Javascript -5

object literals →

- these are like data structures or a type of data structure.
- a little difference between object literals and objects. object literals means to create a object type collection or data structure.
- used to store keyed collections & complex entities

{ • property = (key, value) pair
• object = properties collection

Create object literals →

let/const object variablename = {

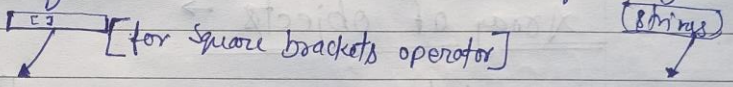
key : value ,
key : value ,
key : value

} ;

- const object literals are like const arrays we can change inside properties because these are work on reference method that store memory address
- We can define array as a property inside object literals

Get value →

- object ["key"]; (prefer for variables)
- object.key ; (prefer for keys)



these keys are converted to strings by JS than we use them and get the value associated with it. this works on only square brackets .in case of dot we can't use number.

add / update value →

update → object.key = Newvalue ;

Add → object.key = Newvalue ;

(access method = New/updated value)

remove → delete object.key ;

DATE

Nested objects → "object of object"

```
const/let objectname = {
        key : {
                key1 : value ;
                key2 : value
        }
}
```

Array of objects →

```
const/let Arrayname = [
        {
                key : value,
                key : value
        },
        {
                key : value,
                key : value
        }
]
```

Math object →

Math.PI; → give the value of Pi. π

Math.PI; → give the value of E (eular number)

Absolute

Math.abs (-12) → give the value of num without any sign of '+' or '-'.
(num)
12

Power

Math.Pow (2,4) → give the value by solve (a^b / 2^4)
2^4

Math.floor (5.6) → give the least value by round off the number (nearest smallest Integer)
5

Math.ceil (5.2) → Largest nearest Integer

Math.random ( ) → A number returned between 0 and 1 (not include 1) (never return 1)
0.24375

Random Integers →

step 1 ▷ let num = Math.random ( );

step 2 ▷ num = num * 10 ; [10 is a max limit of needed random number]

step 3 ▷ num = Math.floor(num);

step 4 ▷ num = num + 1; [include max limit] [remove decimal number]

Short Method ⇒ Let random = Math.floor (Math.random( ) * 10)+1 ;

DATE

In the step 4 we added the plus 1 it can increase the min limit if we plus 20 than the result is between 20 and (20 + max limit [10]) means between 20 and 30.

and here is 30 not included if we want 30 to be included than we have to make the 20 to 21 but this make exclude to 20.