LES FONCTIONS

A quoi servent les fonctions ?

- Permet d'écrire un block de code une fois, et de le réutiliser par la suite dans le script.
- Eviter d'avoir à retaper du code identique, pour effectuer plusieurs tâches similaires dans son script.
- Permet d'aérer et d'améliorer l'ergonomie du script et du code.
- Beaucoup plus facile de faire évoluer son script par la suite, en ajoutant d'autres fonctions.

Syntaxe

- Il existe deux manières de coder une fonction :

```
#!/bin/bash

#Déclarer une fonction en utilisant le mot "function"
function je-suis-une-fonction(){
        command1
        command2
        commandn
}

#Déclarer une fonction sans spécifier le mot "function"
Je-suis-une-fonction(){
        command1
        command2
        commandn
}
```

Appeler une fonction dans un script

- Il suffit d'utiliser le nom de la fonction (sans les parenthèses) pour appeler le code qu'elle contient.

```
#!/bin/bash

function internet() {
ping -c 1 8.8.8.8

if [ $? -eq 0 ]
then
        echo "La connectivité vers internet est
établie"
else
        echo "Pas de connectivité vers internet"
fi
}

internet
```

Les paramètres

- Tout comme les scripts eux-mêmes, les fonctions peuvent accepter des paramètres.
- De la même manière, le premier paramètre est stocké dans le $1, le second dans le $2, etc…

- Attention, le $0 fait référence au nom du script lui-même et non pas au nom de la fonction.

Utilisation de paramètres dans une fonction

```
#!/bin/bash

function internet() {
ping -c $1 $2

if [ $? -eq 0 ]
then
        echo "La connectivité vers internet est
établie"
else
        echo "Pas de connectivité vers internet"
fi
}

Internet "1" "8.8.8.8"
```

Les variables

- Les variables globales peuvent tout à fait être utilisées dans des fonctions à condition qu'elles aient été déclarées avant la fonction.

```
#!/bin/bash

VARIABLE="Jordan"

function demonstration() {
	echo "La variable $VARIABLE est utilisable"
}
```

Les variables (2)

- Cependant les variables déclarées dans une fonction ne peuvent être utilisées qu'une fois que la fonction a été exécutée

```
#!/bin/bash

VARIABLE="Jordan"

function demonstration() {
	echo "La variable $VARIABLE est utilisable"
	VARIABLE_IN_FUNCTION="1"
}
#VARIABLE_IN_FUNCTION n'est pas utilisable

demonstration
#VARIABLE_IN_FUNCTION est désormais utilisable
```

Les variables locales

- Les variables locales ne peuvent qu'être utilisées au sein d'une fonction.
- Utilisation du mot : local

```
#!/bin/bash

VARIABLE="Jordan"

function demonstration() {
        echo "La variable $VARIABLE est utilisable"
        local VARIABLE_IN_FUNCTION="1"
}


demonstration
#VARIABLE_IN_FUNCTION n'est pas utilisable
```

Le code retour d'une fonction

- Les fonctions possèdent un code de retour qui peut être explicitement indiqué grâce à la commande **return**.

```
#!/bin/bash

PRENOM="Jordan"

function demonstration() {
        echo "Bonjour je m'appelle $PRENOM"
        return 0
}
```

- Si ce code retour n'a pas été défini, alors c'est le code retour de la dernière variable exécutée dans la fonction qui sera celui de la fonction par défaut.

Le code retour d'une fonction (2)

- Le code retour d'une fonction est compris entre 0 et 255.
- On peut y accéder grâce à la commande $?

```
#!/bin/bash

function internet_connectivity() {
	ping -c 1 8.8.8.8 && return 0
}

internet_connectivity

if [ $? -eq 0 ]
then
echo "Connectivité vers internet"
fi
```