



# www.dirasats.com



هذا الغلاف لا يعبر عن حقوق الملكية او فحوى الكتاب, فهو مجرد واجهة للموقع المحمل منه

شكرا لك على ثقتك بنا وعلى اختيار موقعنا

www.dirasats.com



من اجل تواصل معنا المرجو زيارة الموقع ستجد جميع المعلومات

www.dirasats.com

# TP1 : SCRIPTS

Un script est une suite d'instruction élémentaire qui sont exécutées de façon séquentielle (les unes après les autres) par le langage de script. Dans cet article nous nous limiterons à l'utilisation du shell comme langage, et en particulier à bash.

## Notions de base

Pour commencer, il faut savoir qu'un script est un fichier texte standard pouvant être créé par n'importe quel éditeur : vi, emacs ou autre. D'autre part, conventionnellement un script commence par une ligne de commentaire contenant le nom du langage à utiliser pour interpréter ce script, pour le cas qui nous intéresse : /bin/sh. Donc un script élémentaire pourrait être :

```
#!/bin/sh
```

Évidemment un tel script ne fait rien ! Changeons cela. La commande qui affiche quelque chose à l'écran est echo. Donc pour créer le script `bonjour_monde` nous pouvons écrire :

```
#!/bin/sh  
echo "Bonjour, Monde !"  
echo "un premier script est né."
```

Comment on l'exécute ? C'est simple il suffit de faire :

```
[user AT became user]$ sh bonjour_monde  
Bonjour, Monde !  
un premier script est né.  
[user AT became user]$ _
```

C'est pas cool, vous préféreriez taper quelque chose comme :

```
[user AT became user]$ ./bonjour_monde  
Bonjour, Monde !  
un premier script est né.  
[user AT became user]$ _
```

C'est possible si vous avez au préalable rendu votre script exécutable par la commande :

```
[user AT became user]$ chmod +x bonjour_monde  
[user AT became user]$ ./bonjour_monde  
Bonjour, Monde !  
un premier script est né.  
[user AT became user]$ _
```

Résumons : un script shell commence par : `/bin/sh`, il contient des commandes du shell et est rendu exécutable par `chmod +x`.

- 1) Quelques conseils concernant les commentaires Dans un shell-script, est considéré comme un commentaire tout ce qui suit le caractère `#` et ce, jusqu'à la fin de la ligne. Utilisez et abusez des commentaires : lorsque vous relirez un script 6 mois après l'avoir écrit, vous serez bien content de l'avoir documenté.

Exemple :

```
#!/bin/sh
# on veut faire une copie de tous les fichiers
for i in * ; do
# sous le nom $i.bak
    cp $i $i.bak
done
```

Là, au moins, on sait ce qu'il se passe. (Il n'est pas encore important de connaître les commandes de ces deux fichiers)