

# Introducción a La Clase String

## La Clase String. Introducción

- Para Java las cadenas de texto son objetos especiales.
- Los textos deben manejarse creando objetos de tipo String.
- Ejemplo:

```
String texto1 = "¡Prueba de texto!";
```

- Las cadenas pueden ocupar varias líneas utilizando el operador de concatenación `“+”`

```
String texto2 ="Este es un texto que ocupa “ + “varias líneas, no obstante se puede “ + “perfectamente encadenar”;
```

## Comparación entre objetos String

- En elementos de la clase String **No se puede usar** el operador de comparación : `==`.
- En su lugar se deben utilizar estas expresiones:
  - `cadena1.equals(cadena2)`.
    - El resultado es true si la cadena1 es igual a la cadena2. Ambas cadenas son variables de tipo String.
  - `cadena1.equalsIgnoreCase(cadena2)`.
    - Como la anterior, pero en este caso no se tienen en cuenta mayúsculas y minúsculas.
  - `s1.compareTo(s2)`.
    - Compara ambas cadenas, considerando el orden alfabético.
      - Si la primera cadena es mayor en orden alfabético que la segunda devuelve 1 o un número positivo,
      - Si son iguales devuelve 0
      - Si es la segunda la mayor devuelve -1 o un número negativo.
  - `s1.compareToIgnoreCase(s2)`.
    - Igual que la anterior, sólo que además ignora las mayúsculas.
- Hay que tener en cuenta que el orden no es el del alfabeto español, sino que usa la tabla ASCII, en esa tabla la letra ñ es mucho mayor que la o.

## El método valueOf

- `String.valueOf(valor)`
- Este método pertenece no sólo a la clase String, sino a otras y siempre es un método que convierte valores de una clase a otra.
- En el caso de los objetos String, permite convertir valores que no son de cadena a forma de cadena.
- **Ejemplo :**

```
//Ejemplo
String numero = String.valueOf(1234);
String fecha  = String.valueOf(new Date());
```

- En el ejemplo se observa que este método pertenece a la clase String directamente.
- No hay que utilizar el nombre del objeto creado (como se verá más adelante, es un método estático).

# Métodos más comunes de la Clase String

//METODO=====	DESCRIPCION=====
char charAt(int index)	Proporciona el carácter que está en la posición dada por el entero index.
int compareTo(String s)	Compara las dos cadenas. Devuelve un valor menor que cero si la cadena s es mayor que la original, devuelve 0 si son igua
int compareToIgnoreCase(String s)	Compara dos cadenas, pero no tiene e cuenta si el texto es mayúsculas o no.
String concat(String s)	Añade la cadena s a la cadena original.
String copyValueOf(char[] data)	Produce un objeto String que es igual al array de caracteres data.
boolean endsWith(String s)	Devuelve true si la cadena termina con el texto s
boolean equals(String s)	Compara ambas cadenas, devuelve true si son iguales
boolean equalsIgnoreCase(String s)	Compara ambas cadenas sin tener en cuenta las mayúsculas y las minúsculas.
byte[] getBytes()	Devuelve un array de caracteres que toma a partir de la cadena de texto
void getBytes(int srcBegin, int srcEnd,char[] dest, int dstBegin);	Almacena el contenido de la cadena en el array de caracteres dest. Toma los caracteresdesde la posición
int indexOf(String s)	Devuelve la posición en la cadena del texto s. Valor -1 si no lo encuentra.
int indexOf(String s, int primeraPos)	Devuelve la posición en la cadena del texto s, empezando a buscar desde la posición PrimeraPos
int lastIndexOf(String s)	Devuelve la última posición en la cadena del texto s. Valor -1 si no lo encuentra.
int lastIndexOf(String s, int primeraPos)	Devuelve la última posición en la cadena del texto s, empezando a buscar desde la posición PrimeraPos
int length()	Devuelve la longitud en caracteres de la cadena.
boolean startsWith(String s)	Devuelve true si la cadena comienza con el texto s.
String <b>replace</b> (char carAnterior, char ncarNuevo)	Devuelve una cadena <b>idéntica</b> al original pero que ha cambiando los caracteres iguales a carAnterior por carNuevo
<b>String</b> replaceFirst( <b>String</b> str1, <b>String</b> str2)	Cambia la primera aparición de la cadena str1 por la cadena str2
<b>String</b> replaceAll( <b>String</b> str1, <b>String</b> str2)	Cambia la todas las apariciones de la cadena uno por la cadena dos
<b>String</b> split( <b>String</b> str[, int max])	Fracciona un <b>String</b> en varios hasta <b>max</b> según el patrón <b>str</b> o una expresión regular en <b>str</b>
<b>String</b> substring(int primeraPos, int segundaPos)	Devuelve el texto que va desde primeraPos a segunaPos.
char[] toCharArray()	Devuelve un <b>array</b> de caracteres a partir de la cadena dada
<b>String</b> toLowerCase()	Convierte la cadena a <b>minúsculas</b>
<b>String</b> toLowerCase(Locale <b>local</b> )	Lo mismo pero siguiendo las instrucciones del argumento <b>local</b>
<b>String</b> toUpperCase()	Convierte la cadena a mayúsculas
<b>String</b> toUpperCase(Locale <b>local</b> )	Lo mismo pero siguiendo las instrucciones del argumento <b>local</b>
<b>String</b> trim()	Elimina los blancos que tenga la cadena tanto por delante como por detrás
<b>Static String</b> valueOf(tipo elemento)	Devuelve la cadena que representa el valor elemento. Si elemento es booleano, por ejemplo devolvería una cadena con el va

## Ejemplos

- Ejemplo :

```
//Ejemplo de Métodos de la clase String
String cadena = "Buenos Días";
int posicion = 1;
System.out.printf( "El caracter número %d de la cadena :\"%s\" es :\"%c\\n", posicion, cadena, cadena.charAt(posicion) );
```

- Salida

- El carácter número 1 de la cadena : "Buenos Días" es : 'u'
  - Los String empiezan en el 0, después el 1, 2, 3.... y así sucesivamente.

Introducción a La Clase String - [Andrés Ramos González](#) - IES Alonso de Avellaneda

**Este obra está bajo una licencia:** [Creative Commons Reconocimiento-CompartirIgual 3.0](#)

