

Série d'exercices 4 : 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 – afficher une adresse email de 5 personnes

Ecrivez une fonction qui cette fois prend 2 arguments en entrée : un prénom et un nom de famille. Puis, cette fonction affiche l'adresse email « prénom.nom@eduvaud.ch » : Appelez ensuite votre fonction 3 fois de suite pour afficher les lignes suivantes :

Marc.Duchemin@eduvaud.ch

Jean.Favre@eduvaud.ch

Patricia.Renaud@eduvaud.ch

Exercice 3 – afficher une adresse email selon l'utilisateur

Reprenez votre fonction de l'exercice 2 telle quelle sans la modifier. Cette fois, ne l'appellez pas sur des prénoms et des noms déjà donnés mais prenez l'input de l'utilisateur. Demandez-lui son prénom et son nom de famille et affichez l'adresse email correspondante.

Exemple d'utilisation :

Donnez votre prénom : Paul

Donnez votre nom de famille : Obama

Paul.Obama@eduvaud.ch

Exercice 4 – Carré ou cube de nombre

Écrivez deux fonctions qui prennent chacune un seul nombre en argument. Une fonction affichera le carré (puissance 2) du nombre passé en argument, l'autre fonction affichera le cube (puissance 3) du nombre passé en argument. Les affichages de chaque fonction se feront dans une phrase sous la forme « *Votre résultat est : ...* »

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 5 – prix avec rabais

Téléchargez le fichier `exercice5.py` et créez la fonction `prix_final` qui prend deux arguments : un prix en francs et un rabais en pourcentage. La fonction calcule le prix final en fonction du rabais. Vous n’avez pas besoin de prendre l’input de l’utilisateur, 4 appels de fonction sont déjà donnés et votre programme doit afficher :

Le prix final de 100 CHF avec 10% de rabais est: 90.0 CHF.

Le prix final de 100 CHF avec 50% de rabais est: 50.0 CHF.

Le prix final de 80 CHF avec 25% de rabais est: 60.0 CHF.

Le prix final de 120 CHF avec 15% de rabais est: 102.0 CHF.

Exercice 6 – surface d’un rectangle

Écrivez un programme qui contient une fonction appelée `surface` et qui prend 2 arguments : une longueur et une largeur en mètres. La fonction calcule la surface du rectangle en fonction de la longueur et de la largeur. Votre programme doit demander une longueur et une largeur à l’utilisateur et utiliser la fonction pour afficher la surface. Exemple d’exécution :

Donnez la longueur en mètres : 10

Donnez la largeur en mètres : 3

La surface est de 30 m2.

Exercice 7 – Jeu textuel

On veut écrire la logique d’un jeu vidéo basé uniquement sur du texte. On commence par indiquer à l’utilisateur la situation :

Vous vous trouvez face à deux portes. Laquelle choisissez-vous ? Celle de gauche ou celle de droite?

Si le joueur choisit « *gauche* » il se retrouvera face à un monstre. S’il choisit « *droite* », on lui proposera un autre choix :

Vous arrivez dans une salle avec quatre portes numérotées de 1 à 4. Laquelle choisissez-vous?

Si le joueur choisit « *2* », il se retrouvera face au même monstre. S’il choisit une autre porte, on lui affichera le texte suivant : « *Vous tombez dans une fosse. Vous avez perdu.* »

Dans les deux cas où le joueur se retrouve face à un monstre, on lui proposera de fuir ou de combattre. S'il choisit « fuir » on affichera : « Vous prenez la fuite. Vous avez perdu. » S'il choisit « combattre », on affichera : « Vous avez battu le monstre. Vous avez gagné! »

Vous devez coder ce jeu en créant une fonction pour gérer la situation du monstre pour ne pas copier-coller à deux endroits cette partie du code.

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_serie4.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_serie4.sh
```

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