**Arithmetic operators :**

Mathematical operations were used in programming such as :

var a=5;

var b=9;

c=a+b;

d=a-b;

e=a\*b;

f=a/b;

g=a\*\*b;

h=a^b;

i=a%b;

console.log(c);

console.log(d);

console.log(e);

console.log(f);

console.log(g);

console.log(h);

console.log(i);

**assignment operator :**

there is a short of mathematical operations in programming :



var a=5;

var b=9;

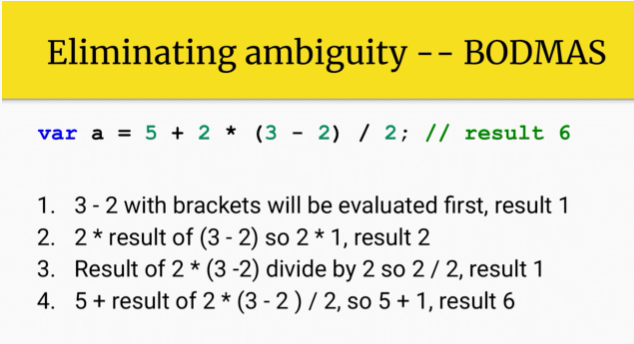
a\*=b;

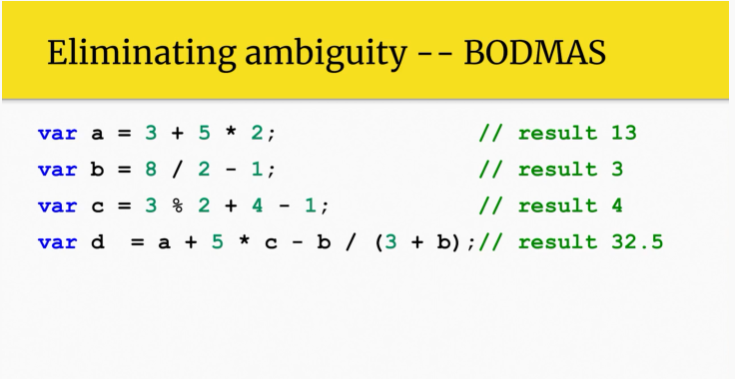
b-=a;

console.log(a);

console.log(b);

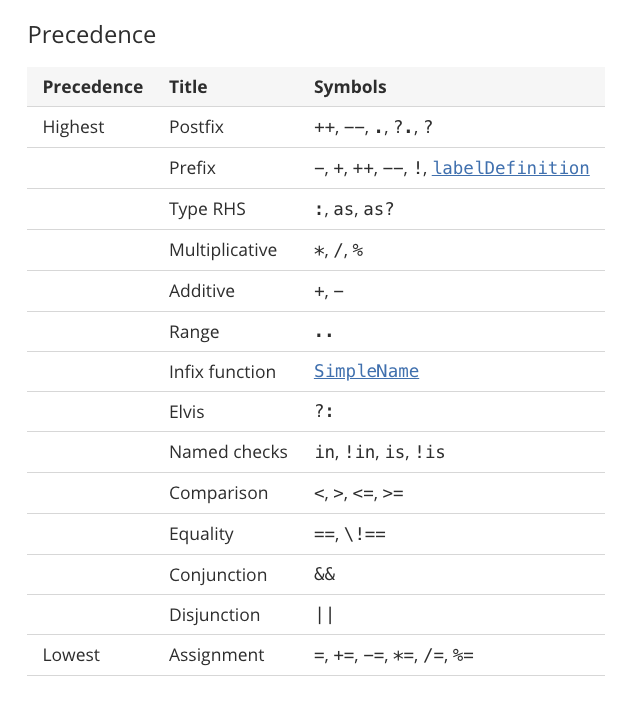
**expression evaluation : (BODMAS)**





**Operator precedence :**

There are other operators apart from mathematical + - / \* etc operators.this operators list are given below:



Examples :

var a=3;

var b=8;

var c = a-- + ++a / b++    // (2+4) / 9 => 6/9=0.6

alert(a);

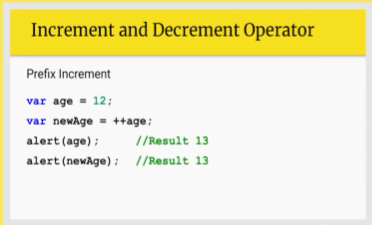
alert(b);

alert(c);

increment / decrement operator also called unary operators.this can be use as postfix a++ and prefix ++a; .this operators used for sigle value increment or decrement.

Prefix increment example :

First value is increment and the variable is then added.



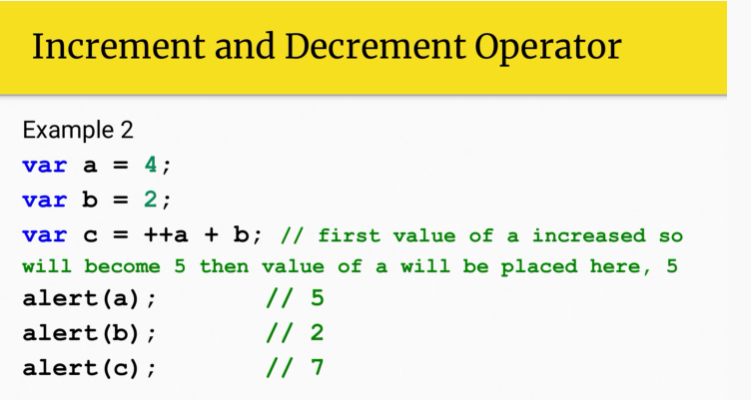
Postfix increment example :

Since value first comes in and then value in variable is incremented.

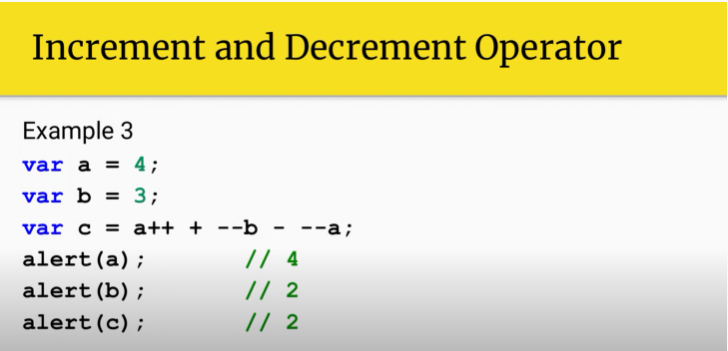
Graphical user interface, text, application

Description automatically generated

More examples :



Text

Description automatically generated  a=2;

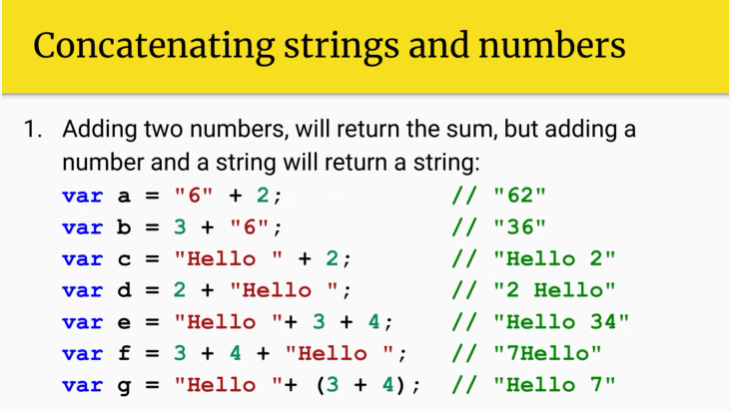
var b=2;

var c=a++ + --b - --a;  //3-1-1 = 2-1 = 1

alert(c);

**concatenation :**

merge two variables together :



Examples :

var a="4"+2;

var b=21+"pak";

var c=5+6+"3"+"so"

var d="try"+(3-2);

document.write(a+" \n");document.write(" \n"+b);document.write(" \n"+c);document.write("\n"+d);

**comparison operator :**

used to compare value and gave output accordingly.when comparing is exactly the same as postdefined value then it return true otherwise false.we can use to do other logical operations based on camparison operator’s result.usually it comes up with loops or mathematical functions alongside.

Table

Description automatically generated

Example:

var a=4;

var b="4";

var c=3;

var d=11;

console.log(a==b);  //true

console.log(a===b);  // false

console.log(a>c);   //true

console.log(c>=d);  // false

console.log(c!=d);  // true

***logical operator :***

it gives you result in true or false which you can use with your numeric data to identify and manipulate your variables accordingly to the logic you applied.

A **logical operator** is a symbol or word used to connect two or more expressions such that the value of the compound expression produced depends only on that of the original expressions and on the meaning of the operator.[[1]](https://press.rebus.community/programmingfundamentals/chapter/logical-operators/#footnote-215-1) Common logical operators include AND, OR, and NOT

var a=50;

var b=40;

var c=a>40 && a<60;

console.log(c); //true

var d=b<a && b>60;

console.log(d); // false

var e=a>30 || a<40;

console.log(e); //true

var f=b>a || b<39;

console.log(f) // false

var g=!(a>40);

console.log(g); // result false actually true

var h=!(b<36);

console.log(h); // resuot true actually false

var i=!false;

console.log(i); // answer will be true but actually false

var j=!!a;

console.log(j); // a is true then it becomes false then it becomes true

why they are called short circuit operators :

in ANDs case it checks first operand if result is not true it will not further checks other expression if available it returns directly false.

Utilization of logical operators , comparison operators comes in loops and if else or switch case conditions.

A picture containing graphical user interface

Description automatically generated