

## Formation Rust

<b>Durée :</b>	3 jours
<b>Public :</b>	Développeurs
<b>Pré-requis :</b>	Notions de programmation
<b>Objectifs :</b>	Connaître l'intérêt et l'utilisation du langage Rust - Etre capable de développer, compiler et tester une application en Rust
<b>Sanction :</b>	Attestation de fin de stage mentionnant le résultat des acquis
<b>Taux de retour à l'emploi:</b>	Aucune donnée disponible
<b>Référence:</b>	PRO101839-F
<b>Note de satisfaction des participants:</b>	Pas de données disponibles

### Découvrir Rust

Rust : présentation du langage, styles supportés  
Usages : programmation système, applications en ligne de commande, services réseaux, ...  
Apports de Rust par rapport au C# ou Java  
Compilateur, packages d'installation  
Détails du système de compilation Cargo  
IDEs utilisables ([rust-lang.org/tools](http://rust-lang.org/tools)), documentation (the book)

**Atelier : Installation de l'environnement de développement (rustup, cargo, rustc, crates.io), structure d'une programme**

### Maîtriser les bases

Types de bases  
Types référence  
Tableaux et vecteurs  
Manipulation de chaînes de caractères  
Ownership  
Structures de contrôles : conditions et boucles  
Manipulation de références  
Ecriture de fonctions et appels  
Surcharge d'opérateurs

**Atelier : Multiples fonctions et passage de paramètres**

### Gérer les erreurs

Gestion des erreurs, propagation  
Capture des exceptions  
Manipulation de types et écrire de type personnalisé

### **Atelier : Gestion des erreurs dans un programme**

#### **Utiliser des outils**

Construction de profils (Crates)  
Utilisation de Modules  
Gestion des chemins  
Construction de blocs (Items)  
Documentation du code  
Gestion des dépendances  
Publication (crates.io)  
Tests unitaires et tests d'intégration

### **Atelier : Manipulation de workspaces, gestion des dépendances, documentation et tests du code**

#### **Manipuler des objets et implémenter la généricité**

Concepts de la POO supportés par Rust  
Utilisation des Structures (struct)  
Attributs et méthodes  
Structures génériques  
Mutabilité interne  
Ecriture d'énumérations  
Traits  
Itérateurs sur des objets ou des chaînes  
Utiliser des collections d'objets : Vect, VecDeque, LinkedList, BinaryHeap, HashMap

### **Atelier : Manipulation de plusieurs structures de données**

#### **Gérer les entrées-sorties**

Lecture et écriture de fichiers et répertoires  
Sérialisation d'objets  
Compression de flux  
Gestion des chemins  
Fonctions spécifiques au système de fichiers  
Mise en réseau

### **Atelier : Ecriture et lecture de plusieurs types de fichiers**

#### **Implémenter la concurrence**

Bases de la programmation concurrente  
Parallélisme et fonctions associées (fork-join, spawn & join)  
Partage de ressources, variables globales atomiques  
Pipeline, Channel

Verrous de synchronisation : Mutex, deadlock, ...  
RwLock

**Atelier : Implémentation de la concurrence avec Rust**