

IT Bootcamp

Introducción a Java

DigitalHouse>



Índice

1. **¿Qué es Java?**
2. **Sintaxis del Lenguaje**
3. **Variables + Tipos de Datos**
4. **Operadores**

1 | ¿Qué es Java?



Java

- Java es un lenguaje de programación y tecnología pensada para el desarrollo de aplicaciones de gran envergadura, que sean altamente escalables, de gran integración con otras tecnologías y por sobre todo muy robustas. se destaca por ser un lenguaje fuertemente tipado.
- Nació con el objetivo de ser un lenguaje de programación de estructura sencilla que pudiera ser ejecutado en diversos sistemas operativos.



Versiones de Java

- **Java** posee dos ediciones principales:
 - **Java Standard Edition (Java SE):** Fue conocido en principio como J2SE, pero a partir de su versión 5.0 fue estandarizado como únicamente Java SE. Es el entorno básico de Java que permite crear programas estándares con las herramientas y colecciones básicas ya incorporadas en el lenguaje.
 - **Java Enterprise Edition (Java EE):** Es utilizado principalmente para el desarrollo de aplicaciones empresariales o de gran tamaño. Contempla como principal ambiente el web, utilizando diferentes APIS como JDBC, JPA, servlets, JSP, EJB (Enterprise Java Beans), etc.



¿Qué es necesario para programar en Java?

- Para poder programar en Java, el dispositivo desde el que estemos trabajando debe contar con dos complementos básicos:
 - **JRE (Java Runtime Environment)**: Es un conjunto de utilidades que provee **Java** para la ejecución de las aplicaciones desarrolladas en este lenguaje. Generalmente JRE viene incluido con la gran mayoría de los sistemas operativos e incluso como complemento de algunos entornos de desarrollo (IDE).
 - **JDK (Java Development KIT)**: Es un software que provee las herramientas necesarias para poder desarrollar aplicaciones o elementos de software mediante Java. Cuenta con diferentes versiones, sin embargo las dos consideradas como más estables son la 8 y la 11.



2 | Sintaxis



Sintaxis

Java se destaca por ser un lenguaje **fuertemente tipado**. ¿Qué significa esto? Que es **muy exigente** tanto con su **sintaxis** como con sus **tipos de datos**.

Entre algunas de esas exigencias tenemos:

- Es **Case sensitive** (distingue mayúsculas de minúsculas)
- Cada línea de código debe finalizar con **;** (**punto y coma**)
- El principio y fin de cada estructura, así como los bloques de código se delimitan con **{ }** (**llaves**)
- Los parámetros se pasan entre **()** (**paréntesis**)



Ejemplo Sintaxis

```
package com.bootcamp.Ejercicio1;
```

```
public class Main {
```

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

```
        System.out.println("Hola Mundo");
```

```
    }
```

```
}
```

Llave de
inicio

Llave de fin

Línea que
termina en ;



3 | Variables & Tipos de Datos



Variables



Una **variable** es una posición en memoria que se destina o reserva con la finalidad de poder guardar un dato dentro de ella.



Tal como en **matemática**, es un valor que puede **cambiar** en un determinado período de tiempo.



En **Java** para poder utilizar una variable o asignarle determinados valores, ésta tuvo que haber sido **declarada** con anterioridad. Para declarar variables utilizamos **Tipos de Datos**.



Tipos de Datos

Los tipos de datos definen qué puede ser almacenado dentro de una variable como así también los límites de tamaño a tener en cuenta para esto. En Java existen **tipos de datos primitivos** como así también clases que actúan como si fuesen tipos de datos.

Numéricos

- **short** (entero, 2 bytes)
- **int** (entero, 4 bytes)
- **long** (entero, 8 bytes)
- **float** (flotante, 4 bytes)
- **double** (flotante, 8 bytes)

Caracteres

- **char** (caracter, 2 bytes)
- **String** (cadena de caracteres, no **es primitivo**, es una **Clase**)

Otros

- **boolean** (lógico, 2 bytes)
- **void** (tipo de dato “nulo”, para funciones que no retornan valor)



¿Cómo declarar una variable?

Nombre de la variable

Representa el nombre que tendrá la variable en cuestión. Se sugiere que sea un nombre representativo de lo que almacenará.

Java

```
tipo de dato nombre_variable;
```

Tipo de Dato

Nos indica el tipo de dato que contendrá la variable a declarar..



Ejemplo de declaración

```
{  
    int numero;  
    boolean bandera;  
    double sueldo;  
    String nombre;  
    char letra;  
}
```



Reglas para nombres de variables



No puede comenzar con un número. Debe comenzar con una letra. Como buena práctica se sugiere una letra minúscula.



No puede utilizar caracteres especiales (% , \$, * , etc) o caracteres reservados para otras operaciones.



No puede incluir espacios en blanco.



No puede llamarse igual que palabras reservadas del lenguaje (como por ejemplo: int, for, if, else, private, entre otras).



4 | Operadores



Operadores aritméticos, lógicos y relacionales

Aritméticos	Relacionales	Lógicos
+ Suma	== Igual	&& (AND o y)
- Resta	>,< Mayor y menor	(OR u o)
* Multiplicación	>=,<= Mayor = y menor =	! (NOT o negación)
/ División	!= Distinto	
	= Asignación	



Ejemplos

//Aritméticos

```
int suma = 3 + 5;  
int resta = 15 - 4;  
int multi = 4 * 6;  
int divi = 9 / 3;
```

//Relacionales

```
boolean mayor = 5 > 3;  
boolean menor = 1 < 9;  
boolean igual = 4 == 4;
```

//Lógicos

```
boolean si = !false;  
boolean no = !true;  
boolean and = true && true;  
boolean or = true || false;
```



Operaciones de lectura y escritura

```
//Escritura  
System.out.println("Hola");
```

*//dependiendo del tipo de dato que
queremos leer utilizamos diferentes
funciones next*

```
//Lectura  
  
Scanner teclado = new Scanner(System.in);  
  
int num = teclado.nextInt();  
String nombre = teclado.next();  
double promedio = teclado.nextDouble();
```

Material extra

Material extra (no obligatorio)



Links de lectura y videos de interés

Lectura: [Java Variables](#)

Video: [Tipos de Datos y Variables en Java](#)

Lectura: [Java Declare Multiple Variables](#)

Lectura: [Java Operators](#)

Video: [Operadores aritméticos en Java \(con práctica\)](#)

DigitalHouse>