

Formation Java SE 8 à 12

■ Durée :	3 jours (21 heures)
■ Tarifs inter-entreprise :	1 975,00 CHF HT (standard) 1 580,00 CHF HT (remisé)
■ Public :	Développeurs Java
■ Pré-requis :	Connaissances en programmation orientée objet Java
■ Objectifs :	Maîtriser les évolutions apportées par les nouvelles versions de Java - Pouvoir migrer du code de la version 8 à 12
■ Modalités pédagogiques, techniques et d'encadrement :	<ul style="list-style-type: none">• Formation synchrone en présentiel et distanciel.• Méthodologie basée sur l'Active Learning : 75 % de pratique minimum.• Un PC par participant en présentiel, possibilité de mettre à disposition en bureau à distance un PC et l'environnement adéquat.• Un formateur expert.
■ Modalités d'évaluation :	<ul style="list-style-type: none">• Définition des besoins et attentes des apprenants en amont de la formation.• Auto-positionnement à l'entrée et la sortie de la formation.• Suivi continu par les formateurs durant les ateliers pratiques.• Évaluation à chaud de l'adéquation au besoin professionnel des apprenants le dernier jour de formation.
■ Sanction :	Attestation de fin de formation mentionnant le résultat des acquis
■ Référence :	JAV100684-F
■ Note de satisfaction des participants:	4,54 / 5
■ Contacts :	commercial@dawan.fr - 09 72 37 73 73

■ Modalités d'accès :	Possibilité de faire un devis en ligne (www.dawan.fr, moncompteformation.gouv.fr, maformation.fr, etc.) ou en appelant au standard.
■ Délais d'accès :	Variable selon le type de financement.
■ Accessibilité :	Si vous êtes en situation de handicap, nous sommes en mesure de vous accueillir, n'hésitez pas à nous contacter à referenthandicap@dawan.fr, nous étudierons ensemble vos besoins

Généralités sur la fréquence de release et le support des versions post Java 8

Support des différents JDK (8 à 11) par Oracle et la communauté autour d'OpenJDK

Rappels des principaux apports de Java 8

Disparition du PermGen Space au profit du Metaspace

API Time

Expressions lambda

Méthodes default dans une interface

Interfaces fonctionnelles

Les Streams

Moteur de scripts Nashorn

Découvrir les nouveautés de Java 9

Modularité du JDK (JEP 200 - Jigsaw/JSR 376)

Inférence du type de variable (JEP 193)

Amélioration de l'usage des collections (JEP 269)

Nouveautés de l'API des processus (JEP 202)

Gestion de la dépréciation (JEP 277)

Amélioration sur les threads : spin-wait (JEP 285)

Variables et méthodes Handle (JEP 193, 274) Support de HTTP2 et des WebSockets (JEP 110)

Moteur JavaScript : ES 6 (JEP 236, 292)

Améliorations de Javadoc pour le support de HTML5 (JEP 224, 225, 254)

Support des images : multi-résolution (JEP 251), formats (JEP 262)

API d'accès aux spécifications du bureau de l'OS (JEP 272)

Améliorations apportées à JavaFx

Unification des logs (JEP 158, 264, 271)

Utiliser les améliorations de Java 10

Inférence du type de variable (JEP 286)

Application Class-Data Sharing (JEP 310)

Versioning de release (JEP 322)

Amélioration liées au Garbage Collector (JEP 307, JEP 304)

JEP 319: Root Certificates

Nouvelles méthodes dans la classe Collectors

Nouvelle méthode dans la classe Optional

Nouvelles méthodes dans la classe Objects

Découvrir les nouveautés de Java 11

Commercialisation du JDK 11 par Oracle

Conséquences du Long Term Support d'Oracle

JDK11 d'Oracle vs Open JDK 11

Utilisation de var dans les lambdas

Standardisation du nouveau client HTTP

Exécution directe d'un fichier source

Suppression des modules JavaFX, Java EE et CORBA

Découvrir les nouveautés de Java 12

Collecteurs : Teeing Collector Améliorations sur les chaînes de caractères Comparaison des fichiers Formatage compact des nombres Adoption de l'Unicode 11 Appeçu des fonctionnalités de prévisualisation Fonctions et API supprimées : Java6, méthodes de finalisation,...

Migration de code de Java 8 à 12