Tech-tutorials 2k21

learn subjects easily

Master Core Java in 30 days

Day-3

Keywords

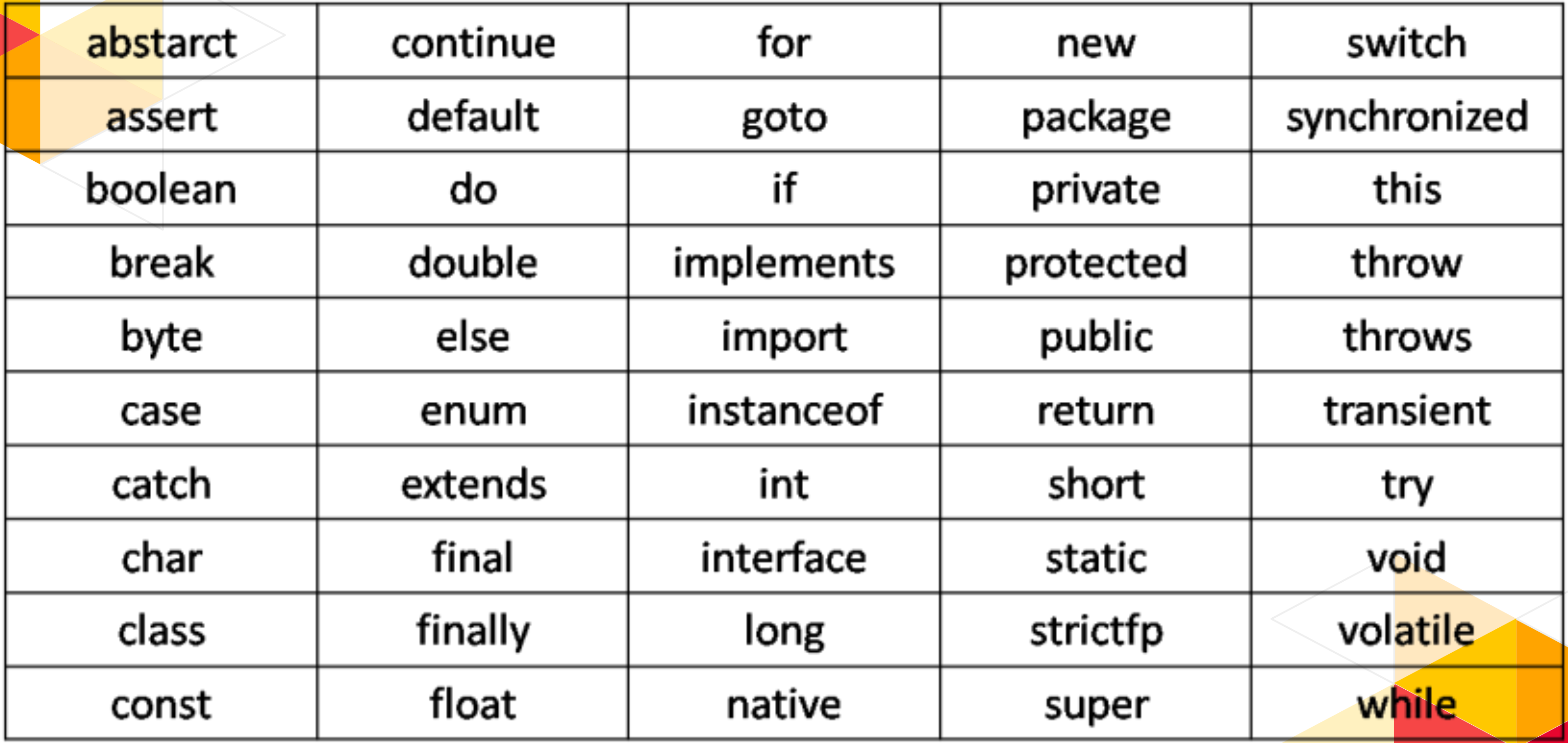
Data types

Operators

Core Java

@6:30 PM Today

Keywords



abstarct	continue	for	new	switch
assert	default	goto	package	synchronized
boolean	do	if	private	this
break	double	implements	protected	throw
byte	else	import	public	throws
case	enum	instanceof	return	transient
catch	extends	int	short	try
char	final	interface	static	void
class	finally	long	strictfp	volatile
const	float	native	super	while

keywords

Java has 64 reserved keywords. We can divide them into the following categories.

Primitive types and void:

boolean,byte,char,short,int,long,float,double,void

Modifiers:

public,private,protected,abstract,static,transient,final,volatile,synchronized,native

Declarations:

class ,interface,implements,extends,enum,package,throws

Control Flow:

if,else,switch.case,break,default,for,while,do,continue,try,catch,finally,throw,return

Miscellaneous:

this,new,super,import,instanceof,null,true,false,strictfp,assert,goto,const

Data Types

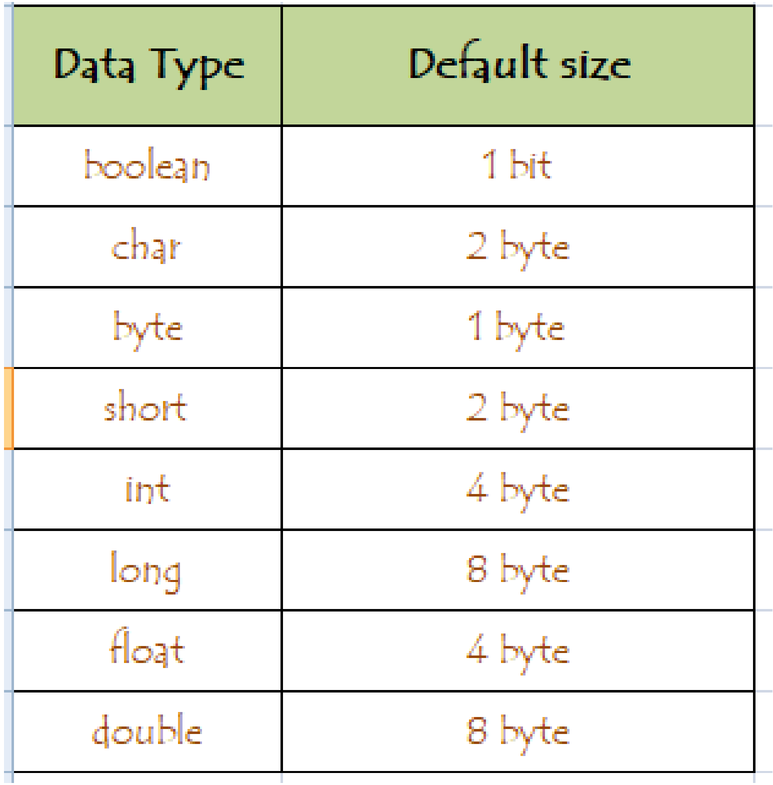
Data types specify the different sizes and values that can be stored in the variable.

There are two types of data types in Java:

Primitive data types: The primitive data types include boolean, char, byte, short, int, long, float and double.

Non-primitive data types: The non-primitive data types include class, interface and array

Sizes of primitive data types



Data Type	Default size
boolean	1 bit
char	2 byte
byte	1 byte
short	2 byte
int	4 byte
long	8 byte
float	4 byte
double	8 byte

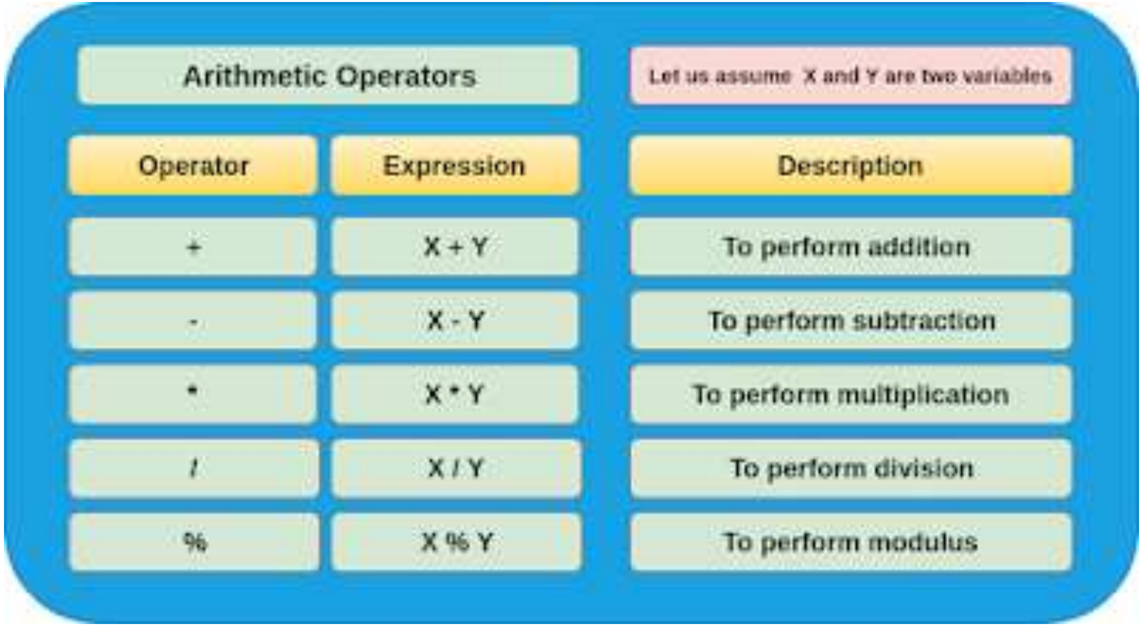
operators

is a symbol that is used to perform operations

1. Unary Operator,
2. Arithmetic Operator,
3. Shift Operator,
4. Relational Operator,
5. Bitwise Operator,
6. Logical Operator,
7. Ternary Operator and
8. Assignment Operator.

Arithmetic Operators

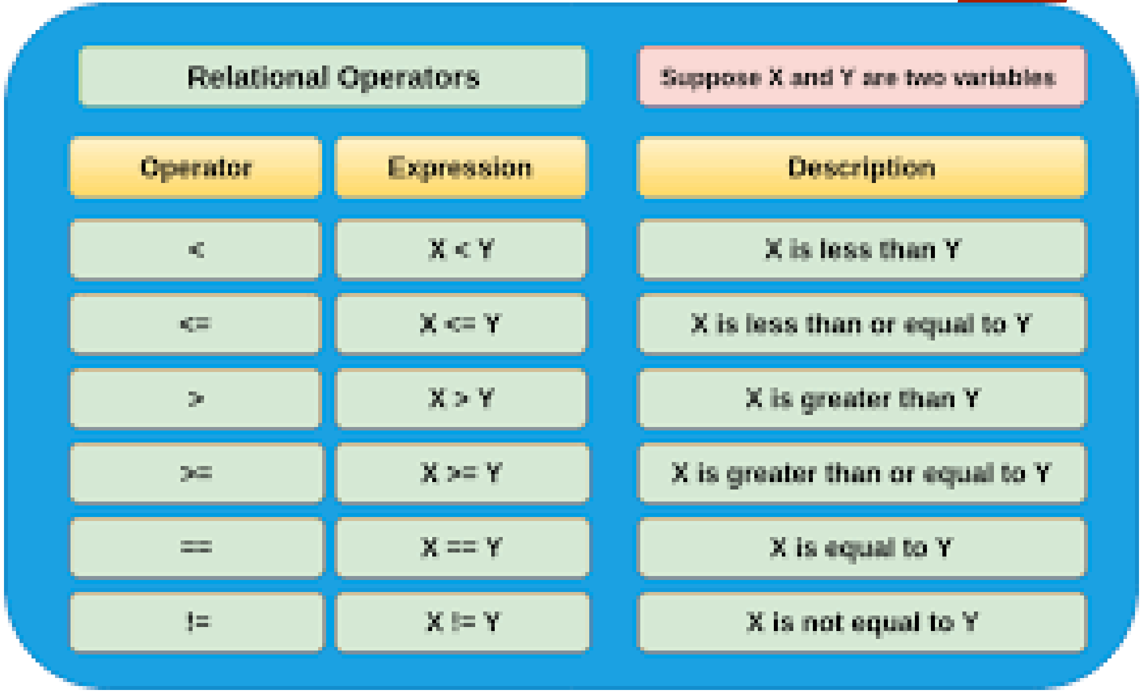
An arithmetic operator performs mathematical operations such as addition, subtraction, multiplication, division etc



Arithmetic Operators		Let us assume X and Y are two variables
Operator	Expression	Description
+	X + Y	To perform addition
-	X - Y	To perform subtraction
*	X * Y	To perform multiplication
/	X / Y	To perform division
%	X % Y	To perform modulus

Relational Operators

A relational operator checks the relationship between two operands. If the relation is true, it returns 1; if the relation is false, it returns value 0.



Relational Operators

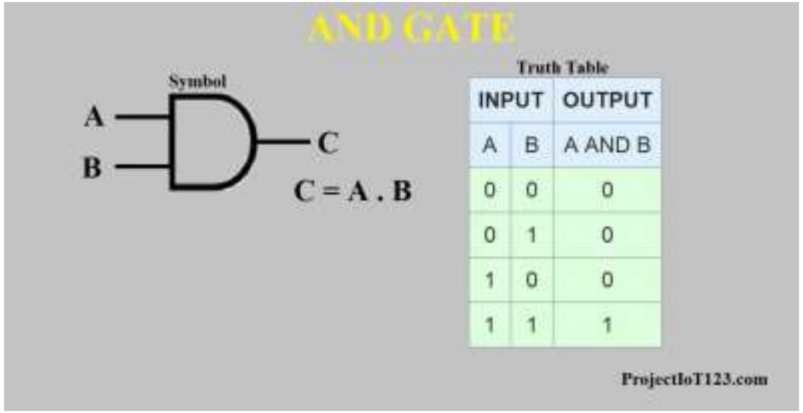
Suppose X and Y are two variables

Operator	Expression	Description
<	X < Y	X is less than Y
<=	X <= Y	X is less than or equal to Y
>	X > Y	X is greater than Y
>=	X >= Y	X is greater than or equal to Y
==	X == Y	X is equal to Y
!=	X != Y	X is not equal to Y

Logical Operators

Logical operators are commonly used in decision making in programming. Used to perform logical operations

OPERATORS	OPERATORS NAME
||	Logical OR operator
&&	Logical AND operator
!	Logical NOT operator

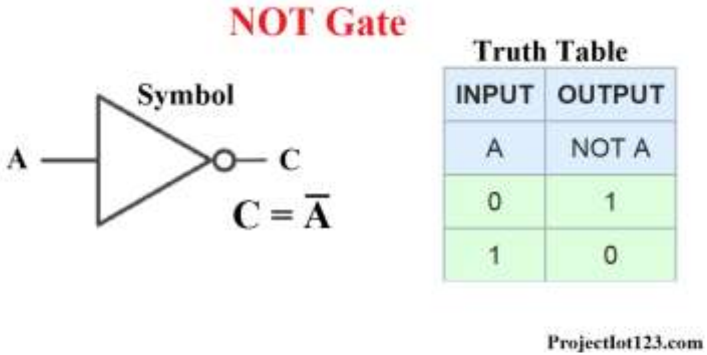


AND GATE

Truth Table

INPUT		OUTPUT
A	B	A AND B
0	0	0
0	1	0
1	0	0
1	1	1

ProjectIoT123.com

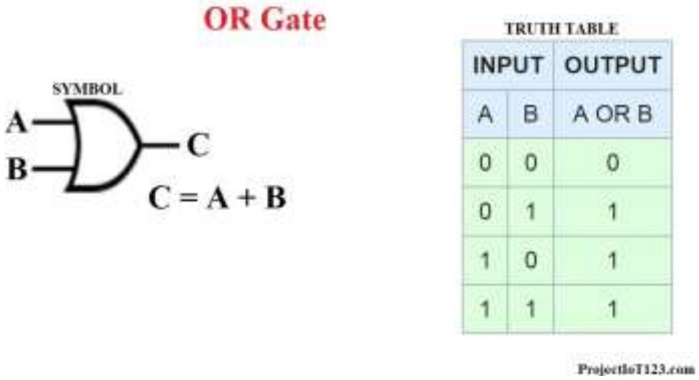


NOT Gate

Truth Table

INPUT	OUTPUT
A	NOT A
0	1
1	0

ProjectIot123.com



OR Gate

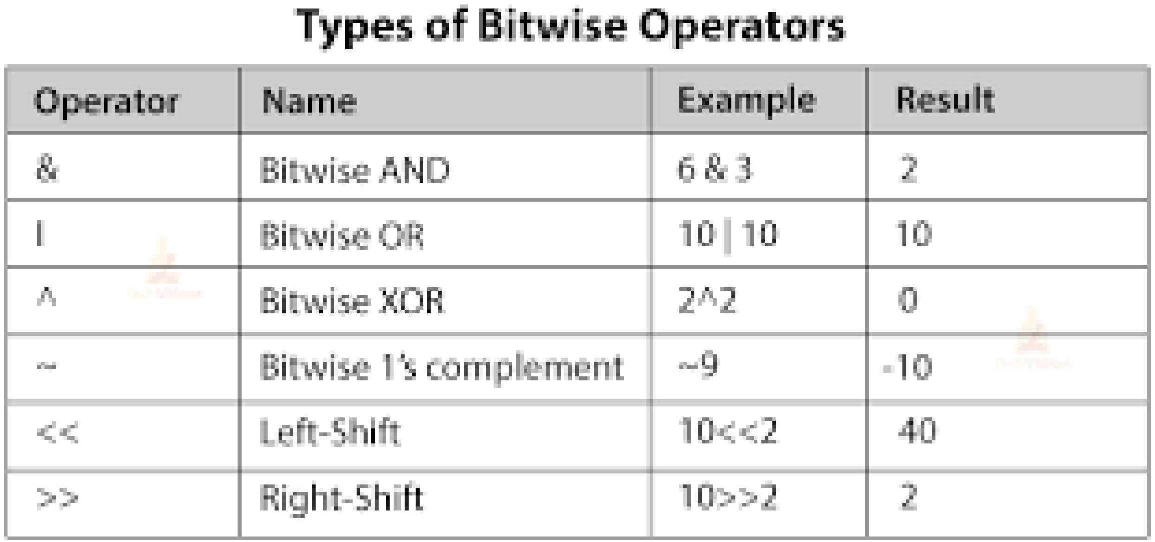
TRUTH TABLE

INPUT		OUTPUT
A	B	A OR B
0	0	0
0	1	1
1	0	1
1	1	1

ProjectIoT123.com

Bitwise Operators

These operators are used to perform operations at bit-Level



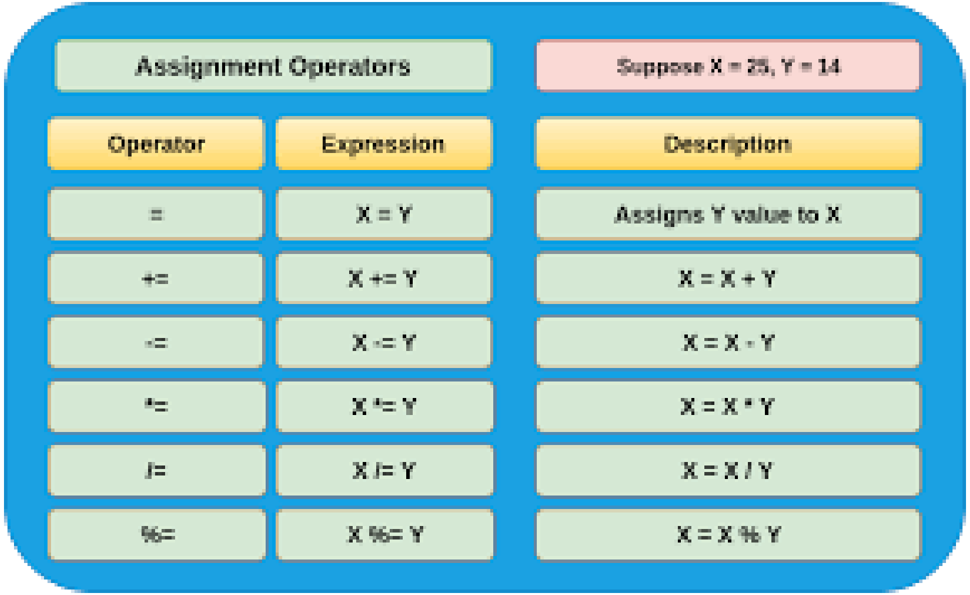
Types of Bitwise Operators

Operator	Name	Example	Result
&	Bitwise AND	6 & 3	2
|	Bitwise OR	10 | 10	10
^	Bitwise XOR	2^2	0
~	Bitwise 1's complement	~9	-10
<<	Left-Shift	10<<2	40
>>	Right-Shift	10>>2	2

Assignment Operators

These operators are used to perform assignment operations

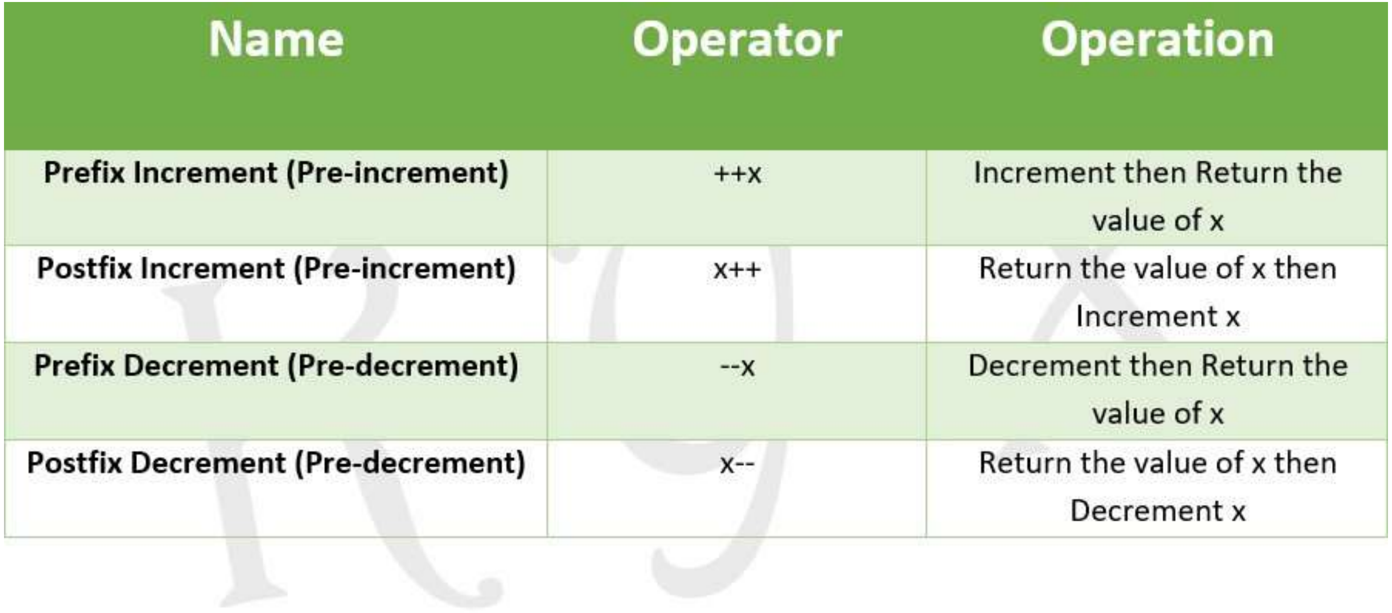
Used for short cut notations in java Language



Assignment Operators		Suppose X = 25, Y = 14
Operator	Expression	Description
=	X = Y	Assigns Y value to X
+=	X += Y	X = X + Y
-=	X -= Y	X = X - Y
*=	X *= Y	X = X * Y
/=	X /= Y	X = X / Y
%=	X %= Y	X = X % Y

Unary Operators

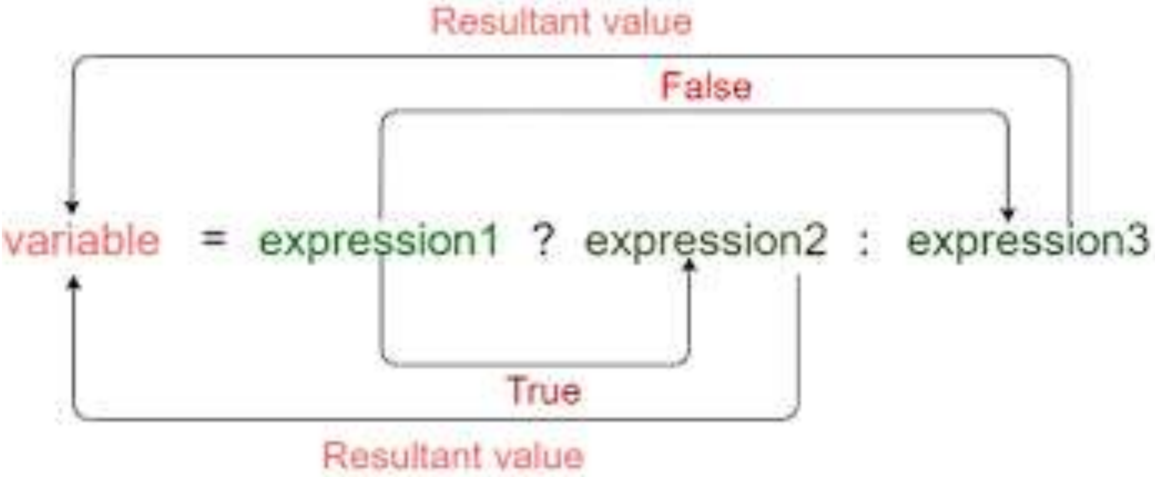
These operators used to perform increment (adding 1) or decrement operation (Subtracts 1) to a variable.



Name	Operator	Operation
Prefix Increment (Pre-increment)	++x	Increment then Return the value of x
Postfix Increment (Pre-increment)	x++	Return the value of x then Increment x
Prefix Decrement (Pre-decrement)	--x	Decrement then Return the value of x
Postfix Decrement (Pre-decrement)	x--	Return the value of x then Decrement x

Conditional Operators

This operator is known as ternary operators
If first exp evaluates true then second exp is true else it is going to evaluate third exp.
Similar to if-else statement



Tech-tutorials 2k21

Thank You

Please Subscribe, Like and Share
My videos