

```
cd /home/event/discovery_piscine
mkdir cell04
cd cell04
mkdir ex04
cd ex04
```

```
vim argv.sh
i
```

```
#!/bin/bash
if [ "$#" -eq 0 ]; then // si le nombre d'arguments passés au script est égal à 0, alors

    // if : si
    // début de la structure conditionnelle (condition + action)

    // [] : bloc contenant le test conditionnel (la condition)
    // attention à bien mettre un espace après le crochet ouvrant et avant le crochet fermant
    // pour que la commande soit bien interprétée !

    // "$#" : nombre d'arguments passés au script
    // -eq : equal (égal) (test d'égalité)
    // point-virgule ; fin de la ligne (techniquement, then est sur une autre ligne)
    // On peut aussi l'écrire sur 2 lignes :
        // if [ "$#" -eq 0 ]
        // then

    // then : alors
    // sépare la condition testée du bloc d'instructions à exécuter

    echo "No arguments supplied" // on affiche à l'écran (sortie standard) : "No arguments supplied"
fi // fin de la structure conditionnelle
if [ -n "$1" ]; then // si $1 n'est pas vide (si l'argument numéro 1 existe), alors

    // $1 : valeur du premier argument passé au script
    // c'est-à-dire la première valeur qui suit le nom du programme
    // lorsqu'on l'exécute
    // dans cet exemple : ./argv.sh 2 40
    // 2 est le premier argument et 40 est le 2ème argument passé au script
```

```

// "" : "en tant que chaîne de caractère"
// penser à l'utiliser pour ne pas que le script interprète
// le 1er argument comme une commande, mais bien comme une chaîne de
// caractère à afficher à l'écran !

echo "$1"
(en chaîne de caractères)
fi
if [ -n "$2" ]; then
    echo "$2"
fi
if [ -n "$3" ]; then
    echo "$3"
fi

echap
:wq
enter

chmod +x argv.sh

// on affiche à l'écran (sortie standard) la valeur du premier argument

// fin de la structure conditionnelle
// idem pour le 2ème argument

// idem pour le 3ème argument

// remarque : une ligne est sautée automatiquement après un echo

// TESTS :

./argv.sh
// résultat : No arguments supplied

./argv.sh 42
// résultat : 42

./argv.sh 2 40
// résultat : 2
//          40

./argv.sh Hello World !
// résultat : Hello
//          World
//          !

./argv.sh Hello World ! 42
// résultat : Hello
//          World
//          !

```