# APPRENDRE À PROGRAMMER EN RUST

# APPRENDRE À PROGRAMMER EN RUST

Guillaume Gomez

DUNOD

**NOUS NOUS ENGAGEONS EN FAVEUR DE L'ENVIRONNEMENT :**


Nos livres sont imprimés sur des papiers certifiés pour réduire notre impact sur l'environnement.

Le format de nos ouvrages est pensé afin d'optimiser l'utilisation du papier.

Depuis plus de 30 ans, nous imprimons 70 % de nos livres en France et 25 % en Europe et nous mettons tout en œuvre pour augmenter cet engagement auprès des imprimeurs français.

Nous limitons l'utilisation du plastique sur nos ouvrages (film sur les couvertures et les livres).


11 rue Paul Bert, 92240 Malakoff
www.dunod.com
ISBN 978-2-10-085285-7

# AVANT-PROPOS

Rust est un langage moderne et puissant qui est de plus en plus populaire auprès des développeurs, notamment en raison de ses performances élevées, de sa sécurité, de sa concurrence et de sa prise en charge des architectures multiplateformes. Cet ouvrage a pour objectif de vous aider à apprendre Rust.

Cependant, je tiens à souligner qu'il est préférable de connaître au moins un autre langage de programmation avant de plonger dans Rust. Ce n'est pas parce que Rust est un langage particulièrement difficile à apprendre, mais plutôt parce que les concepts abordés dans cet ouvrage supposent une certaine familiarité avec la programmation. En d'autres termes, si vous êtes totalement novice en programmation, vous pouvez trouver certains passages de cet ouvrage difficiles à suivre.

Si vous avez déjà une expérience de la programmation dans un autre langage (tel que C, C++, Java, Python, Javascript, Haskell…), vous aurez une longueur d'avance pour apprendre Rust. Vous serez plus en mesure de comprendre les concepts de programmation avancée, et vous pourrez vous concentrer sur les aspects spécifiques de Rust plutôt que de devoir assimiler à la fois les notions de base de la programmation et de Rust.

Néanmoins, même si vous n'avez pas d'expérience préalable en programmation, vous pouvez toujours vous plonger dans cet ouvrage avec enthousiasme et motivation. Vous y trouverez de nombreux exemples de code, d'explications claires et détaillées, ainsi que des astuces et des conseils pour vous aider à avancer. Vous apprendrez non seulement Rust, mais vous améliorerez également vos compétences en programmation en général.

# TABLE DES MATIÈRES



## Les bases de la programmation en Rust

## Les spécificités de Rust

## Pour aller plus loin

# INTRODUCTION

Le développement de Rust a été initié par Graydon Hoare en 2006, notamment dans le but de résoudre les failles de sécurité dans Firefox sans que cela impacte négativement les performances. Sa première version stable, la 1.0, est sortie le 15 mai 2015. En août 2020, Mozilla a arrêté de soutenir le développement du langage, conduisant à la création de la fondation Rust le 8 février 2021, le but de cette fondation n'étant pas de diriger le développement du langage mais de le soutenir financièrement.

Depuis sa première version stable, Rust a été adopté par toutes les plus grosses entreprises de l'informatique telle que Google qui s'en sert pour Android ainsi que son cloud, Microsoft qui s'en sert dans Windows, Amazon, Facebook, Discord, Huawei, Dropbox, Mozilla…

Du côté des projets open source, c'est devenu le troisième langage de programmation utilisé dans le développement du kernel Linux après le C et l'assembleur en 2022. Le projet GNOME a de plus en plus de projets internes utilisant Rust et a déjà réécrit certaines de ses bibliothèques telles que libsvg.

Rust est un langage de programmation système, compilé et multi-paradigme. C'est un croisement entre langage impératif (C), objet (C++), fonctionnel (Ocaml) et concurrent (Erlang). Il s'inspire des recherches en théorie des langages de ces dernières années afin d'atteindre trois objectifs : rapidité, sécurité (en gestion mémoire notamment) et concurrent (partage des données sécurisé entre tâches). Il est notamment utilisé pour de la programmation système, écrire des serveurs webs, faire des applications en ligne de commandes, des applications graphiques et des jeux vidéo.

Ses points forts sont :

- ✓ la gestion de « propriété » (**ownership**) des variables ;
- ✓ la gestion de la mémoire ;
- ✓ le typage statique ;
- ✓ l'inférence de type ;
- ✓ le filtrage par motif (**pattern matching**) ;
- ✓ la généricité.

En premier lieu, nous aborderons la syntaxe de Rust ainsi que ses types et les outils à utiliser pour se faciliter la vie.

Ensuite, nous verrons plus en détail les spécificités de ce langage et notamment les systèmes de « propriété » et de « durée de vie ».

Enfin, nous verrons les bases de concepts et fonctionnalités avancés telles que l'utilisation de bibliothèques écrites en C directement dans votre code Rust ou comment écrire des macros procédurales (proc-macro).

Avant de poursuivre, vous trouverez ci-dessous les principaux sites internet traitant de Rust, notamment son site web officiel, sa documentation ainsi que son dépôt sur github :

- ✓ le site internet : rust-lang.org
- ✓ la documentation officielle : doc.rust-lang.org/stable/std/
- ✓ le dépôt Github du langage : github.com/rust-lang/rust

Il est maintenant temps de commencer.