Java versions 6 and below, strings were stored in PermGen, meaning that once a String was created, it was never garbage collected.

## ****Compressed****String****– Java 6****

The JDK 6 update 21 Performance Release, introduced a new VM option:

|  |  |
| --- | --- |
| 1 | -XX:+UseCompressedStrings |

When this option is enabled, Strings are stored as byte[], instead of char[] – thus, saving a lot of memory. However, this option was eventually removed in JDK 7, mainly because it had some unintended performance consequences.

**Java 8 Features**

In HashMap From Java 8, the linked lists are dynamically replaced with balanced binary search trees in collision resolution after the number of collisions in a given bucket location exceed a certain threshold.

**What is *StringJoiner*?**

[*StringJoiner*](https://www.baeldung.com/java-string-joiner)is a class introduced in Java 8 for joining separate strings into one, like**taking a list of colors and returning them as a comma-delimited string**. We can supply a delimiter as well as a prefix and suffix:

|  |  |
| --- | --- |
|  |  |

**Java 9 Features**

[**https://www.baeldung.com/java-9-compact-string**](https://www.baeldung.com/java-9-compact-string)

1. **Compact Strings**

[According to String’s Javadocs](https://docs.oracle.com/javase/7/docs/api/java/lang/String.html) for versions up to and including Java 8, Strings are stored in the UTF-16 format internally.

The char data type and java.lang.Character objects [are also based on the original Unicode specification](https://docs.oracle.com/javase/6/docs/api/java/lang/Character.html#unicode), which defined characters as fixed-width 16-bit entities.

Starting with JDK 9, Strings that contain only 1-byte characters use Latin-1 encoding, while Strings with at least 1 multi-byte character use UTF-16 encoding.



