

String compareTo () Método en Java con Ejemplo

¿Qué es el método compareTo () en Java?

El método compareTo () se usa para realizar una ordenación natural en una cadena. Clasificación natural significa el orden de clasificación que se aplica al objeto, por ejemplo, orden léxico para Cadena, orden numérico para ordenar enteros, etc.

El orden léxico no es más que orden alfabético. compareTo methods hace una comparación secuencial de letras en la cadena que tienen la misma posición.

compareTo está definido en la interfaz java.lang.Comparable

Sintaxis :

```
1 public int compareTo (String str)
```

Entrada de parámetros:

str – Este método solo acepta un solo tipo de datos de cadena de entrada.

Método de devoluciones:

Este método Java devuelve un tipo de datos int que se basa en la comparación lexicográfica entre dos cadenas.

- devuelve <0, entonces la cadena que llama al método es primero lexicográficamente
- devuelve == 0 entonces las dos cadenas son lexicográficamente equivalentes
- devuelve > 0, entonces el parámetro pasado al método compareTo es lexicográficamente el primero.

Entendamos con un ejemplo:

Ejemplo 1:

```
1 public class Sample_String {
2     public static void main(String[] args) {
3         String str_Sample = "a";
4         System.out.println("Comparar con 'a' b es:" + str_Sample.compareTo("b"));
5         str_Sample = "b";
6         System.out.println("Comparar con 'b' a es:" + str_Sample.compareTo("a"));
7         str_Sample = "b";
8         System.out.println("Comparar con 'b' b es:" + str_Sample.compareTo("b"));
9     }
10 }
```

Salida

Compare con 'a' b es: -1
Compare con 'b' a es: 1
Compare con 'b' b es: 0

Aquí,

- El carácter a viene antes que b alfabéticamente. Por lo tanto, el resultado es -1
- El carácter b viene antes que alfabéticamente. Por lo tanto, la salida es 1
- El carácter b es equivalente, por lo tanto, la salida es 0.

Puede usar el método Use “compareToIgnoreCase” en caso de que no desee que el resultado sea sensible a mayúsculas y minúsculas.

Ejemplo 2:

```
1 public class Sample_String {
2     public static void main(String[] args) {
3         // Compare con una cadena
4         String str_Sample = "RockStar";
5         System.out.println("Comparar con 'ROCKSTAR':" + str_Sample.compareTo("rockstar"));
6         // Comparar con - Ignorar caso
7         System.out.println("Comparar con 'ROCKSTAR' - Caso Ignorado:" + str_Sample.compareToIgnoreCase("rockstar"));
8     }
9 }
```

Salida

Comparar con 'ROCKSTAR': -32
Comparar con 'ROCKSTAR' – Caso ignorado: 0

¿Cuándo usar el método CompareTo ()?

CompareTo () se utiliza para comparar dos cadenas lexicográficamente. Cada carácter de ambas cadenas se convierte en un valor Unicode. Sin embargo, si ambas cadenas son iguales, este método devuelve 0 sino que solo da como resultado un valor negativo o positivo.

En este método, si la primera cadena siempre es lexicográficamente más alta que la segunda cadena, devuelve un número positivo.

si a1 > a2, devuelve un número negativo

si a1 < a2, devuelve un número positivo

si a1 == a2, devuelve 0

Ejemplo 3:

```
1 clase pública Compare {
2     public static void main(String[] args) {
3         Cadena s1 = "Guru1";
4         Cadena s2 = "Guru2";
5         System.out.println("Cadena 1:" + s1);
6         System.out.println("String 2:" + s2);
7
8         // Compara las dos cadenas.
9         int S = s1.compareTo(s2);
10
11        // Muestra los resultados de la comparación.
12
13        if (S < 0) {
14            System.out.println("\n" + s1 + "\n" + " is lexicographically higher than " + s2);
15        } else if (S == 0) {
16            System.out.println("\n" + s1 + "\n" + " is lexicographically equal to " + s2);
17        } else if (S > 0) {
18            System.out.println("\n" + s1 + "\n" + " is lexicographically less than " + s2);
19        }
20    }
21 }
```

Salida:

Cadena 1: Guru1
Cadena 2: Guru2
“Guru1” es lexicográficamente más alta que “Guru2”



Su primera aplicación usando MySQL y PHP



Vistas en MySQL Tutorial: crear, unir y soltar con ejemplos



MySQL JOINS Tutorial: INTERIOR, EXTERIOR, IZQUIERDA, DERECHA, CRUZADA



UNIÓN DE MySQL – Tutorial completo

← Prev

Next →

0

REPLIES

Leave a Reply

Want to join the discussion?
Feel free to contribute!

Name *

Email *

Website

Post Comment