

La clase String en Java

La clase String

En la API de Java tenemos la clase String.

Todas las cadenas que utilizamos en nuestros programas Java son instancias de esta clase.

Por lo tanto podemos afirmar que, en Java, **el tipo String es un tipo de referencia**.

```
String miSaludo = new String("Hola mundo");
```

Pero la clase String, por comodidad, admite la creación de objetos de manera literal, dándonos la falsa sensación de que se trata de valores primitivos:

```
String miSaludo = "Hola mundo"; //utilizar comillas dobles!
```

```
String miSaludo2 = "Quiero decir \"hola\""; //para utilizar el carácter comillas dobles se debe escapar
```

La clase String

Los caracteres que conforman un String se codifican usando Unicode

- Unicode proporciona una forma única de definir cada carácter en cada idioma hablado del mundo asignándole un número único.
- Se puede definir con diferentes codificaciones de caracteres como UTF-8, UTF-16, UTF-32, etc. Entre estos, UTF-8 es el más popular, ya que se utiliza en más del 90% de los sitios web, así como en la mayoría de los sistemas operativos modernos como Windows.

Son inmutables

- Una clase inmutable es simplemente una clase cuyas instancias no se pueden modificar.
- Una vez creado un String no podemos modificarlo!.

Caracteres de escape al usar String

Caracteres de escape	Descripción
\t	Inserta un tabulador
\b	Inserta un retroceso (backspace)
\n	Inserta una nueva línea
\r	Inserta un retorno de carro
\'	Inserta una comilla simple
\"	Inserta una comilla doble
\\	Inserta una barra invertida

Concatenar String

Para concatenar un String utilizaremos el operador suma:

```
String miSaludo = "Hola";
```

```
String miSaludo2 = "Mundo";
```

```
String miSaludoCompleto = miSaludo + " " + miSaludo2;
```

Comparar String

Para comparar dos String por referencia utilizaremos el operador ==

Para comparar dos String por valor utilizaremos el método equals()

```
String miSaludo = "Hola";
```

```
String miSaludo2 = new String("Hola");
```

```
miSaludo == miSaludo2; // false → son diferentes objetos
```

```
miSaludo.equals(miSaludo2); // true → las cadenas de ambos objetos son idénticas
```

Métodos de la clase `String` (I)

método	devuelve	descripción
<code>length()</code>	<code>int</code>	Número de caracteres de una cadena
<code>equals(String a)</code>	<code>boolean</code>	Compara si dos cadenas son iguales por valor
<code>equalsIgnoreCase(String a)</code>	<code>boolean</code>	Compara si dos cadenas son iguales sin tener en cuenta mayúsculas y minúsculas
<code>compareTo(String a)</code>	<code>int</code>	Compara dos cadenas devolviendo: <ul style="list-style-type: none">- Un valor negativo si la cadena es anterior a la cadena pasada en el argumento- 0 si ambas cadenas son iguales (valor)- Un valor positivo si la cadena es posterior a la cadena pasada en el argumento
<code>trim()</code>	<code>String</code>	Crea un nuevo objeto eliminando los espacios en blanco que pudiera haber al principio y al final de la cadena original
<code>charAt(int a)</code>	<code>char</code>	Extrae el carácter de la posición pasada en el argumento

Métodos de la clase `String` (II)

método	devuelve	descripción
<code>toCharArray()</code>	<code>char[]</code>	Convierte una cadena en un array de caracteres
<code>substring(int a,int b)</code>	<code>String</code>	Extrae la sub-cadena entre las posiciones “a” y “b”
<code>substring(int a)</code>	<code>String</code>	Extrae la sub-cadena entre la posición “a” hasta el final
<code>indexOf(String a)</code>	<code>int</code>	Indica en qué posición se encuentra el carácter o la cadena (por primera vez) buscando desde el principio
<code>lastIndexOf(String a)</code>	<code>int</code>	Indica en qué posición se encuentra el carácter o la cadena (por primera vez) buscando desde el final
<code>startsWith(String a)</code>	<code>boolean</code>	Indica si la cadena comienza con la cadena “a”
<code>endsWith(String a)</code>	<code>boolean</code>	Indica si la cadena termina con la cadena “a”
<code>split(String a)</code>	<code>String[]</code>	Divide la cadena en varias sub-cadenas utilizando el patrón “a” como separador