DataTips

Formation

SCALA

DataTipsLearning.com

Tarif :
1200 € HT

Durée :
2 Jours
(14h)

Lieu :
PARIS/
Distancielle/
Votre Local

Contact :

contact@datatipslearning.com
+33 0751386021
Réf. DTPSL-SCALA-2022
S'inscrire

Plus de détails :

Objectifs
Public / Prérequis
Programme détaillé
Méthode pédagogique

Scala est un langage JVM type-safe qui intègre des aspects de programmation orientée objet (POO) et de programmation fonctionnelle (FP). Scala est né en 2003 à l'EPFL en Suisse par Martin Odersky pour fournir un environnement haute performance, son nom est une abréviation du terme SCAlable LANguage.

Scala est compilé en bytecode Java, et il est possible d'utiliser les bibliothèques Java. Scala est utilisé pour alimenter des sites plus fréquentés, notamment Twitter, Netflix.

Objectifs

- Décrire les liens entre Scala, Java et la JVM.
- Reconnaître la base du langage Scala.
- Programmer objet en Scala.
- Les concepts de la programmation fonctionnelle.
- Utiliser le Pattern Matching.
- Les aspects avancés de la programmation en Scala.
- Écrire des fonctions d'ordre supérieur.
- Apprendre à utiliser les structures de données.
- Apprendre à mieux gérer les erreurs.
- Comprendre les implicites.
- Connaitre comment gérer la concurrence avec les Futures.
- Apprendre à utiliser ScalaTest & ScalaCheck.

Public

- Data Engineer
- Data Architectes
- Data Scientist
- Data Analyst

Prérequis

- Pas de connaissances particulières.

Programme détaillé

I. Introduction à Scala
    1. Présentation de Scala
    2. Installation de Scala
    3. Utilisation de SBT
    4. Exécution des outils de ligne de commande Scala
    5. Exécution de Scala REPL dans les IDE
    6. Immuabilité de référence vs de valeur

II. Les bases de Scala

1. Les concepts de la programmation fonctionnelle
2. Déclaration de valeurs et de variables
3. Les types
4. Les méthodes et fonctions
5. If Else
6. Opérateurs conditionnels
7. Les boucles
8. Utilisation de Lazy val
9. Enumérations
10. Chaînes interpolées
11. Packages et imports

III. Pattern Matching
1. Un simple match
2. Valeurs, variables et types dans les matches
3. Matching avec des collections
4. Matching avec les case classe
5. Matching avec les arguments
6. Matching avec les Options
7. Matching avec les expressions régulières
8. Types de matching
9. Sealed et matching exhaustives
10. Guards
11. Extracteurs
12. Notation Infixe dans les clauses case
13. Matching des structures imbriquées
14. Fonctions partielles

IV. Les fonctions
1. Les procédures
2. Les fonctions
3. Les Fonctions récursives
4. Les Fonctions imbriquées
5. Les appels des fonctions
6. Les valeurs par défaut
7. Paramètres vararg
8. Groupes de paramètres
9. Paramètres Type
10. Méthodes et opérateurs
11. Les fonctions comme valeurs
12. Fonctions anonymes
13. Paramètre de type fonction
14. Inférence de paramètres
15. Utilité des fonctions Higher-Order
16. Closures
17. Currying
18. Contrôler les abstractions
19. Le retour d'une fonction

V. Implicites
   1. Arguments implicites
   2. Scénarios pour des arguments implicites
   3. Conversions implicites
   4. Le pattern Type Class
   5. Problèmes techniques avec implicites
   6. Règles de résolution des implicites
   7. Implicites intégrés de Scala

VI. Programmation fonctionnelle en Scala
   1. Présentation de la programmation fonctionnelle
   2. Programmation fonctionnelle en Scala
   3. Récursivité tail recursion
   4. Partially Applied Functions vs fonctions partielles
   5. Currying et autres transformations sur les fonctions
   6. Structures de données fonctionnelles
   7. Traversing, Mapping, Filtering, Folding, & Reducing
   8. Left vs Right Traversals
   9. For Comprehensions
   10. Polymorphisme et fonctions d'ordre supérieur

VII. Programmation orientée objet en Scala
   1. Notions de base des classes et des objets
   2. Types de référence et types de valeur
   3. Value Classes
   4. Parent Types
   5. Constructeurs dans Scala
   6. Fields dans les classes
   7. Propriétés avec Getters et Setters
   8. Validation des inputs
   9. Appel des constructeurs de classe parent
   10. Nested Types
   11. Nested Classes
   12. Singletons
   13. Objets compagnons
   14. Objets étend classe & trait
   15. La méthode apply
   16. Héritage Scala

VIII. Traits
   1. Héritage multiple
   2. Interfaces en Java 8
   3. Utiliser les traits comme des mixins
   4. Stackable Traits
   5. Construire des traits
   6. Choix des classes ou des traits
   7. Implémentations concrètes
   8. Overriding dans les traits

9. Initialisation des champs des fields
10. Classe étend des traits
11. Self Types

IX. Le système d'objets Scala
1. Parameterized Types
2. Hiérarchie des types Scala
3. Nothing et Null
4. Products, Case Classes et Tuples
5. Predef Object
6. Égalité des objets
7. Overriding les membres des classes et des traits
8. Linéarisation de la hiérarchie d'un objet

X. Les collections
1. Les principaux traits des collections
2. Collection générique, mutable, immuable, Concurrente et parallèle
3. Choix de collection
4. Les Arrays
5. Les Map
6. Les Tuples
7. Les Séquences
8. Les Listes
9. Les Sets
10. Zipping
11. Des algorithmes communs
12. Mapping une fonction
13. Reducing, Folding, & Scanning
14. Les itérateurs
15. Les streams
16. Les collectes parallèles

XI. Règles de visibilité
1. Mots clés de visibilité
2. Visibilité publique
3. Visibilité protégée
4. Visibilité privée
5. Visibilité privée et protégée délimitée

XII. Gestion des erreurs
1. Try Catch
2. Option
3. Either
4. Try Succes Failure

XIII. Programmation fonctionnelle avancée
1. Types de données algébriques

2. Théorie des catégories
3. Variance
4. Monade
5. Option
6. Functor

XIV. Concurrence avec les Futures
XV. ScalaTest & ScalaCheck

Méthode pédagogique

La formation se compose d'une partie théorique, et également une partie pratique représentant 60% de de la formation.
La partie pratique contient plusieurs exercices sous forme de notebook Databricks avec les corrections, avec aussi un projet à la fin de la formation comme simulation d'une prod.

Chaque jour, une évaluation rapide des connaissances est effectuée avant de commencer les nouvelles parties de la formation.
A la fin, une synthèse globale est délivré aux stagiaires, renforcé par un projet prod.
Finalement, une évaluation QCM est proposée.

Un support de cours sera remis à chaque stagiaire comprenant les slides, les exercices et les corrigés et un git du projet prod.
Une feuille de présence est fournie en fin de formation avec une certificat de complétion de formation pour chaque stagiaire.

Le formateur est un Data Engineer expert, qui intervient sur le sujet depuis plusieurs années en formation mais aussi en conseil.