

```

/**
 * Rimpiazza.java
 *
 * Rimpiazza application
 *
 * @author 3BI
 * @version 1.00 2018/5/3
 */
class Rimpiazza{
    private String x;
    private String y;

    public Rimpiazza(){
        x = "ciao";
        System.out.println ("La stringa originale: " + x);

        y = new String(x); // crea per copia (se creata con new impossibile poi usare y==x)
        System.out.println ("La stringa copiata: " + y);
    }

    public void ini(){
        for (int i = 0; i<x.length(); i++)
            y = y.replace (x.charAt(i), '-'); // rimpiazza tutte le occorrenze
    } // metodo alternativo: creare accodando o riempiendo

    public void stampa(){
        System.out.println ("La stringa riempita: " + y);
    }

    public static void main (String [] args){
        Rimpiazza o = new Rimpiazza();
        o.ini();
        o.stampa();
    }
}

```



```

General Output
-----Configuration
La stringa originale: ciao
La stringa copiata: ciao
La stringa riempita: ----
Process completed.

```

**nb:** un oggetto **String** visualizza a monitor il contenuto con istruzione

```
System.out.println (<nomeString>);
```

un array di char visualizza a monitor l'indirizzo esadecimale

```
char[] <nomeArray> = y.toCharArray();
System.out.println ("Array di char: " + <nomeArray>);
```

```
|Array di char: [C@15db9742
```

→ Address Bus 32 bit  
se 8 cifre esadecimali  
(infatti 4\*8 = 32 bit)

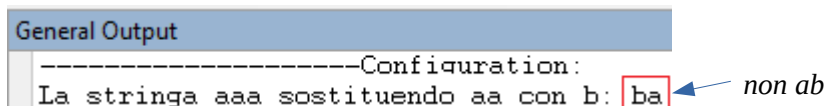
String	<code>replace(char oldChar, char newChar)</code> Returns a string resulting from replacing all occurrences of oldChar in this string with newChar.
String	<code>replace(CharSequence target, CharSequence replacement)</code> Replaces each substring of this string that matches the literal target sequence with the specified literal replacement sequence.
String	<code>replaceAll(String regex, String replacement)</code> Replaces each substring of this string that matches the given <b>regular expression</b> with the given replacement.
String	<code>replaceFirst(String regex, String replacement)</code> Replaces the first substring of this string that matches the given <b>regular expression</b> with the given replacement.

String **replace** (char *old*, char *new*) rimpiazza **tutte** le occorrenze del carattere creando una nuova stringa

String **replace** (*sequenzaChar\_old*, *sequenzaChar\_new*) rimpiazza **tutte** le occorrenze della sequenza di caratteri (anche dinamica) creando una nuova stringa (ver. 1.5) e non interpreta le **espressioni regolari**

scandisce dall'inizio ad es:

```
String s = "aaa";
String n = s.replace ("aa", "b"); // scandendo da inizio aa e poi a
System.out.println ("La stringa " + s + " sostituendo aa con b: " + n);
```

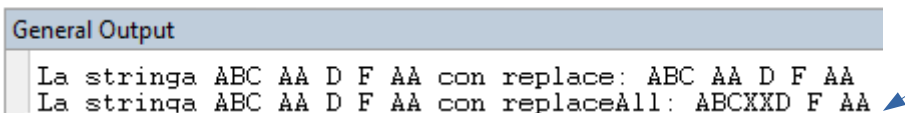


String **replaceAll** (String *old*, String *new*) rimpiazza **tutte** le occorrenze della sequenza di caratteri creando una nuova stringa (ver. 1.4) ed interpreta le **espressioni regolari (regex)**

```
String s2 = "ABC AA D F AA";
String n = s2.replace ("\\WAA\\W", "XX"); // non viene valutata la regex → nessuna sostituzione
System.out.println ("La stringa " + s2 + " con replace: " + n);
n = s2.replaceAll ("\\WAA\\W", "XX")
```

// \\W significa non word-char  
// cioè carattere speciale  
// viene valutato → sostituito

```
System.out.println ("La stringa " + s2 + " con replaceAll: " + n);
```



String **replaceFirst** (String *old*, String *new*) rimpiazza la **prima** occorrenza della sequenza di caratteri creando una nuova stringa (ver. 1.4) ed interpreta le **espressioni regolari**

```
La stringa aaaa con replaceFirst (sostituisce primo aa con b): baa
```

## Usare metodo della classe String per convertire da oggetto String ad array di char

```
char[]
```

```
toCharArray()
```

Converts this string to a new character array.

```
Array di char  
c i a o
```

```
String x = "ciao";  
char [] a = x.toCharArray();  
System.out.println ("\n\nArray di char");  
  
for(int i=0; i<a.length; i++) // stampare contenuto con ciclo  
    System.out.print(a[i] + " ");
```

## Creare un oggetto di tipo String accodando

Metodo alternativo per creare un oggetto di tipo String della stessa lunghezza di una stringa nota, riempito con carattere '-' (anche in tal caso, [impossibile](#) poi usare `y+=x`)

```
String x = "ciao";  
String y = "";  
for (int i=0; i<x.length() ; i++)  
    y = y + '-';  
System.out.println ("La stringa accodata: " + y);
```

## oppure riempiendo con uso di metodo della classe Arrays l'array di char

```
String x = "ciao";  
String y = "";  
char [] a = x.toCharArray();  
Arrays.fill (a, '-');  
System.out.println ("L'array riempito con fill: " + Arrays.toString(a));  
for (int i=0; i<a.length ; i++)  
    y = y + a[i];  
System.out.println ("La stringa riempita con fill: " + y);
```

```
L'array riempito con fill: [-, -, -, -]  
La stringa riempita con fill: ----
```

## Stampare il contenuto di un array

### 1) con ciclo

```
Stampa array di String con ciclo  
ABC AA D
```

```
String [] s3 = {"ABC", "AA", "D"};  
System.out.println ("\nStampa array di String con ciclo");  
for (int i=0; i<s3.length; i++)  
    System.out.print(s3[i] + " ");
```

### 2) con metodo statico della classe [Arrays](#)

```
Uso Arrays [ABC, AA, D]
```

```
String [] s3 = {"ABC", "AA", "D"};  
System.out.println ("\nUso Arrays " + Arrays.toString (s3));
```

## Conversioni di tipo

### 1) da String a valore numerico - **classi wrapper**

Per convertire una variabile da tipo stringa a tipo numerico si usano i metodi **parseInt()** della classe **Integer**, **parseFloat()** della classe **Float** o **parseDouble()** della classe **Double** etc.

```
String s0 ="1";
int i0 = Integer.parseInt(s0);
System.out.println("La stringa " + s0 + " convertita in intero " + i0);
System.out.println("si somma a 3, ottenendo " + (i0+3));
```

```
General Output
-----Configuration:
La stringa 1 convertita in intero 1
si somma a 3, ottenendo 4
```

### 2) da valore numerico a String:

#### conversione implicita con accodamento

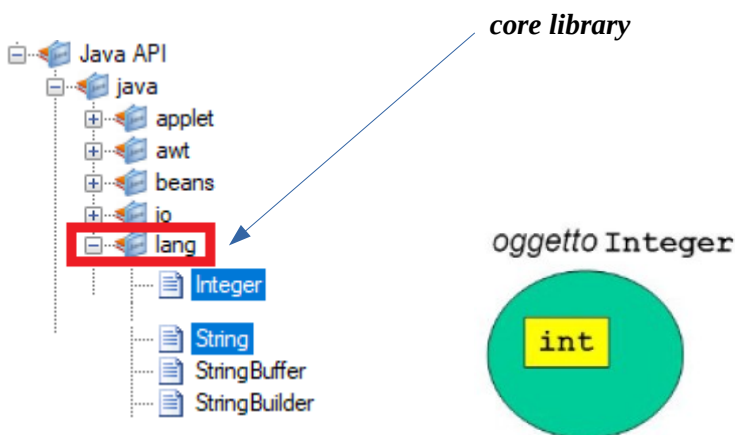
```
General Output
10
Conversione implicita: 10
```

```
int i = 10;
System.out.println(i);
System.out.println("Conversione implicita: " + i);
```

Per convertire una variabile da tipo numerico a tipo stringa si usa il metodo **valueOf()** della classe **String** oppure i metodi **toString()** della classe wrapper opportuna (*Integer* per convertire da intero a tipo stringa etc..)

```
Converto con String.valueOf: 10
String s1 = String.valueOf(i);
System.out.println("Converto con String.valueOf: " + s1);
```

```
Converto con Integer.toString: 10
String s = Integer.toString(i);
System.out.println("Converto con Integer.toString: " + s);
```



Tipo primitivo	Classe "wrapper" corrispondente
boolean	Boolean
char	Character
byte	Byte
short	Short
int	Integer
long	Long
float	Float
double	Double