

```

#!/bin/bash
if [ $# -eq 0 ]; then
    echo "No arguments supplied"
else
    i=1
    while [ $i -le 3 ]
    do
        eval "arg=\${$i}"
        if [ -n "$arg" ]; then
            echo "$arg"
            i=$((i+1))
        else
            break
        fi
    done

    // while : tant que ; tant que la valeur de la variable i est
    // inférieure ou égale (le : less or equal) à 3
    // la suite d'instructions entre do et done sera répétée (3 fois maximum)
    // on utilise eval pour exécuter la commande "arg=\${$i}" dynamiquement
    // cette commande assignera la valeur du nème argument positionnel
    // (avec n = valeur de i)
    // à la variable arg
    // ainsi, on assignera à arg la valeur de l'argument 1
    // quand i sera égal à 1, la valeur de l'argument 2
    // quand i sera égal à 2, etc)
    // on assignera donc $1 à arg,
    // puis $2 à arg (après un tour de boucle), etc

    // il est nécessaire d'échapper (avec un antislash) le premier $
    // pour ne pas que le shell l'interprète directement
    // on veut que cela soit eval qui l'interpète
    // eval traitera ainsi la commande comme arg=$1 puis arg=$2 ...

    // si la valeur de arg n'est pas nulle
    // (si l'argument 1 (au 1er tour de boucle) existe,
    // puis l'argument 2 (au 2ème tour), puis le 3)
    // on l'affiche
    // on incrémente le compteur de 1
    // sinon (si l'argument en cours n'existe pas)
    // on sort de la boucle ; on ne fera pas d'autre tour de boucle
    // permet d'économiser un tour de boucle
    // si 1 seul argument est fourni, le programme sortira de la boucle
    // après avoir vérifié que le 2ème argument est vide, il ne testera
    // pas si le 3ème argument est vide

    // fin de si

    // terminé (fin de la boucle while)

```

fi

```
// si la condition [ $i -le 3 ] est toujours vraie,  
// on refait un tour de boucle  
// fin de si
```