

Sessions garanties disponibles

Formation Rust

Réf. SF33515

Montez en compétences sur Rust, un langage de programmation sûr et ultra-rapide !

3 j Durée	2 100 € HT Inter / participant	5 550 € HT Intra (≤12 pers.)	25 Sessions à venir
--------------	-----------------------------------	---------------------------------	------------------------

Objectifs pédagogiques

- ✓ Comprendre les avantages et le positionnement de Rust
- ✓ Installer Rust et configurer l'environnement de développement
- ✓ Maîtriser la syntaxe et les concepts basiques de Rust
- ✓ Appliquer les concepts d'ownership et de borrowing
- ✓ Utiliser les structs et les enums pour organiser les données
- ✓ Gérer les erreurs en Rust
- ✓ Implémenter la généricité, les traits et les lifetimes
- ✓ Créer et utiliser des closures et des itérateurs

Public visé & prérequis

Public viséArchitectes logiciels, Développeurs

PrérequisDes connaissances en programmation et sur les systèmes Unix.

Programme détaillé

Introduction à la formation Rust

Présentation générale et objectifs de cette formation Rust
Principales caractéristiques du langage, avantages et positionnement
Fonctionnement de Rust (vitesse, erreurs, sûreté, multithreading) et paradigmes de programmation
Installer Rust et Cargo, le gestionnaire de packages pour gérer dépendances et compilation
Démarrer un nouveau projet : crates et structure de base
Outils de développement
Documentation

Activités pratiques :

Tour de table, échanges sur les différents contextes professionnels et les perspectives d'application des nouvelles compétences, installation de rustup et mise en place de différents outils de développement. Codage d'un premier programme en Rust via notre formation.

Syntaxe et concepts basiques de programmation Rust

Blocs de code, accolades, structures... découvrir la syntaxe de Rust
Variables, constantes et (im)mutabilité
Types primitifs : entiers, flottants, strings, tuples...
Expressions if, if else, boucles et boucles conditionnelles
Fonctions : déclaration, paramètres et retours
Définir un module
Tableaux et slices
Introduction aux macros
Commentaires

Activités pratiques :

Création de divers programmes en Rust (conversion de température, génération du nième nombre de la suite Fibonacci...) pour manipuler les concepts de base du langage.

Propriété (ownership) et emprunt (borrowing)

Principe de fonctionnement et intérêt de l'ownership : mémoire et allocation
Comprendre les règles de l'ownership
Transférer la propriété : les moves
Références et borrowing
Opérateurs & et & mut

Activités pratiques :

Résolution de problèmes de programmation pour comprendre et mettre en application les concepts de propriété, emprunt et les slices.
Echanges sur l'apport de tels principes dans les programmes Rust en termes de sécurité et de gestion de la mémoire, comparaison avec d'autres langages.

Types personnalisés : structures (structs) et énumérations (enums)

Utiliser les structs pour organiser la donnée : définition et instanciation
Méthodes et fonctions
Enumérations : Option et autres usages communs
Filtrage par motif (pattern matching) avec l'expression match
Déstructuration : tuples, énumérations, pointeurs...

Activités pratiques :

Écriture de programmes utilisant les structs et les enums, corrections de morceaux de code pour faire passer des tests (rustlings GitHub) et mise en œuvre de quelques patterns et anti-patterns de programmation Rust.

Gestion des erreurs

Les deux types d'erreurs sous Rust : recoverable et unrecoverable
Apports de l'approche Rust pour la gestion des erreurs
Utiliser la macro panic!
Result, principes de fonctionnement
La propagation d'erreurs
Gérer les erreurs dans main()

Activités pratiques :

Utilisation de la macro panic, résolution de problèmes de programmation avec l'enum Option, la méthode unwrap et les combinateurs, l'enum Result et ses alias, gestion de multiples types d'erreur avec try.

Généricité en Rust, traits et lifetimes

Principes de fonctionnement des traits dans Rust
Créer et implémenter un trait : Into, Drop, Iterator, Clone...
L'attribut derive
La surcharge d'opérateurs
Références et concept de durée de vie (lifetime)
Durées de vie statiques et associées
La généricité en Rust : types, fonctions, implémentations, restrictions...

Activités pratiques :

Implémentations de différents types de traits, surcharge de l'opérateur d'addition avec le trait Add, mise en œuvre de types et fonctionnalités génériques, utilisation de la clause where pour expliciter une restriction.

Closures et itérateurs

Les closures (aussi appelées lambdas) : caractéristiques et syntaxe
Closures passées en paramètres
Types anonymes
Fonctions passées en paramètres
Fonctions d'ordre supérieur (HOF)
Créer ses propres itérateurs avec le trait Iterator

Activités pratiques :

Capture de variables par référence, par valeur, utilisation des closures avec des outils de la bibliothèque standard (itérateurs any, find), refactoring et mise en œuvre de la méthode next.

Multi-threading

Créer des threads

Transfert de données entre thread : message passing

Les traits Sync et Send

Activités pratiques :

Création de threads avec spawn, communication entre threads et manipulation de concepts de programmation concurrente.

Quelques concepts de programmation avancée en Rust

Fonctionnement des macros

Contourner la sûreté de Rust avec les blocs unsafe

Interfaçage avec C/C++

Prochaines sessions

DATE	LIEU	STATUT
19/10/2026	à distance	Sur confirmation
19/10/2026	Grenoble	Sur confirmation
19/10/2026	Toulouse	Sur confirmation
19/10/2026	Aix-en-Provence	Sur confirmation
19/10/2026	Lyon	Sur confirmation
17/11/2026	Paris	Sur confirmation

Pourquoi choisir Sparks Formation ?

20 ans d'expertise IT, des formateurs experts et des sessions garanties contre l'annulation.

20 ans
d'expertise

1 068+
formations

11 178
apprenants formés/an

+67
Net Promoter Score

Avis des participants

4,9/5 — 27 avis

Christian M. — Cyril est top formateur, très bonne maîtrise de rust et formation de bon niveau.

Jonathan P. — Le formateur était très sympa, compétent, et didactique. 3 jours est un peu court pour la quantité et la difficulté de ce qu'il y a à apprendre, et...

Votre formateur

Emmanuel D. — Formateur C++ & C#

Emmanuel intervient sur des formations portant sur les langages et écosystèmes backend. Parmi les formations qui lui sont confiées figurent notamment « C# » et « C++ ». Son activité...

Ils nous font confiance

Cette année notre filière Langages & Écosystèmes Backend a accompagné plus de 36 entreprises dont :

RENAULT SAS THALES GLOBAL SERVICES THALES GLOBAL SERVICES SAS ALSTOM ANSYS FRANCE SAS IT LINK FRANCE SII ILE DE FRANCE
STMICROELECTRONICS ACTIA ENGINEERING SERVICES AIRBUS FRANCE +26 autres

Financement & certifications

Cette formation est éligible aux financements OPCO et FNE-Formation. Nos équipes vous accompagnent dans le montage de votre dossier.



Réservez votre place dès maintenant

20 ans d'expertise · 1 000+ formations · NPS +67 · réponse sous 24h ouvrées

0 805 950 800

demande@sparks-formation.com