

Fonctions informatiques

Plan du T.P.

1	Fonctions mathématiques et fonctions informatiques	16
1.1	Analogies	16
1.2	Définition et syntaxe des fonctions informatiques	17
2	Utiliser une librairie de Python	18
2.1	Obtenir la fonction racine carrée dans Python	18
2.2	Distance entre deux points du plan	18
3	Algorithmes	18

[Retour au sommaire](#)

1 Fonctions mathématiques et fonctions informatiques

1.1 Analogies

Exercice 1 :

- Voici une fonction mathématique :

Soit f la fonction définie sur \mathbb{R} par : $f(x) = 2x$

- Voici la fonction informatique correspondante dans Python :

```
1 def f(x) :  
2     return 2*x
```

- a. Saisir cette fonction **dans l'éditeur**.
- b. Lancer la lecture du programme (bouton ▶).
- c. Vous pouvez maintenant utiliser votre fonction **depuis la console** :
- d. Appeler la fonction depuis la console pour obtenir les images par f de : $-5, 12, 133$.

```
1 >>> f(2)  
2 4
```

Fonction informatique recevant plusieurs arguments

Exercice 2 :

Nous souhaitons maintenant écrire une fonction qui, connaissant la base et la hauteur d'un triangle, renverra son aire.

- a. Compléter le programme suivant et le saisir dans l'éditeur :

```
1 def triangle(base, hauteur) :  
2     aire = ( ..... * ..... ) / 2  
3     return .....
```

- b. Enregistrer le programme (Ctrl -S), et lancer sa lecture (bouton ▶).
- c. Tester maintenant votre fonction en l'appelant depuis la **console**: `>>> triangle(5, 3)`, par exemple.

Une fonction de gain dans un jeu de hasard

Exercice 3 :

Un jeu de hasard propose de remporter 3.2 fois sa mise.

- a. Écrire dans l'**éditeur** une fonction `gain(mise)` :
 - prenant en argument la mise,
 - et renvoyant le gain que l'on peut espérer.
- b. Enregistrer le programme (Ctrl -S), et lancer sa lecture (bouton ▷).
- c. Tester maintenant votre fonction en l'appelant avec différentes valeurs depuis la **console**.

1.2 Définition et syntaxe des fonctions informatiques



Définition :

Une **fonction informatique** est un bloc d'instructions, qui porte un **nom**, et qui peut recevoir en entrée plusieurs **arguments**. Elle renvoie un **résultat** au moyen de l'instruction `return` en Python.

En Python

```
1 def nom_de_la_fonction(argument1, argument2, ...) :  
2     instructions  
3     return resultat
```

Remarques :

- Il faut veiller à placer les deux points en fin de première ligne,
- ces deux points introduisent le "corps" de la fonction, qui doit être **indenté** : décalé 4 caractères.
- La fin de l'indentation, ou l'instruction `return`, signalent à Python la fin de l'exécution de la fonction.
- Le résultat à renvoyer est déclaré à l'aide du mot-clef `return`.

Exercice 4 :

Le tarif d'entrée pour une pièce de théâtre est fixé à 24 euros par adulte, et 11 euros par enfant.

- a. Écrire une fonction `prix(adultes, enfants)` renvoyant le prix à payer en fonction du nombre d'adultes et d'enfants.
- b. Utiliser votre fonction pour déterminer le tarif pour un groupe composé de 4 adultes et 5 enfants.

Exercice 5 : fonction de conversion

Un bureau de change propose aujourd'hui le taux de change : 1 \$ = 0.87 €.

- a. Écrire une fonction `conversion(dollar)` :
 - prenant en argument le nombre de dollars à échanger,
 - et renvoyant le nombre d'euros obtenus en échange.,
- b. Utiliser votre fonction pour déterminer le nombre d'euros obtenus en échange de 250 dollars.
- c. Écrire en langage naturel une fonction réciproque permettant de convertir les euros en dollars.

2 Utiliser une librairie de Python

Python propose de très nombreuses librairies contenant, entre autres, des fonctions dont nous aurons besoin dans nos programmes. C'est le cas la fonction racine carrée.

2.1 Obtenir la fonction racine carrée dans Python

Cette dernière s'obtient dans Python en deux temps :

- importer, une fois pour toutes, en première ligne du programme, la fonction `sqrt` depuis la librairie `math` de Python (`sqrt` comme `square root` en anglais) :

```
1 from math import sqrt
```

- on peut alors utiliser la fonction `sqrt` dans le programme, aussi souvent que nécessaire :

```
1 from math import sqrt
2
3 variable_1 = sqrt(81)
4 variable_2 = sqrt(variable_1)
```

- Que contient `variable_2` à l'issue de ce programme ?

2.2 Distance entre deux points du plan

Exercice 6 : fonction distance entre deux points

Dans un repère orthonormé du plan, si $A(x_A, y_A)$ et $B(x_B, y_B)$ sont deux points, la longueur AB est donnée par la formule :

$$AB = \sqrt{(x_B - x_A)^2 + (y_B - y_A)^2}$$

Nous nous proposons dans cet exercice d'écrire une fonction permettant de calculer la distance entre deux points.

- Quelles variables fournir en arguments à notre fonction Python ?
- Écrire une fonction `distance(xA, yA, xB, yB)` renvoyant la distance entre $A(x_A, y_A)$ et $B(x_B, y_B)$.
- Utiliser cette fonction pour déterminer la distance entre $A(-3, 1)$ et $B(1, -2)$.
- Vérifier par un calcul mathématique sur papier.

3 Algorithmes

Un algorithme de cuisine

- mettre 300 grammes de farine dans un saladier,
- ajouter 3 œufs,
- ajouter 3 cuillères à soupe de sucre,
- ajouter 2 cuillères à soupe d'huile,
- ajouter 50 grammes de beurre,
- mélanger délicatement en ajoutant 600 ml de lait au fur et à mesure,
- verser une louche de pâte dans une poêle
- faire cuire à feu doux.



Définition :

Un **algorithme** ⁽¹⁾ est une suite finie d'opérations ou d'instructions permettant de résoudre un problème ou d'obtenir un résultat.

Exercice 7 : un algorithme utilisateur dans Python

Proposer un algorithme permettant d'écrire et d'appeler une fonction dans Python.

-
-
-

Indication :

celui-ci pourra comporter les trois actions suivantes, à ranger dans le bon ordre :

- cliquer sur le bouton exécuter,
- appeler la fonction dans la console,
- écrire la fonction dans l'éditeur.

(1). <https://fr.wikipedia.org/wiki/Algorithme>