

# Ibrahima Khalilou Lahi Samb

## I. [Tout savoir sur le langage de programmation Java et son environnement](#)

- **JAVA** : Inspiré du C++, java est un langage de programmation orienté objet initialement développé par **Sun Microsystems** puis acquis par **Oracle en 2009**.  
Il fait partie de la famille des langages fortement typé autrement dit chaque variable déclarée avec java doit avoir un type bien déterminé.  
Java est utilisé principalement utilisé pour le développement d'applications compatibles avec tous types de système, web (coté serveur) et des applis mobiles.
- **JDK** : Il correspond à l'environnement de développement logiciel utilisé pour développer des applications java. Il se compose du **JRE, un interpréteur, un compilateur, un archiveur** ainsi que tous les outils nécessaires pour développer en java.
- **JRE** : **Java Runtime Environment** fournit une configuration minimale requise pour l'exécution d'une application Java.
- **JVM** : La compilation de fichier java en **bytecode** nous donne un autre fichier avec l'extension **.class** qui pour son execution a besoin de la **Java Virtual Machine** qui est une machine virtuelle avec comme but d'exécuter le bytecode.

## II. [Découvrir Scala](#)

Scala est un langage de programmation multi-paradigme intégrant les paradigmes tels que la **POO** (programmation orientée objet) et la **PF** (programmation fonctionnelle) offrant ainsi au développeur la possibilité de s'orienter vers le paradigme de son choix.

Appelé aussi langage JVM lié notamment au fait que lorsqu'on compile du code Scala, ce dernier est compilé en bytecode java pour ensuite être exécuté par la JVM.

De nos jours, le langage Scala est principalement utilisé dans les applications Big Data grâce notamment à un de ses Framework Apache Spark

## III. [En savoir plus sur Apache Spark](#)

Apache Spark est un Framework ou système de traitement et de calcul distribué développé par Apache Software Foundation.

Ce Framework simplifie le développement et permet d'optimiser l'efficacité des travaux d'analyses de données. La puissance de Spark réside dans son traitement en mémoire qui utilise un pool distribué de nœuds à forte densité de mémoire.

Grace à ce traitement en mémoire, Spark effectue ces tâches 100 fois plus rapidement que les autres Framework qui le font sur disque.

# Ibrahima Khalilou Lahi Samb

**Hadoop** est le rival le plus connu de Spark, mais ce dernier évolue d'une manière beaucoup plus rapide et est plus utilisé par les entreprises du fait de sa rapidité, sa simplicité et le fait qu'il prend en charge de nombreux API Python et R.

Vous trouverez plus de détails sur la comparaison **Hadoop vs Spark** ici [Spark vs Hadoop](#)

## IV. [Les différents cas d'utilisation du Big data](#)

Le Big data correspond à l'ensemble des stratégies et technologies utilisées pour rassembler, organiser et traité de grandes volume de données.

Il peut avoir plusieurs cas d'usage tel que :

- La recommandation de produit
- L'amélioration de l'expérience client
- La création de modèle de Machine Learning

## V. [Les différents cas d'utilisation de Scala](#)

Grace à sa facilité d'utilisation et à sa rapidité, Scala est utilisé dans de nombreux domaines dont ;

- **Data processing**
- **Distribued computing**
- **Web development**

## VI. [Produits célèbres développés avec Scala](#)

- **Akka** : « [Akka](#) est une boîte à outils et un environnement d'exécution permettant de créer des applications hautement concurrentes, distribuées et tolérantes aux pannes sur la JVM. Akka est écrit en [Scala](#) , avec des liaisons de langage fournies à la fois pour Scala et Java. »
- **Play Framework** : c'est une technologie basée sur une architecture Web légère, sans état et conviviale qui tire parti du Framework Akka, [Play](#) fournit une consommation de ressource prévisible et minimale pour écrire une application Web hautement évolutive.
- **Lagom** : Il s'agit d'un Framework Scala pour la création de micro services. C'est une plate-forme qui vous fournit un ensemble complet d'outils pour créer, gérer et surveiller des micro services.

## VII. [La performance chez d'autres langages](#)

Coté performance, Scala cohabite avec d'autres langages qui ont également fait leurs preuves notamment **Python**. Ce dernier malgré sa présence dans de nombreux domaines, python présente toutefois des limites notamment s'il s'agit de construire des systèmes distribués fiables et performant.

Parmi ses limites on peut avoir :

# Ibrahima Khalilou Lahi Samb

- **Membres faiblement ou dynamiquement typés** : Le typage dynamique avec python peut à un moment donner constituer un goulot d'étranglement dû au coût d'interprétation de ces membres tapés dynamiquement.
- **Lenteur** : Python peut être considéré comme relativement lent par rapport à d'autres langages de programmation qui fonctionnent plus prêt du système d'exploitation.  
Selon **Nathan Murthy**, « *Les limites de performance de Python le rendent mal adapté aux systèmes de traitement en temps réel ou aux systèmes basés sur les flux qui déplacent ou manipulent de grands volumes de données sur plusieurs machines avec une faible latence, ce qui est pratiquement tautologique pour les ingénieurs construisant des systèmes distribués à grande vitesse* »
- **Système de construction désorganisé** : Selon toujours **Nathan Murthy**, « *Juste pour faire une simple déclaration d'importation d'une ligne, numpy nécessite 285 Mo à 529 Mo de code de soutien* ». Les outils d'installation des dépendances python placent ces fichiers de manière dispersé dans le système de fichier.

Pour ne citer que ceux-là, python est néanmoins très efficaces pour les petits programmes rapides et sales qui exécutent des taches séquentielles.