

## Sintaxis

```
if (condición)
{
    // Declaraciones para ejecutar si
    // la condición es verdadera
}
```

## If-else

```
if (condición)
{
    // Ejecuta este bloque si
    // la condición es verdadera
}
else
{
    // Ejecuta este bloque si
    // la condición es falsa
}
```

## If anidado

```
if (condicion1)
{
    // Ejecuta cuando condicion1 es verdadero
    if (condition2)
    {
        // Ejecuta cuando condicion2 es verdadero
    }
}
```

## If-else if

```
if (condición)
    declaración;
else if (condición)
    declaración;
.
.
    declaración
else ;
```

## ejemplo

```
int i = 20;

    if (i == 10)
        System.out.println("i es 10");
    else if (i == 15)
        System.out.println("i es 15");
    else if (i == 20)
        System.out.println("i es 20");
    else
        System.out.println("i no está presente");
```

## switch-case

La instrucción **switch** es una declaración de bifurcación de múltiples vías (selección múltiple). Proporciona una forma sencilla de enviar la ejecución a diferentes partes del código en función del valor de la expresión.

### sintaxis

```
switch (expresión)
{
    case valor1:
        declaracion1;
        break;
    case value2:
        declaracion2;
        break;
    .
    .
    case valorN:
        declaracionN;
        break;
    default:
        declaracionDefault;
}
```

- La expresión puede ser de tipo **byte**, **short**, **int**, **char** o una enumeración. A partir de JDK7, la expresión también puede ser de tipo **String**.
- Los valores duplicados de case no están permitidos.

### Ejemplo

```
int i = 9;
    switch (i)
    {
        case 0:
```

```
        System.out.println("i es cero.");
        break;
    case 1:
        System.out.println("i es uno.");
        break;
    case 2:
        System.out.println("i es dos.");
        break;
    default:
        System.out.println("i es mayor que 2.");
    }
}
```

do- while

La sentencia (hacer mientras) crea un bucle que ejecuta una sentencia especificada, hasta que la condición de comprobación se evalúa como falsa. La condición se evalúa después de ejecutar la sentencia, dando como resultado que la sentencia especificada se ejecute al menos una vez.

Ejemplo

```
Int i=0;

do {

    i = i + 1;

    result = result + i;

} while (i < 5);
```