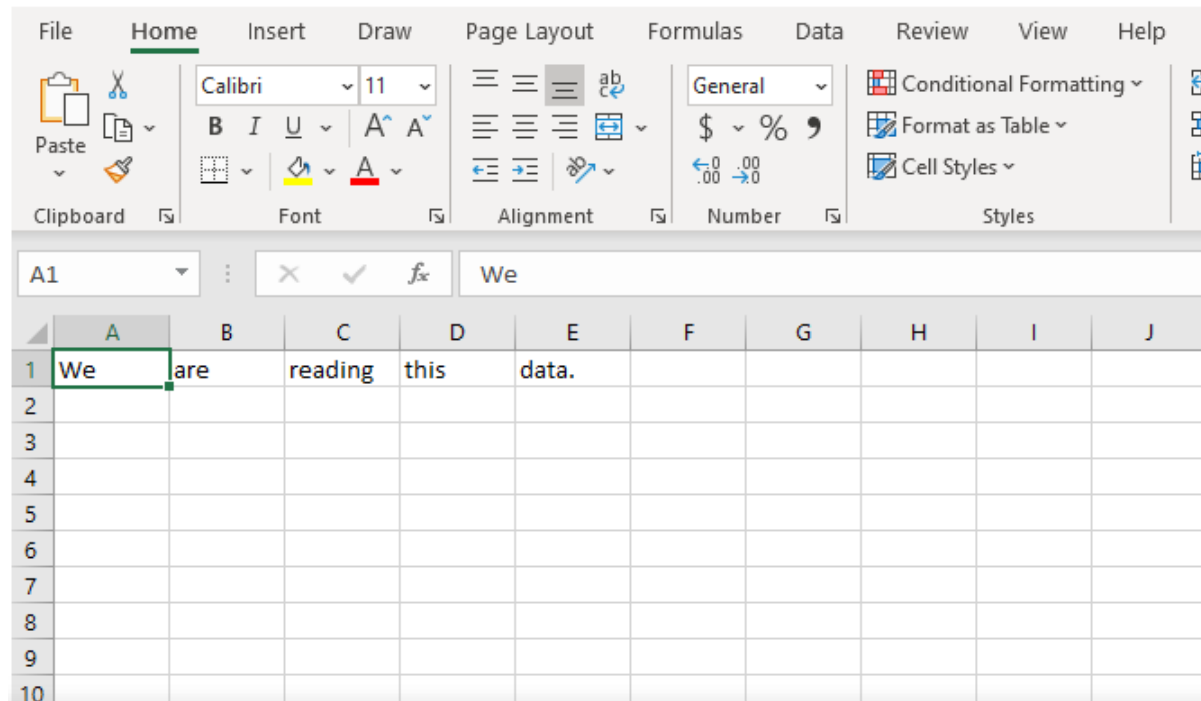
Correction



- Ainsi, l'ID d'automatisation de la cellule A1 est A1. Nous pouvons donc maintenant l'utiliser. Maintenant, nous allons utiliser ces identifiants d'automatisation pour lire et écrire des données dans une feuille Excel.

Automatisation de la lecture des données à partir d'une feuille Excel

- Nous lirons ces données à partir d'un fichier Excel et les imprimerons sous forme de phrase.



Activité 1 : automatisation de tâches redondantes

Correction



- Puisque "We" est écrit dans la cellule A1 et que son identifiant d'automatisation est également A1, il est maintenant tout à fait évident que l'identifiant d'automatisation de la cellule B1 est B1, celui de la cellule C1 est C1, et ainsi de suite.
- Par conséquent, le code pour lire les données est :

```
text = ""
for i in range(65, 65+5):
    e_id = str(chr(i))+str(1)
    dlg.child_window(auto_id = e_id).click_input()
    ei = dlg.child_window(auto_id=e_id)
    ei.click_input()
    text+=(ei.legacy_properties()['Value'])
    text+=" "
print("Text read from sample.xlsx is:",(text))
```

- Sortie :

```
Text read from sample.xlsx is: We are reading this data.
```

Activité 1 : automatisation de tâches redondantes

Correction



- Explication du code :
 - À l'intérieur de la boucle for, e_id est l'ID d'automatisation que nous créons pour chaque cellule. child_window() est utilisé ici pour obtenir un élément particulier où l'ID d'automatisation est passé en paramètre.
 - Ensuite, nous faisons un clic de souris dessus en utilisant click_input() pour saisir cet élément. Maintenant, nous voulons lire la valeur de la cellule dans Excel, donc pour cela, utilisez legacy_properties() ['Valeur'].

Maintenant, avançons et essayons d'écrire dans la ligne suivante de la feuille Excel ci-dessus.

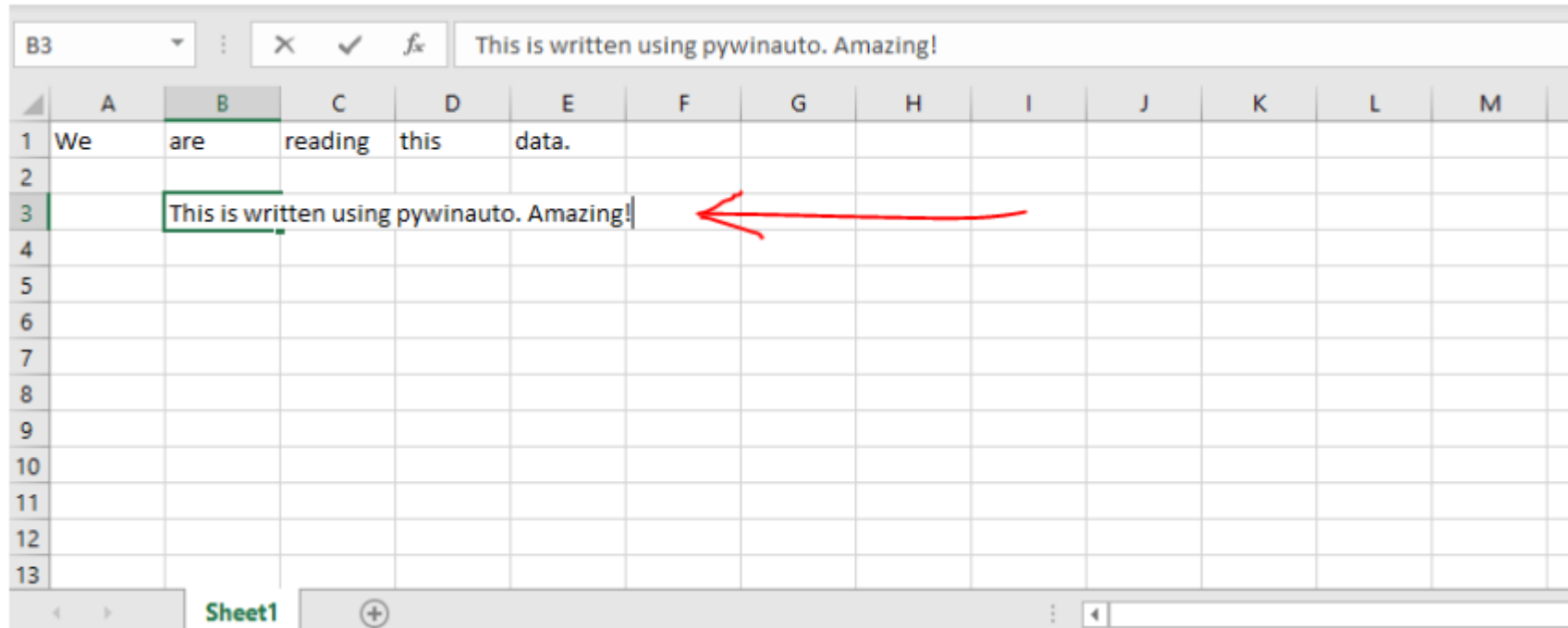
- **Automatisation de l'écriture de données dans une feuille Excel**
 - Nous écrirons des données dans la cellule B3 de la feuille Excel. Ici aussi, vous pouvez utiliser l'outil inspect.exe pour récupérer son ID d'automatisation. Utilisez ensuite ce code pour écrire des données.

```
write_data = "This is written using pywinauto. Amazing!«  
window = dlg.child_window(auto_id="B3")  
window.click_input()  
window.type_keys(write_data, with_spaces=True)
```

Activité 1 : automatisation de tâches redondantes

Correction

- Sortie :



- Explication du code :

- write_data est la chaîne que nous devons écrire. Dans la seconde, nous saisissons à nouveau cet élément et cliquons dessus en utilisant click_input(). Maintenant, pour écrire des données dans cette cellule B3, la méthode type_keys() est utilisée. Il faut 2 paramètres, l'un est la chaîne à écrire et l'autre est de vérifier l'espace, c'est-à-dire with_spaces s'il est conservé vrai, les espaces seront également saisis lors de l'écriture des données de la chaîne.

ACTIVITÉ n° 2

Planification des tâches récurrentes

Compétences visées :

- Planifier des tâches avec les outils système

Recommandations clés :

- Lire attentivement l'énoncé et les questions
- Se référer au cours



02 heures





WEBFORCE
BE THE CHANGE

CONSIGNES

1. Pour le formateur :

- Laisser à l'apprenant l'occasion de comprendre seul l'énoncé.
- S'assurer de la bonne compréhension du contexte avant de lui laisser le temps de réfléchir et répondre.
- Discuter les réponses des apprenants avant de donner la solution.

2. Pour l'apprenant :

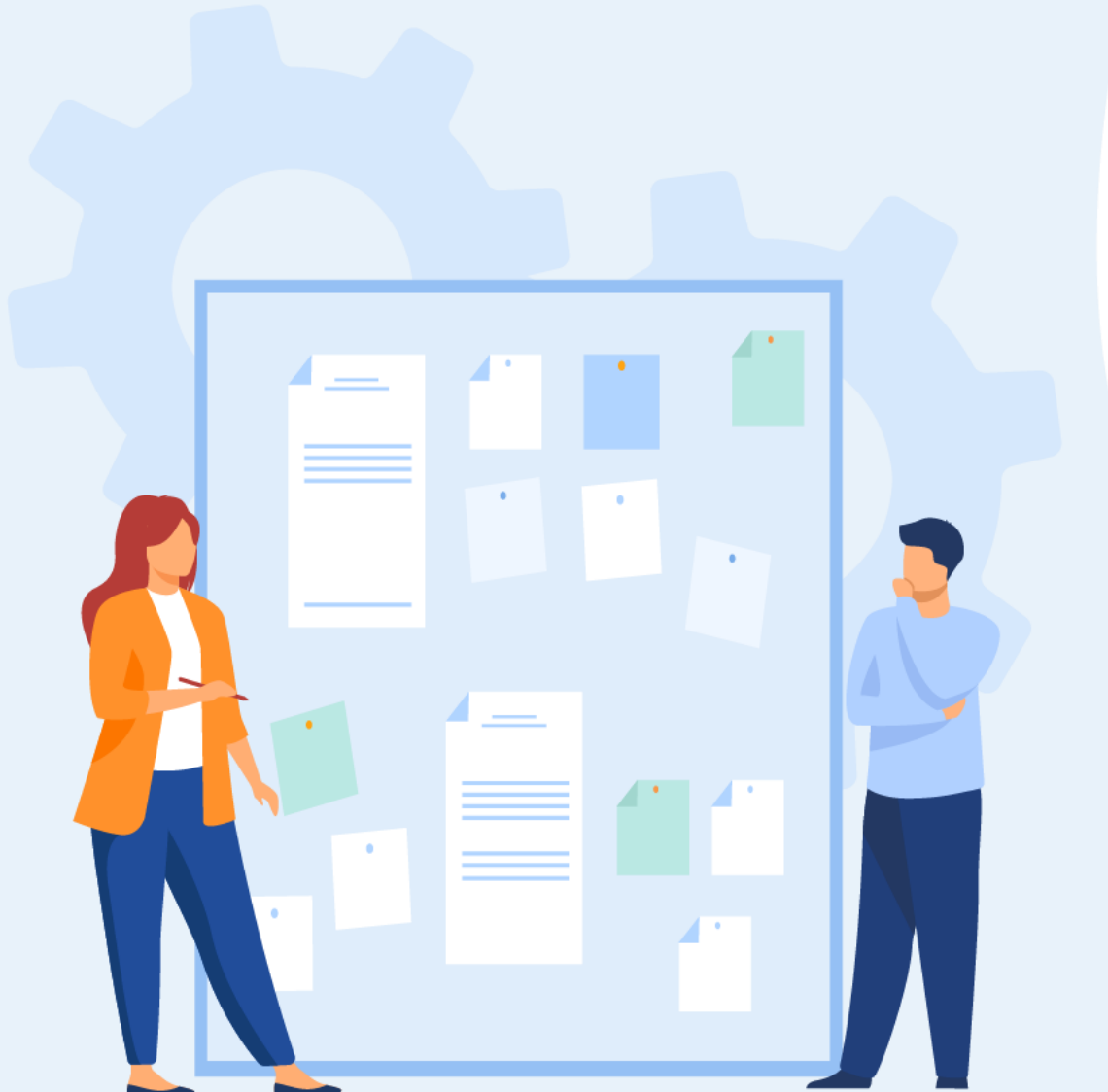
- Lire et bien comprendre la question.
- Essayer de trouver par vous-même une réponse à cette question et la noter.
- Parcourir les réponses proposées.
- Pour chaque réponse : comparez-la à votre réponse et cochez-la si elle lui correspond ou bien est compatible.

3. Conditions de réalisation :

- Individuel ou par groupes (2 ou 3 maximum).
- Support de résumé théorique accompagnant.
- Stylo et feuille de papier.

4. Critères de réussite :

- Le stagiaire est-il capable de :
 - Réaliser des tâches par l'automatisation ?
- Réponses correctes pour au moins 70 % des questions.



Activité 2 : planification des tâches récurrentes

Exercice



WEBFORCE
BE THE CHANGE

Exercice :

Planifier des tâches de courrier électronique tous les jours à l'aide de crontab.



Activité 2 : planification des tâches récurrentes

Correction



Rappel :

- **Crontab** est une commande Linux qui affiche la liste des commandes (tâches) qui doivent être automatiquement planifiées pour des tâches spécifiques afin que les tâches s'exécutent à une heure précise de manière répétée ou aident à créer une tâche planifiée.
- Un planificateur de tâches "cron" est utilisé pour exécuter des tâches (jobs) dans crontab. "cron" tire son nom du mot grec chronos qui signifie temps. "cron" est le processus système qui effectuera automatiquement des tâches pour vous selon un calendrier défini.

Commandes crontab :

- Lister toutes les tâches :

crontab -l : affiche la liste des tâches définies qui doivent être exécutées régulièrement.

- Modifier/créer des tâches :

crontab -e : crée ou modifie les tâches.

Format des tâches crontab :

Le format d'exécution d'une tâche dans un calendrier spécifique est écrit ci-dessous:

[Minute] [heure] [Jour_du_Mois] [Mois_de_l'Année] [Jour_de_la_Semaine] [commande]

Activité 2 : planification des tâches récurrentes

Correction



Exemples (format des tâches crontab) :

- `* /5 * * * * python myemail.py` : cette tâche exécutera votre script toutes les 5 minutes.
- `* 4,5 * * * *` : cette tâche exécutera votre script toutes les minutes après les heures 4 et 5.
- `* */1 * * * *` python myemail.py : cette tâche exécutera votre script chaque minute après chaque heure.
- `5 0 * 8 * *` python myemail.py : cette tâche exécutera votre script à 00 h 05 en août.

Maintenant, essayons d'utiliser crontab pour envoyer un e-mail à un utilisateur tous les jours en envoyant de belles citations. Nous nous enverrons un devis tous les jours à 6 h 00.

Activité 2 : planification des tâches récurrentes

Correction



- Voici un script Python pour envoyer un email.

```
#myemail.py

import smtplib

from string import Template
from email.mime.multipart import
MIMEMultipart
from email.mime.text import MIMEText

import datetime
import wikiquote

def main():
    names = ['YOUR_NAME']
    emails = ['YOUR_EMAIL']
    message_template =
read_template('message.txt')
```

Activité 2 : planification des tâches récurrentes

Correction



```
# configurer le serveur SMTP
s = smtplib.SMTP(host='smtp.gmail.com', port=587)
s.starttls()
s.login('GMAIL_EMAIL', 'GMAIL_PASSWORD')

# Pour chaque contact, envoyez l'e-mail :
for name, email in zip(names, emails):
    msg = MIMEMultipart()
# créer un message
    # ajouter le nom actuel de la personne au modèle de message
    message = message_template.substitute(PERSON_NAME=name.title(),
```

Activité 2 : planification des tâches récurrentes

Correction



```
# Imprimer le corps du message
print(message)

# configurer les paramètres du message
msg['From']=name
msg['To']=email
msg['Subject']="This is TEST"

# ajouter dans le corps du message
msg.attach(MIMEText(message, 'plain'))

# envoyer le message via le serveur configuré précédemment
s.send_message(msg)
del msg
```

Activité 2 : planification des tâches récurrentes

Correction



```
# Terminer la session SMTP et fermer la connexion
s.quit()

if __name__ == '__main__':
    main()
```

- Le fichier message .txt :

```
Dear ${PERSON_NAME},
QUOTE OF THE DAY: ${QUOTE}
Yours Truly,
```

- Tapez la commande : **crontab -e**
- Ajoutez la ligne ci-dessous à la fin du fichier.

```
06 * * * python directory_location/myemail.py
```

- Cela exécutera le script myemail.py tous les jours à 06 h 00.

Activité 2 : planification des tâches récurrentes

Correction



- Lorsque vous écrivez votre code, l'interpréteur compile et convertit le code dans un format compréhensible par votre machine. Le code ne peut pas être interprété et analysé s'il y a des erreurs de syntaxe non valides.
- Les erreurs de syntaxe sont détectées pendant la compilation du programme, une fois qu'une erreur est trouvée, cela empêchera l'exécution du code. Habituellement, les erreurs sont explicites. Alors que certaines erreurs ne sont pas aussi corporatives.
- La bonne chose à propos des erreurs de syntaxe est que le compilateur indique où le problème pourrait être.
- Examinons quelques-unes des causes les plus courantes d'erreurs de syntaxe :
 - Mots-clés réservés mal orthographiés
 - Espaces requis manquants
 - Guillemets manquants
 - Utilisation abusive des instructions de bloc (if-else, boucles)
 - Opérateur d'affectation manquant (=)
 - Déclaration de variables invalides
 - Appel ou définition de fonction non valide

ACTIVITÉ n° 3

Correction de scripts

Compétences visées :

- Identifier les erreurs dans les scripts Python

Recommandations clés :

- Lire attentivement l'énoncé et les questions
- Se référer au cours



02 heures



CONSIGNES

1. Pour le formateur :

- Laisser à l'apprenant l'occasion de comprendre seul l'énoncé.
- S'assurer de la bonne compréhension du contexte avant de lui laisser le temps de réfléchir et répondre.
- Discuter les réponses des apprenants avant de donner la solution.

2. Pour l'apprenant :

- Lire et bien comprendre la question.
- Essayer de trouver par vous-même une réponse à cette question et la noter.
- Parcourir les réponses proposées.
- Pour chaque réponse : comparez-la à votre réponse et cochez-la si elle lui correspond ou bien est compatible.

3. Conditions de réalisation :

- Individuel ou par groupes (2 ou 3 maximum).
- Support de résumé théorique accompagnant.
- Stylo et feuille de papier.

4. Critères de réussite :

- Le stagiaire est-il capable de :
 - Corriger les erreurs dans des scripts ?
- Réponses correctes pour au moins 70 % des questions.



Activité 3 : correction de scripts

Exercice



Exercice : identifier l'erreur générée dans chaque cas.

1.

```
prin("hi")
```

2.

```
name="code leaks"  
if name == "code leaks":  
print("hello")  
else:  
    print("who?")
```


Activité 3 : correction de scripts

Exercice



3.

```
print(hi)
```

4.

```
name="code leaks"  
if name == "code leaks"  
    print("hello")  
else:  
    print("who?")
```

5.

```
game == "me"
```

Activité 3 : correction de scripts

Exercice



6.

```
123game="me"
```

```
@game="me"
```

```
"me" = game
```

7.

```
def func():  
    print("hi")  
  
func(1)
```

1. Mots-clés réservés mal orthographiés

```
: prin("hi")  
-----  
NameError                                Traceback (most recent call last)  
<ipython-input-2-091467a1e20e> in <module>()  
----> 1 prin("hi")  
  
NameError: name 'prin' is not defined
```

- Le compilateur a lancé un message d'erreur comme "prin not defined". Il n'est pas défini comme un mot-clé défini par l'utilisateur ou intégré, il confond donc le compilateur sur l'endroit où se trouve ce mot.

2. Espaces requis manquants

- Contrairement à d'autres langages de programmation, Python requiert un bloc indenté. C'est pourquoi de nombreux programmeurs ont du mal à s'approprier ce concept dans les premières étapes.

```
name="code leaks"  
if name == "code leaks":  
print("hello")  
else:  
    print("who?")
```

File "<ipython-input-4-0c3ad72d11a0>", line 3

```
print("hello")
```

^

IndentationError: expected an indented block

3. Guillemets manquants

```
print(hi)
```

```
-----  
NameError                                 Traceback (most recent call last)  
<ipython-input-6-6122400d05e8> in <module>()  
----> 1 print(hi)  
  
NameError: name 'hi' is not defined
```

- Lorsqu'il manque des guillemets dans une chaîne, le compilateur confond le but de la chaîne et ne l'identifie pas.
- Notez comment l'erreur est "name hi not defined", même s'il est censé être une chaîne, pas une variable. Le compilateur l'a confondu avec une variable non définie et n'a deviné nulle part la possibilité d'une chaîne
- Remarquez comment l'erreur est devenue l'erreur littérale lorsqu'un guillemet est ajouté. Le compilateur l'a reconnu comme une chaîne et l'erreur END OF LINE est renvoyée.

```
print('hi)
```

```
File "<ipython-input-9-8878a92e9096>", line 1  
    print('hi)
```

```
      ^  
SyntaxError: EOL while scanning string literal
```

4. Utilisation abusive des instructions de blocs (IF-ELSE, LOOPS)

```
name="code leaks"  
if name == "code leaks"  
    print("hello")  
else:  
    print("who?")
```

```
File "<ipython-input-10-325017325d9b>", line 2  
    if name == "code leaks"
```

```
SyntaxError: invalid syntax
```

- L'erreur est "invalid syntax". Pas très descriptif, c'est pourquoi un if-else deux-points (:) est nécessaire pour corriger ce bug. C'est la même chose avec les blocs while.

5. Opérateur d'affectation manquant (=)

```
game == "me"
```

```
-----  
NameError                                Traceback (most recent call last)  
<ipython-input-12-b8eee24ed53f> in <module>()  
----> 1 game == "me"  
  
NameError: name 'game' is not defined
```

- Notez que le compilateur ne génère pas d'erreur pour indiquer que l'opérateur d'affectation est mal utilisé. Mais il essaie de comparer la variable game à la chaîne "me" en trouvant une variable indéfinie.

6. Déclaration invalide des variables

- Il existe de nombreuses façons de violer la convention de dénomination des variables.
- Vous ne pouvez pas utiliser de caractères spéciaux pour attendre un trait de soulignement (), ou utiliser un nombre au début de la variable.

```
123game="me"
```

```
File "<ipython-input-17-958acc91b980>", line 1
```

```
123game="me"
```

```
^
```

```
SyntaxError: invalid syntax
```

```
@game="me"
```

```
File "<ipython-input-18-d11eedd16ed1>", line 1
```

```
@game="me"
```

```
^
```

```
SyntaxError: invalid syntax
```


Activité 3 : correction de scripts

Correction



- Voici une autre façon invalide de déclarer une variable.

```
"me"= game
```

```
File "<ipython-input-13-ecdc4778675f>", line 1
```

```
"me"= game
```

```
^
```

```
SyntaxError: can't assign to literal
```

7. Appel ou définition de fonction invalide

- Comme toute autre instruction de bloc, la déclaration de fonction a également une syntaxe.
- Des espaces appropriés et l'utilisation du deux-points (:) sont nécessaires.
- Une erreur dans la syntaxe empêchera l'exécution. L'exemple suivant montre une fonction exécutable sans erreur.

```
def func():  
    print("hi")
```

```
func()
```

```
hi
```

Activité 3 : correction de scripts

Correction



- L'appel de fonction doit avoir les précautions requises pour éviter les bogues. L'erreur suivante est causée par l'argument fourni qui n'est pas défini dans la déclaration.

```
def func():  
    print("hi")  
  
func(1)
```

```
-----  
TypeError                                 Traceback (most recent call last)  
<ipython-input-22-d8d855e2d99a> in <module>()  
      2     print("hi")  
      3  
----> 4 func(1)  
  
TypeError: func() takes 0 positional arguments but 1 was given
```

ACTIVITÉ n° 4

Test temps d'exécution des scripts

Compétences visées :

- Réduire le temps de réalisation de tâches par l'automatisation

Recommandations clés :

- Lire attentivement l'énoncé et les questions
- Se référer au cours



01 heure



CONSIGNES

1. Pour le formateur :

- Laisser à l'apprenant l'occasion de comprendre seul l'énoncé.
- S'assurer de la bonne compréhension du contexte avant de lui laisser le temps de réfléchir et répondre.
- Discuter les réponses des apprenants avant de donner la solution.

2. Pour l'apprenant :

- Lire et bien comprendre la question.
- Essayer de trouver par vous-même une réponse à cette question et la noter.
- Parcourir les réponses proposées.
- Pour chaque réponse : comparez-la à votre réponse et cochez-la si elle lui correspond ou bien est compatible.

3. Conditions de réalisation :

- Individuel ou par groupes (2 ou 3 maximum).
- Support de résumé théorique accompagnant.
- Stylo et feuille de papier.

4. Critères de réussite :

- Le stagiaire est-il capable de :
 - Tester le temps d'exécution des scripts ?
- Réponses correctes pour au moins 70 % des questions.



Activité 4 : test temps d'exécution des scripts



- Il existe de nombreux modules Python comme `time`, `timeit` et `datetime` en Python qui peuvent stocker l'heure à laquelle une section particulière du programme est en cours d'exécution. En manipulant ou en obtenant la différence entre les heures de début et de fin auxquelles une section particulière est exécutée, nous pouvons calculer le temps qu'il a fallu pour exécuter la section.
- Les méthodes suivantes peuvent être utilisées pour calculer la différence de temps :
 - ❑ Le module **Time** de Python fournit diverses fonctions liées au temps. Ce module fait partie des modules utilitaires standard de Python. La méthode **`time.time()`** du module `Time` est utilisée pour obtenir le temps en secondes depuis l'époque. La gestion des secondes intercalaires dépend de la plateforme.
 - ❑ La bibliothèque Python **`datetime`** définit une fonction qui peut être principalement utilisée pour obtenir l'heure et la date actuelles. La fonction `now()` renvoie la date et l'heure locales actuelles qui sont définies dans le module `datetime`.
 - ❑ **`timeit`** exécute votre extrait de code des millions de fois (la valeur par défaut est 1000000) afin que vous obteniez la mesure statistiquement la plus pertinente du temps d'exécution du code.

Activité 4 : test temps d'exécution des scripts



Exemple 1 : calcul de l'heure à l'aide du module de temps et de la fonction time.time()

```
# program to compute the time
# of execution of any python code
import time

# we initialize the variable start
# to store the starting time of
# execution of program
start = time.time()

# we can take any program but for
# example we have taken the below
# program
a = 0
for i in range(1000):
    a += (i**100)

# now we have initialized the variable
# end to store the ending time after
# execution of program
end = time.time()

# difference of start and end variables
# gives the time of execution of the
# program in between
print(" Le temps d'exécution du programme est :", end-start)
```

Sortie :

Le temps d'exécution du programme est : 0.001995563507080078

Activité 4 : test temps d'exécution des scripts

Exercices



Exercice :

1. Vérifier les temps d'exécution du programme de l'exemple 1 pour différents nombres de calculs : 100, 200, 300, 400, 500, 600, 700, 800, 900 et 1000 ordre de calcul.
2. Modifier l'exemple 1 en utilisant la fonction `datetime.now()`
3. Modifier l'exemple 1 en utilisant la méthode `timeit()`

Activité 4 : test temps d'exécution des scripts

Corrections



1.

```
# program to compute the time
# of execution of any python code
# for different number of computations
import time

# we initialize a for loop and in each
# iterations store the time of start
# and end of the iterations
for j in range(100, 1100, 100):
    start = time.time()

    # program to iterate the range of
    # below loop increasing the value
    # in each iterations
    a = 0
    for i in range(j):
        a += (i**100)

    # the end variable to store the
    # ending time after execution of
    # program after each iterations
    end = time.time()

    # difference of start and end variables

# gives the time of execution of the program
# in between in each iterations
print("Time for execution of program for {} order of computations: {}".format(
    j, round(end-start, 10)))
```

Sortie :

```
Time for execution of program for 100 order of computaions: 0.0
Time for execution of program for 200 order of computaions: 0.0
Time for execution of program for 300 order of computaions: 0.0
Time for execution of program for 400 order of computaions: 0.0
Time for execution of program for 500 order of computaions: 0.000995636
Time for execution of program for 600 order of computaions: 0.0010249615
Time for execution of program for 700 order of computaions: 0.001994133
Time for execution of program for 800 order of computaions: 0.0019664764
Time for execution of program for 900 order of computaions: 0.0029964447
Time for execution of program for 1000 order of computaions: 0.0019903183
```

Activité 4 : test temps d'exécution des scripts

Corrections



2.

```
# program to compute the time of
# execution of any python code
from datetime import datetime

# we initialize the variable start to
# store the starting time of execution
# of program
start = datetime.now()

# we can take any program but for
# example we have taken the below
# program
a = 0
for i in range(1000):
    a += (i**100)

# now we have initialized the variable
# end to store the ending time after
# execution of program
end = datetime.now()

# difference of start and end variables
# gives the time of execution of the
# program in between
print(" Le temps d'exécution du programme est :",
      str(end-start)[5:])
```

Sortie :

Le temps d'exécution du programme est : 0.001996

Activité 4 : test temps d'exécution des scripts

Corrections



3.

```
# program to compute the time of
# execution of any python code using timit

# importing the required module
import timeit

# code snippet to be executed only once
mysetup = "from math import sqrt"

# code snippet whose execution time
# is to be measured
mycode = '''
def example():
    mylist = []
    for x in range(100):
        mylist.append(sqrt(x))
...

# timeit statement
print (" Le temps d'exécution du programme est :",
      timeit.timeit(setup = mysetup,
                    stmt = mycode,
                    number = 10000))
```

Sortie :

Le temps d'exécution du programme est : 0.0023286999994525104

Activité 4 : test temps d'exécution des scripts

Corrections



3.

- `timeit()` nous donnerait le temps d'exécution de n'importe quel programme. Ce module fournit un moyen simple de trouver le temps d'exécution de petits morceaux de code Python.
- Il fournit la méthode `timeit()` pour faire de même.
- La fonction de module `timeit.timeit(stmt, setup, timer, number)` accepte quatre arguments :
 - `stmt` qui est l'instruction que vous voulez mesurer ; par défaut, il est "pass".
 - `setup` qui est le code que vous exécutez avant d'exécuter le `stmt` ; par défaut, il est "pass". Nous l'utilisons généralement pour importer les modules requis pour notre code.
 - `timer` qui est un objet `timeit.Timer` ; il a généralement une valeur par défaut raisonnable.
 - `number`: est le nombre d'exécutions que vous souhaitez exécuter le `stmt`.



PARTIE 5

Créer des fichiers logs

Dans ce module, vous allez :

- Comprendre la persistance des données
- Manipuler les fichiers logs
- Tester le fonctionnement des scripts



06 heures

ACTIVITÉ n° 1

Manipulation de fichiers

Compétences visées :

- Création, lecture et suppression dans un fichier

Recommandations clés :

- Lire attentivement l'énoncé et les questions
- Se référer au cours



02 heures



CONSIGNES

1. Pour le formateur :

- Laisser à l'apprenant l'occasion de comprendre seul l'énoncé.
- S'assurer de la bonne compréhension du contexte avant de lui laisser le temps de réfléchir et répondre.
- Discuter les réponses des apprenants avant de donner la solution.

2. Pour l'apprenant :

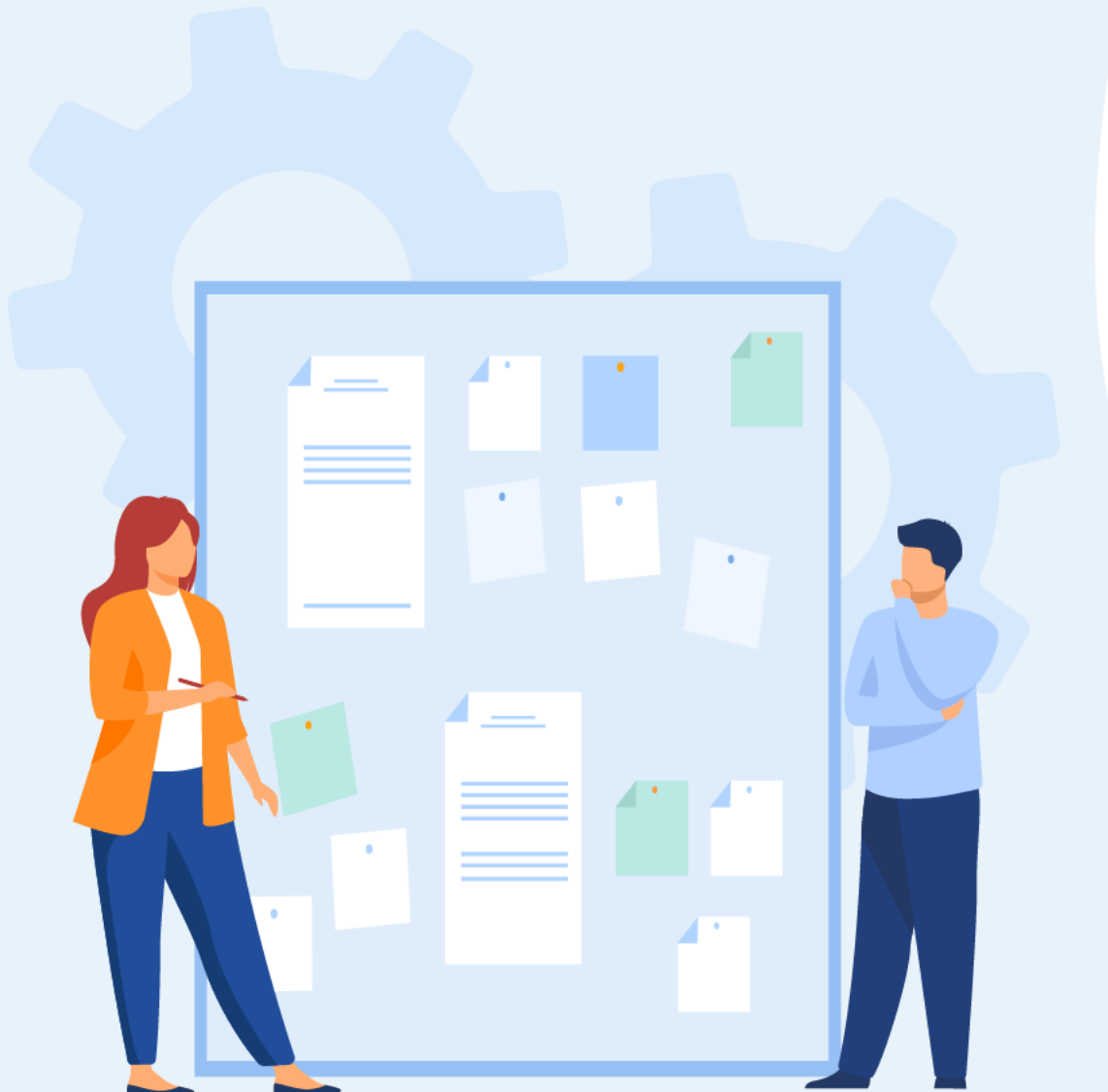
- Lire et bien comprendre la question.
- Essayer de trouver par vous-même une réponse à cette question et la noter.
- Parcourir les réponses proposées.
- Pour chaque réponse : comparez-la à votre réponse et cochez-la si elle lui correspond ou bien est compatible.

3. Conditions de réalisation :

- Individuel ou par groupes (2 ou 3 maximum).
- Support de résumé théorique accompagnant.
- Stylo et feuille de papier.

4. Critères de réussite :

- Le stagiaire est-il capable de :
 - Créer, lire et supprimer dans un fichier ?
- Réponses correctes pour au moins 70 % des questions.



Activité 1 : manipulation de fichiers

Exercices



Exercice 1 : calcul de moyenne

1. Écrire une fonction PYTHON **remplir_fichier(nom_fichier,n)** qui permet de créer et de remplir un fichier texte contenant les noms et les notes d'un groupe d'étudiants. Chaque ligne du fichier contient deux colonnes séparées par " :". La première colonne représente le nom de l'étudiant et la deuxième colonne représente la note.
2. Écrire une fonction PYTHON **calculer_moyenne(nom_fichier)** qui permet de lire un fichier texte contenant des valeurs sur deux colonnes séparées par le caractère " :" sous la forme **nom :note**. Cette fonction doit retourner la moyenne des notes des étudiants du fichier.

Exercice 2 : traçage de courbes - Matplotlib

1. Écrire une fonction PYTHON **remplir_fichier(nom_fichier)** qui permet de créer et de remplir un fichier texte contenant trois colonnes séparées par un espace. La première colonne représente des valeurs de x telles que $x \in [-3\pi, 3\pi]$ et les deux dernières colonnes représentent respectivement $\sin(x)$ et $\cos(x)$.
2. Écrire une fonction PYTHON **lire_fichier(nom_fichier)** qui permet de lire un fichier texte contenant des valeurs sur trois colonnes séparées par un espace. Cette fonction doit retourner trois tableaux X, Y et Z contenant respectivement les valeurs des trois colonnes du fichier.
3. Tracer les courbes des fonctions sinus et cosinus à partir des valeurs du fichier en utilisant la bibliothèque matplotlib.

Correction exercice 1 : calcul de moyenne

- On va écrire une fonction PYTHON `remplir_fichier(nom_fichier,n)` qui permet de créer et de remplir un fichier texte contenant les noms et les notes d'un groupe d'étudiants. Chaque ligne du fichier contient deux colonnes séparées par " : ". La première colonne représente le nom de l'étudiant et la deuxième colonne représente la note. Voici deux versions de cette fonction.

In [1] :

```
#version1 - on écrit ligne par ligne avec write
def remplir_fichier1(nom_fichier,n):
    f=open(nom_fichier,'w')
    for i in range(n):
        nom=input("Donner le nom ")
        note=float(input("Donner la note "))
        f.write(nom+'!'+str(note)+'\n')
    f.close()
```

In [2] :

```
remplir_fichier1("Notes.txt",3)
```

Out [2] :

```
Donner le nom Ali
Donner la note 12.5
Donner le nom Samia
Donner la note 11.75
Donner le nom Walid
Donner la note 9.25
```



Remarque

- Un fichier intitulé "Notes.txt" sera créé dans notre dossier de travail. Essayons de lire ce fichier avec `readlines` pour voir son contenu.

Activité 1 : manipulation de fichiers

Corrections



Correction exercice 1 : calcul de moyenne (suite)

In [3] :

```
f=open("Notes.txt",'r')
print(f.readlines()) f
.close()
```

Out [3] :

```
['Ali:12.5\n', 'Samia:11.75\n', 'Walid:9.25\n']
```

In [7] :

```
#version2 - On écrit toutes les lignes en utilisant une liste L avec writelines
def remplir_fichier2(nom_fichier,n): f=open(nom_fichier,'w') L=[] for i in range(n):
nom=input("donner le nom") note=float(input("donner la note"))
L.append(':'+str(note)+'\n') f.writelines(L) f.close()
```

Correction exercice 1 : calcul de moyenne (suite)

2. Maintenant, on va écrire une fonction PYTHON `calculer_moyenne(nom_fichier)` qui permet de lire le fichier texte (qu'on vient de créer), contenant des valeurs sur deux colonnes séparées par le caractère ":" sous la forme `nom :note`. Cette fonction doit retourner la moyenne des notes des étudiants du fichier.

```
def calculer_moyenne1(nom_fichier):
```

```
    f=open(nom_fichier,'r')
```

```
    L=f.readlines()
```

```
    f.close()
```

```
    s=0
```

```
    for ch in L:
```

```
        E=ch.split(":")
```

```
        s=s+float(E[1])
```

```
    return(s/len(L))
```

```
moy=calculer_moyenne1("Notes.txt")
```

```
print('La moyenne est',moy)
```

Correction exercice 2 : Traçage de courbes – Matplotlib

1. On va écrire une fonction PYTHON remplir_fichier(nom_fichier) qui permet de créer et de remplir un fichier texte contenant trois colonnes séparées par un espace. La première colonne représente des valeurs de x tel que $x \in [-3\pi, 3\pi]$ et les deux dernières colonnes représentent respectivement $\sin(x)$ et $\cos(x)$. On commence par charger les modules nécessaires à cet exercice, à savoir numpy et matplotlib. Ensuite, on donnera deux versions de la fonction remplir_fichier.

```
import numpy as np
import matplotlib.pyplot as plt
def remplir_fichier(nom_fichier):
    fich=open(nom_fichier,'w')
    x=np.linspace(-3*np.pi,3*np.pi,100)
    y=np.sin(x)
    z=np.cos(x)
    #écriture ligne par ligne avec write
    for i in range(len(x)):
        fich.write(str(x[i])+ ' '+str(y[i])+ ' '+str(z[i])+'\n')
```

```
fich.close()
```

```
def remplir_fichier_version2(nom_fichier):
    fich=open(nom_fichier,'w')
    x=np.linspace(-3*np.pi,3*np.pi,10)
    y=np.sin(x)
    z=np.cos(x)
    L=[]
    for i in range(len(x)):
        L.append(str(x[i])+ ' '+str(y[i])+ ' '+str(z[i])+'\n')
    #écriture de toutes lignes avec une liste et writelines
    fich.writelines(L)
    fich.close()
remplir_fichier("res.txt")
```

Correction exercice 2 : Traçage de courbes – Matplotlib (suite)



Remarque

- Un fichier intitulé "Notes.txt" sera créé dans notre dossier de travail. Essayons de lire ce fichier avec readlines pour voir son contenu.

```
f=open("res.txt",'r') print(f.readlines()) f.close()
```

2. Maintenant, on va écrire une fonction lire_fichier(nom_fichier) qui permet de lire un fichier texte contenant des valeurs sur trois colonnes séparées un espace. Cette fonction doit retourner trois tableaux X et Y et Z contenant respectivement les valeurs des trois colonnes du fichier. On donnera deux versions de cette fonctions.

```
def lire_fichier_version1(nom_fichier):
```

```
    # Lecture de toutes les lignes
```

```
    fich=open(nom_fichier,'r')
```

```
    l1=list()
```

```
    l2=list()
```

```
    l3=list()
```

```
    L=fich.readlines()
```

```
for i in L:
```

```
    LL=i.split()
```

```
    l1.append(float(LL[0]))
```

```
    l2.append(float(LL[1]))
```

```
    l3.append(float(LL[2]))
```

```
x=np.array(l1)
```

```
y=np.array(l2)
```

```
z=np.array(l3)
```

```
fich.close()
```

```
return(x,y,z)
```

```
def lire_fichier_version2(nom_fichier):
```

```
    # Lecture ligne par ligne
```

```
    fich=open(nom_fichier,'r')
```

```
    l1=list()
```

```
    l2=list()
```

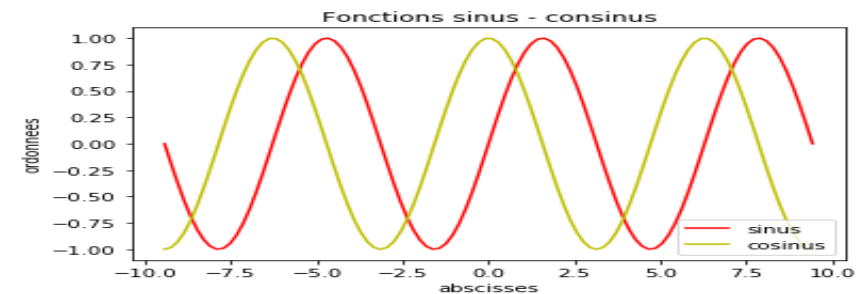
```
    l3=list()
```

Correction exercice 2 : Traçage de courbes – Matplotlib (suite)

```
while True:
    L=fich.readline()
    if L!="":
        LL=L.split()
        l1.append(float(LL[0]))
        l2.append(float(LL[1]))
        l3.append(float(LL[2]))
    else:
        break;
x=np.array(l1)
y=np.array(l2)
z=np.array(l3)
fich.close()
return(x,y,z)
```

3. Maintenant, on essaie de lire le contenu de notre fichier créé précédemment avec la fonction `lire_fichier`. Ensuite, nous allons tracer les courbes des fonctions sinus et cosinus à partir des valeurs du fichier, en utilisant la bibliothèque `matplotlib`.

```
T=lire_fichier_version2("res.txt")
pl.plot(T[0],T[1],'r')
pl.plot(T[0],T[2],'y')
pl.xlabel('abscisses')
pl.ylabel('ordonnees')
pl.title('Fonctions sinus - cosinus')
pl.legend(['sinus','cosinus'])
pl.show()
```



ACTIVITÉ n° 2

La création de fichiers logs

Compétences visées :

- Écrire des scripts pour la création de logs

Recommandations clés :

- Lire attentivement l'énoncé et les questions
- Se référer au cours



02 heures



CONSIGNES

1. Pour le formateur :

- Laisser à l'apprenant l'occasion de comprendre seul l'énoncé.
- S'assurer de la bonne compréhension du contexte avant de lui laisser le temps de réfléchir et répondre.
- Discuter les réponses des apprenants avant de donner la solution.

2. Pour l'apprenant :

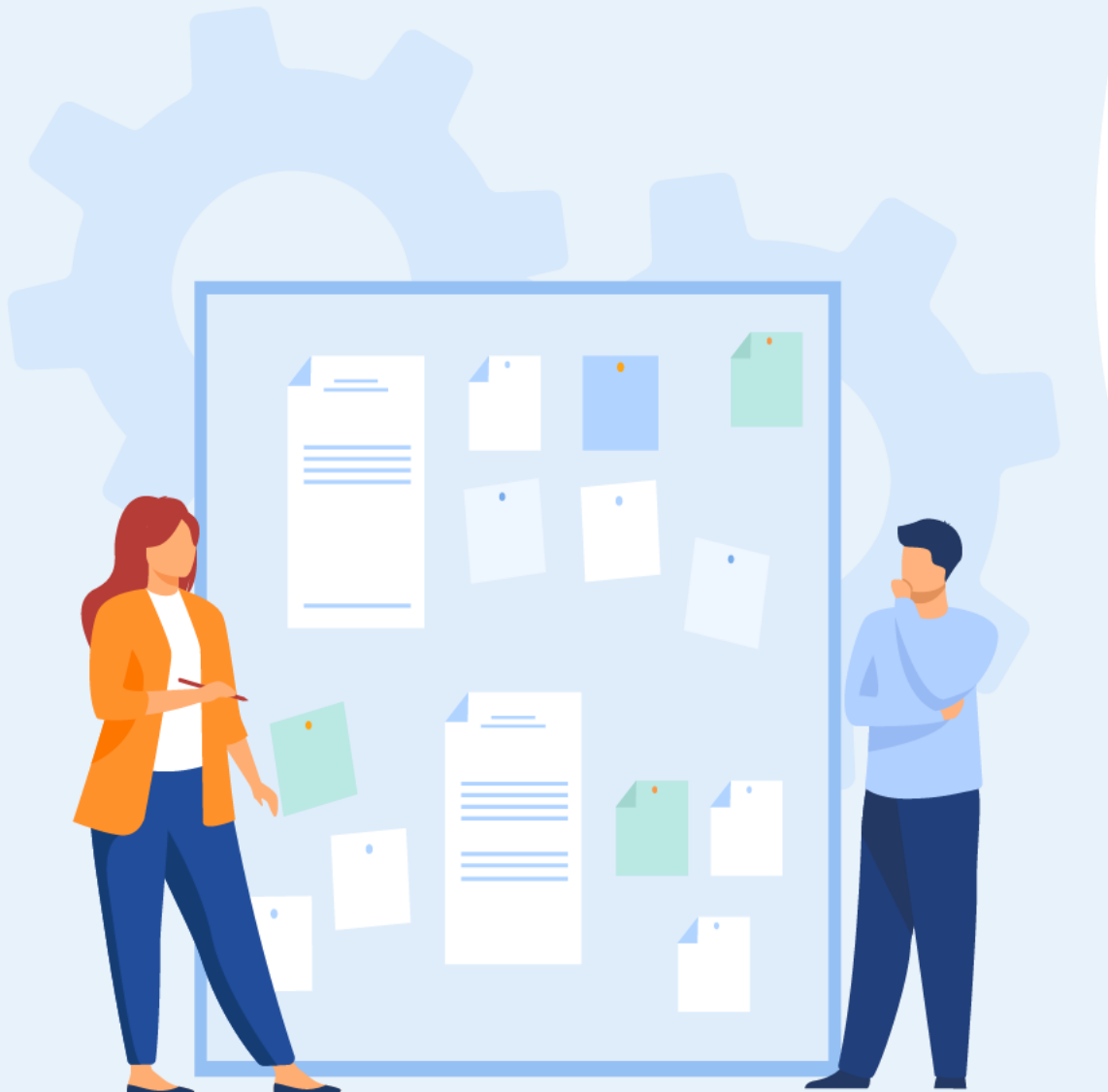
- Lire et bien comprendre la question.
- Essayer de trouver par vous-même une réponse à cette question et la noter.
- Parcourir les réponses proposées.
- Pour chaque réponse : comparez-la à votre réponse et cochez-la si elle lui correspond ou bien est compatible.

3. Conditions de réalisation :

- Individuel ou par groupes (2 ou 3 maximum).
- Support de résumé théorique accompagnant.
- Stylo et feuille de papier.

4. Critères de réussite :

- Le stagiaire est-il capable de :
 - Manipuler les fichiers logs ?
- Réponses correctes pour au moins 70 % des questions.



Activité 2 : la création de fichiers logs

Exercices



Exercice :

1. Configurer un logging vers la sortie standard pour Systemd.
2. Configurer un logging vers Syslog.
3. Configurer et envoyer les messages logs à un fichier (fichier log).

1. Configurer un logging vers la sortie standard pour Systemd

- C'est l'option la plus simple et probablement la meilleure pour configurer la journalisation de nos jours.
- Lors de l'utilisation de systemd pour exécuter un démon, les applications peuvent simplement envoyer des messages logs à stdout ou stderr et demander à systemd de transférer les messages vers journald et syslog.
- Comme avantage supplémentaire, cela ne nécessite même pas la capture d'exceptions, car Python les écrit déjà dans l'erreur standard. Cela dit, suivez la convention appropriée et gérez vos exceptions.

```
import logging
import os
logging.basicConfig(level=os.environ.get("LOGLEVEL", "INFO"))
exit(main())
```

- L'application enregistrera désormais tous les messages de niveau INFO ou supérieur dans stderr, en utilisant un format simple :
- ERROR:the.module.name:The log message
- L'application peut même être configurée pour inclure des messages DEBUG, ou peut-être uniquement ERROR, en définissant la variable d'environnement LOGLEVEL.

2. Configurer un logging vers Syslog

- L'alternative est de l'envoyer directement à syslog.
- C'est idéal pour les systèmes d'exploitation plus anciens qui n'ont pas systemd.
- Python nécessite une configuration un peu plus élaborée pour pouvoir envoyer des messages logs Unicode.

Voici un exemple d'implémentation :

```
import logging
import logging.handlers
import os
class SyslogBOMFormatter(logging.Formatter):
def format(self, record):
result = super().format(record)
return "ufeff" + result
handler = logging.handlers.SysLogHandler('/dev/log')
formatter = SyslogBOMFormatter(logging.BASIC_FORMAT)
handler.setFormatter(formatter)
root = logging.getLogger()
root.setLevel(os.environ.get("LOGLEVEL", "INFO"))
root.addHandler(handler)
try:
    exit(main())
except Exception:
    logging.exception("Exception in main()") exit(1)
```

3. Configurer et envoyer les messages logs à un fichier (fichier log)

- La dernière option consiste à enregistrer les messages directement dans un fichier. Ceci est rarement utile de nos jours, car les administrateurs peuvent configurer syslog pour écrire certains messages dans des fichiers spécifiques, ou en cas de déploiement à l'intérieur de conteneurs, il s'agit d'un anti-pattern. De plus, si vous utilisez le logging centralisé, le fait de devoir gérer des fichiers logs supplémentaires est une préoccupation supplémentaire. Mais c'est une option qui est toujours disponible.
- Lors de la journalisation dans des fichiers, la principale chose dont il faut se méfier est que les fichiers logs doivent être alternés régulièrement. L'application doit détecter le fichier log renommé et gérer cette situation.
- Bien que Python fournisse son propre gestionnaire de rotation de fichiers, il est préférable de laisser la rotation des logs à des outils dédiés tels que logrotate. Le WatchedFileHandler gardera une trace du fichier log et le rouvrira s'il est tourné, ce qui le fait fonctionner correctement avec logrotate sans nécessiter de signaux spécifiques.

Voici un exemple d'implémentation :

```
import logging
import logging.handlers
import os
handler = logging.handlers.WatchedFileHandler( os.environ.get("LOGFILE",
"/var/log/yourapp.log"))
formatter = logging.Formatter(logging.BASIC_FORMAT)
handler.setFormatter(formatter)
root = logging.getLogger()
root.setLevel(os.environ.get("LOGLEVEL", "INFO"))
root.addHandler(handler)
try:
    exit(main())
except Exception:
    logging.exception("Exception in main()")
    exit(1)
```

ACTIVITÉ n° 3

Évaluation de fonctionnement des scripts

Compétences visées :

- Évaluation des cas de test

Recommandations clés :

- Lire attentivement l'énoncé et les questions
- Se référer au cours



02 heures



CONSIGNES

1. Pour le formateur :

- Laisser à l'apprenant l'occasion de comprendre seul l'énoncé.
- S'assurer de la bonne compréhension du contexte avant de lui laisser le temps de réfléchir et répondre.
- Discuter les réponses des apprenants avant de donner la solution.

2. Pour l'apprenant :

- Lire et bien comprendre la question.
- Essayer de trouver par vous-même une réponse à cette question et la noter.
- Parcourir les réponses proposées.
- Pour chaque réponse : comparez-la à votre réponse et cochez-la si elle lui correspond ou bien est compatible.

3. Conditions de réalisation :

- Individuel ou par groupes (2 ou 3 maximum).
- Support de résumé théorique accompagnant.
- Stylo et feuille de papier.

4. Critères de réussite :

- Le stagiaire est-il capable de :
 - Tester un code Python ?
- Réponses correctes pour au moins 70 % des questions.



Activité 3 : évaluation de fonctionnement des scripts



Les étapes pour tester le code Python à l'aide du module unittest sont :

1. Écrire le code.
2. Importer le unittest module .
3. Créer un fichier commençant par le mot-clé test. Par exemple, testprime.py. Le mot-clé test est utilisé pour identifier les fichiers de test.
4. Créer une classe étendant la classe unittest.TestCase
5. Écrire des méthodes (tests) à l'intérieur de la classe. Chaque méthode contient différents cas de test en fonction de vos besoins. Il faut nommer la méthode commençant par test mot-clé.
6. Exécuter les tests.

Nous pouvons exécuter les tests de différentes manières.

- Exécuter la commande Python `-m unittest test_filename.py`
- Nous exécutons les fichiers de test comme des fichiers Python généraux avec la commande Python `test_filename.py`. Pour que cette méthode fonctionne, nous devons invoquer le main méthode de unittest dans le fichier d'essai.
- Et enfin, en utilisant le discover, nous pouvons exécuter automatiquement les tests en utilisant la commande Python `-m unittest discover` sans mentionner le nom de fichier du test. Il trouvera les tests en utilisant la convention de nommage que nous avons suivie. Nous devons donc nommer nos fichiers de test avec le mot-clé test au démarrage.

Activité 3 : évaluation de fonctionnement des scripts



1. Ouvrez un dossier dans votre favori éditeur de code et créez un fichier appelé `utils.py`. Collez le code suivant dans le fichier.

```
import math
def is_prime(n):
    if n < 0:
        return 'Negative numbers are not allowed'

    if n <= 1:
        return False

    if n == 2:
        return True

    if n % 2 == 0:
        return False
```

```
for i in range(2, int(math.sqrt(n)) + 1):
    if n % i == 0:
        return False
    return True

def cubic(a):
    return a * a * a

def say_hello(name):
    return "Hello, " + name
```

Activité 3 : évaluation de fonctionnement des scripts



- Nous avons trois fonctions différentes dans le `utils.py` déposer. Maintenant, nous devons tester chaque fonction avec différents cas de test.
- Écrivons les tests pour la première fonction `is_prime` :
 1. Créez un fichier appelé `test_utils.py` dans le dossier d'exemples en tant que `utils.py`.
 2. Importez le `utils unittest` module.
 3. Créez une classe avec le nom `TestUtils` extension `unittest.TestCase` classer. Le nom de la classe peut être n'importe quoi. Essayez de donner à la classe un nom significatif.
 4. Dans la classe, écrivez une méthode appelée `test_is_prime` qui accepte `self` comme argument.
 5. Écrivez différents cas de test avec des arguments pour le `is_prime` et comparez la sortie avec la sortie attendue.
 6. Exemple de cas de test `self.assertFalse(utils.is_prime(1))`.
 7. Nous attendons la sortie du `is_prime(1)` sera faux dans le cas ci-dessus.
 8. Semblable au cas ci-dessus, nous allons différents cas de test en fonction de la fonction que nous testons.

Activité 3 : évaluation de fonctionnement des scripts



Voyons les tests :

```
import unittest

import utils

class TestUtils(unittest.TestCase):
    def test_is_prime(self):
        self.assertFalse(utils.is_prime(4))
        self.assertTrue(utils.is_prime(2))
        self.assertTrue(utils.is_prime(3))
        self.assertFalse(utils.is_prime(8))
        self.assertFalse(utils.is_prime(10))
        self.assertTrue(utils.is_prime(7))
        self.assertEqual(utils.is_prime(-3),
                          "Negative numbers are not allowed")
```

Activité 3 : évaluation de fonctionnement des scripts



```
if __name__ == '__main__':  
    unittest.main()
```

- Nous invoquons la méthode main du module unittest pour exécuter les tests en utilisant la commande Python filename.py. Maintenant, lancez les tests.
- Vous verrez une sortie similaire à la sortie ci-dessous.

```
$ python test_utils.py
```

```
.
```

```
-----  
Ran 1 test in 0.001s
```

```
OK
```

Activité 3 : évaluation de fonctionnement des scripts



- Maintenant, essayez d'écrire les cas de test pour d'autres fonctions également.
- Pensez à différents cas pour les fonctions et écrivez des tests pour elles.
- Jetez un œil aux tests suivants qui sont ajoutés à la classe initiale :

```
...
class TestUtils(unittest.TestCase):
    def test_is_prime(self):
        ...

    def test_cubic(self):
        self.assertEqual(utils.cubic(2), 8)
        self.assertEqual(utils.cubic(-2), -8)
        self.assertNotEqual(utils.cubic(2), 4)
        self.assertNotEqual(utils.cubic(-3), 27)
```

Activité 3 : évaluation de fonctionnement des scripts



```
def test_say_hello(self):  
    self.assertEqual(utils.say_hello("Geekflare"), "Hello, Geekflare")  
    self.assertEqual(utils.say_hello("Chandan"), "Hello, Chandan")  
    self.assertNotEqual(utils.say_hello("Chandan"), "Hi, Chandan")  
    self.assertNotEqual(utils.say_hello("Hafeez"), "Hi, Hafeez")  
  
...
```

- Nous n'avons utilisé que certaines des fonctions de comparaison du module unittest. Vous pouvez trouver la liste complète.
- Nous avons appris à écrire des tests unitaires à l'aide du module unittest. Maintenant, il est temps de voir différentes manières d'exécuter les tests.