



Fundamentos de Java

Glosario. Sintaxis, expresiones
y flujo de control en Java



Transformación Digital
Agencia de Transformación Digital y Telecomunicaciones



TECNOLÓGICO
NACIONAL DE MÉXICO

» 1

Asignación (Assignment)

Operación que asigna un valor a una variable. En Java, se realiza con el operador **=**. Las variables locales deben inicializarse explícitamente antes de su uso, mientras que las variables de instancia se inicializan automáticamente con valores predeterminados (como **0** para enteros o **null** para referencias).

Ejemplo: **int x = 5;**

» 2

Bloque (Block)

Colección de sentencias delimitadas por llaves **{ }**. Sirve para agrupar código y definir ámbitos locales. Puede anidarse y se utiliza en estructuras como **if**, **while** o clases.

Ejemplo: **{ int x = 1; System.out.println(x); }**

» 3

Casting

Conversión explícita de un tipo de dato a otro, requerida cuando puede haber pérdida de información (por ejemplo, de **long** a **int**). Se indica entre paréntesis.

Ejemplo: **int squashed = (int) 99L;**

Es crucial para evitar errores de compilación en asignaciones incompatibles.

» 4

Clase (Class)

Conjunto de objetos que comparten una estructura común y un comportamiento común. Una clase actúa como una plantilla o contrato que define la naturaleza de los objetos.

» 5

Comentarios (Comments)

Texto no ejecutable para documentar el código. Java soporta tres estilos:

// para una línea, /* */ para múltiples líneas y / * / para documentación.

Ejemplo: // Esto es un comentario.

Fomenta buenas prácticas de codificación para mejorar la legibilidad.

» 6

Declaración (Declaration)

Definición de una variable especificando su tipo y nombre, sin necesariamente asignar un valor.

Ejemplo: int x;.

Las variables locales deben declararse e inicializarse antes de usarse, mientras que las de instancia se inicializan por defecto.

» 7

Do-While

Estructura de repetición que ejecuta el bloque al menos una vez antes de evaluar la condición. Sintaxis: do { ... } while (condición);.

Ejemplo: do { System.out.println(i); i++; } while (i < 10);.

Ideal cuando se necesita una ejecución inicial garantizada.

» 8

Espacio en Blanco (White Space)

Cualquier cantidad de espacios, tabulaciones o saltos de línea permitidos en el código para mejorar la legibilidad. Java ignora el exceso de espacios en blanco.

Ejemplo: int x=1; es equivalente a int x = 1;.

Recomendación educativa: usa indentación consistente (2 o 4 espacios).

» 9

Expresión (Expression)

Combinación de variables, operadores y valores que se evalúan para obtener un único resultado. Pueden ser booleanas, aritméticas o de otro tipo.

Ejemplo: **x + y > 10**.

Las expresiones booleanas son requeridas en estructuras de control como **if** o **while**.

» 10

Flujo de Control (Flow Control)

Mecanismos para alterar el orden secuencial de ejecución del código, como condicionales (**if**, **switch**) y bucles (**for**, **while**, **do-while**). Incluye *break* y *continue* para control especial.

Educativamente, permite decisiones lógicas y repeticiones eficientes.

» 11

For

Bucle de repetición con inicialización, condición y alteración integradas.

Sintaxis: **for (init; test; alter) { ... }**.

Ejemplo: **for (int i = 0; i < 10; i++) { ... }**.

Útil para iteraciones con contador conocido.

» 12

Identificadores (Identifiers)

Nombres asignados a variables, clases o métodos. Deben comenzar con letra, _ o **\$**, ser sensibles a mayúsculas y no coincidir con palabras clave. Ejemplo:

userName.

» 13

If-Else

Estructura de decisión que ejecuta un bloque si una condición booleana es verdadera, y opcionalmente otro si es falsa.

Sintaxis: `if (condición) { ... } else { ... }`. Puede encadenarse como **if-else-if**.

Ejemplo: `if (x < 10) { ... } else { ... }`.

Enseña toma de decisiones condicionales.

» 14

Inicialización (Initialization)

Asignación de un valor inicial a una variable. Las variables de instancia se inicializan automáticamente (ej. `0` para **int**, `null` para referencias), pero las locales requieren inicialización explícita para evitar errores de compilación.

» 15

Instancia (Instance)

Sinónimo de objeto.

» 16

Método (Method)

Función o procedimiento definido dentro de una clase que describe el comportamiento de los objetos.

» 17

Objeto (Object)

Entidad que tiene estado, comportamiento e identidad.

Estado: Las propiedades del objeto y sus valores actuales (generalmente dinámicos).

Comportamiento: Cómo actúa y reacciona el objeto (sus operaciones).

Identidad: Lo que distingue a un objeto de todos los demás (incluso si su estado es idéntico).

» 18

Operadores (Operators)

Símbolos para realizar operaciones. Incluyen aritméticos (+, -, *, /, %), lógicos (!, ^, &&, ||), y de asignación (=, +=, etc.). La precedencia determina el orden de evaluación.

Ejemplo: **x + y**.

» 19

Palabras Clave (Keywords)

Términos reservados en Java que no pueden usarse como identificadores.

Lista: **abstract, assert, boolean, break, byte, case, catch, char, class, const, continue, default, do, double, else, enum, extends, final, finally, float, for, goto, if, implements, import, instanceof, int, interface, long, native, new, package, private, protected, public, return, short, static, strictfp, super, switch, synchronized, this, throw, throws, transient, try, void, volatile, while**.

Literales reservados: **null, true, false**.

» 20

Paso por Valor (Pass-by-Value)

Mecanismo de Java donde los argumentos se pasan por valor (copia). Para objetos, se copia la referencia, permitiendo modificar el objeto pero no la referencia original.

» 21

Promoción (Promotion)

Conversión automática de tipos más pequeños a más grandes en expresiones (ej. **int a long**).

Asegura compatibilidad en asignaciones.

» 22

Referencia *this*

Referencia implícita al objeto actual dentro de una clase. Útil para resolver ambigüedades entre parámetros y atributos, o pasar el objeto a otros métodos.
Ejemplo: `this.day = day;`

» 23

Sentencia (*Statement*)

Línea de código terminada por punto y coma (;). Puede ser simple o compuesta (bloque).

» 24

Switch

Estructura de decisión basada en valores constantes. Sintaxis: `switch (expresión) { case valor: ... break; default: ... }`. La ejecución cae a casos subsiguientes sin *break*. Solo para tipos integrales, `enum` o `String` (desde Java 7).

» 25

Tipos Primitivos (*Primitive Types*)

Tipos básicos de datos: `boolean` (true/false), `char` (carácter Unicode), `byte` (-128 a 127), `short` (-32768 a 32767), `int` (-2^31 a 2^31-1), `long` (-2^63 a 2^63-1), `float` (32 bits), `double` (64 bits).

No son objetos.

» 26

Tipos de Referencia (*Reference Types*)

Tipos que almacenan referencias a objetos (ej. clases, arrays). Inicializados a `null` por defecto.

Ejemplo: `MyDate fecha = new MyDate();`

» **27**

Variables (Variables)

Espacios nombrados para almacenar datos.

Tipos: locales (ámbito en método, requieren inicialización), de instancia (ámbito en clase, inicializadas por defecto).

» **28**

Variable de instancia (campo)

Variable definida dentro de una clase (pero fuera de los métodos) que pertenece a un objeto específico. Cada objeto tiene su propia copia independiente de estas variables.

Ejemplo: **nombre** de un Usuario específico.

» **29**

While

Bucle de repetición que evalúa la condición antes de ejecutar.

Sintaxis: **while (condición) { ... }**.

Ejemplo: **while (i < 10) { ... i++; }**.

Útil para repeticiones con condición incierta.

» **Referencias:**

- 1.** Deitel, P. J., y Deitel, H. M. (2017). *Java: How to Program, Early Objects* (Undécima edición). Pearson.
- 2.** Eckel, B. (2006). *Thinking in Java* (Cuarta edición). Prentice Hall.
- 3.** Oracle. (2023). *The Java® Language Specification: Java SE 21 Edition* <https://docs.oracle.com/javase/specs/>
- 4.** Oracle. (s.f.). *The Java™ Tutorials*. Recuperado el 27 de noviembre de 2024 de <https://docs.oracle.com/javase/tutorial/>

Elaboró contenido:

Ulises Juárez Martínez
Manuel Panzi Utrera