CADENAS DE TEXTO EN JAVA.

ÍNDICE

INTRODUCCIÓN

La clase String nos permitirá almacenar cadenas de caracteres. Realmente es una secuencia de 0 a n caracteres. Es dinámica, por lo tanto le podemos asignar diferentes valores (cadenas), con diferente tamaño. Esto se debe a la asignación dinámica que se hace en memoria.

Creación

Aunque como ya dije anteriormente es una Clase, podemos tratarla como un dato primitivo inicialmente. Una vez asignado un valor este es inmodificable. Cada vez que yo asigno un nuevo valor (nueva cadena), realmente provoca una nueva instanciación interna pero es transparente para nosotros.

```
public class usoString{
  public static void main(String[] args){
    String modulo="programación";
    String ciclo= new String();
    ciclo="Desarrollo de Aplicaciones Web";
  }
}
```

Asignación de valores

Las cadenas de caracteres o Strings en Java, se representan con comillas dobles. Si queremos que la propia " sea un carácter en sí, y no el final de la cadena, hay que anteponer la secuencia de escape .

```
System.out.println("*****ASIGNACION DE VALORES*****");
String introduccion;
introduccion="Trataremos inicialmente los 'String' para el manejo de cadenas. Son
muy \"IMPORTANTES\"");
```

También podemos utilizar los String de las siguientes maneras:

```
System.out.println("TBO");
String TBO = "TBO";
System.out.println("Me gusta " + TBO);
String c = "Me gusta".substring(3, 6);//nos muestra la cadena gus
String d = TBO.substring(0,2);//nos muestra la cadena TB
```

Extracción de un carácter individual

Cada uno de los caracteres que forman parte de la cadena, tiene asignado una posición. El primer carácter de la cadena ocupa la posición 0, el segundo carácter la posición 1, y así sucesivamente.

Con el método charAt(posición), podemos acceder al carácter que ocupa dicha posición.

```
public class usoString{
  public static void main(String[] args){
    String modulo="programación";
    String ciclo= new String();
    ciclo="Desarrollo de Aplicaciones Web";

    char c=ciclo.charAt(0);
    System.out.println("El primer caracter de ciclo es: "+c);
    System.out.println("El quinto caracter de ciclo es: "+ciclo.charAt(4));
  }
}
```

Obtener la longitud de la cadena

Otra función interesante en el manejo de cadenas es la que nos proporciona la longitud de la cadena. Incluye también los espacios en blanco ocupados dentro de la cadena. Para ello hacemos uso del método length().

```
public class usoString{
  public static void main(String[] args){
    String modulo="programación";
    String ciclo= new String();
    ciclo="Desarrollo de Aplicaciones Web";
    System.out.println("*****Longitud de la cadena*****);
    System.out.println("La longitud de la cadena ciclo es: "+ciclo.length());
  }
}
```

Descomposición de una cadena

Es un método que permite extraer una subcadena dada, indicando la posición inicial y final de los caracteres a extraer. El método es substring(posIni, posFin); posIni indica la posición desde donde empiezas a extraer la cadena posFin es el número de elementos que quieras recuperar desde la posición inicial

```
System.out.println("*****Extraccion de una subcadena*****");
System.out.println("Subcadena de la cadena ciclo de las cuatro primeros caracteres
que podemos nombrar como 0, 1, 2, 3 (el número de caracteres es cuatro). El
carácter final indicado en la llamada al método, el número 4, queda excluido del
substring.: "+ciclo.substring(0,4);
```

Igualdad de dos cadenas

Si queremos averiguar si dos cadenas son iguales, o lo podemos hacer con el operador == como con el resto de los tipos simples. Para ello tenemos que utilizar el método equals, que devuelve true si ambas cadenas son iguales.

```
System.out.println("*****COMPARACIÓN ENTRE CADENAS*****);
String texto1="Buenos días";
String texto2=new String("Buenos días");
System.out.println("¿Las cadenas son iguales?: "+texto1.equals(texto2));
System.out.println("¿Las cadenas son iguales?: " + texto1==texto2); // El
resultado de esta instrucción puede ser incorrecto.
```

Al preguntar si dos cadenas son iguales, distingue si está en mayúsculas o minúsculas. Si necesitamos preguntar sin distinguir mayúsculas de minúsculas, utilizamos equalsIgnoreCase().

```
System.out.println("*****COMPARACIÓN ENTRE CADENAS*****);
String texto1="Buenos días";
String texto2=new String("BUENOS DÍAS");
System.out.println("¿Las cadenas son iguales?: "+texto1.equalsIgnoreCase(texto2));
```

Otro modo de comparar cadenas, para saber si una es mayor que la otra, es con el método compareTo. Este método devuelve un entero >0, <0 o =0. Veamos un ejemplo:

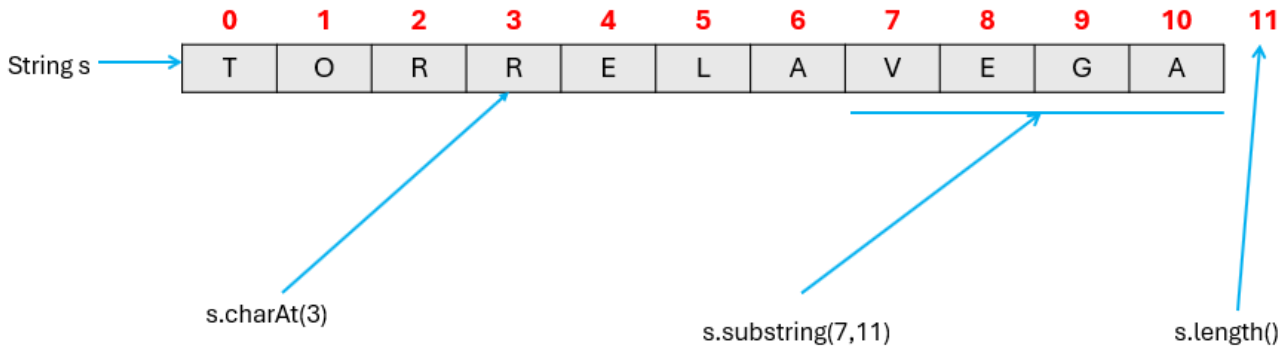
```
System.out.println("*****COMPARACIÓN ENTRE CADENAS*****);
String texto1="Buenos días";
String texto2=new String("Buenos dias");
if (texto1.compareTo(texto2)>0)
  System.out.println("texto1 es superior a texto2");
else if(texto1.compareTo(texto2)<0)
  System.out.println("texto1 es inferior a texto2");
else
  System.out.println("texto1 y texto2 son iguales");
```

Otras funciones de Strings

Existen un montón de funciones para trabajar con Strings en Java. Si os vais a la API de Java, podéis buscar la clase String y veréis todas las que hay. Algunas de ellas son las siguientes:

- **startsWith(subcadena)** y **endsWith(subcadena)**: para comprobar si una cadena comienza o finaliza con una subcadena determinada.
- **trim()**: elimina los espacios en blanco de una cadena que tenga por delante o por detrás. No elimina los espacios intermedios.
- **toUpperCase()** y **toLowerCase()**: me permite cambiar todos los caracteres por mayúsculas o minúsculas.
- **indexOf(cadenaABuscar)**: permite buscar una cadena dentro de otra.
- **indexOf(cadenaABuscar,posicion)**: igual que la anterior, pero desde una posición determinada.
- **replace(cadenaABuscar, cadenaSustituta)**: permite reemplazar una cadena por otra.
- **lastIndexOf (String cad)**: Retorna la posición de la ultima ocurrencia de la cadena dada como parámetro.
- **lastIndexOf (String cad,int ini)**: Retorna la posición de la última ocurrencia de la cadena dada como parámetro buscando en retroceso a partir de la posición dada como parámetro.

Como ya hemos comentado, existen muchas más funciones que podéis probar e investigar por vuestra cuenta. Sirva esta imagen como guía.



EJERCICIOS

Hoja de ejercicios 1