03-Ternary

The **ternary operator** is a concise way to write an if-else statement. It involves three operands: a condition, a result for when the condition is true, and a result for when the condition is false. This operator can help reduce the lines of code required for simple conditional assignments.

Syntax of Ternary Expression

```
condition ? result_if_true : result_if_false;
```

Implementation of Ternary Expression

Checking if a Number is Even or Odd: Using Simple if-else Approach

```
int n = 4;
int res = 0;

if (n % 2 == 0) {
    res = 10;
} else {
    res = 20;
}

System.out.println(res); // Output: 10
```

Checking if a Number is Even or Odd: Using Ternary Operator

```
int n = 4;
int res = (n % 2 == 0) ? 10 : 20;
System.out.println(res); // Output: 10
```

Example with Different Input:

```
int n = 5;
int res = (n % 2 == 0) ? 10 : 20;
```

```
System.out.println(res); // Output: 20
```

The ternary operator provides a one-line solution for conditional expressions, making the code more compact. However, it may not always be suitable for complex if-else statements that require multiple actions or complex logic.

Key Points

- **Conciseness:** The ternary operator reduces the number of lines needed for simple condition checks.
- **Readability:** While the ternary operator can make the code more compact, it should be used carefully to maintain readability, especially for more complex conditions.
- **Limitations:** It is best suited for straightforward if-else scenarios and may not work well for cases involving multiple statements or complex logic.