Content Menu ▼

Immutable String in Java

sunsilk
FREE
PERFECT STRAIGHT CONDITIONER
ARE YOU READY TO GO PERFECT STRAIGHT?
BUY NOW

In java, **string objects are immutable**. Immutable simply means unm
unchangeable.

Once string object is created its data or state can't be changed but a ne
object is created.

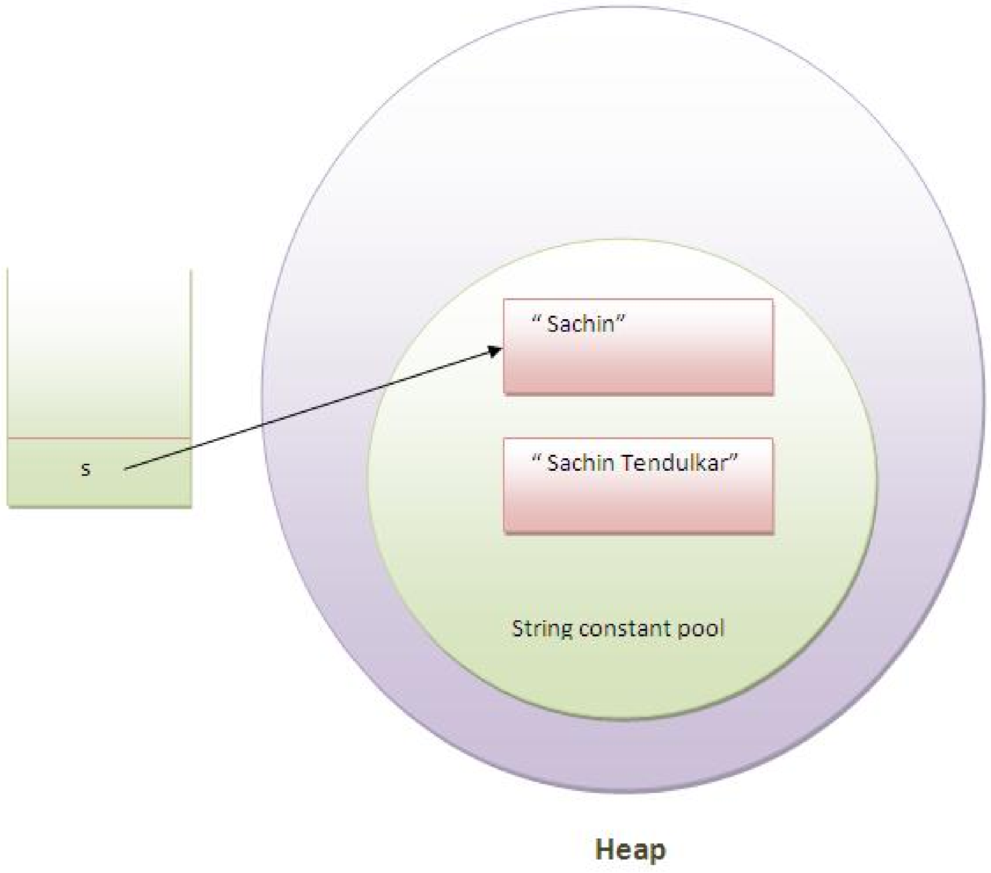
Let's try to understand the immutability concept by the example given b

```
class Testimmutablestring{
 public static void main(String args[]){
   String s="Sachin";
   s.concat(" Tendulkar");//concat() method appends the string at the
   System.out.println(s);//will print Sachin because strings are immuta
 }
}
```

Test it Now

```
Output:Sachin
```

Now it can be understood by the diagram given below. Here Sachin is n
but a new object is created with sachintendulkar. That is why string is k
immutable.



As you can see in the above figure that two objects are created but s reference variable still refers to "Sachin" not to "Sachin Tendulkar".

But if we explicitely assign it to the reference variable, it will refer to "Sachin Tendulkar" object.For example:

```
class Testimmutablestring1{
 public static void main(String args[]){
   String s="Sachin";
   s=s.concat(" Tendulkar");
   System.out.println(s);
 }
}
```

Test it Now

```
Output:Sachin Tendulkar
```

In such case, s points to the "Sachin Tendulkar". Please notice that still sachin object is not modified.

Why string objects are immutable in java?

Because java uses the concept of string literal.Suppose there are 5 reference variables,all referes to one object "sachin".If one reference variable changes the value of the object, it will be affected to all the reference variables. That is why string objects are immutable in java.

« prev next »