

## Série d'exercices 8 : fonctions

### Exercice 1 – print dans une fonction

Écrivez une fonction qui prend un argument en entrée et l'affiche au sein du texte suivant : « Bonjour x! » où x est votre argument. Appelez ensuite votre fonction 5 fois de suite pour afficher les lignes suivantes :

*Bonjour Marc!*

*Bonjour Jean!*

*Bonjour Patricia!*

*Bonjour Louise!*

*Bonjour André!*

### Exercice 2 – appel de fonction en boucle

Copiez-collez votre fonction de l'exercice 1 et cette fois utilisez la avec une boucle sur chaque élément du tableau suivant : ["Marc", "Jean", "Patricia", "Louise", "André", "Marlène", "Victor", "Sophia"].

### Exercice 3 – Carré ou cube de nombre

Écrivez deux fonctions qui prennent chacune un seul nombre en argument. Une fonction retournera le carré (puissance 2) du nombre passé en argument, l'autre fonction retournera le cube (puissance 3) du nombre passé en argument.

Puis, écrivez un programme qui demande à l'utilisateur un nombre, puis qui demande à l'utilisateur s'il veut calculer le carré ou le cube du nombre (l'utilisateur tapera 2 ou 3 pour indiquer son choix). Le programme appellera ensuite soit l'une soit l'autre fonction sur le nombre donné par l'utilisateur en fonction de son choix.

Exemple d'utilisation :

*Donnez un nombre : 6*

*Voulez-vous calculer le carré (2) ou le cube (3) de votre nombre? 2*

*Votre résultat est : 36*

Si l'utilisateur donne un choix invalide, votre programme devra afficher « *****Vous devez choisir 2 ou 3***** ».

### Exercice 4 – carrés ou cubes d'un tableau affichés

Modifiez légèrement votre programme de l'exercice 3 de la façon suivante : cette fois l'utilisateur ne donne pas un nombre, il choisit s'il veut calculer les carrés ou les cubes sur le tableau suivant de nombres : [104, 200, 36, 470, -97, 23, 18, 74, 9, -4]. Vous devrez appeler soit l'une soit l'autre fonction en boucle sur ce tableau, puis afficher le résultat ligne par ligne pour chaque élément du tableau.

### Exercice 5 – carrés ou cubes d'un tableau modifié

Modifiez légèrement votre programme de l'exercice 4 pour qu'au lieu d'afficher chaque résultat, votre programme modifie chaque élément du tableau en remplaçant le nombre par son carré ou son cube. Tout à la fin du programme, afficher simplement le tableau modifié en utilisant `print()` une fois.

### Exercice 6 – Lister et ajouter des contacts

Écrivez un programme qui demande en boucle à l'utilisateur de taper seulement 1 lettre pour indiquer ce qu'il veut faire :

- S'il tape « Q » le programme devra afficher « Au revoir » et s'arrêter.
- S'il tape « A » le programme devra appeler une fonction qui ajoute un contact téléphonique à une liste.
- S'il tape « L » le programme devra appeler une fonction qui liste les contacts téléphoniques.

Le programme continuera de demander à l'utilisateur ce qu'il veut faire tant que la réponse n'est pas « Q ». Conseil : commencez par coder un programme qui correspond à cette donnée et qui appelle des fonctions qui ne font rien à part afficher un message. Une fois que c'est fait, modifiez les deux fonctions pour qu'elles correspondent aux consignes suivantes :

La fonction qui ajoute un contact téléphonique devra demander à l'utilisateur le nom du nouveau contact (par exemple « Marc »), puis son numéro de téléphone (par exemple « 0230761213 »). Chacune de ces deux informations devra être ajoutée dans un tableau différent. Puis le programme affichera le message : « Marc : 0230761213 ajouté ».

La fonction qui liste les contacts téléphoniques devra parcourir les deux tableaux (celui des noms des contacts et celui des numéros de téléphone) et simplement les afficher ligne par ligne ainsi : « Marc : 0230761213 ».

Évidemment que si l'utilisateur commence par taper « L » le programme n'affichera rien puisque l'utilisateur n'a encore ajouté aucun contact. Si l'utilisateur tape « A », puis donne un

nom et un numéro, puis tape « L », le programme affichera une seule ligne avec le nom et le numéro du contact ajouté.

## Exercice 7 – Lister, ajouter et modifier des contacts

Reprenez votre programme de l'exercice 6 et ajoutez une nouvelle fonctionnalité : l'utilisateur peut désormais taper « M » pour indiquer qu'il veut modifier le numéro de téléphone d'un contact existant. La fonction appelée dans ce cas devra d'abord demander à l'utilisateur le nom du contact qu'il souhaite modifier (par exemple « Marc »). Ensuite, il faudra rechercher dans le tableau des noms de contacts si ce nom existe déjà. S'il n'existe pas, votre programme devra afficher « Ce nom de contact n'existe pas. ». S'il existe, votre programme devra demander à l'utilisateur le nouveau numéro de ce contact (par exemple « 0187658978 ») et modifier le tableau des numéros de téléphone au bon endroit puis afficher « Marc : 0187658978 modifié ».

Pour simplifier, on part du principe que chaque nom de contact dans la liste est unique à tout moment.

Pour valider votre code, essayez de taper « A » pour ajouter un contact, puis « L » pour le lister, puis « M » pour le modifier et enfin de nouveau « L » pour vérifier si le contact est listé avec le nouveau numéro et non pas l'ancien.

## Vérification

Vérifiez vos fichiers en les nommant « exercice » suivi du numéro de l'exercice et en les plaçant sur le **Bureau** et en téléchargeant le fichier `python_checker_serie8.sh` depuis Moodle et en le copiant sur le **Bureau**. Ouvrez ensuite le **terminal** (touches espace + commande et tapez « ter » pour voir apparaître l'application terminal), entrez les 2 commandes suivantes :

```
cd ~/Desktop
```

```
sh python_checker_serie8.sh
```

Pour chaque exercice, le terminal devrait afficher « **exercice est correct !** ». Toute erreur sera affichée avec un message en **rouge**.