



# Trabajando con variables y la consola en JAVA

Nuevo tipos de variables

# Imprimiendo variables en la consola de JAVA



Hasta ahora hemos aprendido que en JAVA existen variables de tipo primitivas, su tipo empieza con minúscula (boolean, char, byte, short, int, float y long) y se le pueden asignar valores a través de las literales asignadas al momento de su declaración.

¿Ahora más allá de asignarles valor como podemos ver efectivamente el valor almacenado en la variable?

Para ello nos apoyaremos en una línea de código que nos proporciona JAVA para tal función, si recién comienzas en el mundo de la programación esto te resultará algo confuso, pero no nos detendremos analizar (por el momento) como es que esto funciona, simplemente lo utilizaremos para llevar adelante las practicas de programacion.

La sintaxis será la siguiente:

```
System.out.println( elNombreDeMiVariable );
```

No olvides que cada instrucción en java debe finalizar con un ; el IDE se quejara de todas maneras si te la olvidas.

# Veamos un ejemplo:

Nuestro código debería verse así:

```
public static void main(String[] args) {  
  
    int miVariable = 2;  
    System.out.println(x: miVariable);  
  
}
```

Y la salida de nuestra consola debería verse así:

```
run:  
2  
BUILD SUCCESSFUL (total time: 0 seconds)
```

## Veamos ejemplos con otros tipos de variables:

```
boolean miVariableBooleana = false;
System.out.println(x: miVariableBooleana);
char miVariableCaracter = 'a';
System.out.println(x: miVariableCaracter);
byte miVariableByte = 2;
System.out.println(x: miVariableByte);
short miVariableShort = 10;
System.out.println(x: miVariableShort);
int miVariableEntera = 66;
System.out.println(x: miVariableEntera);
float miVariableFloat = 2.3F;
System.out.println(x: miVariableFloat);
long miVariableLarga = 5_000_000_999_888L;
System.out.println(x: miVariableLarga);
```

La salida de nuestra consola será la siguiente:

```
run:
false
a
2
10
66
2.3
5000000999888
```

# Operador de asignación =



El operador de asignación como ya sabemos permite asignar el valor de la literal a la variable.

Sin embargo, una vez declarada la variable con su tipo e inicializada con la literal nos permitirá cambiar el valor almacenado por dicha variable sólo utilizando el identificador.

```
[tipo] [identificador] = [literal];
```

```
int mi_Variable = 2 ;
```

```
mi_Variable = 3;
```

Si la variable fuese una constante al querer cambiar el valor de dicha variable nos marcaría con rojo y una advertencia el IDE diciendo que la asignación no está permitida.

# Ejemplos:

```
int miVariable;  
miVariable = 2;  
System.out.println(x: miVariable);
```

declaramos la variable  
asignamos el valor literal 2  
enviamos a imprimir la variable

```
miVariable = 3;  
System.out.println(x: miVariable);
```

nuevamente asignamos valor a la variable, un 3  
volvemos a imprimir por consola el valor de la variable

```
run:  
2  
3
```

Esta será la salida de los valores en la consola, observemos que primero imprimió un 2 y luego un 3.

Observemos como el IDE subraya toda la instrucción. Ya que la variable al tener la palabra final antes del tipo se convierte en constante y al ya tener asignado el valor 3.14, no puede cambiar a 0.  
Si quitamos esa línea de código, el programa compilara e imprimirá por pantalla 3.14

```
final double MI_CONSTANTE = 3.14;  
System.out.println(x: MI_CONSTANTE);  
  
MI_CONSTANTE = 0;
```