

Ejemplo de Variables con Java



📌 Guía de Variables en Java con IntelliJ IDEA ⚡

Introducción

En esta lección aprenderás a **declarar, inicializar y manipular variables en Java** utilizando IntelliJ IDEA.



💡 Recuerda: Una **variable** es un espacio en memoria donde podemos almacenar y modificar información durante la ejecución del programa.

🚀 1. Creación del Proyecto en IntelliJ

1. Abrir **IntelliJ IDEA** y seleccionar la opción de **Nuevo Proyecto**.
2. **Nombre del proyecto:** Variables
3. **Ubicación:**

C:\cursos\Java

4. **Dejar los valores por defecto** y hacer clic en **Crear**.

📁 2. Creación de la Clase Java 📁

1. Ubicar la carpeta **src** en el panel izquierdo.
 2. Clic derecho > New > Java Class
 3. Nombre de la clase: **Variables**
 4. Hacer clic en Finalizar.
-

📝 3. Creación del Método Principal 💻

Dentro de la clase **Variables**, agregamos el **método main**, que es el punto de entrada de nuestra aplicación en Java.

```
public class Variables {  
    public static void main(String[] args) {  
        // Aquí agregaremos el código de variables  
    }  
}
```

📌 Explicación:

- ◆ **public** → Indica que el método es accesible desde cualquier parte del programa.
- ◆ **static** → Permite ejecutarlo sin necesidad de crear un objeto de la clase.
- ◆ **void** → No retorna ningún valor.
- ◆ **main** → Es el nombre obligatorio del método principal en Java.

💡 Ahora estamos listos para trabajar con **variables** en Java. ¡Vamos a definirlas! 🚀

12 4. Declaración e Inicialización de Variables 12

Ahora declararemos nuestra primera variable:

```
int edad = 31;
```

📌 Explicación:

- ◆ **int** → Es el **tipo de dato**, en este caso un número entero.
- ◆ **edad** → Es el **nombre de la variable**.

- ◆ = → Significa **asignación**, es decir, guardamos un valor en la variable.
- ◆ 31 → Es el **valor inicial** asignado a `edad`.

💻 ¿Qué sucede en memoria?

Cuando ejecutamos esta línea, en memoria se guarda la variable `edad` con el valor 31:

Variable	Valor	Tipo de Dato
edad	31	int

💰 5. Agregando más Variables 💰

Vamos a definir más variables de diferentes tipos de datos:

```
double precio = 300.40;
boolean disponible = true;
char genero; // Primero se crea la variable sin un valor
genero = 'M'; // Asignamos el valor después de declararla
```

📌 Explicación de cada tipo:

- ✓ **double** → Guarda valores decimales (`precio = 300.40`).
- ✓ **boolean** → Solo puede almacenar `true` (verdadero) o `false` (falso).
- ✓ **char** → Guarda un solo carácter ('M' o 'F').

💻 ¿Qué sucede en memoria?

Variable	Valor	Tipo de Dato
edad	31	int
precio	300.40	double
disponible	true	boolean
genero	'M'	char

🖨 6. Accediendo a los Valores de las Variables 🖼

Para mostrar en pantalla los valores almacenados en nuestras variables, usamos `System.out.println()`.

```
System.out.println(edad);
System.out.println(precio);
System.out.println(disponible);
System.out.println(genero);
```

❖ Salida esperada en consola:

```
31
300.4
true
M
```

7. Modificación del Valor de una Variable

Ahora cambiaremos el valor de `edad = 35;`

```
edad = 35;
System.out.println(edad);
```

💻 ¿Qué sucede en memoria?

En memoria se sobreescribe el valor de 31 y ahora el nuevo valor es 35.

Variable	Valor antes	Valor después
edad	31	35

❖ Salida esperada en consola:

```
31
35
```

💡 Nota:

- ◆ No necesitamos volver a declarar el tipo de dato cuando modificamos una variable, solo escribimos su nombre y asignamos el nuevo valor.
-

🛠 Código final Variables.java

```
public class Variables {
    public static void main(String[] args) {
        // Variables
        int edad = 31;
        double precio = 300.40;
        boolean disponible = true; // true/false
        char genero;
        genero = 'M'; // M-Masculino / F-Femenino

        // Acceder a las variables
    }
}
```

```
System.out.println(edad);
// Modificar el valor
edad = 35;
System.out.println(edad);
// Imprimir el resto de las variables
System.out.println(precio);
System.out.println(disponible);
System.out.println(genero);
}
}
```

🎯 8. Ejercicio Práctico

💡 Reto:

- ◆ Modifica el valor de genero a 'F'.
- ◆ Cambia disponible a false.
- ◆ Agrega una nueva variable String nombre = "Juan"; e imprimela.

```
genero = 'F';
disponible = false;
String nombre = "Juan";
System.out.println(genero);
System.out.println(disponible);
System.out.println(nombre);
```

📌 Salida esperada en consola:

```
F
false
Juan
```

🎯 Conclusión

- ✓ Aprendiste a definir diferentes tipos de variables en Java.
- ✓ Entendiste cómo se almacenan los datos en memoria.
- ✓ Practicaste cómo modificar e imprimir valores en consola.

📚 ¡Sigue adelante y nos vemos en la siguiente lección! 🚀

Ing. Marcela Gamiño e Ing. Ubaldo Acosta

Fundadores de GlobalMentoring.com.mx