

	Institut Supérieur Informatique	
	Types et Instructions de base	Prof. M. SY Periode: 2020 – 2021
	Classe(s): Licence 2 GL	Durée:

Exercice 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.

Exercice 2

Cet exercice a pour but de calculer l’aire d’une couronne (c’est-à-dire l’aire comprise entre 2 disques de même centre mais de rayons différents), et de travailler sur la notion d’hypothèse.

1. Donner une définition ainsi qu’un jeu de tests de la fonction *aire_disque* qui calcule l’aire $\pi \times r^2$ d’un disque de rayon r

Remarque : En python, la constante π est déjà définie dans le module *math*. Pour l’utiliser, il faut déclarer l’utilisation de ce module en tête du programme avec l’instruction suivante *import* :

2. Donner une définition ainsi qu’un jeu de tests de la fonction *aire_couronne* qui, étant donné deux nombres r_1 et r_2 , calcule l’aire de la couronne de rayon intérieur r_1 et de rayon extérieur r_2 .

Par hypothèse, on considère que le rayon intérieur est inférieur ou égal au rayon extérieur

Exercice 3

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 4:

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

1. triez et affichez la liste
2. ajoutez l’élément 12 à la liste et affichez la liste
3. renversez et affichez la liste
4. affichez l’indice de l’élément 17
5. enlevez l’élément 38 et affichez la liste
6. affichez la sous-liste du 2^e au 3^e élément
7. affichez la sous-liste du début au 2^e élément
8. affichez la sous-liste du 3^e élément à la fin de la liste
9. affichez la sous-liste complète de la liste
10. affichez le dernier élément en utilisant un indiciage négatif.

Exercice 5: (Comparaison des premier et dernier caractères d'une chaîne de caractères)

Écrire une fonction *first_last* d'argument *c* une chaîne de caractères ; qui retourne *True* si la chaîne est vide ou si son premier caractère coïncide avec son dernier caractère et *False* sinon.

Exercice 6:

Dans cet exercice, nous nous intéressons aux conversions de chaînes de caractères vers des listes et vice-versa.

1. Dans cette question nous proposons de convertir une liste de chaînes de caractères vers une chaîne de caractère.

Le principe est simplement d'effectuer une jonction (ou concaténation) des chaînes de la liste en les séparant avec un caractère séparateur.

2. Dans cette question, nous souhaitons convertir une chaîne de caractères en une liste de chaînes de caractères selon la spécification suivante :

Exercice 7:(Liste d'entiers rangés dans l'ordre croissant)

Écrire une fonction *is_increasing*

- d'argument *L* une liste (non vide) de nombres (de type *int* ou *float*)
- qui retourne *True* si les éléments de la liste *L* sont rangés dans l'ordre croissant (au sens strict) et *False* sinon.

Exercice 8:

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

1. les entiers de 0 à 3
2. les entiers de 4 à 7
3. 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