

TD 1 Programmation Python

Exercice 1

1. Affectez les variables temps et distance par les valeurs 6.892 et 19.7. Calculez et affichez la valeur de la vitesse. Améliorez l’affichage en imposant un chiffre après le point décimal.
2. Saisir un nom et un âge en utilisant l’instruction input(). Les afficher. Refaire la saisie du nom, mais avec l’instruction raw_input(). L’afficher. Enfin, utilisez la « bonne pratique » : recommencez l’exercice en transtypant les saisies effectuées avec l’instruction raw_input()

Exercice 2

1. Saisissez un flottant. S’il est positif ou nul, affichez sa racine, sinon affichez un message d’erreur.
2. L’ordre lexicographique est celui du dictionnaire. Saisir deux mots, comparez-les pour trouver le « plus petit » et affichez le résultat. Refaire l’exercice en utilisant l’instruction ternaire.
3. On désire sécuriser une enceinte pressurisée. On se fixe une pression seuil et un volume seuil : pSeuil = 2.3, vSeuil = 7.41. On demande de saisir la pression et le volume courant de l’enceinte et d’écrire un script qui simule le comportement suivant :
 - si le volume et la pression sont supérieurs aux seuils : arrêt immédiat ;
 - si seule la pression est supérieure à la pression seuil : demander d’augmenter le volume de l’enceinte;
 - si seul le volume est supérieur au volume seuil : demander de diminuer le volume de l’enceinte
 - sinon déclarer que « tout va bien ».

Ce comportement sera implémenté par une alternative multiple.

Exercice 3

Initialisez deux entiers : a = 0 et b = 10. Écrire une boucle affichant et incrémentant la valeur de a tant qu’elle reste inférieure à celle de b. Écrire une autre boucle décrémentant la valeur de b et affichant sa valeur si elle est impaire. Boucler tant que b n’est pas nul.

Exercice 4

Écrire une saisie filtrée d’un entier dans l’intervalle 1 à 10, bornes comprises. Affichez la saisie.

Exercice 5

Affichez chaque caractère d’une chaîne en utilisant une boucle for. Affichez chaque élément d’une liste en utilisant une boucle for.

Exercice 6

Affichez les entiers de 0 à 15 non compris, de trois en trois, en utilisant une boucle for et l’instruction range().

Exercice 7

Utilisez l’instruction continue pour modifier une boucle for d’affichage de tous entiers de 1 à 10 compris, sauf lorsque la variable de boucle vaut 5.

Exercice 8

Utilisez une exception pour calculer, dans une boucle évoluant de -3 à 3 compris, la valeur de $\sin(x)/x$.

TD 1 Programmation Python B2

Exercice 9

La clause else des boucles. Dans cet exercice, effectuez les saisies avec des integerbox et les affichages avec des msgbox, tous deux appartenant au module easygui. Initialisez une liste avec 5 entiers de votre choix puis saisissez un entier. Dans une boucle for, parcourez la liste. Si l'entier saisi appartient à la liste, sauvez-le et interrompez la boucle (puisque vous l'avez trouvé). Si la boucle s'est bien terminée, utilisez une clause else pour afficher un message l'annonçant. Entrez maintenant un autre entier, cette fois-ci positif. Écrivez une boucle while pour déterminer si cet entier est premier. S'il ne l'est pas, la boucle devra afficher le premier diviseur trouvé et s'interrompre. S'il est premier, l'afficher dans une clause else.

Exercice 10

Écrire une procédure table avec quatre paramètres : base, debut, fin et inc. Cette procédure doit afficher la table des base, de debut à fin, de inc en inc. Tester la procédure par un appel dans le programme principal.

Exercice 11

Écrire une fonction cube qui retourne le cube de son argument.

Écrire une fonction volumeSphere qui calcule le volume d'une sphère de rayon r fourni en argument et qui utilise la fonction cube. Tester la fonction volumeSphere par un appel dans le programme principal.

Exercice 12

Écrire une fonction maFonction qui retourne $f(x) = 2x^3 + x - 5$.

Écrire une procédure tabuler avec quatre paramètres : fonction, borneInf, borneSup et nbPas. Cette procédure affiche les valeurs de fonction, de borneInf à borneSup, tous les nbPas. Elle doit respecter $f < \text{borneSup}$. Tester cette procédure par un appel dans le programme principal après avoir saisi les deux bornes dans une floatbox et le nombre de pas dans une integerbox (utilisez le module easyguiB).

Exercice 13

Écrire une fonction volMasseEllipsoide qui retourne le volume et la masse d'un ellipsoïde grâce à un tuple. Les paramètres sont les trois demi-axes et la masse volumique. On donnera à ces quatre paramètres des valeurs par défaut. On donne : $v = \frac{4}{3} \pi abc$ Tester cette fonction par des appels avec différents nombres d'arguments.

Exercice 14

Écrire une fonction somme avec un argument « tuple de longueur variable » qui calcule la somme des nombres contenus dans le tuple. Tester cette fonction par des appels avec différents tuples d'entiers ou de flottants.

Exercice 15

Écrire une autre fonction somme avec trois arguments, et qui renvoie leur somme. Dans le programme principal, définir un tuple de trois nombres, puis utilisez la syntaxe d'appel à la fonction qui décompresse le tuple. Affichez le résultat.

Exercice 16

Écrire une fonction unDictionnaire avec un argument « dictionnaire de longueur variable », et qui affiche son argument. Dans le programme principal, définir un dictionnaire, puis utilisez la syntaxe d'appel à la fonction qui décompresse le dictionnaire. Affichez le résultat.

TD 1 Programmation Python B2

Exercice 17

définir la liste : `liste = [17, 38, 10, 25, 72]`, puis effectuez les actions suivantes :

- triez et affichez la liste ;
- ajoutez l'élément 12 à la liste et affichez la liste ;
- renversez et affichez la liste ;
- affichez l'indice de l'élément 17 ;
- enlevez l'élément 38 et affichez la liste ;
- affichez la sous-liste du 2^e au 3^e élément ;
- affichez la sous-liste du début au 2^e élément ;
- affichez la sous-liste du 3^e élément à la fin de la liste ;
- affichez la sous-liste complète de la liste ;
- affichez le dernier élément en utilisant un indicage négatif.

Bien remarquer que certaines méthodes de liste ne retournent rien.

Exercice 18

Initialisez `truc` comme une liste vide, et `machin` comme une liste de cinq flottants nuls. Affichez ces listes.

Utilisez la fonction `range()` pour afficher :

- les entiers de 0 à 3 ;
- les entiers de 4 à 7 ;
- les entiers de 2 à 8 par pas de 2.

Définir `chose` comme une liste des entiers de 0 à 5 et testez l'appartenance des éléments 3 et 6 à `chose`.

Exercice 19

Utilisez une liste en compréhension pour ajouter 3 à chaque élément d'une liste d'entiers de 0 à 5.

Exercice 20

Utilisez une liste en compréhension pour ajouter 3 à chaque élément d'une liste d'entiers de 0 à 5, mais seulement si l'élément est supérieur ou égal à 2.

Exercice 20

Utilisez une liste en compréhension pour obtenir la liste `['ad', 'ae', 'bd', 'be', 'cd', 'ce']` à partir des chaînes "abc" et "de".

Indication : utilisez deux boucles `for` imbriquées.

Exercice 21

Utilisez une liste en compréhension pour calculer la somme d'une liste d'entiers de 0 à 9.

Exercice 22

Soit `A` un entier naturel. Écrire un programme qui permet de savoir si l'année `A` est bissextile ou non. On rappelle qu'une année est bissextile si elle est soit divisible par 4 mais non divisible par 100, soit divisible par 400.

TD 1 Programmation Python B2

Exercice 23

Soit m un entier naturel non nul donné. Écrire un programme qui affiche la valeur de la somme

$$1^5 + 2^5 + \dots + m^5.$$

Exercice 24

Écrire un programme qui affecte une valeur à un taux de TVA, une valeur à un montant TTC puis calcule le montant HT correspondant.

Exercice 25

Écrivez un programme qui permet à l'utilisateur de saisir un nombre, puis affiche si le nombre est pair ou impair.

Exercice 26

Généralisez l'exercice précédent : saisir un nombre entier P et un nombre entier Q pour dire si Q divise P ou non.

Exercice 27

Écrire un programme qui permet à l'utilisateur de saisir un nombre entier, puis affiche ses diviseurs.