

PREMIÈRES ÉTAPES

Ce que vous allez apprendre !

- Qu'est-ce qu'un script ?
- Les composants principaux d'un script.
- Comment utiliser des variables.
- De quelle manière peut-on effectuer des tests et des prises de décision.
- Comment donner des arguments à votre script.
- Comment permettre à l'utilisateur de rentrer des données qui vont pouvoir être utilisées par le script.
- Téléchargement de **GIT BASH** pour windows et de **notepad++**

Un script

- Contient une série de commandes.
- Ces commandes sont exécutées par un interpréteur (le shell pour les scripts que nous allons voir) les unes après les autres.
- Tout ce que vous pouvez taper en ligne de commande peut être inclus dans un script.
- Le scripting est la méthode idéale pour l'automatisation de tâches et notamment la mouvance DevOPS.

Rendre un script exécutable

- Il est important de mettre les droits sur le fichier contenant le script de manière à ce que celui-ci devienne exécutable.

```
chmod a+x script.sh
```

- Le `chmod a+x` permet de rajouter l'exécution sur tous les utilisateurs pour le fichier `script.sh`

Exécuter un script

- Pour exécuter un script, vous avez le choix de, soit vous trouver dans le répertoire où se situe votre script, soit d'utiliser le chemin absolu.
- Imaginons que notre script soit dans le répertoire jordan contenu dans le répertoire home.

```
Chemin absolu :  
/home/jordan/script.sh
```

```
Chemin relatif (si je suis dans le même répertoire que  
le script):  
./script.sh
```

Le Shebang !

- Au début de chaque script bash, il est important de faire figurer le shebang :

```
#!/bin/bash
```

- Il commence par un « # » (ce qui correspond la plupart du temps à des commentaires) puis d'un « ! » et enfin le chemin menant au programme Bash Shell.
- En effet, nous y indiquons le chemin de l'interpréteur qui va lire le script. Il en existe d'autres :

```
#!/bin/sh  
#!/bin/csh  
#!/bin/zsh  
...
```

Le Shebang ! (2)

- Si vous décidez de ne pas faire figurer de Shebang au début de votre script, alors les commandes à l'intérieur du script seront exécutées en utilisant votre propre shell.
- Attention car certaines syntaxes varient lorsque l'on utilise le shell bash par rapport au shell zsh. C'est pourquoi il est fortement recommandé d'être propre et de spécifier l'interpréteur que votre script devra utiliser.

Le Shebang ! (3)

- Comme vous l'avez compris, le shebang fait référence à l'interpréteur qui sera utilisé.
- De ce fait, on peut lui spécifier n'importe quel interpréteur comme par exemple l'interpréteur Python, et écrire du code Python à l'intérieur du script :

```
#!/usr/bin/python  
print('Bonjour je m'appelle Jordan')
```


Les variables

- Ce sont des espaces de stockage qui possèdent un nom.
- Imaginez que vous associez à un nom, un contenu qui peut varier au fur et à mesure de votre script. C'est cela que l'on appelle variable !

```
NOM_DE_LA_VARIABLE="valeur"
```

- Les variables sont sensibles à la casse et par convention, on les met toujours en majuscules.
- Attention à ne pas mettre d'espace entre la variable, le signe = et les ".

Les commentaires

- En bash, les commentaires sont des éléments figurants dans le script qui ne vont pas être interprétés par l'interpréteur.
- En effet, lorsque le script va détecter le caractère **#**, il ne va pas interpréter les éléments figurant après sur la même ligne excepté pour le shebang.

```
#!/bin/bash
echo "Bonjour je m'appelle Jordan"

#Ceci est un commentaire
#Ceci ne sera pas interprété
```