> *A boolean expression is an expression that evaluates to either true or false.*

It typically involves logical and/or relational operators and is used to make decisions or control the flow of a program.

Boolean expressions are commonly encountered in programming languages and are used in conditions, loops, and conditional statements.

Some key components of boolean expressions:

1. Relational Operators:

Relational operators compare two values and evaluate to either true or false. Common relational operators include:

- Equality (==): Checks if two values are equal.
- Inequality (!=): Checks if two values are not equal.
- Greater than (>): Checks if the left operand is greater than the right operand.
- Less than (<): Checks if the left operand is less than the right operand. Greater than or equal to (>=): Checks if the left operand is greater than or equal to the right operand.
- Less than or equal to (<=): Checks if the left operand is less than or equal to the right operand.

2. Logical Operators:

Logical operators combine boolean expressions and produce a boolean result. Common logical operators include:

- Logical AND (&&): Evaluates to true if both operands are true.
- Logical OR (||): Evaluates to true if at least one of the operands is true.
- Logical NOT (!): Negates the boolean value of the operand.

3. Boolean Variables:

Boolean variables are variables that can hold boolean values (true or false). They are often used to store the result of boolean expressions or to control program flow.

4. Boolean Constants:

Boolean constants directly represent boolean values. In many programming languages, the constants true and false are used to represent the boolean values.

Examples of boolean expressions:

- x > 5: Evaluates to true if the value of x is greater than 5.
- y != 0 && x > 10: Evaluates to true if y is not equal to 0 and x is greater than 10.
- a == true || b == false: Evaluates to true if either a is true or b is false.
- !(x <= 3): Evaluates to true if x is not less than or equal to 3.