

El tipus de dades String

Més coses d'aquelles que en algun lloc s'han d'explicar...

El tipus de dades String

A més dels tipus de dades ja coneguts:

int, double, char, boolean...

Que són els que anomenem tipus de dades **primitius**, existeix un altre tipus de dades anomenat **String**

Les variables de tipus **String** **poden** **contenir** **cadena**s **de** **caràcters** **de** **qualsevol** **longitud**.

La classe String està dins del paquet java.lang

El tipus de dades String

```
public static void main (String [] args) {  
  
    // es poden declarar variables de tipus String  
    String cad1, cad2;  
  
    // els valors literals s'escriuen entre cometes dobles  
    cad1 = "Hola, que tal estàs ";  
  
    // es poden llegir amb Keyboard.readString  
    cad2 = Keyboard.readString();  
  
    // de valors literal ja n'havíem vist abans:  
    System.out.println("Resultat: ");  
  
    // es poden escriure amb print i println  
    System.out.print(cad1);  
    System.out.println(cad2);  
  
    // es poden concatenar amb l'operador +  
    System.out.println("Resultat: "+cad1+cad2);  
}
```

El tipus de dades String

String vs. char

El tipus de dades **String** no s'ha de confondre amb el tipus **char**. **Són tipus diferents!!!!**

- Les variables **String** poden contenir cadenes de qualsevol longitud.
- Les variables **char** només poden contenir **un** caràcter.
- Els literals **String** van entre dobles cometes
- Els literals **char** van entre cometes simples

El tipus de dades String

String vs. char

```
public static void main (String [] args) {  
  
    char car;  
    String cadCar;  
  
    car = 'a' ;           // CORRECTE un sol caràcter  
    car = ' ' ;           // CORRECTE un sol caràcter (blanc)  
    cadCar = "a" ;        // CORRECTE cadena d'un sol caràcter  
    cadCar = "" ;         // CORRECTE "" és la cadena buida  
    cadCar = " " ;        // CORRECTE cadena d'un sol caràcter  
    cadCar = "hola" ;     // CORRECTE cadena de múltiples caràcters  
  
    car = ' ' ;           // INCORRECTE no es el mateix ' ' que ' '  
    car = "a" ;           // INCORRECTE tipus incompatibles  
    car = 'hola' ;        // INCORRECTE char només admet 1 caràcter  
    car = cadCar ;         // INCORRECTE tipus incompatibles  
    cadCar = 'a' ;         // INCORRECTE tipus incompatible  
    cadCar = car ;         // INCORRECTE tipus incompatible  
  
    cadCar = cadCar + car; // CORRECTE + és l'operador de concatenació  
}
```

String i els operadors de comparació

- Els operadors de comparació `<`, `<=`, `>`, i `>=` **no es poden utilitzar** amb variables de tipus `String`.
- Els operadors `==` i `!=` **si** que es poden utilitzar però **no proporcionen el resultat esperat**. No comparen les cadenes de caràcters sinó quelcom diferent. En parlem després
- El tipus `String` és una classe. **L'API de Java proporciona funcions (mètodes) per manipular-les**. En teniu un llistat a continuació:

La classe String

Method Summary

char	charAt (int index) Returns the char value at the specified index.
int	codePointAt (int index) Returns the character (Unicode code point) at the specified index.
int	codePointBefore (int index) Returns the character (Unicode code point) before the specified index.
int	codePointCount (int beginIndex, int endIndex) Returns the number of Unicode code points in the specified text range of this String.
int	compareTo (String anotherString) Compares two strings lexicographically.
int	compareToIgnoreCase (String str) Compares two strings lexicographically, ignoring case differences.
String	concat (String str) Concatenates the specified string to the end of this string.
boolean	contains (CharSequence s) Returns true if and only if this string contains the specified sequence of char values.
boolean	contentEquals (CharSequence cs) Compares this string to the specified CharSequence.

tipus
return

nom

parametres

Explicació del què fa

La classe String

boolean	<code>contains(CharSequence s)</code> Returns true if and only if this string contains the specified sequence of char values.
boolean	<code>contentEquals(CharSequence cs)</code> Compares this string to the specified CharSequence.
boolean	<code>contentEquals(StringBuffer sb)</code> Compares this string to the specified StringBuffer.
static String	<code>copyValueOf(char[] data)</code> Returns a String that represents the character sequence in the array specified.
static String	<code>copyValueOf(char[] data, int offset, int count)</code> Returns a String that represents the character sequence in the array specified.
boolean	<code>endsWith(String suffix)</code> Tests if this string ends with the specified suffix.
boolean	<code>equals(Object anObject)</code> Compares this string to the specified object.
boolean	<code>equalsIgnoreCase(String anotherString)</code> Compares this String to another String, ignoring case considerations.
static String	<code>format(Locale l, String format, Object... args)</code> Returns a formatted string using the specified locale, format string, and arguments.
static String	<code>format(String format, Object... args)</code> Returns a formatted string using the specified format string and arguments.

La classe String

	each character starting at the specified index.
int	<code>length()</code> Returns the length of this string.
boolean	<code>matches(String regex)</code> Tells whether or not this string matches the given regular expression .
int	<code>offsetByCodePoints(int index, int codePointOffset)</code> Returns the index within this String that is offset from the given index by codePointOffset code points.
boolean	<code>regionMatches(boolean ignoreCase, int toffset, String other, int ooffset, int len)</code> Tests if two string regions are equal.
boolean	<code>regionMatches(int toffset, String other, int ooffset, int len)</code> Tests if two string regions are equal.
String	<code>replace(char oldChar, char newChar)</code> Returns a new 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 regular expression with the given replacement.

El tipus de dades String

- Aquestes “funcions” (mètodes) de la classe String són **diferents** a les que heu usat de la classe Math!!!. No porten el prefix de static

- Sintaxi **No és el nom de la classe**

objecte.nomMetode(arguments)

- Exemples

```
String frase=“Hola bon dia”;
```

```
System.out.println(frase.length()); → 12
```

int	<u>length()</u> Returns the length of this string.
-----	---

Comparació de String

```
public class ExempleString {  
    public static void main(String args[]){  
        String mot1,mot2,mot3;  
        int r;  
        mot1="Ordinador";  
        mot2=Keyboard.readString();  
        System.out.println("Els mots són: " + mot1 + " " + mot2);  
        mot3=mot1+mot2;  
        System.out.println("El contingut de la suma és: " + mot3);  
        r= mot1.compareTo(mot2);  
        System.out.println("El resultat de la comparació és " + r);  
        System.out.println("El resultat de la comparació és " + mot1.equals(mot2));  
        System.out.println("La llargaria de la cadena mot3 és " + mot3.length());  
    }  
}
```

<terminated> ExempleString [Java Application] C:\Program Files\Java\jre1.8.0_101\bin\javaw

Co

bon dia avui és dilluns

Els mots són: Ordinador bon dia avui és dilluns

El contingut de la suma és: Ordinadorbon dia avui és dilluns

El resultat de la comparació és -19

El resultat de la comparació és false

La llargaria de la cadena mot3 és 32

```
public class Exemp
    public static
        String mot
        int r;
        mot1="Ordinador";
        mot2=Keyboard.readString();
        System.out.println("Els mots són: " + mot1 + " " + mot2);
        mot3=mot1+mot2;
        System.out.println("El contingut de la suma és: " + mot3);
        r= mot1.compareTo(mot2);
        System.out.println("El resultat de la comparació és " + r);
        System.out.println("El resultat de la comparació és " + mot1.equals(mot2));
        System.out.println("La llargaria de la cadena mot3 és " + mot3.length());
    }
}
```

Comp

<terminated> ExempleString [Java Application] C:\Program Files\Java\jre1

Avui fa sol

Els mots són: Ordinador Avui fa sol

El contingut de la suma és: OrdinadorAvui fa sol

El resultat de la comparació és 14

El resultat de la comparació és false

La llargaria de la cadena mot3 és 20

```
public class ExempleString
{
    public static void main
    {
        String mot1,mot2,mot3;
        int r;
        mot1="Ordinador";
        mot2=Keyboard.readString();
        System.out.println("Els mots són: " + mot1 + " " + mot2);
        mot3=mot1+mot2;
        System.out.println("El contingut de la suma és: " + mot3);
        r= mot1.compareTo(mot2);
        System.out.println("El resultat de la comparació és " + r);
        System.out.println("El resultat de la comparació és " + mot1.equals(mot2));
        System.out.println("La llargaria de la cadena mot3 és " + mot3.length());
    }
}
```