

Exercice n°1 :

Affectez les variables temps et distance par les valeurs 12.872 et 32.5.

Calculez et affichez la valeur de la vitesse.

Améliorez l’affichage en imposant un chiffre après le point décimal.

Exercice n°2 :

Saisir un nom et un âge en utilisant l’instruction `input()` . Les afficher.

Refaire la saisie du nom, mais avec l’instruction `raw_inpu()` . L’afficher.

Enfin, utilisez la « bonne pratique » : recommencez l’exercice en transtypant les saisies effectuées avec l’instruction `raw_input()`

Exercice n°3 :

Saisissez un flottant. S’il est positif ou nul, affichez sa racine, sinon affichez un message d’erreur.

(Fonction racine : `sqrt()` ; il faut importer la fonction `sqrt()` du module `math`)

Exercice n°4 :

Saisir deux nombres, comparez-les pour trouver le « plus petit » et affichez le résultat.

Faire l’exercice en utilisant l’instruction ternaire (3 éléments):

`<res> = <a> if <condition> else `

Exercice 5 :

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 6 :

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 7 :

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

Exercice 8 :

Affichez chaque caractère d'une chaîne en utilisant une boucle `for`.

Affichez chaque élément d'une liste en utilisant une boucle `for`.

Exercice 9 :

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

Exercice 10 :

Utilisez l'instruction `break` pour interrompre une boucle `for` d'affichage des entiers de 1 à 10 compris, lorsque la variable de boucle vaut 5.

Exercice 11 :

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 12 :

Utilisez une exception pour calculer, dans une boucle évoluant de -3 à 3 compris, la valeur de $\sin(x)/x$. (Importer la fonction `sin` du module `math`) « `from math import sin` »

Exercice 13 :

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 14 :

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.

Utiliser l'import : `from math import pi`

Exercice 15 :

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 16 :

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 17 :

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