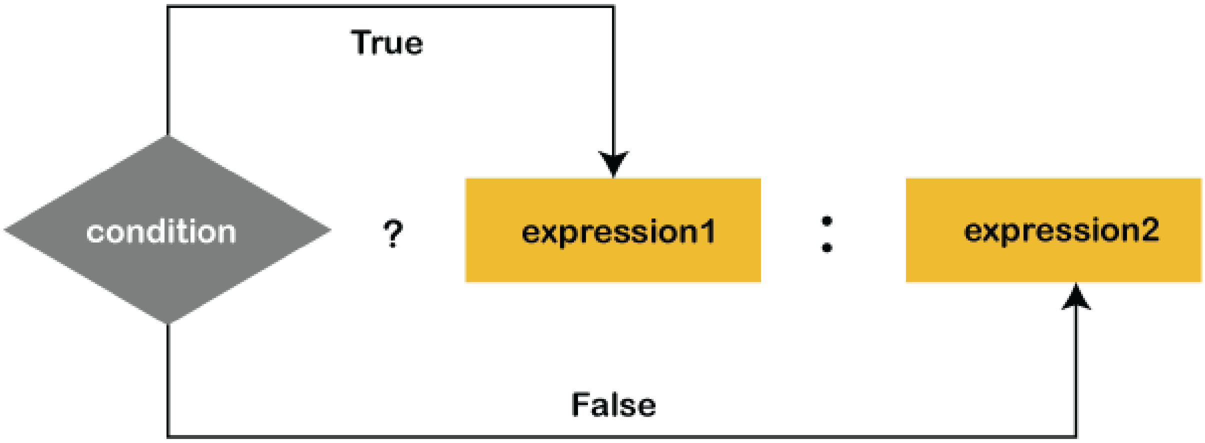
Ternary 0perator

We already learnt about this conditional statement so let us recall this in brief. The Java ternary operator lets you write concise *if…else* statements. Ternary statements get their name because they take three conditions. A ternary operator evaluates whether a statement is true or false and returns a specified value depending on the result of the operator.



Here is the syntax for a ternary operator in Java:

```
variable = (expression) ? expressionIsTrue : expressionIsFalse;
```

The origin of the name “ternary” refers to how a ternary operator has three parts. Our statement takes in three operands:

- **expression** is the expression the operator should evaluate
- **expressionIsTrue** is the value assigned to *variable* if the expression is true
- **expressionIsFalse** is the value assigned to *variable* if the expression is false

You do not need to assign the contents of the ternary operator to a variable. For instance, you could write a ternary operator in a System.out.println() statement. This will let you see the result of your ternary operator in the Java console.

Unlike an “if” statement, the ternary operator does not accept an “else” keyword. The ternary statement uses the colon (:) to represent an “else” condition.

Let’s use an example to show this operator in action.

Ternary Operator Java Example

Suppose that we are building a shopping website. We only want people to be allowed to buy products if they are aged 16 or over.

To verify the age of our customers, we could use a ternary operator. This is more efficient than using an "if" statement because a user can only be under 16 or aged 16 and over. Here is an example program that would accomplish the task of verifying the age of a user:

```
public class EvaluateAge {
        public static void main(String[] args) {
                int age = 22;
                String result = (age >= 16) ? "This user is over 16." : "This user is under 16."
                System.out.println(result);
        }
}
```

Our code evaluates our ternary. Our condition is true so our code returns:

```
This user is over 16.
```

First, we define a class called EvaluateAge. Then, we declare a Java variable called *age*. This variable stores the value of our customer's age. *age* is assigned the value 22.

We declare a variable called "*result*" whose value is equal to the result of our ternary operator. The ternary operator evaluates whether the user's "*age*" is equal to or greater than 16 ("*age >= 16*").

Note: Research more about ternary operator and how to use multiple conditions in it.