

Compte rendu TP2

RIOUAL Matthieu :computer:

MORNEAU Hugo :computer:

Exercice 1

1. Les commandes sont cherchés dans les fichiers contenus dans le **PATH**
2. Le chemin absolu de notre répertoire ourant est stocké dans le **HOME**
3. Quelques variables d'environnement bien utiles :
 - **LANG** : langue utilisée par les logiciels
 - **PWD** : repertoire courant
 - **OLDPWD** : repertoire courant precedent
 - **SHELL** : emplacement du bash
 - **_** : printenv
4. Une variable locale est crée ainsi : `MY_VAR="bonjour"`
5. Lorsque l'on tape `bash` dans le shell, la variable local disparaît => Cette commande permet d'ouvrir un nouvel environnement bash. si l'on tape `exit` on retourne dans l ancien environnement et **MY_VAR** réapparaît.
6. Pour transformer notre variable **MY_VAR** en une variable d'environnement : `export MY_VAR`. Cette fois ci la commande `bash` conserve dans son environnement la référence de la variable : on peut accéder à sa valeur.
7. Pour créer et afficher une variable d'environnement, le code est le suivant : `export NOMS="MORNEAU RIOUAL|echo $NOMS"`
8. Pour afficher la variable dans une phrase on effectue la commande suivant : `echo "Bonjour a vous deux, $NOMS !"`
9. La commande `unset` supprime la référence tandis que affecter une valeur vide ne le fait pas
10. La commande est la suivante : `echo '$HOME :' $HOME`

Exercice 2

Avant de taper nos scripts, nous créons un dossier de travail et nous l'ajoutons à la variable **PATH** avec la commande suivante : `cd ~|mkdir script|echo 'PATH=$PATH:~/script'>> ~/.bashrc`. Ce qui a pour conséquence d'ajouter le chemin ~/script dans la variable **PATH** à chaque démarrage.

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

```
PASSWORD="TP"
```

```
read -s -p 'veuillez donner le mot de passe' passe
if [ \ $pass=$PASSWORD ]; then
    echo 'bien vu'
else
    echo 'dommage'
fi
```

Exercice 3:

```
#!/bin/bash

function is_number()
{
    re='^[+-]?[0-9]+([.][0-9]+)?$'
    if ! [[ $1 =~ $re ]] ; then
        return 1
    else
        return 0
    fi
}

is_number $1
if [ $? -eq 0 ]; then
    echo 'reel'
else
    echo 'pas reel'
fi
```

Exercice 4:

```
#!/bin/bash

if [ $# -eq 0 ]; then
    echo "Utilisation : $0 nom_utilisateur"
else
    while(( "$#" )); do
        a=$(grep -w "$1" /etc/passwd|wc -l)
        if [ $a -eq 0 ]; then
            echo "utilisateur non existant"
        else
            echo "utilisateur existant"
        fi
        shift
    done
fi
```

Exercice 5:

```
#!/bin/bash

fact=1
for i in $(seq 2 $1); do
    fact=$((fact * $i))
done
echo $fact
```

Exercise 6:

```
#!/bin/bash

try=1001
nb=$RANDOM
while [ $nb -gt 1000 ]
do
    nb=$RANDOM
done

while [ $try -ne $nb ]
do
    read -p 'choisissez un nombre!' try
    if [ $try -lt $nb ]; then
        echo -e '\nEssayez plus grand!\n'
    elif [ $try -gt $nb ]; then
        echo -e '\nEssayez plus petit!\n'
    fi
done
echo -e '\nBravo, le nombre est :' $nb
```

Exercise 7:

```
#!/bin/bash

fin=0
i=2
tab=()
val =1

function is_number()
{
    re='^[+-]?[0-9]+([.][0-9]+)?$'
    if ! [[ $1 =~ $re ]] ; then
        return 1
    else
        return 0
    fi
}
```

```
}

is_number val
while [ $? -eq 1 ]
do
    read -p 'Entrez une valeur numérique' val
    is_number $val
done
tab[1]=$val
while [$fin -ne 1 ]
do
    read -p 'Entrez une valeur, ou exit pour sortir' val
    is_number val
    if [ $? -eq 0 ]; then
        min=
        max=
        sum=
        tab[i]=$val
        for j in $(seq 1 $i)
        do
            if [ ${tab[j]} -lt $min ]; then
                min = ${tab[j]}
            elif [ ${tab[j]} -gt $max ]; then
                max=${tab[j]}
            fi
            sum=$(( $sum + ${tab[j]} ))
        done
    else
        if [ $val = 'exit' ]; then
            fin=1
        else
            echo 'nombre non valide'
        fi
    fi
done
echo 'Vous êtes sorti'
```