

```

#!/bin/bash
if [ $# -eq 0 ]; then
    echo "No arguments supplied"
else
    i=1
    // sinon (si on a fourni au moins 1 argument au script)
    // on initialise la variable i à 1 ; elle servira de compteur
    // dans la boucle for (elle sera incrémentée plus tard de 1 après chaque
    // affichage de la valeur d'un argument à l'écran)

    for arg in "$@"
        // pour (for) chaque élément (arg) dans (in) la liste
        // de tous les arguments passés au script ("$@")
        // on débute donc une boucle for, qui parcourera
        // tous les arguments passés au script
        // for ... in : pour ... dans ;
        // boucle (répète les instructions entre do (fais) et done (terminé)
        // pour chacun des éléments (arg) dans la liste (in)
        // de tous les arguments passés au script ("$@")
        // arg est le nom de la variable qui stockera la valeur
        // de chaque argument
        // passé au script, l'un après l'autre (le 1er au premier tour de boucle,
        // le deuxième au deuxième tour, etc..)
        // $@ : tous les arguments passés au script
        // on indique $@ entre double guillemets ("$@") pour pouvoir traiter
        // chaque argument comme une chaîne de caractères

    do
        // do : fais
        // ce qui suit sera répété pour chaque arg
        // (pour chaque argument passé au script, un par un),
        // en commençant par le premier argument
        // après done (terminé), le script reviendra à cet endroit (à do),
        // exécutera à nouveau les instructions jusqu'à done,
        // mais cette fois avec l'argument suivant (le deuxième), etc...

        if [ $i -le 3 ]; then
            // si la valeur de la variable i est inférieure ou égale
            // (le : less or equal) à 3, alors
            echo "$arg"
            // on affiche la valeur de l'argument en cours
            // (celui du 1er au 1er tour de boucle, du 2ème au 2ème tour, etc)
            // sous la forme d'une chaîne de caractères
            // (grâce aux doubles guillemets)
        fi
    done
fi

```

```

i=$((i+1)) // $(( )) est la syntaxe pour effectuer des opérations arithmétiques :
            // ici, elle calcule i+1, et l'affecte à i (grâce au signe =)
            // elle augmente donc (elle incrémente) la valeur de i de 1
            // comme i est égal à 1 au début du else, sa valeur passera à 2
            // grâce à cette opération
            // au 2ème tour de boucle, elle passera à 3, etc...
            // donc, après avoir affiché la valeur du 1er argument à l'écran,
            // i sera égal à 2, puis à 3 après avoir affiché la valeur du 2ème...
            // d'où le terme de compteur
fi          // fin de si

done        // terminé : on refait un tour de boucle s'il reste un argument
            // à traiter dans la liste des arguments ("$@")

            // ATTENTION : A 42, L'UTILISATION DES BOUCLES FOR EST INTERDIT !

fi          // fin de si

```