

## Table des matières

Qu'est-ce qu'un JDK ?.....	6
Téléchargez un JDK.....	6
Téléchargez et installez Eclipse .....	7
En résumé .....	7

*Source : Open Classroom*

# Démarrez votre projet avec Java

---

Avant de commencer par du concret, nous allons débiter ce chapitre par une brève introduction **historique et théorique sur Java**. Je vous rassure, **rien de trop long** 😊.

## Un peu d'histoire autour de Java

Le Java est une technologie développée à la base par **Sun Microsystems** dont les débuts datent de **1990**. Ses deux créateurs, **James Gosling et Patrick Naughton**, voulaient apporter deux grands principes avec ce nouveau langage :

- des applications pouvant fonctionner sur **n'importe quel matériel** (hardware),
- un langage **beaucoup plus simple** à appréhender que le C++.

Contrairement au C++, avec Java les développeurs n'ont plus besoin de **gérer la mémoire de leurs programmes**. Cela facilite l'apprentissage et évite de nombreuses erreurs.

Java doit son succès principalement au fait qu'il a grandi en même temps que **l'arrivée d'internet**. De nombreux outils intégrés dans Java permettent de **développer facilement des projets web**.

Dans la fin des années 90 et au début des années 2000, une technologie Java est très présente sur le Web : **les applets Java**. Ce sont des applications Java, exécutées **directement dans le navigateur**, qui permettent d'avoir beaucoup plus de fonctionnalités qu'avec les versions d'HTML , CSS et JavaScript disponibles à l'époque.

Aujourd'hui, les applets ne sont plus vraiment utilisées. De nouveaux outils, s'appuyant également sur Java, ont pris le dessus et sont très présents sur l'ensemble des infrastructures informatiques et web. Depuis **2009** c'est l'entreprise **Oracle** qui est en charge du développement de Java.

On retrouve Java côté **Backend** dans de nombreux secteurs, tels que **le big data, le web, le mobile ou encore dans les serveurs de jeux en ligne**.

En développement, on distingue le développement **Frontend** et le développement **Backend**.

- **Le développement Frontend** regroupe tout ce qui est **visible** par l'utilisateur. Les boutons, les animations, les textes affichés à l'écran, etc. Dans ce contexte, **le design et l'expérience utilisateur** sont très importants.
- **Le développement Backend** regroupe les traitements effectués **"en coulisse"**, ceux que l'utilisateur ne voit pas forcément. Par exemple le mécanisme de gestion des amis dans un réseau social.

## Pourquoi apprendre Java ?

Java est aujourd'hui le langage **le plus répandu** dans les entreprises selon l'index [TIOBE](#), cela est dû à :

- son **grand succès** dans les années 90 ;
- les outils très **riches, variés et matures** dont ils disposent ;
- sa très **grande communauté**, qui permet de facilement trouver de l'aide si besoin ;
- sa **facilité** de prise en main.

Java, depuis plusieurs années, est largement **en tête des offres d'emplois** dans le secteur informatique.

Le salaire moyen pour **un développeur Java junior** se situe entre **35 000€ et 42 000€** selon le cabinet de recrutement [Urban Linker](#) . Maîtriser ce langage, c'est l'assurance **d'obtenir facilement un travail, et ce dans beaucoup de secteurs différents** (santé, banque, publicité, assurance, etc.).

Java est un langage destiné à créer des applications **riches, robustes, maintenables dans le temps et développées par de grosses équipes**. C'est un langage à **taille d'entreprise**. Le Java n'est **pas destiné** à être utilisé pour de **petites applications** ou de **simples scripts** exécutant une tâche.

Quand on parle de Java, on ne parle pas seulement du langage mais aussi de tout **l'écosystème** qui va autour. Des **centaines d'outils** associés à Java en font l'un des **3 grands piliers** du **Backend web**. Parmi ces piliers on trouve :

- l'écosystème **Java** ;
- les technologies Microsoft **.NET** ;
- le **reste** : PHP, NodeJS, Ruby, etc.

## Le fonctionnement de Java

En développement logiciel, il existe différents types de langages : **compilés, interprétés** ou **intermédiaires**.

C'est la façon dont votre ordinateur va **comprendre et exécuter le code** qui va déterminer la classification d'un langage dans une catégorie ou dans une autre.

Quand vous développez un programme, vous allez écrire un langage qui est **humainement compréhensible**. Votre ordinateur **ne comprend pas directement** le code que vous écrivez.

## Langage compilé

L'étape de compilation, c'est l'étape durant laquelle le code que vous avez écrit **va être transformé en code compréhensible par votre ordinateur** (plus exactement votre processeur). Cette étape se fait **une seule et unique fois**. Le logiciel qui va effectuer cette transformation s'appelle un **compilateur**.

Comme langage compilé, il y a par exemple **le C, le C++** ou encore **le Cobol**.

Quand un programme est compilé sur un ordinateur, il ne va **pas forcément fonctionner sur un autre ordinateur** (si le système d'exploitation diffère par exemple). Il faudra le **re-compiler** sur cet ordinateur pour qu'il fonctionne. Ce processus est parfois long et peut prendre plusieurs dizaines de minutes.

## Langage interprété

Dans les langages interprétés, un nouveau type de logiciel vient remplacer les compilateurs, il s'agit d'un **interpréteur**. C'est un logiciel qui va **lire votre code ligne par ligne et l'exécuter au fur et à mesure**. Cette étape se fait à chaque fois que le programme est démarré.

Ce procédé impacte forcément les performances puisqu'il ajoute une **étape supplémentaire** pendant l'exécution du programme. Le gros avantage est que le code peut fonctionner sur **n'importe quel système sans devoir le re-compiler**.

**JavaScript, Python et PHP** sont des langages interprétés.

Aujourd'hui les interpréteurs sont **optimisés** et vont effectuer une étape de "**compilation**" à la **première lecture** d'une ligne de code pour éviter de devoir relire la ligne à chaque fois.

## Langage intermédiaire

Les langages intermédiaires sont plus rares. **Java fait partie de cette catégorie**. C'est un **compromis** entre les deux précédents types.

Dans un premier temps, un compilateur va être utilisé pour transformer le code dans un code **pseudo-compilé**. Ce code pseudo-compilé n'est **pas directement compréhensible** par la machine mais est beaucoup plus rapide à lire par celle-ci.

Dans un deuxième temps, un **interpréteur** est utilisé pour faire fonctionner ce langage. Étant donné que le code est pseudo-compilé, l'interpréteur est **beaucoup plus rapide**.

On a donc l'avantage d'un langage **fonctionnant partout et avec de bonnes performances**.

Comme pour les langages interprétés, le pseudo-code est bien souvent lui aussi **compilé définitivement** à la première lecture ou au bout d'un certain nombre de lectures. C'est le cas en Java.

## C'est quoi la JVM ?

Le pseudo-langage en Java s'appelle le **ByteCode**. L'interpréteur Java s'appelle la **Java Virtual Machine (JVM)**. Cet interpréteur comprend de nombreux outils :

- un **interpréteur** de ByteCode ;
- un outil de **gestion de la mémoire** ;
- un ensemble de **fonctionnalités** déjà codées ;
- un **optimiseur** de code.

La JVM existe pour **Windows, Linux, Mac OS X, Android ou encore Solaris et Amiga**.

Dans le prochain chapitre, nous allons installer les outils utiles au développement Java. Ces outils comprennent principalement le **compilateur Java** vers **Bytecode** et la **JVM**.

La JVM est un outil **puissant** et **performant**. C'est pourquoi des développeurs ont créé des outils capables de compiler d'autres langages vers du **ByteCode**. On peut donc par exemple rendre compatible du **Python et du Java**. Cela grâce à un outil nommé **Jython**.

# Installez votre environnement de développement Java avec Eclipse

---

Vous êtes prêt à installer Eclipse ? Parfait ! Mais avant toute chose, nous devons installer un JDK (Java Development Kit).

## Qu'est-ce qu'un JDK ?

Le kit de développement Java, ou *JDK*, permet de **compiler** et d'**exécuter** le code Java sur votre ordinateur. Nous ne rentrerons pas ici dans les détails techniques (si vous êtes impatients d'en savoir plus, vous pouvez parcourir [le cours Java](#)), mais sachez simplement que ce JDK contient :

- **un compilateur** permettant de traduire le code source Java en un code binaire optimisé pour n'importe quel système Windows, Linux, Mac ;
- **une JVM**, ou *machine virtuelle Java*. Il s'agit d'un logiciel sachant lire ce code binaire et l'exécuter sur le système de votre ordinateur.

## Téléchargez un JDK

Si vous avez déjà téléchargé Java sur votre ordinateur, vous avez peut-être remarqué que vous l'aviez fait **gratuitement** depuis le site **Oracle**, sous le nom "Java SE". Cependant, depuis janvier 2019, la politique tarifaire a changé, et pour une utilisation commerciale, il n'est plus possible d'utiliser gratuitement la JDK d'Oracle. Si vous travaillez en entreprise ou même en freelance sur des projets Java, vous êtes concerné par ce problème !

Heureusement, la communauté développe des versions libres de la JDK. Les deux projets libres principaux sont :

- **OpenJDK** : c'est un peu le projet *source* de Java. Il contient toutes les dernières évolutions et corrections du compilateur et de la JVM. Mais vous ne pouvez pas l'utiliser directement sur votre ordinateur, en particulier s'il est sous Windows, car ce sont les codes sources du logiciel de Java et non les logiciels exécutables sur votre ordinateur ;
- **AdoptOpenJDK** : il s'agit un projet communautaire visant à fournir les logiciels exécutables de Java pour n'importe quel système Windows, Linux ou Mac, en incorporant au mieux les dernières mises à jour.

C'est donc sur [AdoptOpenJDK](#) que nous allons télécharger la JDK. Le site détecte normalement votre système. Vous pouvez donc laisser les paramètres proposés par défaut, mais pensez tout de même à :

- vérifier le système proposé (Windows, Linux, Mac en x86 ou x64) ;
- choisir OpenJDK 8 et HotSpot.

Une fois le fichier téléchargé, cliquez dessus pour le lancer et l'installer, en gardant toutes les options par défaut.

## Téléchargez et installez Eclipse

Maintenant que vous avez téléchargé une JDK qui fonctionnera avec Eclipse, téléchargez votre IDE et installez-le pour commencer à coder en Java.

Voici les étapes de base :

- cliquez sur ce lien pour accéder au [site Eclipse](#) ;
- cliquez sur **Desktop IDEs**. Les liens qui apparaissent incluent les **plugins préinstallés** dont vous aurez besoin, selon le langage que vous allez utiliser ;
- sélectionnez **Eclipse IDE for Enterprise Java Developers**, qui comprend les plugins préinstallés non seulement pour un IDE Java, mais aussi pour le développement de projets web ;
- téléchargez la version **adaptée** à votre système et installez-la. Quand vous cliquez sur le **fichier exécutable** que vous avez téléchargé, conservez les paramètres par défaut. Eclipse vous fournit un espace de travail, c'est-à-dire un emplacement où vous pouvez placer vos fichiers. Vous pouvez choisir l'emplacement par défaut ou créer un nouveau dossier ;
- une fois l'installation terminée, lancez Eclipse depuis le programme d'installation en cliquant dessus.

## En résumé

Dans ce chapitre, vous avez découvert que :

- un **JDK** est nécessaire pour faire fonctionner Java sur votre ordinateur ;
- un JDK permet de compiler puis d'exécuter votre code via un logiciel nommé **JVM** ;
- il est possible d'installer un JDK depuis le site **AdoptOpenJDK** ;
- il est possible d'installer Eclipse depuis le site Eclipse.

*À présent, il vous est possible de créer et d'exécuter votre premier programme Java sur Eclipse.*

## **Travail personnel**

- Débogage de code dans Eclipse
- Test de méthodes avec JUnit dans Eclipse
- Refactoring de code dans Eclipse
- Structurer un projet Java dans Eclipse avec Maven
- Utilisation de Git avec Eclipse et GitHub
- Envoi de corrections sur GitHub à l'aide d'Eclipse