http://www.coderanch.com/t/494246/java/java/Absence-NULL-string

|  |  |
| --- | --- |
| **Author** | **Absence of NULL in string ?** |
| **maninder pal singh**  Greenhorn   Joined: May 04, 2010 Posts: 1 | posted Tuesday, May 04, 2010 13:22:39 private message  http://cache-www.coderanch.com/templates/default/images/thumbs-up.gif  Quote |
| Please justify why null character < as a termination signal>is absent in [String](http://www.coderanch.com/t/410859/java/java/String-StringBuffer-StringBuilder-Performance) in Java where it is present in other languages like C / C++ ? |
| Regards  Maninder Pal Singh | |
| http://cache-www.coderanch.com/templates/default/images/spacer.gif | |  |
| **Rob Spoor**  Sheriff   Joined: Oct 27, 2005 Posts: 19028  [I like...](http://www.coderanch.com/how-to/java/BumperStickers) Android Eclipse IDE Opera | posted Tuesday, May 04, 2010 15:02:57 private message  http://cache-www.coderanch.com/templates/default/images/thumbs-up.gif  Quote |  |
| Because [java.lang.String](http://docs.oracle.com/javase/7/docs/api/java/lang/String.html" \o "Java API" \t "_new) has other ways of specifying where it ends. The length() method is the main one.   The reason C uses the NULL character\* is because there is no other way to indicate how large an array is. Arrays in C have no hard bounds like arrays in Java have. You can easily try to read "elements" beyond its border. These are usually the cause for many (security) bugs in programs (buffer overruns). Java has solved this using the bounds checking (with an ArrayIndexOutOfBounds being thrown if the index is invalid) and the length field all arrays have.    \* The NULL character is actually still available in Java. It's the character '\0', or (char)0. It just isn't necessary for ending strings. |  |