**==================Answer#1=======================**

**Types of Operator In Dart Language:**

There are various type of operator but major is **Three…  
1) Arithmetic Operators  
2) Equality and Relational Operators  
3) Logical Operators**

**1. Arithmetic Operators:**

These class of operators contain those operators which are used to perform arithmetic operation on the operands. They are binary operators i.e they act on two operands. They go like this:

| **Operator Symbol** | **Operator Name** | **Operator Description** |
| --- | --- | --- |
| + | Addition | Use to add two operands |
| – | Subtraction | Use to subtract two operands |
| -expr | Unary Minus | It is Use to reverse the sign of the expression |
| \* | Multiply | Use to multiply two operands |
| / | Division | Use to divide two operands |
| ~/ | Division | Use two divide two operands but give output in integer |
| % | Modulus | Use to give remainder of two operands |

**Example: Using Arithmetic Operators in the program**

**Code:**

void main()

{

int a = 2;

int b = 3;

// Adding a and b

var c = a + b;

print("Sum of a and b is $c");

// Subtracting a and b

var d = a - b;

print("The difference between a and b is $d");

// Using unary minus

var e = -d;

print("The negation of difference between a and b is $e");

// Multiplication of a and b

var f = a \* b;

print("The product of a and b is $f");

// Division of a and b

var g = b / a;

print("The quotient of a and b is $g");

// Using ~/ to divide a and b

var h = b ~ / a;

print("The quotient of a and b is $h");

// Remainder of a and b

var i = b % a;

print("The remainder of a and b is $i");

}

**Output:**

Sum of a and b is 5

The difference between a and b is -1

The negation of difference between a and b is 1

Product of a and b is 6

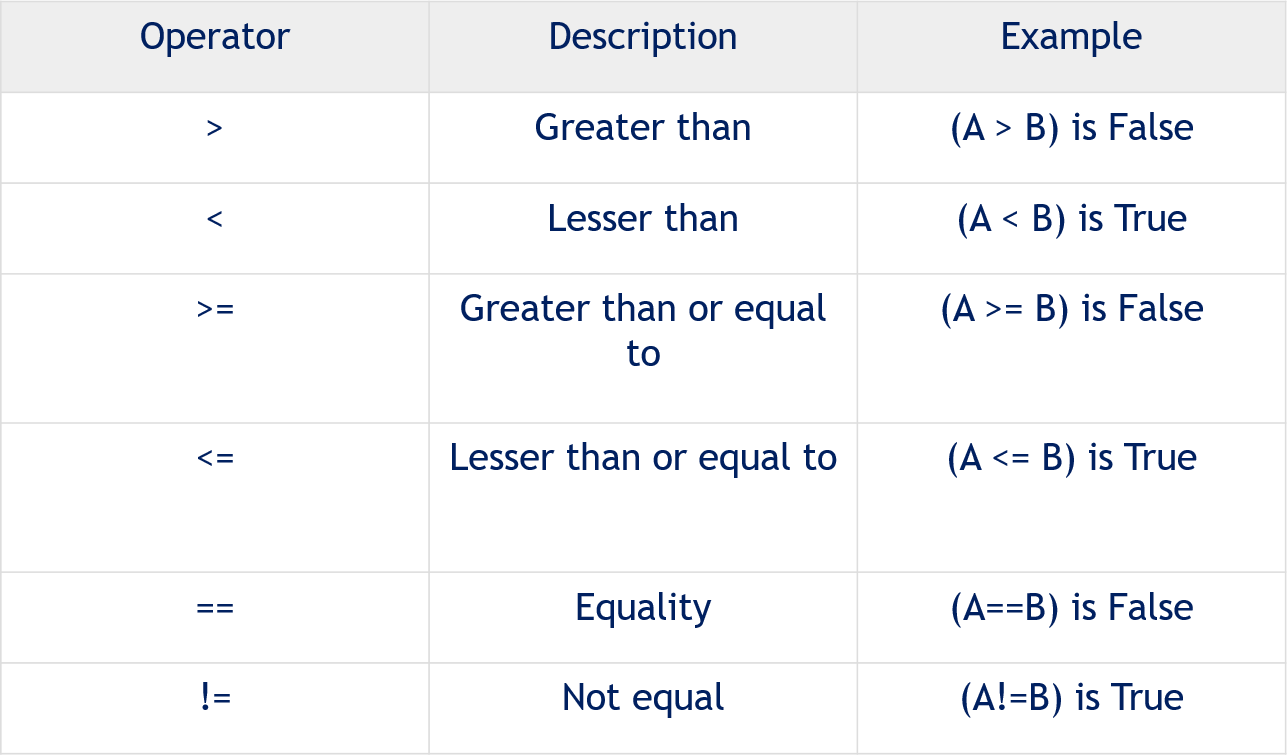
The quotient of a and b is 1.5

The quotient of a and b is 1

The remainder of a and b is 1

**Equality and Relational Operators:**

These class of operators contain those operators which are used to perform relational operation on the operands.



**Code:**

void main()

{

int a = 2;

int b = 3;

// Greater between a and b

var c = a > b;

print("a is greater than b is $c");

// Smaller between a and b

var d = a < b;

print("a is smaller than b is $d");

// Greater than or equal to between a and b

var e = a >= b;

print("a is greater than b is $e");

// Less than or equal to between a and b

var f = a <= b;

print("a is smaller than b is $f");

// Equality between a and b

var g = b == a;

print("a and b are equal is $g");

// Unequality between a and b

var h = b != a;

print("a and b are not equal is $h");

}

**Output:**

a is greater than b is false

a is smaller than b is true

a is greater than b is false

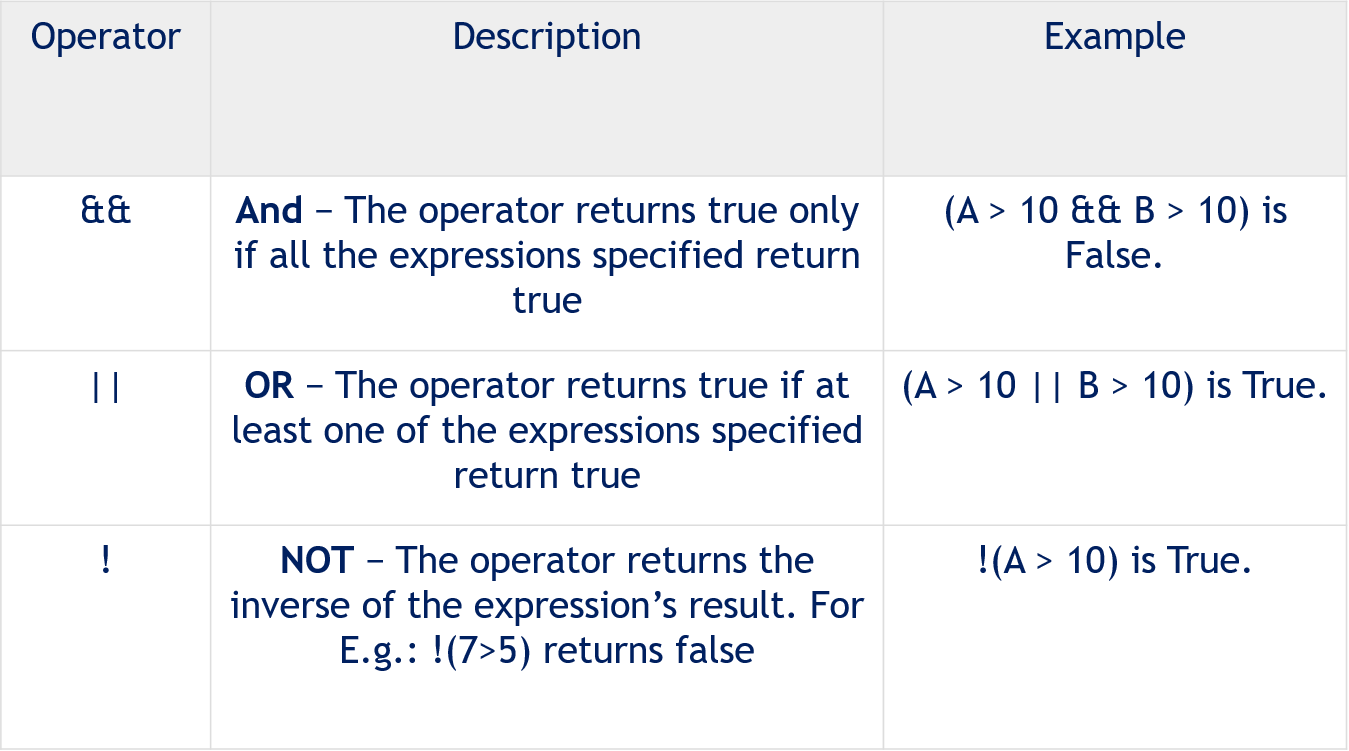
a is smaller than b is true

a and b are equal is false

a and b are not equal is true

**3) Logical Operators:**

These class of operators contain those operators which are used to logically combine two or more conditions of the operands.



**Code:**

void main()

{

int a = 5;

int b = 7;

// Using And Operator

bool c = a > 10 && b < 10;

print(c);

// Using Or Operator

bool d = a > 10 || b < 10;

print(d);

// Using Not Operator

bool e = !(a > 10);

print(e);

}

**Output:**

false

true

true

**==================== Answer#4==========================**

## **Conditional Expressions**

Dart has two operators that let you evaluate expressions that might otherwise require ifelse statements −

### **condition ? expr1 : expr2**

If condition is true, then the expression evaluates **expr1** (and returns its value); otherwise, it evaluates and returns the value of **expr2**.

### **expr1 ?? expr2**

If **expr1** is non-null, returns its value; otherwise, evaluates and returns the value of **expr2**

### **Example**

The following example shows how you can use conditional expression in Dart −

**Code:**

void main() {

var a = 10;

var res = a > 12 ? "value greater than 10":"value lesser than or equal to 10";

print(res);

}

**Output**

Vale is lesser than or equal to 10

### **Example**

Let’s take another example −

void main() {

var a = null;

var b = 12;

var res = a ?? b;

print(res);

}

**Output**

12

**==================Answer#6=======================**

**Types of Operator In Dart Language:**

Like other languages **(**[**C**](https://www.geeksforgeeks.org/c-programming-language/)**,**[**C++**](https://www.geeksforgeeks.org/c-plus-plus/)**,**[**Java**](https://www.geeksforgeeks.org/java/)**)**, whenever a variable is created, each variable has an associated data type. In **Dart language**, there is the type of values that can be represented and manipulated in a programming language. The data type classification is as given below:

| Data Type | Keyword | Description |
| --- | --- | --- |
| Number | int, double, num | Numbers in Dart are used to represent numeric literals |
| Strings | String | Strings represent a sequence of characters |
| Booleans | bool | It represents Boolean values true and false |
| Lists | List | It is an ordered group of objects |
| Maps | Map | It represents a set of values as key-value pairs |

**Number:**

**Code:**

void main() {

// declare an integer

int num1 = 2;

// declare a double value

double num2 = 1.5;

// print the values

print(num1);

print(num2);

var a1 = num.parse("1");

var b1 = num.parse("2.34");

var c1 = a1+b1;

print("Product = ${c1}");

}

**Output:**

2

1.5

Product = 3.34

**String:**

**Code:**

void main() {

String string = 'Hamza ''Ahmed ''Siddiqui';

String str = 'Coding is ';

String str1 = 'Fun';

print (string);

print (str + str1);

}

**Output:**

Hamza Ahmed Siddiqui

Coding is Fun

**Boolean:**

**Code:**

void main() {

String str = 'Coding is ';

String str1 = 'Fun';

bool val = (str==str1);

print (val);

}

**Output:**

False

**List:**

**Code:**

void main()

{

List gfg = new List(3);

gfg[0] = 'Hamza';

gfg[1] = 'Ahmed';

gfg[2] = 'Siddiqui';

print(gfg);

print(gfg[0]);

}

**Output:**

[Hamza, Ahmed, Siddiqui]

Hamza

**Map:**

**Code:**

void main() {

Map gfg = new Map();

gfg['First'] = 'Hamza';

gfg['Second'] = 'Ahmed';

gfg['Third'] = 'Siddiqui';

print(gfg);

}

**Output:**

{First: Hamza, Second: Ahmed, Third: Siddiqui}