

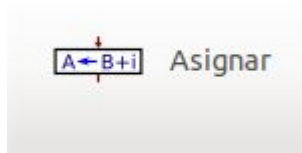
0. INICIO / FIN ALGORITMO.

Algoritmo test

FinAlgoritmo

```
public class test {  
    public static void main(String[] args) {  
  
    }  
}
```

1. ASIGNAR



```
variable <- 5|
```

```
byte variable_byte = 5;  
short variable_short = 5;  
int variable_int = 5;  
long variable_long = 5;  
float variable_float = 5;  
double variable_double = 5.0;  
char variable_char = 'a';  
boolean variable_boolean = true;
```

| Java Primitive Data Types | | | | |
|---------------------------|-------------------------|---------|------------------------------|---|
| Type | Values | Default | Size | Range |
| byte | signed integers | 0 | 8 bits | -128 to 127 |
| short | signed integers | 0 | 16 bits | -32768 to 32767 |
| int | signed integers | 0 | 32 bits | -2147483648 to 2147483647 |
| long | signed integers | 0 | 64 bits | -9223372036854775808 to 9223372036854775807 |
| float | IEEE 754 floating point | 0.0 | 32 bits | +/-1.4E-45 to +/-3.4028235E+38, +/-infinity, +/-0, NaN |
| double | IEEE 754 floating point | 0.0 | 64 bits | +/-4.9E-324 to +/-1.7976931348623157E+308, +/-infinity, +/-0, NaN |
| char | Unicode character | \u0000 | 16 bits | \u0000 to \uFFFF |
| boolean | true, false | false | 1 bit used in 32 bit integer | NA |

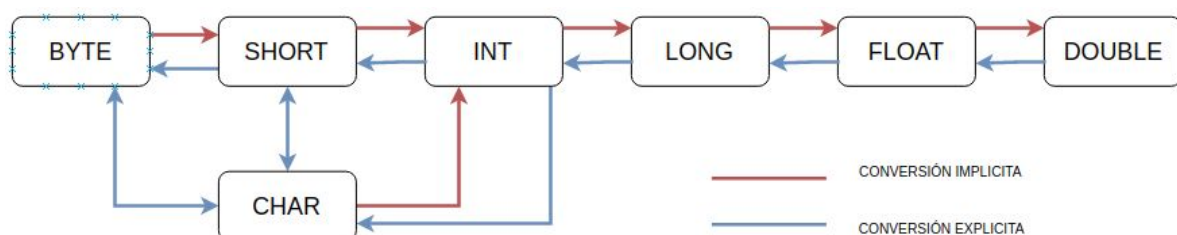
1.1. CAST - CONVERTIR DE UN TIPO DE VARIABLE A OTRO.

1.1.1. CONVERSIÓN IMPLÍCITA. Automático.

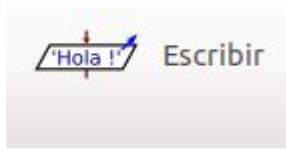
Mera asignación `int value_int = value_short`

1.1.2. CONVERSIÓN EXPLÍCITA.

Forzado por el programador. Uso de (tipo de variable). `int value_int = (int)value_long`



2. ESCRIBIR



Escribir "Escribir valor de salida"
Escribir valor_salida

```
int valor_salida = 5;  
System.out.println("Escribir valor de salida");  
System.out.println(valor_salida);
```

3. LEER

Leer valor_entrada

Leer cadena_entrada

```
// Creamos escaner de entrada.  
Scanner escaner_entrada = new Scanner(System.in);  
  
// Variables numéricas  
boolean valor_entrada_boolean = escaner_entrada.nextBoolean();  
byte valor_entrada_byte = escaner_entrada.nextByte();  
short valor_entrada_short = escaner_entrada.nextShort();  
int valor_entrada_int = escaner_entrada.nextInt();  
long valor_entrada_long = escaner_entrada.nextLong();  
float valor_entrada_float = escaner_entrada.nextFloat();  
double valor_entrada_double = escaner_entrada.nextDouble();  
  
// Cadena de caracteres  
String valor_entrada_cadena = escaner_entrada.nextLine();
```

4. SI

Si expresion logica Entonces
..... acciones por verdadero
Fin Si

```
if (expresion_logica) {  
    // acciones_por_verdadero  
}
```

5. SI-ENTONCES

Si expresion logica Entonces
..... acciones por verdadero
SiNo
..... acciones por falso
Fin Si

```
if (expresion_logica) {  
    // acciones_por_verdadero  
} else {  
    // acciones_por_falso  
}
```

6. SEGÚN

```
Segun variable_numerica Hacer
1:
    secuencia_de_acciones_1
2:
    secuencia_de_acciones_2
3:
    secuencia_de_acciones_3
De Otro Modo:
    secuencia_de_acciones_dom
Fin Segun
```

```
switch (variable_numerica) {
case 1:
    // secuencia_de_acciones_1
    break ;
case 2:
    // secuencia_de_acciones_2
    break;
case 3:
    // secuencia_de_acciones_3
    break;

default:
    // secuencia_de_acciones_dom
    break;
}
```

7. MIENTRAS

```
Mientras expresion_logica Hacer
    secuencia_de_acciones
Fin Mientras
```

```
while (expresion_logica) {
    // Secuencia de acciones
}
```

8. REPETIR

```
Repetir
    secuencia_de_acciones
Hasta Que expresion_logica
```

```
do {
    // Secuencia de acciones
} while (expresion_logica);
```

9. PARA

```
Para indice<-0 Hasta 59 Con Paso 1 Hacer
    secuencia_de_acciones
Fin Para
```

```
for (int indice = 0; indice < 59; indice++) {
    // Secuencia de acciones
}
```

10. PROCEDIMIENTO

```
Algoritmo Procedimientos
    Escribir "Tabla de multiplicación"
    Escribir "Ingrese un número:"
    Leer input
    tablaMultiplicacion(input)
FinAlgoritmo

Subproceso tablaMultiplicacion(value)
    Escribir "Procesando tabla de multiplicación..."
    Para index<-0 Hasta 10 Con Paso 1 Hacer
        calculo <- (value * index)
        Escribir index,":",calculo
    Fin Para
FinSubproceso
```

```
public class Procedimientos {

    public static void main(String[] args) {
        System.out.println("Tabla de multiplicacion");
        System.out.println("Ingrese un número");
        Scanner escaner_entrada = new Scanner(System.in);
        int input = escaner_entrada.nextInt();
        tablaMultiplicacion(input);
    }

    public static void tablaMultiplicacion(int value) {
        System.out.println("Procesando tabla de multiplicación");
        for (int index = 0; index < 11; index++) {
            int calculo = value * index ;
            System.out.println(index+" : "+calculo);
        }
    }
}
```

11. FUNCIÓN.

```
Funcion resultado <- Maximo( n1, n2)
  Si n1 > n2 Entonces
    resultado <- n1
  Sino
    resultado <- n2
  FinSi
FinFuncion
```

```
Algoritmo Funciones01
  Escribir "El maximo de 5 y 12 es:"
  Escribir Maximo(5,12)
  Escribir "El maximo de 25 y 12 es:"
  Escribir Maximo(25,12)
FinAlgoritmo
```

```
public class Funciones {

    public static void main(String[] args) {
        System.out.println("El máximo de 5 y 12 es:");
        System.out.println(Maximo(5,12));

        System.out.println("El máximo de 25 y 12 es:");
        System.out.println(Maximo(25,12));
    }

    public static int Maximo(int n1, int n2) {
        if (n1>n2) {
            return n1;
        }else {
            return n2;
        }
    }
}
```
