


ayckinn

[Fiche publique](#)

[Mon profil](#)

► [Authentification](#)

[Déconnexion](#)

Langue : 

Plan du site

Progresser

- [Présentation](#)
- [Cours et problèmes](#)
- [Questions fréquentes](#)
- [Forum d'entraide](#)

Enseigner

- [Présentation](#)
- [Groupes et classes](#)

Concourir

- [Présentation](#)
- [Classement](#)
- [Épreuves de concours](#)
- [Résultats](#)

Olympiades

- [Présentation](#)
- [Sélection](#)
- [Résultats](#)
- [Autres olympiades](#)

L'association

- [Présentation](#)
- [Historique](#)
- [Stages d'entraînement](#)
- [Nous contacter](#)
- [Signaler une anomalie](#)

Problèmes résolus

[irotras](#) ²⁰²⁴ 22 h 54

Casernes de pompiers

[zodiack](#) ²⁰³³ 22 h 53

Le secret du Goma

[meems](#) ²⁰²² 22 h 53

Amitié entre gardes

C++

Fonctions : introduction

Cours



Parallèle avec les fonctions mathématiques

La notion de fonction utilisée en programmation et algorithmique se rapproche de celle des mathématiques sur certains points. En mathématiques, une fonction associe à un paramètre une valeur. À partir de cette association, une fois qu'elle est définie et posée, on peut faire une étude sur les diverses propriétés de la fonction.

Comme c'est souvent le cas avec les notions, en informatique, le principe de fonction va ressembler à celui des mathématiques. L'objectif n'est cependant pas de faire une étude, mais de fonctionner d'une certaine manière avec l'ordinateur.

Nom et paramètres

Voici l'exemple d'une fonction mathématique prenant un paramètre x :

$$f : x \mapsto 4x + 1$$

On peut aussi écrire l'égalité :

$$f(x) = 4x + 1$$

En programmation, on utilise généralement des noms longs, tout comme pour les variables, car c'est encore une occasion de mieux décrire ce que fait notre programme. Ensuite, une fonction gèrera en principe un nombre fixe de paramètres (ou *arguments*), qui peut être nul ou multiple. Au sein de la fonction, ces paramètres seront naturellement des variables (avec un nom les décrivant si possible).

Valeur de retour

En mathématiques, une fonction associe à ses paramètres une valeur. En programmation, cela pourra être le cas : on dit alors que la fonction retourne une valeur. Néanmoins, une fonction pourra également se contenter d'exécuter des instructions.

Écriture

[mathieuntrc](#) ²⁰²² 22 h 52
Mathématiques de base

[defalex](#) ²⁰²⁴ 22 h 52
Mot de passe du village

[decellasmarcof1](#) ²⁰²⁴ 22 h 52
Département d'architect...

[nad](#) ²⁰²² 22 h 52
Amitié entre gardes

[pilouliz](#) ²⁰²² 22 h 52
Bornes kilométriques

[manon0428](#) ²⁰²² 22 h 52
Jeu de dames

[celiandronnier](#) ²⁰²⁴ 22 h 52
Amitié entre gardes

[lisarncy26](#) ²⁰²² 22 h 52
Calcul des dénivelées

[sachagognet](#) ²⁰²² 22 h 51
Étape la plus longue

Dans la plupart des langages de programmation (généralement impératifs ; tous ceux du site sauf OCaml), la définition d'une fonction ne ressemble pas du tout à celle que l'on fait en mathématiques. En revanche, une requête à une fonction, dite « appel », en est très proche : par exemple `fonction(arg1, arg2, arg3)` pour une fonction à trois paramètres.

Cette écriture vous éveille certainement quelque chose ? Et oui : nous avons déjà fait appel à de nombreuses fonctions, notamment pour les opérations terminales de notre robot : afficher du texte, récupérer la saisie, se déplacer, bouger des objets.

Et la ligne `int main()` que vous avez dû recopier sans plus y réfléchir depuis toujours... En fait, *main* est une fonction, qui est appelée à l'exécution du programme. Dans ce chapitre, nous allons enfin écrire du code en dehors de ce bloc !

Allons-y !

Dans le chapitre, nous commencerons par manipuler des fonctions sans aucun argument et sans valeur de retour. Nous verrons ensuite que les fonctions peuvent être bien plus intéressantes lorsqu'elles sont paramétrables ; et enfin, nous écrirons des fonctions qui retournent une valeur.

Fonctions : introduction

Cours

