

## CLASE MATH JAVA

<code>Math.abs( x )</code>	para <code>int</code> , <code>long</code> , <code>float</code> y <code>double</code>
<code>Math.sin( double a )</code>	devuelve el seno del ángulo <code>a</code> en radianes
<code>Math.cos( double a )</code>	devuelve el coseno del ángulo <code>a</code> en radianes
<code>Math.tan( double a )</code>	devuelve la tangente del ángulo <code>a</code> en radianes
<code>Math.asin( double r )</code>	devuelve el ángulo cuyo seno es <code>r</code>
<code>Math.acos( double r )</code>	devuelve el ángulo cuyo coseno es <code>r</code>
<code>Math.atan( double r )</code>	devuelve el ángulo cuya tangente es <code>r</code>
<code>Math.atan2(double a,double b)</code>	devuelve el ángulo cuya tangente es <code>a/b</code>
<code>Math.exp( double x )</code>	devuelve <code>e</code> elevado a <code>x</code>
<code>Math.log( double x )</code>	devuelve el logaritmo natural de <code>x</code>
<code>Math.sqrt( double x )</code>	devuelve la raíz cuadrada de <code>x</code>
<code>Math.ceil( double a )</code>	devuelve el número completo más pequeño mayor o igual que <code>a</code>
<code>Math.floor( double a )</code>	devuelve el número completo más grande menor o igual que <code>a</code>
<code>Math rint( double a )</code>	devuelve el valor <code>double</code> truncado de <code>a</code>
<code>Math.pow( double x,double y )</code>	devuelve <code>y</code> elevado a <code>x</code>
<code>Math.round( x )</code>	para <code>double</code> y <code>float</code>
<code>Math.random()</code>	devuelve un <code>double</code>
<code>Math.max( a,b )</code>	para <code>int</code> , <code>long</code> , <code>float</code> y <code>double</code>
<code>Math.min( a,b )</code>	para <code>int</code> , <code>long</code> , <code>float</code> y <code>double</code>
<code>Math.E</code>	para la base exponencial, aproximadamente 2.72
<code>Math.PI</code>	para PI, aproximadamente 3.14

Algunos ejemplos:

Funciones Matemáticas	Significado	Ejemplo de uso	Resultado
<code>abs</code>	Valor absoluto	<code>int x = Math.abs(2.3);</code>	<code>x = 2;</code>
<code>atan</code>	Arcotangente	<code>double x = Math.atan(1);</code>	<code>x = 0.78539816339744;</code>
<code>sin</code>	Seno	<code>double x = Math.sin(0.5);</code>	<code>x = 0.4794255386042;</code>
<code>cos</code>	Coseno	<code>double x = Math.cos(0.5);</code>	<code>x = 0.87758256189037;</code>
<code>tan</code>	Tangente	<code>double x = Math.tan(0.5);</code>	<code>x = 0.54630248984379;</code>
<code>exp</code>	Exponenciación neperiana	<code>double x = Math.exp(1);</code>	<code>x = 2.71828182845904;</code>
<code>log</code>	Logaritmo neperiano	<code>double x = Math.log(2.7172);</code>	<code>x = 0.99960193833500;</code>
<code>pow</code>	Potencia	<code>double x = Math.pow(2.3);</code>	<code>x = 8.0;</code>
<code>round</code>	Redondeo	<code>double x = Math.round(2.5);</code>	<code>x = 3;</code>
<code>random</code>	Número aleatorio	<code>double x = Math.random();</code>	<code>x = 0.20614522323378;</code>

**METODOS PARA LA CLASE STRING**

**Int**     **length( )** Retorna la cantidad de caracteres que tiene la String.

**int**     **IndexOf(Stringóchar)** Retorna el índice o posición donde aparece por primera vez la String o char que se recibe como parámetro, o un -1 si no la encuentra.

**Int**     **indexOf(String s, intprimeraPos)** 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*

**Int**     **lastIndexOf(String s, intprimeraPos)** Devuelve la última posición en la cadena del texto *s*, empezando a buscar desde la posición *PrimeraPos*

**Char**   **charAt(intindex)** Proporciona el carácter que está en la posición dada por el entero *index*.

**Boolean**     **equals(Stringstr)** Compara la hilera que invoca al método con la del parámetro. Si son iguales retorna true y si no, retorna false. Es sensitivo a mayúsculas y minúsculas.

**Boolean**     **endsWith(String s)** Devuelve **true** si la cadena termina con el texto *s*

**Boolean**     **equalsIgnoreCase (Stringstr)** Compara la hilera que invoca al método con la del parámetro. Si son iguales retorna true y si nó, retorna false. No es sensitivo a mayúsculas y minúsculas.

**Int**     **compareTo(String str1)** Compara 2 cadenas y retorna un valor numérico. Cero (0): si ambas cadenas son iguales. Un número < 0 si la primera hilera es menor a la segunda. Un número > 0 si la primera hilera es mayor a la segunda.

**Int**     **compareToIgnoreCase(String s)** Compara dos cadenas, pero no tiene cuenta si el texto es mayúsculas o no.

**String concat(String s)** Añade la cadena *s* a la cadena original.

**byte[ ]**   **getBytes( )** Devuelve un array de caracteres que toma a partir de la cadena de texto

**void**   **getBytes(intsrcBegin, intsrcEnd, char[]dest, intdstBegin);** Almacena el contenido de la cadena en el array de caracteres *dest*. Toma los caracteres desde la posición *srcBegin* hasta la posición *srcEndy* les copia en el array desde la posición *dstBegin*

**String**   **replace(charcarAnterior, charncarNuevo)** Devuelve una cadena idéntica al original pero que ha cambiando los caracteres iguales a *carAnterior* por *carNuevo*



## METODOS MÁS UTILIZADOS

---

**String replaceAll (StringviejaString, StringnuevaString)** Reemplaza en la hilera que invoca al método la vieja String por la nueva String. Se utiliza para reemplazar subhileras.

**String replaceFirst(String str1, Stringstr2)** Cambia la primera aparición de la cadena str1 por la cadena str2

**String replaceFirst(String str1, Stringstr2)** Cambia la primera aparición de la cadena uno por la cadena dos

**Stringtrim():** Elimina espacios en blanco al principio y al final de la cadena. No elimina los espacios situados entre las palabras.

**Stringsubstring(intindiceInicial, intindiceFinal)** Retorna el segmento de la String entre la posición 'indiceInicial' inclusive hasta una posición anterior a 'indiceFinal'.

**charcharAt (intindice)**Devuelve el carácter que ocupa la posición indicada por el índice.

**String.valueOf**Convierte valores de una clase a otra. Permite convertir valores que no son de tipo String a valores de tipo String. Este método pertenece a la clase String directamente, no hay que utilizar el nombre del objeto creado ( es un método estático).

**String copyValueOf(char[ ] data)** Produce un objeto **String**que es igual al array de caracteres *data*.

**String startsWith(String s)** Devuelve **true** si la cadena comienza con el texto *s*.

**String substring(intprimeraPos, intsegundaPos)** Devuelve el texto que va desde *primeraPosa* *segundaPos*

**String toLowerCase( )** Convierte la cadena a minúsculas

**String toUpperCase( )** Convierte la cadena a mayúsculas

## EJEMPLOS

**int a = cadena.length();**

**cad1.compareTo(cad2)**

**cadena = cadena.toLowerCase();**

**cadena.substring(11,20)**

**if (cadena1.equals(cadena2))**

**charcharacter = cadena.charAt(7);**



### MÉTODOS DE LA CLASE CHARACTER

- boolean**      **isLetter(char character)** Retorna un verdadero si el carácter del parámetro es una letra.
- Boolean**      **isDigit(char character)** Retorna un verdadero si el carácter del parámetro es un dígito.
- Boolean**      **isUpperCase(char character)** Retorna un verdadero si el carácter del parámetro es una letra mayúscula.
- Boolean**      **isLowerCase(char character)** Retorna un verdadero si el carácter del parámetro es una letra minúscula.

### EJEMPLOS DE USO

```
if (Character.isLowerCase(caracter))
```

```
if (Character.isLetter(caracter))
```

```
if (Character.isUpperCase(caracter))
```



### CONVERSIONES ENTRE TIPOS DE DATOS

`int valor = Integer.parseInt(String) // Pasar de String a int`

`long valor = Long.parseLong(String) // Pasar de String a long`

`byte valor = Byte.parseByte(String) // Pasar de String a byte`

`short valor = Short.parseShort(String) // Pasar de String a short`

`float valor = Float.parseFloat(String) // Pasar de String a float`

`double valor = Double.parseDouble(String) // Pasar de String a double`

`String resultado = Integer.toString(12345) // Pasar de int a String`

`String resultado = Double.toString(12345.25) // Pasar de double a String`

`String resultado = Float.toString(12345.70) // Pasar de float a String`

`String hilera = Character.toString(caracter) // Pasar de caracter a hilera`

`String numero = String.valueOf(1234);`

`String fecha = String.valueOf(new Date());`