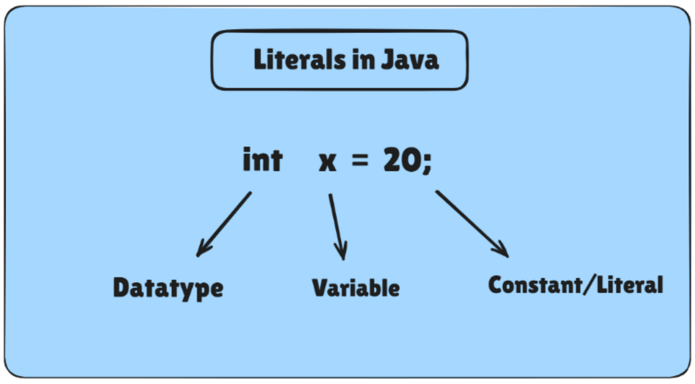
01-Literals

Literals in Java

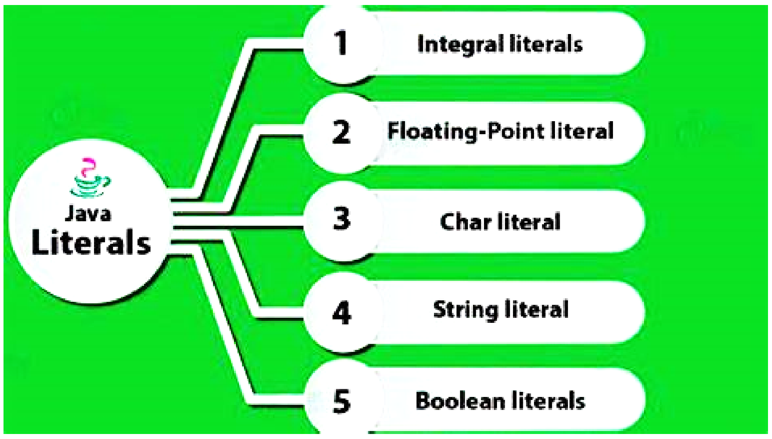
Literals in Java represent fixed values directly in the source code. They are constants that can be assigned to variables and come in various types.



Types of Literals in Java

Java supports several types of literals, including:

1. Integer Literals
2. Character Literals
3. Boolean Literals
4. String Literals
5. Floating Point Literals



Detailed Explanation of Each Literal Type

1. Integer Literals

Integer literals represent whole numbers and can be expressed in four different number systems:

- Decimal (Base 10): Uses digits 0-9.

int decimal = 101; // Decimal literal

- Octal (Base 8): Uses digits 0-7, prefixed with 0.

int octal = 0146; // Octal literal (equals 102 in decimal)

- Hexadecimal (Base 16): Uses digits 0-9 and letters a-f (or A-F), prefixed with 0x or 0X.

```
int hex = 0x123Face; // Hexadecimal literal
```

- Binary (Base 2): Uses digits 0 and 1, prefixed with 0b or 0B.

```
int binary = 0b1111; // Binary literal (equals 15 in decimal)
```

2. Character Literals

Character literals are single characters enclosed in single quotes. They can also represent special characters using escape sequences.

```
char letter = 'a';

char symbol = '%';

char unicodeChar = '\u000d'; // Unicode representation
```

3. Boolean Literals

Boolean literals represent truth values and can only be true or false.

```
boolean isJavaFun = true;

boolean isFishMammal = false;
```

4. String Literals

String literals are sequences of characters enclosed in double quotes.

```
String name = "Jack";

String number = "12345";

String newLine = "\n"; // Newline character
```

5. Floating Point Literals

Floating point literals represent numbers with fractional parts and can be of type float or double.

- Float Literals: Ends with F or f.

float price = 19.99f;

- Double Literals: Ends with D or d (optional).

double weight = 65.7;

double scientific = 1.234e2; // Exponent notation

Invalid Literals and Restrictions

Using underscores in numeric literals can enhance readability, but there are restrictions:

- Cannot start or end a number with an underscore.
- Cannot place an underscore adjacent to a decimal point in a floating-point literal.
- Cannot place an underscore adjacent to F or L suffixes.

Invalid Examples:

int invalid = 77_; // Invalid: underscore at the end

float invalidFloat = 6_.674F; // Invalid: underscore before decimal

Why Use Literals?

Literals are used to directly assign values to variables without needing to define constants separately. They simplify code by embedding constant values within the instructions.

FAQs on Literals

1. What are literals in Java?
    - Literals are fixed values assigned directly to variables in the source code.
2. Can literals be changed during program execution?

- No, literals are constants and cannot be changed once defined.

3. What is a real literal?
   - Real literals represent floating-point numbers, like 12.34 or 1.23e3.
4. What is a null literal?
   - A null literal represents the absence of an object reference, commonly assigned as null.