# **What is the difference between JDK 7 and JDK 6?**

|  |  |
| --- | --- |
| **JDK 6** | JDK7 |
| JDK 6 was released on 11th December 2006. | JDK 7 was released on 07th July 2011 |
| Project Mustang is the code name of JDK 6. | Project Dolphin is the code name of JDK 7. |
| The new features and enhancements introduced in JDK 6 are:   1. Creation of additional classes in java.awt package, 2. Extended support for internationalization, Improvements in javax.swing, 3. Introduction of system tray capability, Introduction to new type of compilers and lot more. 4. JDK 6 also includes enhancements in Debugging, Avoiding memory leaks, and Security support. 5. Support for older win9x versions dropped. 6. Scripting lang support: Generic API for integration with scripting languages, & built-in mozilla javascript rhino integration - Dramatic performance improvements for the core platform, and swing. 7. Improved web service support through JAX-WS 8. JDBC 4.0 support 9. Java compiler API: an API allowing a java program to select and invoke a java compiler programmatically. 10. - Upgrade of JAXB to version 2.0: including integration of a stax parser. 11. - Support for pluggable annotations   - Many GUI improvements, such as integration of swingworker in the API, table sorting and filtering, and true swing double-buffering (eliminating the gray-area effect). | The new features and enhancements introduced in JDK 7 are:   1. Enhanced support for dynamically typed languages 2. Strict type checking verifier, 3. Upgrading of class loader architecture, Updates in collections and concurrency, Locale enhancement, 4. Support for Unicode 6.0 and lot more. 5. JDK 7 also includes enhancements and improvements in I/O and Networking, Security, Cryptography, Database Connectivity, Client, Web, and Management. 6. Features Added:   Support for dynamically-typed languages (InvokeDynamic): Extensions to the JVM, the Java language, and the Java SE API to support the implementation of dynamically-typed languages at performance levels near to that of the Java language itself  - Strict class-file checking: Class files of version 51 (SE 7) or later must be verified with the typechecking verifier; the VM must not fail over to the old inferencing verifier.  - Small language enhancements (Project Coin): A set of small language changes intended to simplify common, day-to-day programming tasks: Strings in switch statements, try-with-resources statements, improved type inference for generic instance creation ("diamond"), simplified varargs method invocation, better integral literals, and improved exception handling (multi-catch)  - Upgrade class-loader architecture: A method that frees the underlying resources, such as open files, held by a URLClassLoader  - Concurrency and collections updates: A lightweight fork/join framework, flexible and reusable synchronization barriers, transfer queues, concurrent linked double-ended queues, and thread-local pseudo-random number generators.  - Internationalization Upgrade: Upgrade on Unicode 6.0, Locale enhancement and Separate user locale and user-interface locale.  - More new I/O APIs for the Java platform (NIO.2), NIO.2 filesystem provider for zip/jar archives, SCTP, SDP, TLS 1.2 support.  - Security & Cryptography implemented Elliptic-curve cryptography (ECC).  - Upgrade to JDBC 4.1 and Rowset 1.1.  - XRender pipeline for Java 2D, Create new platform APIs for 6u10 graphics features, Nimbus look-and-feel for Swing, Swing JLayer component, Gervill sound synthesizer.  - Upgrade the components of the XML stack to the most recent stable versions: JAXP 1.4, JAXB 2.2a, and JAX-WS 2.2.  - Enhanced MBeans. |

**Difference between JSP & Servlet**

|  |  |
| --- | --- |
| **Servlets** | JSP |
| [Servlets are Java programs](http://www.withoutbook.com/Technology.php?tech=3&subject=Servlets%20Interview%20Questions%20and%20Answers) that are already compiled which also creates dynamic web content. | [JSP is a](http://www.withoutbook.com/Technology.php?tech=2&subject=JSP%20Interview%20Questions%20and%20Answers) [web page](http://www.google.com/url?q=http%3A%2F%2Fwww.withoutbook.com%2FTechnology.php%3Ftech%3D2%26subject%3DJSP%2520Interview%2520Questions%2520and%2520Answers&sa=D&sntz=1&usg=AFQjCNEKcSVR77dtQL5P3dWl5YzC0IpJGg) [scripting language](http://www.withoutbook.com/Technology.php?tech=2&subject=JSP%20Interview%20Questions%20and%20Answers) that can generate dynamic content. |
| Servlets run faster compared to JSP. | JSP run slower compared to Servlet as it takes compilation time to convert into Java Servlets. |
| Its little much code to write here. | It’s easier to code in JSP than in Java Servlets. |
| In MVC, servlet act as a controller. | In MVC, jsp act as a view. |
| servlets are best for use when there is more processing  and manipulation involved. | JSP are generally preferred when there is not much processing of data required. |
| There is no such custom tag facility in servlets. | The advantage of JSP programming over servlets is that we can build [custom tags](http://www.withoutbook.com/InterviewQuestionAnswer.php?tech=2&quesId=458&subject=JSP%20Interview%20Questions%20and%20Answers) which can directly call [Java beans](http://www.withoutbook.com/Technology.php?tech=35&subject=Java%20Beans%20Interview%20Questions%20and%20Answers). |
| There are no such methods for servlets. | We can achieve functionality of JSP at client side by running [JavaScript](http://www.withoutbook.com/Technology.php?tech=8&subject=JavaScript%20Interview%20Questions%20and%20Answers) at client side. |

**Difference between Struts 1 and Struts 2**

|  |  |
| --- | --- |
| **Struts 2** | **Struts 1** |
| **Action Classes:** An Struts 2 Action may implement an Action interface, along with other interfaces to enable optional and custom services. Struts 2 provides a base ActionSupport class to implement commonly used interfaces. Albeit, the Action interface is not required. Any POJO object with a execute signature can be used as an Struts 2 Action object. | **Action Classes:** Struts 1 requires Action classes to extend an abstract base class. A common problem in Struts 1 is programming to abstract classes instead of interfaces. |
| Threading Model: Struts 2 Action objects are instantiated for each request, so there are no thread-safety issues. (In practice, servlet containers generate many throw-away objects per request, and one more object does not impose a performance penalty or impact garbage collection.) | Threading Model: Struts 1 Actions are singletons and must be thread-safe since there will only be one instance of a class to handle all requests for that Action. The singleton strategy places restrictions on what can be done with Struts 1 Actions and requires extra care to develop. Action resources must be thread-safe or synchronized. |
| Servlet Dependency: Struts 2 Actions are not coupled to a container. Most often the servlet contexts are represented as simple Maps, allowing Actions to be tested in isolation. Struts 2 Actions can still access the original request and response, if required. However, other architectural elements reduce or eliminate the need to access the HttpServetRequest or HttpServletResponse directly. | Servlet Dependency: Struts 1 Actions have dependencies on the servlet API since the HttpServletRequest and HttpServletResponse is passed to the execute method when an Action is invoked. |
| Testability: Struts 2 Actions can be tested by instantiating the Action, setting properties, and invoking methods. Dependency Injection support also makes testing simpler. | Testability: A major hurdle to testing Struts 1 Actions is that the execute method exposes the Servlet API. A third-party extension, Struts TestCase, offers a set of mock object for Struts 1. |
| Harvesting Input: Struts 2 uses Action properties as input properties, eliminating the need for a second input object. Input properties may be rich object types which may have their own properties. The Action properties can be accessed from the web page via the taglibs. Struts 2 also supports the ActionForm pattern, as well as POJO form objects and POJO Actions. Rich object types, including business or domain objects, can be used as input/output objects | Harvesting Input: Struts 1 uses an ActionForm object to capture input. Like Actions, all ActionForms must extend a base class. Since other JavaBeans cannot be used as ActionForms, developers often create redundant classes to capture input. DynaBeans can used as an alternative to creating conventional ActionForm classes, but, here too, developers may be redescribing existing JavaBeans. |
| The ModelDriven feature simplifies taglb references to POJO input objects. |  |
| Expression Language: Struts 2 can use JSTL, but the framework also supports a more powerful and flexible expression language called "Object Graph Notation Language" (OGNL). | Expression Language: Struts 1 integrates with JSTL, so it uses the JSTL EL. The EL has basic object graph traversal, but relatively weak collection and indexed property support. |
| Binding values into views: Struts 2 uses a "ValueStack" technology so that the taglibs can access values without coupling your view to the object type it is rendering. The ValueStack strategy allows reuse of views across a range of types which may have the same property name but different property types. | Binding values into views: Struts 1 uses the standard JSP mechanism for binding objects into the page context for access. |
| Type Conversion: Struts 2 uses OGNL for type conversion. The framework includes converters for basic and common object types and primitives. | Type Conversion: Struts 1 ActionForm properties are usually all Strings. Struts 1 uses Commons-Beanutils for type conversion. Converters are per-class, and not configurable per instance. |
| Validation: Struts 2 supports manual validation via the validate method and the XWork Validation framework. The Xwork Validation Framework supports chaining validation into sub-properties using the validations defined for the properties class type and the validation context. | Validation: Struts 1 supports manual validation via a validate method on the ActionForm, or through an extension to the Commons Validator. Classes can have different validation contexts for the same class, but cannot chain to validations on sub-objects. |
| Control of Action Execution: Struts 2 supports creating different lifecycles on a per Action basis via Interceptor Stacks. Custom stacks can be created and used with different Actions, as needed. | Control of Action Execution: Struts 1 supports separate Request Processors (lifecycles) for each module, but all the Actions in the module must share the same lifecycle. |

**Difference Between Jboss and Tomcat**

|  |  |
| --- | --- |
| **Tomcat** | **Jboss** |
| Tomcat is a servlet container. It implements only the servlets and jsp specification. | JBoss is a server application based on Java. It implements full JEE servers (including stuff like EJB, JMS, ...). |
| Tomcat makes use of Sun Microsystems specific specifications. | JBoss makes use of the Java EE specification. |

Difference between Linux and Unix

|  |  |
| --- | --- |
| Unix | Linux |
| Unix is an operating system that is very popular in universities, companies, big enterprises etc. | Linux is an example of Open Source software development and Free Operating System (OS). |
| Cost: Different flavors of Unix have different cost structures | Cost: Linux can be freely distributed, downloaded freely, distributed through magazines, Books etc. There are priced versions for Linux also, but they are normally cheaper than Windows. |
| Price: Some free for development use (Solaris) but support is available for a price. | Price: Free but support is available for a price. |
| User: Unix operating systems were developed mainly for mainframes, servers and workstations. The Unix environment and the client-server program model were essential elements in the development of the Internet. | User: Everyone. From home users to developers and computer enthusiasts alike. |
| Manufacturer: Three bigest distributions are Solaris (Oracle), AIX (IBM) & HP-UX Hewlett Packard. | Manufacturer: Linux kernel is developed by the community. Linus Torvalds oversees things. |
| Usage: The UNIX operating system is used in internet servers, workstations & PCs. Backbone of the majority of finance infastructure and many 24x365 high availability solutions. | Usage: Linux can be installed on a wide variety of computer hardware, ranging from mobile phones, tablet computers and video game consoles, to mainframes and supercomputers. |
| Processors: x86/x64, Sparc, Power, Itanium, PA-RISC and many others. | Processors: Dozens of different kinds. |
| Processors: is available on PA-RISC and Itanium machines. Solaris also available for x86/x64 based systems. | Architectures: Originally developed for Intel's x86 hardware, ports available for over two dozen CPU types including ARM |
| Killer Features: ZFS - Next generation filesystem Dtrace - dynamic kernel tracing | Killer Features: Ksplice - kernel update without reboot |
| GUI: Initially Unix was a command based OS, but later a GUI was created called Common Desktop Environment. Most distributions now ship with Gnome. | GUI: Linux typically provides two GUIs, KDE and Gnome. But Linux GUI is optional. |
| File System Support: jfs,gpfs,hfs,ufs,xfs,zfs forma | File System Support: Ext2, Ext3, Ext4, Jfs, ReiserFS, Xfs, Btrfs, FAT, FAT32, NTFS |
| Text mode interface: Originally the Bourne Shell. Now it's compatible with many others including BASH, Korn & C. | Text mode interface: BASH (Bourne Again SHell) is the Linux default shell. It can support multiple command interpreters. |
| Security: A rough estimate of UNIX viruses is between 85 -120 viruses reported till date. | Security: Linux has had about 60-100 viruses listed till date. None of them actively spreading nowadays. |
| Development and Distribution: Unix systems are divided into various other flavors, mostly developed by AT&T as well as various commercial vendors and non-profit organizations. | Development and Distribution: Linux is developed by Open Source development i.e. through sharing and collaboration of code and features through forums etc and it is distributed by various vendors such as Debian, Red Hat, SUSE, Ubuntu, GentuX etc. |
| Thread detection and solution: Because of the proprietary nature of the original Unix, users has to wait for a while, to get the proper bug fixing patch. But these are not as common. | Threat detection and solution: In case of Linux, threat detection and solution is very fast, as Linux is mainly community driven and whenever any Linux user posts any kind of threat, several developers start working on it from different parts of the world |
| Inception: In 1969, it was developed by a group of AT&T employees at Bell Labs and Dennis Ritchie. It was written in “C” language and was designed to be a portable, multi-tasking and multi-user system in a time-sharing configuration. | Inception: Inspired by MINIX (a Unix-like system) and eventually after adding many features of GUI, Drivers etc, Linus Torvalds developed the framework of the OS that became LINUX in 1992. The LINUX kernel was released on 17th September, 1991 |

Difference between Struts and Spring

|  |  |
| --- | --- |
| Spring | Struts |
| Springs is a Lightweight Inversion of Control and Aspect Oriented Container Framework. Every work in the last sentence carry the true purpose of the Spring framework. It is just not a framework to integrate / plug in at the presentation layer. It is much more to that. It is adaptible and easy to run light weight applications, it provides a framework to integrate OR mapping, JDBC etc., Infact Struts can be used as the presentation tier in Spring. | Struts is a sophisticated framework offering the easy 2 develop, structured view/presentation layer of the MVC applications. Advanced, robust and scalable view framework underpinning reuse and seperation of concerns to certain extent. |
| Spring provides Aspect Oriented programming, it also solves the seperation of concerns at a much bigger level. It allows the programmer to add the features (transactions, security, database connectivity components, logging components) etc., at the declaration level. Spring framework takes the responsibility of supplying the input parameters required for the method contracts at runtime reducing the coupling between various modules by a method called dependency injection / Inversion of Control. | Struts features strictly associate with presentation stuff. It offers Tiles to bring in reuse at presentation level. It offers Modules allowing the application presentation to segregate into various modules giving more modularity there by allowing each module to have its own Custom/Default Request Processor. |
| Spring does not route the request in a specific way like this, rather it allows to you to design in your own way however in allowing to exploit the power of framework, it allows you to use the Aspect Oriented Programming and Inversion of Control in a great way with great deal of declarative programming with the XML. Commons framework can be integrated to leverage the validation in spring framework too. Morethan this, it provides all features like JDBC connectivity, OR Mapping etc., just to develop & run your applications on the top of this. | Struts is developed with a Front Controller and dispatcher pattern. Where in all the requests go to the ActionServlet thereby routed to the module specific Request Processor which then loads the associated Form Beans, perform validations and then handovers the control to the appropriate Action class with the help of the action mapping specified in Struts-config.xml file. |
| Spring is not. | Struts 2 integrates with Dojo AJAX framework closely and provides many complex UI components out-of-box, such as datepicker, tooltips, etc. |
| Spring has not. | Struts 2 has AJAX theme. |
| Spring is not. | Struts 2 tags are stylesheet-driven, making it easier to develop consistent pages. |
| Spring is not. | Struts 2 checkboxes are stateful, and require no special handling. |
| Spring is a complete J2EE framework having seven independent layers which can be used all together or few layers integrated with some other framework. Spring provides declarative transaction management and AOP. One layer of spring is for presentation purpose like struts but for only presentation layer, struts is better than spring. | Struts is mainly a presentation layer framework, like redirecting to a particular page ,doing client side validations etc which otherwise very tedious using jsp and servlets. |
| Whereas Spring is not in this area. | If you want a bunch of taglibs that generate form fields and so forth, Struts is probably the better choice. |
| In Spring this problem does not exist. | Our UI is mostly click-driven and light on data and validation. It seems to me that most people run into difficulties with Struts when they start moving a lot of data from HTTP into the model. |

Difference Between JBoss and Web-logic

|  |  |
| --- | --- |
| **WebLogic** | JBoss |
| WebLogic application server is developed by Oracle. | JBoss application server is a free and open source product. |
| But the latest release of WebLogic server only supports Java EE 5. | Latest version of JBoss server supports Java EE 6 Web Profile. |
| You can change console requirements depending on the requirements in WebLogic, as Self Console 7001 is included. | But since JBoss is dependent on Tomcat Server, this is not possible in JBoss. |
| Multiple ways of deployment is possible in Web Logic. | While Ant alone could be used for deployment in JBoss, and it is very quick and easy. |
| WebLogic is an expensive product, it has several features that are not provided in JBoss. For example, WebLogic’s web-based administrator console can be used for the configuration of JMS, Data Sources, and security settings, etc. | Configuration and administration is pretty simple in JBoss, but a UI is not provided. |
| While, clustering is supported for all the APIs in WebLogic. WebLogic offers JMS clustering. | Clustering is supported only for some of the features in JBoss. Whereas, JBoss does not offers JMS clustering. |
| Standard JDBC API is used for database connectivity in WebLogic. | But database connectivity is available in JBoss only through jca-jdbc wrappers, which means that sometimes the programmer has to write his own code. |

Difference Between CSS2 and CSS3

|  |  |
| --- | --- |
| **CSS 3** | **CSS 2** |
| **Features added/updated:**  **- CSS 3 Selectors**  **- Rounded Corners**  **- border-image**  **- box-shadow**  **- text-shadow**  **- Gradient**  **- RGBA: Color, Now with Opacity**  **- Transform (Element Rotation)**  **- Multicolumn Layout**  **-Web Fonts**  **- border-radius**  **- background-origin and background-clip**  **- background-size e.g background-size: auto 125px;**  **- Multiple Backgrounds e.g background-image: url(sheep.png), url(betweengrassandsky.png);** | Features added/updated:  - Font selection  - Tables  - Positioning  - Cursors  - [\*] matches any tag, e.g \*{color: red}.  - [>] Parent-Child relationship, e.g body>div.header>p{color: red}.  - Space can be used to specify that a tag must have other tags as its ancestor. e.g table a img {border:none}.  - can be used to specify some restriction on sibling tags. e.g img + p {color:red}.  There are special syntax to match some mouse hovering, focusing, behaviors. e.g a:link {color:red}, a:visited {color:green}, a:hover {color:yellow}.  - Can be used to match a tag only if it is the first child. e.g li:first-child {color:red}. |
| **- HSL colors**  **-HSLA colors**  **- opacity**  **- RGBA colors**  **- text-overflow**  **- word-wrap**  **- box-sizing**  **- resize**  **- outline**  **- attribute selectors**  **- nav-top, nav-right, nav-bottom, nav-left**  **- overflow-x, overflow-y**  **- content** | - :first-letter and :first-line can be used. e.g p:first-letter {color:red}, p:first-line {color:blue}.  - You can match the existence of a attribute, by the syntax tagName[attributeName]. e.g a[title] {color:red}.  -You can match a tag's attribute's value, using the syntax tagName[attributeName="valueString"]. e.g img[alt="icon"] {border:solid thin red}.  - A word in the value of a attribute can be matched, by using the operator ~=. e.g img[alt~="house"] {border:solid thin red}.  - Layout are done with attributes “display” and “position”. e.g BBB {display:block; position:absolute; top:300px, left:50px}.  - Has the ability to specify a table format. e.g BBB {display: table-row}, CCC {display: table-cell}.  - “white-space” has a new behavior spec: nowrap. e.g DDD {white-space: nowrap}.  Text can be inserted at the beginning or end of specified tag, using :before and :after. e.g AA:before {content:'Proof: '}, AA:after {content:'End of Proof.'}.  - Background Image can be repeated, or just horizontally or vertically. e.g AAA {background-repeat:repeat-x} |

Difference Between HTML4 and HTML5

|  |  |
| --- | --- |
| HTML5 | HTML4 |
| HTML 5 can use SVG and MathML inline. | HTML 4 cannot. |
| HTML 5 allows the storage and use of offline applications. | HTML 4 does not. |
| HTML 5 has many new elements that are not present in HTML 4 e.g <nav>, <footer>. | Added more elements to HTML 5 which does not exist in HTML 4. |
| HTML 5 has dropped certain elements. | The elements exist in HTML 4 has been removed from HTML 5 and few deprecated tags are <font> or <centre>. |
| Is being developed by web hypertext application technology working group (WHATWG) and W3C HTML WG. | Was developed by World Wide Web consortium and WHATWG (web hypertext application technology working group). |
| It includes multimedia elements (<audio> and <video>). | No multimedia support without third party software. |

## Application Server vs Web Server

1**. Application Server** supports **distributed transaction and EJB**. While Web Server only supports Servlets and JSP.

2. Application Server can contain web server in them. most of App server e.g. JBoss or WAS has Servlet and JSP container.

3. Though its not limited to Application Server but they used to provide services like **Connection pooling**, **Transaction management**, messaging, clustering, load balancing and persistence. Now Apache tomcat also provides connection pooling.

4. In terms of l*ogical difference between web server and application server*. web server is supposed to provide http protocol level service while application server provides support to web service and expose business level service e.g. EJB.

5. Application server are more heavy than web server in terms of resource utilization.