


```
cat > mon_script.sh << FIN
echo "Bonjour, ceci est mon premier script"
echo "Arguments: $*"
FIN
```

2. Vous devriez obtenir un fichier `mon_script.sh`. Visualisez-le. Que remarquez-vous?
3. Recommencez, avec cette fois-ci un « \ » pour protéger le « \$ ».
4. Ajouter le sheband en début de script (avec un éditeur interactif). Rendre le script exécutable (`chmod`). Puis exécuter le script en donnant des arguments.

4 Second script

Dans la suite de ce TP, il pourrait être plus intéressant d'utiliser un langage de script, plutôt que le shell BASH et des commandes systèmes. Mais le but est de vous faire manipuler et non de vous apprendre un langage de programmation. Donc n'ayez pas peur si vos solutions vous semblent inefficaces!

1. Créez un fichier « `ingredients.txt` » contenant le texte suivant, permettant de réaliser une recette bien de chez nous :

```
patates:1:kg
lardons:200:g
oignons:200:g
reblochon:1
huile:2:cuillieres
ail:1:gousse
sel:1:pincee
poivre:1:pincee
```

2. Faites un script qui parcourt ce fichier et affiche les ingrédients sous une forme humainement lisible :

```
1kg de patates
200g de lardons
200g de oignons
1 reblochon
1 gousse de ail
1 pincee de sel
1 pincee de poivre
```

Le script utilisera le shell `/bin/bash` et devra s'interrompre en cas d'erreur. Prenez garde à bien traiter le cas particulier du reblochon qui n'a pas d'unité!

3. Modifiez votre script afin qu'il prenne un argument qui sera un multiplicateur pour la quantité d'ingrédients (par exemple, l'argument « 2 » multiplie par 2 toutes les quantités)
4. Nous voulons tester si l'argument donné est bien numérique. Un moyen de faire est de réaliser une opération qui génère une erreur lorsque l'argument n'est pas numérique et nous allons traiter cette erreur. Proposez un tel test en début de script, afin d'afficher un message « L'argument doit être un nombre » en cas d'argument non numérique.