

# Udemy | Programación Java SE desde 0

---

<b>Fundamentos y estructuras básicas</b>	<b>3</b>
Tipos de datos en Java	3
Variables	3
Estructura de carpeta programa java	3
Constantes	4
Operadores en Java	4
Clases en Java	5
Primitivos vs Clases	6
Clase String	7
Métodos más usados	7
Casting	8
Paquetes	9
Entrada / Salida	9
Formas de introducir datos en un programa Java	9
Entrada / Salida con Scanner	9
JOptionPane	10
Condicionales	10
IF	10
SWITCH	10
Operador tenario	10
Bucles	11
while	11

# Fundamentos y estructuras básicas

## Tipos de datos en Java

### (Tipos Primitivos)

- **Enteros**
  - **Long**: 8 bytes - Sufijo L
  - **Int** : 4 bytes - Desde -2,147,483,648 hasta 2,147,483,647
  - **Short** : 2 bytes - Desde -32,768 hasta 32,767
  - **Byte**: 1 byte - Desde -128 hasta 127
- **Coma flotante** ( decimales )
  - **Float**: 4 bytes - Aproximadamente 6 a 7 cifras decimales significativas. Sufijo F.
  - **Double**: 8 bytes - Aproximadamente 15 cifras decimales significativas.
- **Char** - Para representar caracteres ( 'z', 'a' )
- **Boolean** - 2 únicos valores. **True** | **False**

## Variables

Es un espacio en la memoria del ordenador donde se almacenará un valor que podrá cambiar durante la ejecución del programa

Se crean especificando el tipo de dato que almacenará más el nombre de la variable

```
Int salario;
```

Se inicia al darle un valor.

```
salario = 2000;
```

## Estructura de carpeta programa java

- **.settings**
- **bin** | Aquí se guarda el archivo que lee la JVM. En formato .class
- **src** | El archivo con extensión .java se almacena en esta carpeta

**Código bytecode.** Archivo compilado que se guarda con la extensión .class. Está en un lenguaje a mitad de camino entre Java y el código máquina.

## Constantes

Espacio en la memoria del ordenador donde se almacenará un valor que **no podrá cambiar** durante la ejecución del programa

Se crean agregando la palabra final y a continuación especificando el tipo de dato que almacenará en su interior y finalmente el nombre de la constante

```
Final double a_pulgadas = 2.54;
```

Es posible declalarla y luego iniciarla, pero normalmente se realiza todo de una vez.

## Operadores en Java

- **Aritméticos**
  - **+** Suma
  - **-** Resta
  - **\*** Multiplicación
  - **/** División
- **Lógicos, relacionales y booleanos**
  - **>** mayor que
  - **<** menor que
  - **<>** mayor o menor que
  - **!=** distinto que
  - **==** igual que
  - **&&** y lógico
  - **||** o lógico
- **Incremento y decremento**
  - **++** incremento
  - **--** decremento
  - **+=** n° incremento
  - **-=** n° decremento
- **Concatenación**
  - **+** une o concatena

## Clases en Java

- Clases
  - **Propias**
  - **Predefinidas**
    - String
    - Math
    - Array
    - Thread
    - [...]
    -

Las clases predefinidas se encuentran en la **API de Java**.

- Paquetes
  - **Java**
    - Java.awt
    - Java.util
      - Java.util.regex
    - Java.io
    - [...]
  - **Javax**
    - Java.activity
    - Java.annotation
    - [...]
  - [...]

## Primitivos vs Clases

- **Primitivos**
  - Byte
  - Short
  - Int
  - Double
  - Float
  - Long
  - Char
  - boolean
- **Clases** (Objetos)
  - Math
  - String
  - Image
  - File
  - JButon
  - JFrame
  - JTable
  - Rectangle
  - [...]

## Clase String

```
String miNombre;  
String mNombre = Cobo
```

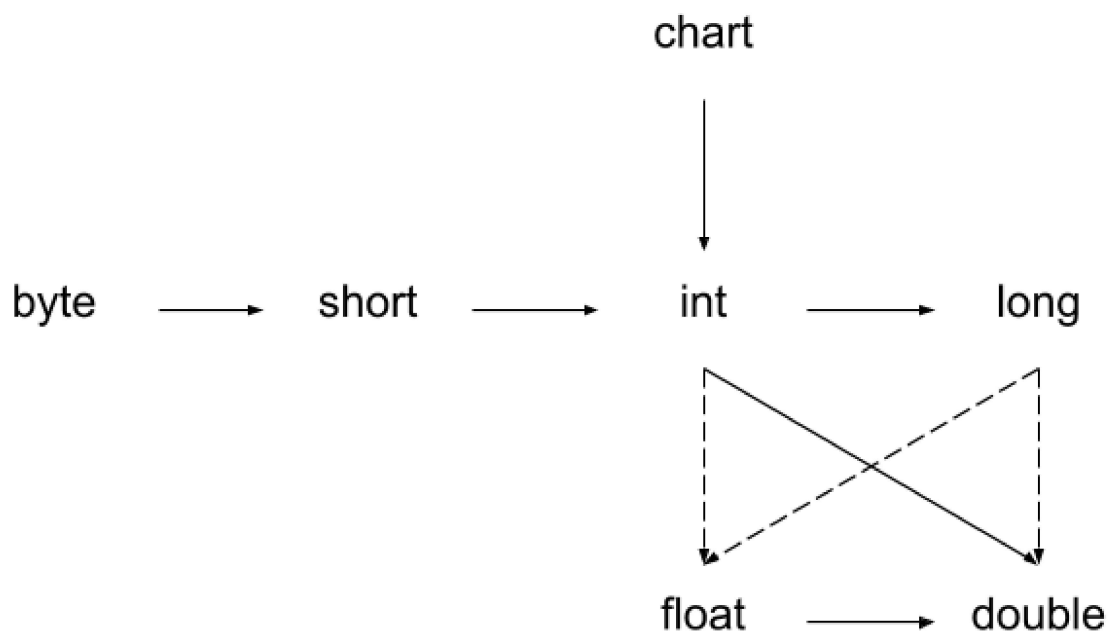
### Métodos más usados

```
length(str);  
substring(int index1, int index2);  
charAt(int index);  
equals(String);  
equalsIgnoreCase(String);
```

## Casting

```
int resultado = (int)Math.round(numero);
```

Hay que tener en cuenta que con tipos primitivos se puede perder precisión.





## Paquetes

Se usan para organizar las clases, evitar conflictos de nombres y controlar la visibilidad de las clases.

Al crear paquetes propios:

- El nombre del paquete debe estar en minúsculas
- Por convención, se usa como nombre del paquete un nombre de dominio al revés.

```
Com.cursojava.entradadatos
```

```
Com.cursojava.vistas
```

## Entrada / Salida

### Formas de introducir datos en un programa Java

- A través de una G.U.I.
- Usando la clase Scanner.  
Scanner permite introducir información usando la consola .
  - Scanner
    - `nextLine()` - Texto
    - `nextInt()` - Números enteros
    - `nextDouble()` - Números decimales
- Usando la clase JOptionPane.  
Crea una ventana en la que se puede introducir información.
  - JOptionPane
    - `showInputDialog()` - Se trata de un método estático

### Entrada / Salida con Scanner

1.- Se importa la clase Scanner

```
import java.util.Scanner
```

2.- Se crea un objeto de la clase Scanner

```
Scanner sc = new Scanner(System.in);
```

3.- Se crea una variable en la que guardar el dato introducido

```
Int age = sc.nextInt();  
String name m= sc.nextLine();  
[...]  
sc.close();
```

## JOptionPane

- Pertenece al paquete **javax.swing**
- El método más utilizado es **showInputDialog()**
- Es un método **estático**
  - `JOptionPane.showInputDialog();`

## Condicionales

### IF

### SWITCH

```
Switch (valor a evaluar){  
    Case valor1:  
        Código a ejecutar;  
        break;  
    Case valor1:  
        Código a ejecutar;  
        break;  
    Case valor1:  
        Código a ejecutar;  
        break;  
    Default:  
}
```

- El valor a evaluar solo puede ser un **char, byte, short, String o un enum**;
- En los case no se permiten operadores relacionales. **Solo se puede evaluar igualdad**
- La opción break es opcional

### Operador ternario

## Bucles

- Indeterminados
  - While
  - Do-While
- Determinados
  - For
  - Foreach

### While

Repite las líneas de código en el bucle mientras la condición sea verdadera. Si no se diera una condición falsa en ningún momento, se crearía un bucle infinito

```
while(Condición){  
    Código a ejecutar  
    [...]  
    [...]  
    [...]  
}
```

### Do-while

Similar al bucle while, salvo que **ejecuta el código del interior al menos uan vez**

```
do(Condición){  
    Código a ejecutar  
    [...]  
    [...]  
    [...]  
}while
```