

## Introduction à la programmation fonctionnelle en Haskell

### DESCRIPTION

Au cours des dernières années, la programmation fonctionnelle est devenue un paradigme incontournable. Un nombre croissant de langages de programmation intègrent des “capacités” fonctionnelles (fermetures, composition de fonction par exemple), et certains framework, tels que rx, sont entièrement fondés sur ce modèle.

Les enjeux sont importants. En effet, l'approche fonctionnelle permet d'écrire un code à la fois plus sûr, plus clair et plus modulaire.

Cette formation d'initiation à la programmation fonctionnelle, vous propose de vous familiariser avec ses concepts clés, tels que transparence référentielle (fonctions pures, immutabilité, fonctions d'ordre supérieur (Higher order functions), fermetures, types de données ou encore pattern Matching. Elle vous permettra également d'apprendre les bases de Haskell, langage-étendard de la programmation fonctionnelle.

### OBJECTIFS PEDAGOGIQUES

- Découvrir le paradigme de la programmation fonctionnelle (FP : Functional Programming)
- Reconnaître des patterns de programmation fonctionnelle et les mettre en œuvre
- Se lancer dans l'apprentissage du langage Haskell
- Enrichir sa culture de programmation avec des concepts de programmation fonctionnelle utilisables dans tous les langages

### PUBLIC CIBLE

Développeurs de tous niveaux, sans expérience de la programmation fonctionnelle.

### PRE-REQUIS

Expérience de base du développement de logiciel.

#### Stage pratique

Qualité du logiciel - Software  
Craftsmanship

Code :

**HASKE**

Durée :

**2 jour(s) (14,00 heures)**

Exposés : **20 %**

Cas pratiques : **60 %**

Echanges d'expérience : **20 %**

#### Inter-entreprises :

Prochaines sessions  
disponibles [sur notre site web](#).

Tarif : 1 650,00 € HT /  
participant

#### Intra-entreprise :

Tarifs et dates sur demande.

### **METHODE PEDAGOGIQUE**

Formation avec apports théoriques, échanges sur les contextes des participants et retours d'expérience pratique du formateur, complétés de travaux pratiques et de mises en situation.

### **PROFIL DES INTERVENANTS**

### **MODALITÉS D'ÉVALUATION ET FORMALISATION À L'ISSUE DE LA FORMATION**

L'évaluation des acquis se fait tout au long de la session au travers des ateliers et des mises en pratique. Une évaluation à chaud sur la satisfaction des stagiaires est réalisée systématiquement en fin de session et une attestation de formation est délivrée aux participants mentionnant les objectifs de la formation, la nature, le programme et la durée de l'action de formation ainsi que la formalisation des acquis.

### **PROGRAMME PEDAGOGIQUE DETAILLE**

#### **Jour 1**

#### **GUIDES DE SURVIE**

- Ghci
- Stack

#### **NOTATION DES TYPES EN HASKELL**

#### **FONCTIONS**

- Définition générale
- Fonction pure
- Effets de bord
- Transparence référentielle

#### **IMMUTABILITÉ**

#### **COMPOSITION DE FONCTIONS**

#### **FONCTIONS À L'ORDRE SUPÉRIEUR**

#### **TYPES**

- Définition générale
- Garantie d'expressions valides
- Outil de conception

**TYPES ALGÈBRIQUES**

- Types sommes
- Types produits

**WRAP UP DE FIN DE JOURNÉE**

**Jour 2**

**PATTERN MATCHING**

- Cas d'usage
- Cases
- Guards
- Totalité des fonctions

**TYPES PARAMÉTRÉS**

- Richesse sémantique
- Type Maybe
- Polymorphisme
- Foncteur
- Enchaîner des fonctions dans un foncteur
- Lier des fonctions via bind

**BILAN ET CLÔTURE DE LA FORMATION**

