



FORMATION EN PRESENTIEL ET/OU EN DISTANCIEL (SELON LE THEME)

RUST

Dernière mise à jour le 10/10/2023.

OBJECTIF

Comprendre les notions fondamentales du langage RUST

COMPÉTENCES DÉVELOPPÉES

- Savoir créer des applications évolutives et performantes.
- Savoir concevoir la puissance de Rust
- Développer ce langage

PUBLIC CONCERNÉ

Développeurs et architectes

MODALITÉS ET DÉLAIS D'ACCÈS

Vous pouvez vous inscrire pour suivre une de nos formations jusqu'à la veille de la date de démarrage si la formation est financée directement par votre entreprise ET si le nombre maximum de participants n'est pas atteint. Si la formation est financée via un OPCO, vous devez au préalable avoir obtenu un accord de ce dernier.

MODALITÉS DE DÉROULEMENT DE L'ACTION DE FORMATION

Formation présentielle ou distancielle dispensée par un formateur expérimenté. La formation alterne des exposés théoriques, des démonstrations et la mise en pratique au travers d'exercices et de cas concrets.

Tarif

1850 € HT
par participant

Réf.

RST

Contact

0465260114
nathalie.husson@univlearn.fr
<https://univlearn.fr/>

PRÉ-REQUIS

- Assurer d'une connaissance de base dans un langage de développement.

ACCESSIBILITÉ AUX PERSONNES EN SITUATION DE HANDICAP

Nos locaux sont accessibles aux personnes à mobilité Réduite PMR. De plus, nos conseillers sont disponibles pour vous accompagner dans vos démarches à travers nos partenaires. Nous sommes en mesure de mobiliser les expertises, les outils nécessaires pour vous accueillir, vous accompagner et vous former.



Niveau

Autres formations
professionnelles continues



Modalité

Présentiel



Effectif par session

2 mini > 6 maxi



Durée

21 heures / 3 jour(s)



PROGRAMME

Module 1 - Introduction

- Débuter avec Rust
- Les avantages
- Installation
- Premier programme
- Outils de développement
- La librairie standard

Module 2 - Variables et Types

- Utilisation de variables et de types
- Commentaires
- Constantes globales
- Types primitifs
- Liaison de variables à des valeurs
- Portée des variables
- Type checking et conversions
- Expressions
- La pile et le tas (stack et heap)

Module 3 - Instructions et fonctions

- Instruction conditionnelle
- Iteration
- Fonctions
- Attributes (annotations)
- Tests

Module 4 - Structures de données

- Strings
- Tableaux, vectors et slices
- Tuples
- Structures
- Enumérations
- Matching patterns

Module 5 - Fonctions avancées et gestion d'erreur

- Notions avancées sur les fonctions



- La récursivité
- Les closures
- Les Iterators
- Consumers et adapters
- Généricité
- Gestion d'erreur

Module 6 - Programmation orientée objet

- Association fonctions et structures
- Méthode
- Constructeur
- Builder
- Méthodes avec des tuples et des enums
- Traits
- Static and dynamic dispatch
- Surchage
- POO
- Héritage
- Visitor pattern

Module 7 - Gestion de la mémoire

- Pointeurs et references
- Propriété (ownership) et référence
- Boxes
- Compteur de reference
- Notion de pointeurs

Module 8 - Organisation du Code et Macros

- Organisation
- Macros
- Modules et Crates

Module 9 - Programmation concurrente

- Coder pour plusieurs cœurs
- Threads
- Accés concurents
- Espaces partagés
- Communication avec des channels

Module 10 - La Standard Library



- Exploration
- Le module prelude
- Collections (hashmaps et hashsets)
- Manipulation de fichiers

INTERVENANT(S)

Formateur - Consultant expérimenté

ÉVALUATION

Chaque module de cours est concrétisé par un TP afin de permettre l'acquisition d'un vrai savoir-faire sur tous les points abordés, Questions posées par le formateur tout au long de la formation à l'oral ou à travers un QCM

ATTRIBUTION FINALE

Attestation Individuelle de Formation