## Assignment3

## Q1. Write the Code for all Data types.

/\*///data types coding//

let a=25

console.log(a)

console.log(typeof a)// number

//console.log(b)//not define

let c;

console.log(c)// undefined

let d=null

console.log(d)// null/// not understand the use of null

console.log(typeof d)// object

let e="Mohit"

console.log(typeof e)// string

\*/

//\*\*\*operators coding \*/

//  Arithmetic operators\*\*//

/\*let a=20;

let b=10;

let add= a+b;

console.log(add)//30

let sub=a-b;

console.log(sub)//10

let multi=a\*b;

console.log(multi)//200

let divd=a/b;

console.log(divd)//2

let c=5

let d=3

let mode=c%d;

console.log(mode)//2

let exponational=c\*\*\*d;

console.log(exponational)//125

\*/

//\*\*compare operators\*\*///

/\*a=10

b=20

console.log(a>b)// false

console.log(a<b)//true

console.log(a>=b)// false

console.log(a<=b)//true

console.log(a==b);// false

console.log(a=b);//20 but b value in a

console.log(a)//20 ac to upper line code

console.log(a===b)//true because a=b in 43 line//\*/

//\*\*\*Assignment operator \*/

//let a=10

//let b=20

//console.log(a+=b)// working same a+b

//console.log(a-=b)// working same a-b

//console.log(a\*=b)// working same as a\*b

//console.log(a/=b)// working same as a/b

//console.log(a%=b)

//console.log(a\*\*=b)// a\*\*b working

//\*\* Shifting Assignment operators \*/

//let a=20;

//console.log(a>>=2)// 20/2=10/2=5

//console.log(a<<=2)//20\*2=40\*2=80

//console.log (a>>>=2)// samajh nhii aaya

//\*\*\*increment decrement operators \*/

//let a=5

//console.log(++a)//6 first increment then execute

//console.log(a++)//5 first execute then increment

 //console.log(--a)//4 first decrement then execute

 //console.log(a--)// 5 first execute then decremen

## Q2. What are the uses of JavaScript Operators?

JavaScript operators are symbols that are used to perform operations on operands (values and variables). [There are different types of JavaScript operators such as **arithmetic**, **assignment**, **comparison**, **logical**, **conditional** and **type** operators](https://www.w3schools.com/js/js_operators.asp).

Q3. Explain the Assignment Operator.

The **assignment operator** in JavaScript is represented by the symbol =. [It is used to assign a value to a variable or property 1](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Operators/Assignment). The syntax for the assignment operator is as follows:

Assignment operators assign values to JavaScript variables.

|  |  |  |
| --- | --- | --- |
| **Operator** | **Example** | **Same As** |
| = | x = y | x = y |
| += | 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 |
| \*\*= | x \*\*= y | x = x \*\* y |

Shift Assignment Operators

|  |  |  |
| --- | --- | --- |
| **Operator** | **Example** | **Same As** |
| <<= | x <<= y | x = x << y |
| >>= | x >>= y | x = x >> y |
| >>>= | x >>>= y | x = x >>> y |

Bitwise Assignment Operators

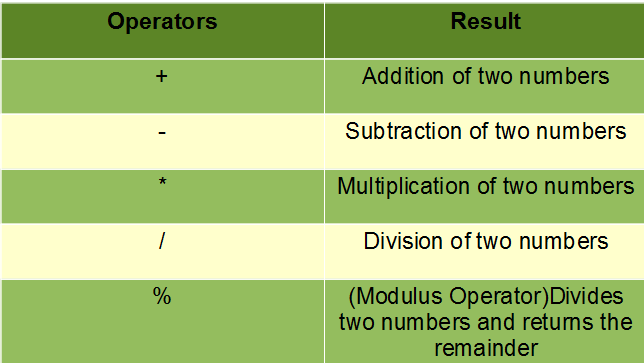
|  |  |  |
| --- | --- | --- |
| **Operator** | **Example** | **Same As** |
| &= | x &= y | x = x & y |
| ^= | x ^= y | x = x ^ y |
| |= | x |= y | x = x | y |

Logical Assignment Operators

|  |  |  |
| --- | --- | --- |
| **Operator** | **Example** | **Same As** |
| &&= | x &&= y | x = x && (x = y) |
| ||= | x ||= y | x = x || (x = y) |
| ??= | x ??= y | x = x ?? (x = y) |

## Q4. Explain the all Arithmetic Operators.

These operators involve the mathematical operators that can be used to perform various simple or advanced arithmetic operations on the primitive data types referred to as the operands. These operators consist of various unary and binary operators that can be applied on a single or two operands. Let’s look at the various operators that Java has to provide under the arithmetic operators.



## Q5. What are the Binary Operators in JS.

Binary operators are these operators required to

* Multiplicative Operators
  + The \* Operator
  + The / Operator
  + The % Operator
* Additive Operators
  + The Addition operator (+)
  + The Subtraction Operator (-)
* Bitwise Shift Operators
  + The Left Shift Operator (<<)
  + The Signed Right Shift Operator (>>)
  + The Unsigned Right Shift Operator (>>>)
* Relational Operators
  + The Less-than Operator (<)
  + The Greater-than Operator (>)
  + The Less-than-or-equal Operator (<=)
  + The Greater-than-or-equal Operator (>=)
  + The instanceof operator
  + The in operator
* Equality Operators
  + The Equals Operator (==)
  + The Does-not-equals Operator (!=)
  + The Strict Equals Operator (===)
  + The Strict Does-not-equal Operator (!==)
* Binary Bitwise Operators (&, ^, |)
* Binary Logical Operators (&&, ||)

Technically speaking, also the assignment and comma operators are binary.

## Q6. Explain the Pre-Increment and Post Increment in JS.

Pre-increment means first increment then execute like a=5 , ++a=6

Post increment means first execute then increment like a=5, a++ =5 after then 6