

Der **String Constant Pool** (auch **String Pool** genannt) ist ein spezieller Speicherbereich in Java, der dazu verwendet wird, **Speicherplatz zu sparen** und die **Effizienz** beim Umgang mit **String**-Objekten zu verbessern. Im Wesentlichen speichert der String Pool **einmal erstellte String-Objekte** und ermöglicht es, dass diese wiederverwendet werden, anstatt sie bei jeder String-Erstellung erneut anzulegen.

Wie funktioniert der String Pool?

1. String-Literale:

- Wenn du einen String als Literal deklarierst, wird dieser String in den **String Pool** gelegt, falls er dort noch nicht existiert.
- Wenn bereits ein String mit demselben Inhalt im Pool vorhanden ist, wird kein neues String-Objekt erstellt. Stattdessen wird eine Referenz auf das bereits existierende String-Objekt zurückgegeben.

Beispiel:

```
String str1 = "Hello";  
String str2 = "Hello"; // str2 verweist auf dasselbe Objekt wie str1
```

Hier zeigt sowohl `str1` als auch `str2` auf das **gleiche** String-Objekt im String Pool, da beide dasselbe Literal "Hello" verwenden. Kein neues Objekt wird erstellt.

2. Speicheroptimierung:

- Durch die Wiederverwendung von String-Literalen spart Java Speicher. Anstatt mehrere String-Objekte mit demselben Inhalt zu erstellen, wird nur eine Instanz im Pool gespeichert, auf die mehrere Referenzen zeigen können.

3. new String():

- Wenn ein String mit dem Konstruktor `new` erstellt wird, z. B. `new String("Hello")`, wird ein **neues** Objekt im normalen Heap-Speicher erstellt, auch wenn bereits ein String-Objekt mit demselben Wert im Pool existiert. Dieses neue Objekt wird **nicht** im String Pool gespeichert, sondern separat im Heap.

Beispiel:

```
String str1 = "Hello"; // im String Pool  
String str2 = new String("Hello"); // neues Objekt im Heap, nicht im Po
```

In diesem Fall sind `str1` und `str2` verschiedene Objekte, obwohl sie denselben Inhalt haben.

intern() Methode:

- Die Methode `intern()` kann verwendet werden, um einen String explizit in den String Pool zu legen, falls er noch nicht vorhanden ist. Wenn du einen String mit `intern()` aufrufst, wird geprüft, ob ein String mit demselben Wert im Pool existiert. Falls ja, wird die Referenz darauf zurückgegeben. Falls nicht, wird der String in den Pool gelegt und dann die Referenz zurückgegeben.

Beispiel:

```
String str1 = new String("Hello");  
String str2 = str1.intern(); // str2 verweist jetzt auf das Objekt im P
```