Java SE vs Java ee

Java peut être diviser en trois parties:

* Java SE
* Java EE
* Java ME

1. Java SE (Java Standard Edition): Java SE est utilisé pour le développement d’applications autonomes qui s’exécutent uniquement sur le système. Dans Java SE, vous pouvez apprendre les bases de Java comme: l’héritage, le polymorphisme, etc.

2. Java EE (Java Enterprise Edition): Java EE est utilisé pour développer des applications Web qui s’exécutent à l’aide du navigateur Web. Dans Java EE, vous pouvez apprendre les rubriques avancées telles que: Servlet, JSP, EJB, etc.

3. Java ME (Java Micro Edition): Java ME est utilisé pour le développement d’applications mobiles, vous en apprendrez davantage sur le développement d’applications Android.

Liste des fonctionnalités de J2EE

Au niveau client, J2EE prend en charge le code HTML pur, ainsi que les applets ou les applications Java. Il s’appuie sur le JSP et le code de servlet pour créer du HTML ou d’autres données formatées pour le client.

Les EJB (Enterprise JavaBeans) fournissent une autre couche dans laquelle la logique de la plate-forme est stockée. Un serveur EJB fournit des fonctions telles que le threading, la concurrence, la sécurité et la gestion de la mémoire. Ces services sont transparents pour le programmeur.

La connectivité JDBC (Java Database Connectivity), qui est l’équivalent de Java ODBC, constitue l’interface standard pour les bases de données Java.

L’API de servlet améliore la cohérence pour les développeurs sans nécessiter d’avoir une interface utilisateur graphique.

Table de comparaison

| **Java** | **J2EE** |
| --- | --- |
| Java est un acronyme de Java Standard Edition (Java SE) qui est utilisé pour mieux décrire le Core Java. | J2EE signifie Java 2 Entreprise Edition, ou simplement « Java EE ». |
| Java est un langage de programmation de haut niveau qui tire une grande partie de sa syntaxe auprès de C et C++. | J2EE est une plate-forme basée sur Java, essentiellement une extension de Java Standard Edition (Java SE). |
| Il est principalement utilisé pour développer des applications desktop. | Il est principalement utilisé pour le développement des applications d’entreprise multi-niveaux sur le Web. Il peut être aussi utilisé pour les applications desktop et Web. |
| C’est un langage basé sur la POO qui simplifie le développement logiciel. | Il s’agit d’un ensemble d’API Java qui ciblent les technologies d’entreprise telles que les EJB, servlets, JSP, etc. |
| C’est le début de Core Java qui commence par les concepts de base de Java | C’est le niveau suivant de Java qui implémente une architecture à deux niveaux (client et serveur). |

Conclusion

Java est un langage de programmation à usage général qui est lié un peu au C++ et qui est en fait un descendant direct du langage C. Comme une grande partie de syntaxe Java est héritée de C et C++, il serait sage de penser à Java comme une version Web du C++. Cependant, les deux partagent différents points, donc comparer les deux ne serait pas une bonne idée non plus. Java n’a pas été conçu pour remplacer le C++, mais pour surmonter les complexités du C++. En revanche, Java EE (Java Platform Enterprise Edition), anciennement J2EE, est un standard de développement en ligne d’applications Web côté serveur. Il s’agit d’un ensemble d’API Java appartenant à Oracle et dotés d’un ensemble puissant de bibliothèques pouvant être utilisées pour créer des applications d’entreprise.  
 

The Java Platform, Enterprise Edition (Java EE) is a collection of Java APIs owned by [Oracle](https://www.techtarget.com/searchoracle/definition/Oracle) that software developers can use to write server-side applications. It was formerly known as Java 2 Platform, Enterprise Edition, or [J2EE](https://www.theserverside.com/definition/J2EE-Java-2-Platform-Enterprise-Edition).

[Sun Microsystems](https://www.theserverside.com/definition/Sun-Microsystems) (together with industry partners such as [IBM](https://www.techtarget.com/searchitchannel/definition/IBM-International-Business-Machines)) originally designed Java EE to simplify application development in a [thin-client](https://www.techtarget.com/searchnetworking/definition/thin-client)-tiered environment. Java EE simplifies app development and decreases the need for programming by creating standardized, reusable modular components and by enabling the tier to handle many aspects of programming automatically.

Java EE applications are hosted on application servers, such as  IBM's [WebSphere](https://searchdomino.techtarget.com/definition/WebSphere), Oracle's GlassFish or [Red Hat](https://www.techtarget.com/searchdatacenter/definition/Red-Hat)'s WildFly server, all of which run either in the cloud or within a corporate [data center](https://www.techtarget.com/searchdatacenter/definition/data-center). While Java EE apps are hosted on the server side, examples of Java EE clients include an [internet of things (IoT)](https://internetofthingsagenda.techtarget.com/definition/Internet-of-Things-IoT) device, smartphone, [RESTful](https://www.techtarget.com/searchapparchitecture/definition/REST-REpresentational-State-Transfer) web service, standard web-based application, [WebSocket](https://www.techtarget.com/whatis/definition/WebSocket) or even [microservices](https://www.techtarget.com/searchapparchitecture/definition/microservices) running in a [Docker](https://www.techtarget.com/searchitoperations/definition/Docker) container.

### Java EE architecture goals

The Java EE architecture provides services that simplify the most common challenges facing developers when building modern applications, in many cases through [APIs](https://www.techtarget.com/searchapparchitecture/definition/application-program-interface-API?_gl=1*kek3uo*_ga*MTUxNTA2ODk5OS4xNjY4OTQyMDU2*_ga_TQKE4GS5P9*MTY2ODk0MjA1Ni4xLjEuMTY2ODk0MjE5NS4wLjAuMA..&_ga=2.111427524.113174200.1668942056-1515068999.1668942056), thus making it easier to use popular [design patterns](https://www.techtarget.com/searchsoftwarequality/definition/pattern) and industry-accepted best practices.

For example, one common challenge enterprise developers face is how to handle requests coming in from web-based clients. To simplify this challenge, Java EE provides the [Servlet](https://www.techtarget.com/whatis/definition/servlet) and [JavaServer Pages (JSP)](https://www.theserverside.com/definition/Java-Server-Page-JSP) APIs, which provide methods for activities like finding out what a user typed into a text field in an online form or storing a [cookie](https://www.techtarget.com/searchsoftwarequality/definition/cookie) on a user's browser.

Another common task is how to store and retrieve information in a database. To address this goal, Java EE provides the Java Persistence API (JPA,) which makes it easy to map data used within a program to information stored in the tables and rows of a database. Also, creating web services or highly [scalable](https://www.techtarget.com/searchdatacenter/definition/scalability) logic components is simplified through the use of the [Enterprise JavaBeans (EJB)](https://www.theserverside.com/definition/Enterprise-JavaBeans-EJB) specification. All of these APIs are well tested, relatively easy for Java developers to learn and can greatly simplify some of the hardest parts of enterprise development.

### Java EE core technologies

Along with the four aforementioned APIs, there are more than 30 Java APIs included as Java EE core technologies, with that number to approach 50 with the eventual release of Java EE 8. These Java EE core technologies broadly fall into the following file categories:

* **HTTP client technologies**. For dealing with [HTTP](https://www.techtarget.com/whatis/definition/HTTP-Hypertext-Transfer-Protocol)-based clients, Java EE includes the Java API for WebSocket programming, an API for [JSON](https://theserverside.techtarget.com/definition/JSON-Javascript-Object-Notation) Processing, the JSF and Servlet APIs and the JSP Standard Tag Library (JSTL).
* **Database and resource access technologies**. For interacting with external and back-end systems, Java EE includes JavaMail, a standard connector architecture, a [Java Message Service (JMS)](https://www.theserverside.com/definition/Java-Message-Service-JMS) API and a Java Transaction API (JTA) for enforcing two-phase commits.
* **REST and web service technologies:**To help with the development and deployment of REST-, [SOAP](https://www.techtarget.com/searchapparchitecture/definition/SOAP-Simple-Object-Access-Protocol)-, XML- and JSON-based web services, the Java APIs for RESTful Web Services (JAX-RS) and [XML-based web services (JAX-WS)](https://www.theserverside.com/definition/JAX-WS-Java-API-for-XML-Web-Services) are included, along with APIs for XML messaging and XML registries (JAXR).
* **Java EE security and container management**: For implementing custom Java EE security and managing Java EE containers, software developers have access to the Java Authorization Contract for Containers and the Java Authentication Service Provider Interface for Containers.