

---

**VER SI STRING CONTIENE UN CARACTER EN ESPECÍFICO O UNA SECUENCIA DE CARACTERES (POR ESO VA CON “ ”)**

```
String clave = "45fde7F#";  
System.out.println("Tiene caracter: " + clave.contains("#"));
```

---

**VER SI STRING CONTIENE UN CARACTER EN ESPECÍFICO (SOLO SIRVE PARA 1 CARACTER)**

```
boolean tieneCaracterParticular = false;  
for (int i = 0; i < clave.length(); i++) {  
    if (clave.charAt(i) == '#') {  
        tieneCaracterParticular = true;  
    }  
}  
System.out.println("Tiene caracter: " + tieneCaracterParticular);
```

---

**VER SI STRING CONTIENE UN NUMERO**

```
boolean tieneNumero = false;  
for (int i = 0; i < clave.length(); i++) {  
    if (Character.isDigit(clave.charAt(i))) {  
        tieneNumero = true;  
    }  
}  
System.out.println("Tiene numero: " + tieneNumero);
```

---

**VER SI STRING CONTIENE UNA LETRA**

```
boolean tieneLetra = false;  
for (int i = 0; i < clave.length(); i++) {  
    if (Character.isLetter(clave.charAt(i))) {  
        tieneLetra = true;  
    }  
}  
System.out.println("Tiene letra: " + tieneLetra);
```

---

**VER SI STRING CONTIENE MAYUSCULA**

```
boolean tieneMayuscula = false;  
for (int i = 0; i < clave.length(); i++) {  
    if (Character.isUpperCase(clave.charAt(i))) {  
        tieneMayuscula = true;  
    }  
}  
System.out.println("Tiene mayúscula: " + tieneMayuscula);
```

---

**VER SI STRING CONTIENE CARACTER ESPECIAL**

```
boolean tieneEspecial = false;  
for (int i = 0; i < clave.length(); i++) {  
    if (Character.isLetterOrDigit(clave.charAt(i)) == false) {  
        tieneEspecial = true;  
    }  
}  
System.out.println("Tiene caracter especial: " + tieneEspecial);
```

---

**AÑO ACTUAL**

```
int anio = LocalDate.now().getYear();  
System.out.println("Estamos en el " + anio);
```

## NUMERO RANDOM ENTRE DOS VALORES (inclusive)

```
int numero = (int)( Math.random()*(MAX - MIN +1) + MIN );
```

```
//EJEMPLO NUMERO ENTRE 0 Y 100  
int numero = (int) ( Math.random()* (100-0 +1) + 0 );  
System.out.println("Numero random: " + numero);
```

---

## CAMBIAR EL TO STRING

```
@Override  
  
public String toString() {  
    return "Marca: " + marca + ", Modelo: " + modelo;  
}
```

---

## PASAR NUMERO ENTERO A UN ARRAY DONDE CADA DIGITO ESTÁ EN UNA POSICION

```
//1° lo paso a un string      String numeroString = String.valueOf(numeroEntero);  
  
//2° meto el string en el array  for(int i = 0; i < dimension; i++) {  
    array[i] = numeroString.charAt(i);  
}
```

---

## ORDENAR ARRAY

```
public Coche[] ordenarFlota() {  
    for (int i = 1; i < flota.length; i++) {  
        for (int j = 0; j < flota.length-1; j++) {  
            if (flota[j] != null && flota[j+1] != null && flota[j].getKM() > flota[j+1].getKM()) {  
                Coche auxiliar = flota[j];  
                flota[j] = flota[j+1];  
                flota[j+1] = auxiliar;  
            }  
        }  
    }  
    return flota;  
}
```

---

## BUSCAR POR...

```
public Coche[] buscarVehiculo(double precio) {  
    Coche [] cochesBuscados = new Coche[flota.length];  
    for (int i = 0; i < flota.length; i++) {  
        if (flota[i] != null && flota[i].getPrecio() == precio) {  
            cochesBuscados[i] = flota[i];  
        }  
    }  
    return cochesBuscados;  
}
```

---

## CONTAR OBJETOS CON UNA CARACTERÍSTICA

```
public int contarCochesDeDeterminadoModelo(String modelo) {  
    int contador = 0;  
    for (int i = 0; i < flota.length; i++) {  
        if (flota[i] != null && flota[i].getModelo().equals(modelo)) {  
            contador++;  
        }  
    }  
    return contador;  
}
```

## ENCONTRAR EL MENOR/MAYOR

```
public Coche encontrarCeroKilometroDeMenorPrecio() {  
    //inicializo  
    Coche cochePrecioMin = null;  
  
    //busco el primer objeto no nulo y con km 0  
    for (int i=0; i < flota.length; i++) {  
        if (flota[i]!=null && flota[i].cantidadKM()==0) {  
            cochePrecioMin = flota[i];  
            break;  
        }  
    }  
  
    //busco si hay otro objeto con km 0 y un precio menor  
    for (int i=0; i<flota.length; i++) {  
        if (flota[i]!=null && flota[i].cantidadKM()==0 && flota[i].getPrecio()<cochePrecioMin.getPrecio()) {  
            cochePrecioMin = flota[i];  
        }  
    }  
  
    return cochePrecioMin;  
}
```

---

## CONTAR CANTIDAD DE STRINGS DIFERENTES

```
public int calcularCantidadDeMarcasDistintas() {  
    int contador = 0;  
    String marcas = "";  
  
    for(int i=0; i<flota.length;i++) {  
        if(flota[i]!=null && marcas.contains(flota[i].getMarca())==false) {  
            marcas+= flota[i].getMarca();  
            contador++;  
        }  
    }  
    return contador;  
}
```

---

## ELIMINAR OBJETO DEL ARRAY

```
public void eliminarCoche(String identificador) {  
    //eliminar  
    for (int i = 0; i < flota.length; i++) {  
        if (flota[i] != null && flota[i].getID().equals(identificador)) {  
            flota[i] = null;  
        }  
    }  
  
    //ordenar los nulos para el fondo (condición: si el primero es nulo y el siguiente no es nulo)  
    for (int i = 1; i < flota.length; i++) {  
        for (int j = 0; j < flota.length-1; j++) {  
            if ( flota[j] == null && flota[j+1] != null ) {  
                Coche auxiliar = flota[j];  
                flota[j] = flota[j+1];  
                flota[j+1] = auxiliar;  
            }  
        }  
    }  
}
```

## EJEMPLO EJERCICIO CON ENUM

```
public enum UnidadesTemperatura {  
    CELSIUS, KELVIN, FAHRENHEIT  
}
```

---

```
public class Temperatura {  
    private final double KELVIN = 273.15;  
    private final byte FAHRENHEIT = 32;  
    private double valor;  
  
    public Temperatura (double valor, UnidadesTemperatura unidad) {  
        switch(unidad) {  
            case CELSIUS:  
                this.valor = valor;  
                break;  
            case KELVIN:  
                this.valor = valor - this.KELVIN;  
                break;  
            case FAHRENHEIT:  
                this.valor = (valor - this.FAHRENHEIT)* 5/9;  
                break;  
        }  
    }  
    public double getValor() {  
        return this.valor;  
    }  
    public double convertir(UnidadesTemperatura unidad) {  
        double temperatura = 0;  
        switch(unidad) {  
            case CELSIUS:  
                temperatura = getValor();  
                break;  
            case KELVIN:  
                temperatura = this.valor + this.KELVIN;  
                break;  
            case FAHRENHEIT:  
                temperatura = (this.valor * 9/5) + this.FAHRENHEIT;  
                break;  
        }  
        return temperatura;  
    }  
}
```

---

```
public class PruebaTemperatura {  
  
    public static void main(String[] args) {  
  
        Scanner teclado = new Scanner (System.in);  
  
        menu();  
  
        UnidadesTemperatura unidad = UnidadesTemperatura.values()[teclado.nextInt()];  
        System.out.println("Eligió: " + unidad);  
        System.out.println("Ingrese el valor de la temperatura:");  
        double valor = teclado.nextDouble();  
        Temperatura temperatura = new Temperatura (valor, unidad);  
  
        menu();  
  
        unidad = UnidadesTemperatura.values()[teclado.nextInt()];  
        System.out.println("Eligió: " + unidad);  
        System.out.println("La temperatura es: " + temperatura.convertir(unidad));  
    }  
    private static void menu() {  
        System.out.println("Elija una unidad:"  
            + "\n0. " + UnidadesTemperatura.values()[0]  
            + "\n1. " + UnidadesTemperatura.values()[1]  
            + "\n2. " + UnidadesTemperatura.values()[2]);  
    }  
}
```