1. What is String in Java?  
   Ans. In Java, a string is an object that represents a number of character values. Each letter in the string is a separate [character](https://www.techtarget.com/whatis/definition/character) value that makes up the Java string object. Characters in Java are represented by the char class. Users can write an array of char values that will mean the same thing as a string.
2. Types of String in Java are?  
   Ans. **Primitive strings.** These are string literals or string calls from a nonconstructor context. A constructor is [a special method used to initialize objects](https://www.theserverside.com/video/Full-Java-constructors-tutorial). [Primitives](https://www.techtarget.com/whatis/definition/primitive) are not objects and have no methods or properties. They are represented at the lowest level of language implementation.

**Object strings.** These are strings created using the new operator. Object strings create two objects, whereas primitives create just one. Object strings create the string literal and the variable to refer to it.

1. In how many ways can you create String objects in java?  
   Ans. There are two ways to create a String object:
2. **By string literal** : Java String literal is created by using double quotes.  
   For Example: String s=“Welcome”;
3. **By new keyword** : Java String is created by using a keyword “new”.  
   For example: String s=new String(“Welcome”);    
   It creates two objects (in String pool and in heap) and one reference variable where the variable ‘s’ will refer to the object in the heap.

4.What is String constant pool?

Ans. **String pool** is nothing but a storage area in [Java heap](https://www.javatpoint.com/java-heap) where string literals stores. It is also known as **String Intern Pool** or **String Constant Pool**. It is just like object allocation. By default, it is empty and privately maintained by the [**Java String**](https://www.javatpoint.com/java-string) class. Whenever we create a string the string object occupies some space in the heap memory. Creating a number of strings may increase the cost and memory too which may reduce the performance also.

The JVM performs some steps during the initialization of string literals that increase the performance and decrease the memory load. To decrease the number of String objects created in the JVM the String class keeps a pool of strings.

When we create a string literal, the JVM first check that literal in the String pool. If the literal is already present in the pool, it returns a reference to the pooled instance. If the literal is not present in the pool, a new String object takes place in the String pool.