

Bien débiter en Java

par Baptiste Wicht ([home](#))

Date de publication : 3 Aout 2007

Dernière mise à jour :

Vous voulez commencer à programmer en Java et vous êtes perdu, vous ne savez pas par quoi commencer ? Alors cet article est fait pour vous. Ce n'est pas un tutoriel du langage Java, c'est plutôt un guide qui va vous aider à débiter, à choisir vos outils et vous indiquer quelles ressources vous pourriez lire pour avancer.


- I - Qu'est-ce que le Java ?
- II - Je débute
- III - Je veux aller plus loin
 - III-A - Comment développer des interfaces graphiques avec Java ?
 - III-B - Comment gérer les bases de données avec Java ?
 - III-C - Comment faire de la programmation web avec Java ?
 - III-D - Comment développer pour un mobile avec Java ?
 - III-E - Comment faire de la 2D/3D en Java ?
 - III-F - Quels sont les autres outils que je peux utiliser ?
- IV - Comment tester mes connaissances ?
- V - J'ai tout lu, mais pas tout compris, où poser mes questions ?
- VI - Conclusion
 - VI-A - Remerciements

I - Qu'est-ce que le Java ?

Pour commencer, Java est un langage de programmation informatique orienté objet créé par James Gosling et Patrick Naughton de Sun Microsystems. Mais c'est également un environnement d'exécution.

Java peut être séparé en deux parties. D'une part, votre programme écrit en langage Java et d'autre part, une machine virtuelle (JVM) qui va se charger de l'exécution de votre programme Java.

C'est cette plateforme qui garantit la portabilité de Java. Il suffit qu'un système ait une machine virtuelle Java pour que tout programme écrit en Java puisse fonctionner.

Avec le langage Java, vous pouvez développer, des applications Desktop, développer des applets pour vos sites web, développer des sites en  **JSP**, des applications pour téléphone mobile. La première chose à faire est bien évidemment d'apprendre à faire des applications standalones simples.


II - Je débute

Il n'est pas facile de commencer à développer si on n'a jamais fait de programmation auparavant. C'est pourquoi, il est très important de commencer avec des bonnes ressources et de ne pas se lancer à la va-vite dans le langage.

Pour les personnes connaissant déjà la programmation et les langages objets, l'apprentissage de Java ne devrait pas se révéler trop compliqué.

Vous pouvez commencer par lire l'un des **tutoriels** d'initiation à Java de la rubrique **Java** de Developpez.com. Je vous conseille d'en lire au moins un parmi ceux là ou alors un des livres proposés :

- **Penser en Java V2**
- **Développez en Java**
- **Cours sur le langage Java**
- **Le livre de Java premier langage**
- **Programmer en Java**
- **Java en concentré**

Il vous faudra bien évidemment un ou des outils pour développer en Java. Néanmoins pour commencer, je vous conseille d'utiliser un simple bloc-note et, de compiler et d'exécuter vos programmes en ligne de commandes pour commencer. Cela vous permettra de bien comprendre le fonctionnement de Java. Ensuite, vous pourrez bien sûr développer avec un  **EDI** plus évolué.



Je vous conseille d'utiliser **Eclipse** ou **Netbeans** pour développer vos applications Java. Tous deux sont multiplateformes, gratuits et très puissants.

III - Je veux aller plus loin

Le langage Java peut être utilisé dans de nombreux domaines que je vous invite à découvrir en parcourant les différents articles Java de Developpez.com. Les chapitres suivants reprennent quelques thèmes que vous pourriez approfondir.

III-A - Comment développer des interfaces graphiques avec Java ?

Pour développer des interfaces graphiques en Java, plusieurs choix s'offrent à vous :



-  **AWT** : C'est le premier framework graphique de Java. Il utilise les composants systèmes pour dessiner l'interface graphique, il est donc limité en composants.
-  **Swing** : C'est le successeur de AWT. Tous les composants sont dessinés directement en Java. Ce qui fait qu'il est plus lent que AWT, mais cette différence est très faible dans les dernières versions de Java
- **SWT** : Ce framework graphique n'est pas disponible avec l'api standard de Java. C'est l'équipe de développement d'Eclipse qui l'a créé. Comme AWT, il utilise directement les composants systèmes mais émule les composants non disponibles en les dessinant lui-même.

Je vous conseille d'utiliser Swing. En effet ce framework est très simple d'utilisation et très complet. De plus, la communauté autour du framework est énorme et permet d'ajouter de nombreuses fonctionnalités à Swing. Vous pouvez aussi tester SWT pour vous faire une idée.

Voici de quoi débiter avec ces frameworks :

- **Création de votre première interface graphique avec Swing**
- **Les principes de base du GUI avec Swing**
- **Le développement d'interfaces graphiques avec SWT**
- **Le livre "Swing la synthèse"**
- **Le livres "Les cahiers du programmeur Swing"**






III-B - Comment gérer les bases de données avec Java ?

La première manière de gérer les bases de données est d'utiliser l'api  **JDBC** qui est en standard avec Java. Ensuite, pour aller plus loin, vous pouvez vous orienter vers les frameworks de mapping objet/relationnel tel que  **Hibernate** ou encore utiliser un framework de persistance plus petit tel qu'iBatis.

Voici des ressources qui vous seront fort utiles pour votre apprentissage :

- **Comment utiliser un pilote de base de données JDBC**
- **Tutoriel JDBC**
- **La persistance des données avec Hibernate 2.1.8**
- **La persistance en base de données avec JPA (Java Persistence API)**
- **Le livre "Hibernate 3.0 - Gestion optimale de la persistance dans les applications Java/J2EE"**
- **Le site officiel d'iBatis**


III-C - Comment faire de la programmation web avec Java ?

La programmation  **Java EE** est un thème très large. En effet, il existe une grande quantité de frameworks webs ainsi que différentes manières de programmer pour le web ( **JSP**,  **Servlets**,  **Applets**) et de nombreux outils (serveur d'application (JBoss, Weblogic) ou conteneur Web Java ( **Tomcat** par exemple)). Je vous conseille donc de bien étudier le sujet avant de vous y lancer.

Voici quelques ressources qui pourraient vous être utiles :

- [Java Standard Tag Library \(JSTL\)](#)
- [Introduction à Struts](#)
- [Programmation web avec Java](#)
- [Les cahiers du programmeur Java, conception et déploiement J2EE](#)
- [Struts par la pratique](#)

III-D - Comment développer pour un mobile avec Java ?

Il existe peu de documentation sur le développement  **Java ME**. Vous pouvez tout de même lire cet article : [Débuter en J2ME avec le profil MIDP](#) ou alors vous pouvez consulter la [documentation de Sun](#) qui est très complète mais n'existe qu'en anglais.

III-E - Comment faire de la 2D/3D en Java ?

En Java, vous pouvez aussi faire du dessin (graphisme) 2D et 3D. Vous pouvez utiliser différentes APIs pour faire ceci : Java2D, Java3D, JOGL, LWJGL et encore quelques autres moteurs graphiques. Voici quelques ressources qui vous aideront à démarrer avec le graphisme :

- [Introduction au dessin en Java](#)
- [La programmation OpenGL en Java : introduction à JOGL](#)
- [Premiers pas avec la librairie Java 3D](#)
- [Les apis 3D pour Java](#)

III-F - Quels sont les autres outils que je peux utiliser ?

Une fois passé le stade des premiers programmes, il sera utile de voir d'autres programmes utilitaires pour Java et de tester de nouveaux EDIs. Découvrez les Edis les plus populaires sur notre page [outils](#).

Vous pourrez aussi être amené à devoir créer un exécutable de votre application. Pour cela, retrouvez toutes les techniques possibles dans [cet article](#).

Pour la gestion de vos builds, où vous pourriez aussi être amené à utiliser des utilitaires dédiés, vous pouvez vous diriger vers [Ant](#) ou [Maven](#).

IV - Comment tester mes connaissances ?

Faire une série d'exercices pour tester vos connaissances peut se révéler très intéressant. En effet, ceci vous permettra de voir si vous avez bien acquis les informations sur Java.

C'est pourquoi l'équipe Java vous propose une série de petits exercices sur le langage Java : **Série d'exercices en Java**.

Une autre façon de s'entraîner est de créer une petite application soi-même. Il ne faut pas viser trop gros sans quoi vous risqueriez de ne jamais arriver à bout de votre application. Voici quelques petits exemples d'applications à développer soi-même qui peuvent se révéler intéressantes :

- **Une calculatrice** : Pour commencer, développer une calculatrice se révèle assez simple, mais vous pouvez aussi aller assez loin suivant les fonctionnalités que vous lui ajoutez.
- **Un petit jeu** : Vous pouvez aussi développer un petit jeu, par exemple un puissance 4 ou un jeu d'échec. Vous n'avez pas encore besoin de travailler profondément sur le graphisme, mais vous pouvez déjà faire de belles choses. Si vous voulez aller un peu plus loin, vous pouvez vous lancer dans un jeu de la vie.
- **Une bibliothèque** : Vous pouvez également développer une petite bibliothèque. Cela vous permettra d'aborder les bases de données ainsi que les formulaires

Bonne chance !

V - J'ai tout lu, mais pas tout compris, où poser mes questions ?

Il est possible que vous n'ayez pas compris une notion de Java ou que vous ayez un problème pour implémenter quelque chose. La première chose à faire est de vérifier dans les tutoriels Java si vous ne trouvez pas la réponse. Ensuite, vous pouvez consulter les différentes Faqs de la rubrique :

- **FAQ Général Java** : Pour tout ce qui concerne le langage en général et les notions principales de Java
- **FAQ Java GUI** : Pour tout ce qui concerne les interfaces graphiques en Java
- **FAQ Java XML** : Pour tout ce qui concerne la manipulation de XML en Java
- **FAQ Java EE** : Pour tout ce qui concerne Java EE
- **FAQ Java ME** : Pour tout ce qui concerne Java ME
- **FAQ JDBC** : Pour tout ce qui concerne la manipulation des bases de données
- **FAQ Hibernate** : Pour tout ce qui concerne Hibernate
- **FAQ Struts** : Pour tout ce qui concerne le framework struts
- **FAQ Eclipse** : Pour tout ce qui concerne Eclipse
- **FAQ Netbeans** : Pour tout ce qui concerne Netbeans
- **FAQ JCreator** : Pour tout ce qui concerne JCreator
- **FAQ Maven 2** : Pour tout ce qui concerne Maven 2

Si votre réponse ne se trouve pas dans une de ces pages, vous pouvez poser votre question sur les forums Java :

- **Général Java** : Pour toutes vos questions générales Java
- Et ses sous-forums :
 - **Langage** : Langage Java, syntaxe, POO, conventions
 - **Apis** : Forum sur les API standard et API complémentaires
 - **JDBC** : Pour vos questions sur l'api JDBC
 - **Persistance** : Pour toutes vos questions sur la gestion de la persistance en Java
 - **Spring** : Pour vos questions sur le framework Spring
 - **XML** : Pour vos questions sur la manipulation des fichiers XML
 - **Java ME** : Pour vos questions sur le développement pour mobiles
- **Interfaces graphiques en Java** : Pour toutes vos questions sur les interfaces graphiques en Java
- Et ses sous-forums :
 - **AWT/Swing** : Pour toutes vos questions sur les apis Swing et AWT
 - **SWT/JFace** : Pour toutes vos questions sur les apis SWT et JFace
 - **Graphisme** : Pour toutes vos questions sur le graphisme (2D, 3D, OpenGL, ...) en Java
- **Developpement web avec Java** : Pour toutes vos questions sur la programmation web en Java
- **Serveurs d'applications pour Java/Java EE** : Pour toutes vos questions sur la programmation web en Java
- **EDI/Outils pour Java** : Pour toutes vos questions sur les différents outils pour Java

Lorsque vous posez une question, soyez toujours le plus précis possible et évitez les questions du type "Urgent problème db". Faites aussi attention à poster dans le bon sous-forum. Plus votre question sera claire et plus il y aura de gens qui la liront et prendront la peine d'essayer d'y répondre.

Eh voilà, j'espère que vous trouverez les réponses à vos questions.

VI - Conclusion

J'espère que cet article aura pu vous aider à vous lancer dans le langage Java plus facilement et que vous aurez acquis de nouvelles connaissances.

L'équipe Java vous souhaite une bonne programmation ;)

VI-A - Remerciements

Je tiens à remercier **Dutmatlab** pour ses corrections.

