

# Programació I

## Tema 2 (continuació)- Classe String de Java



UNIVERSITAT DE  
BARCELONA

Grau en Enginyeria Informàtica  
Facultat de Matemàtiques i Informàtica  
Curs 18-19

# Classe String

<https://docs.oracle.com/javase/8/docs/api/java/lang/String.html>

## Declaració

- Classe de Java que representa un conjunt de caràcters (cadena de caràcters).
- Declaració: `String nomVariable;`  
`//nomVariable encara no té memòria assignada i no té cap valor`

# Classe String

## Assignació d'un valor

- Formes diferents d'assignar un valor a un objecte String, `String str;`

- 1 Assignar la cadena null: `str = new String ();`
- 2 Assignar la cadena "Hola": `str = new String ("Hola");`
- 3 La forma `str = "Hola";` és equivalent a l'anterior.
- 4 Assignar la cadena entrada per teclat (mètode `nextLine()`):

```
Scanner sc;  
sc = new Scanner(System.in);  
str = new String(sc.nextLine());
```

- 5 Assignar la cadena, fins al primer espai en blanc, de l'entrada per teclat (mètode `next()`):

```
Scanner sc;  
sc = new Scanner(System.in);  
str = new String(sc.next());
```

# Classe String

## Consultes sobre un String

- El mètode `length()` retorna el nombre de caràcters d'una cadena:

```
String str = "Hola";  
int longitud;  
longitud = str.length();
```

```
//longitud val 4
```

- Per a consultar el valor del caràcter que està en la posició `i`, considerant que el primer caràcter de tots està a la posició zero:

```
String str = "Hola";  
char c;  
int i = 0;  
c = str.charAt(i);
```

```
//per a i igual a 0 assigna a c el caràcter 'H', per a i igual a 1 assignaria  
'o', etc.
```

# Classe String

## Comparació de String

- Per a saber si dos cadenes (dos objectes de la classe `String`) són iguals, s'ha de cridar al mètode `equals()`:

```
String str1 = "Hola";  
String str2 = "Hola";  
System.out.println(str1.equals(str2));  
//CORRECTE. imprimeix true
```

- **NO ES POT utilitzar l'operador `==`.** Si per error utilitzéssim `==`, estaríem comparant referències, NO contingut, i nosaltres volem comparar el contingut!

```
String str1 = "Hola";  
String str2 = "Hola";  
System.out.println(str1 == str2);  
//INCORRECTE. imprimeix false
```

# Classe String

## Concatenació de String

- Per a concatenar Strings s'utilitza l'operador +:

```
String str1 = "Hola";  
String str2 = "Adeu";  
System.out.println(str1 + " " + str2);
```

# Classe String

## Immutable

- Tot objecte de la classe `String` és **immutable** (inalterable), això vol dir que el seu estat no pot modificar-se després de la seva creació.
- Dibuixem a la pissarra què passa quan fem:  

```
String str = new String("Hola"); str += "Adeu";
```
- Existeix una classe alternativa, `StringBuffer`, de Strings mutables (alterables).

<https://docs.oracle.com/javase/8/docs/api/java/lang/StringBuffer.html>.

- Ara, dibuixem què passa quan fem:

```
Ex: StringBuffer str = new StringBuffer("Hola");  
str.append("Adeu");
```

# Codi Java amb exemples de crides a mètodes de la classe String I

```
public class ClasseString {
    public static void main (String[] args) {
        String s1 = String.valueOf(1);
        String s2 = String.valueOf(2);
        System.out.println("s1+s2=" + s1 + s2);           // 12
        String s3 = new String("Hola");
        String s4 = "Hola";
        System.out.println("s3.length=" + s3.length());   // 4
        System.out.println("s3.majúscules=" + s3.toUpperCase()); // HOLA
        System.out.println("s3.endsWith(a)=" + s3.endsWith("a")); // true
        System.out.println("s3.startsWith(a)=" + s3.startsWith("a")); // false
        System.out.println("substring(1,2)=" + s3.substring(1,2)); // o
        System.out.println("s3.equals_s4=" + s3.equals(s4)); // true
        System.out.println("s3_==_s4=" + (s3 == s4));     // false
        System.out.println("Caràcter_a_la_posició_0:" + s3.charAt(0)); // H
        System.out.println("minúscules_" + s3.toLowerCase()); // hola
    }
}
```