Lets understand about 𝑾𝒉𝒚 𝑺𝒕𝒓𝒊𝒏𝒈 𝒊𝒔 𝒊𝒎𝒎𝒖𝒕𝒂𝒃𝒍𝒆 𝒊𝒏 [#java](https://www.linkedin.com/feed/hashtag/?keywords=java&highlightedUpdateUrns=urn%3Ali%3Aactivity%3A6945673862067941376)  
  
What does mean by Immutability?  
An object is considered as immutable if its state cannot change after it is constructed.  
  
String is Immutable in Java because of below benefits,  
Caching,  
Security,  
Synchronization,  
Performance.  
  
𝑪𝒂𝒄𝒉𝒊𝒏𝒈  
String literals are stored into a special memory region called 𝑺𝒕𝒓𝒊𝒏𝒈 𝑷𝒐𝒐𝒍 (𝑺𝑷) inside HEAP memory by JVM. Caching the String literals and reusing them saves lot of heap space because different String variables refer to the same Object in SP.  
  
Due to immutability, JVM optimizes the amount of memory allocated for Strings by storing distinct objects in SP. This process is called as "interning".  
  
𝑬𝒙𝒂𝒎𝒑𝒍𝒆  
String s1 = "Shiva";  
String s2 = "Shiva";  
  
First "Shiva" will get stored into SP and s1 refers to that, Since "Shiva" is already present in SP s2 will also refer to same Object, By this process JVM saving heap space.  
  
𝑺𝒆𝒄𝒖𝒓𝒊𝒕𝒚  
In a typical Java application we often use String in so many places to store usernames, connectionurls, etc. Assume we have a method called 𝒖𝒑𝒅𝒂𝒕𝒆𝑼𝒔𝒆𝒓𝑷𝒂𝒔𝒔𝒘𝒐𝒓𝒅𝑻𝒐𝑫𝒆𝒇𝒂𝒖𝒍𝒕(𝑺𝒕𝒓𝒊𝒏𝒈 𝒖𝒔𝒆𝒓𝑵𝒂𝒎𝒆) and it performs below operations,  
  
1) Verifies given username for security purpose  
2) Prepares a Strong password with the help of Java's default API's.  
3) Update the password for given userName in DB tbl.  
  
The method caller still has reference to userName object, If Strings are mutable by the time we execute update, we can't sure the String we received, because it may get changed by our caller in between thus making our query prone to SQL injection in this case. So mutable Strings can impact heavily on security over the time.  
  
𝑺𝒚𝒏𝒄𝒓𝒐𝒏𝒊𝒛𝒂𝒕𝒊𝒐𝒏  
Being immutable automatically makes the String thread safe, that means we no need to worry about other threads because they won't be able change our String.  
Hence, It can be shared across multiple threads running simultaneously because if a thread changes the value, then instead of modifying the same, a new String would be created in the String pool, hence Strings are safe in multi-threading.  
  
The immutability guarantees Strings that their value won’t change. So the hashCode() method is overridden in String class to facilitate caching, such that the hash is calculated and cached during the first hashCode() call and the same value is returned ever since.  
  
This, in turn, improves the performance of collections that uses hash implementations when operated with String objects.  
  
𝑷𝒆𝒓𝒇𝒐𝒓𝒎𝒂𝒏𝒄𝒆  
Due to String pool, it enhances the performance by saving the heap memory and faster access of hash implementations when operated with Strings.  
  
Since String is the most widely used data structure, improving the performance of String have a considerable effect on improving the performance of the whole application in general.