­­To convert anything into array :

Cosnt arr=[data] // data is which we want to convert it to array .

CONCEPT OF SPREAD OPERATOR :

To generate random number :

Math.random() // going to give no between 0-1 exclude 1;

Math.random()\*10 // going to give floating no between 0 ,9;

Math.floor(Math.random()\*10) // going to give integer no between 1 – 9 ;

Note : by using spread operator we can divide a string into array of single character

E . g :

*const* arr1="123456789";

*const* arr2=[...arr1];

console.log(arr2);

🡪 MAKING AN EMPTY ARRY OF GIVEN LENGTH

cons newer=new Array (7);

console.log(newer);

note :

const arr=[“item1” , “item2”]

const arr1=[“item1” ,”item2”]

note we can add both arr and arr1 in set because both of them arr different though they have same arrays value because both of them will have different refrence

let set=new Set();

set.add([1,2,3,4,5]);

set.add([1,2,3,4,5]);

console.log(set);

* Difference between map and object :

Object : in object we have only key of type of string or symbol (we do not have a key of number or other data types ) e . g .

const obj={name: "ajaj",

age :17,

1:34};

*for*(let i in obj){

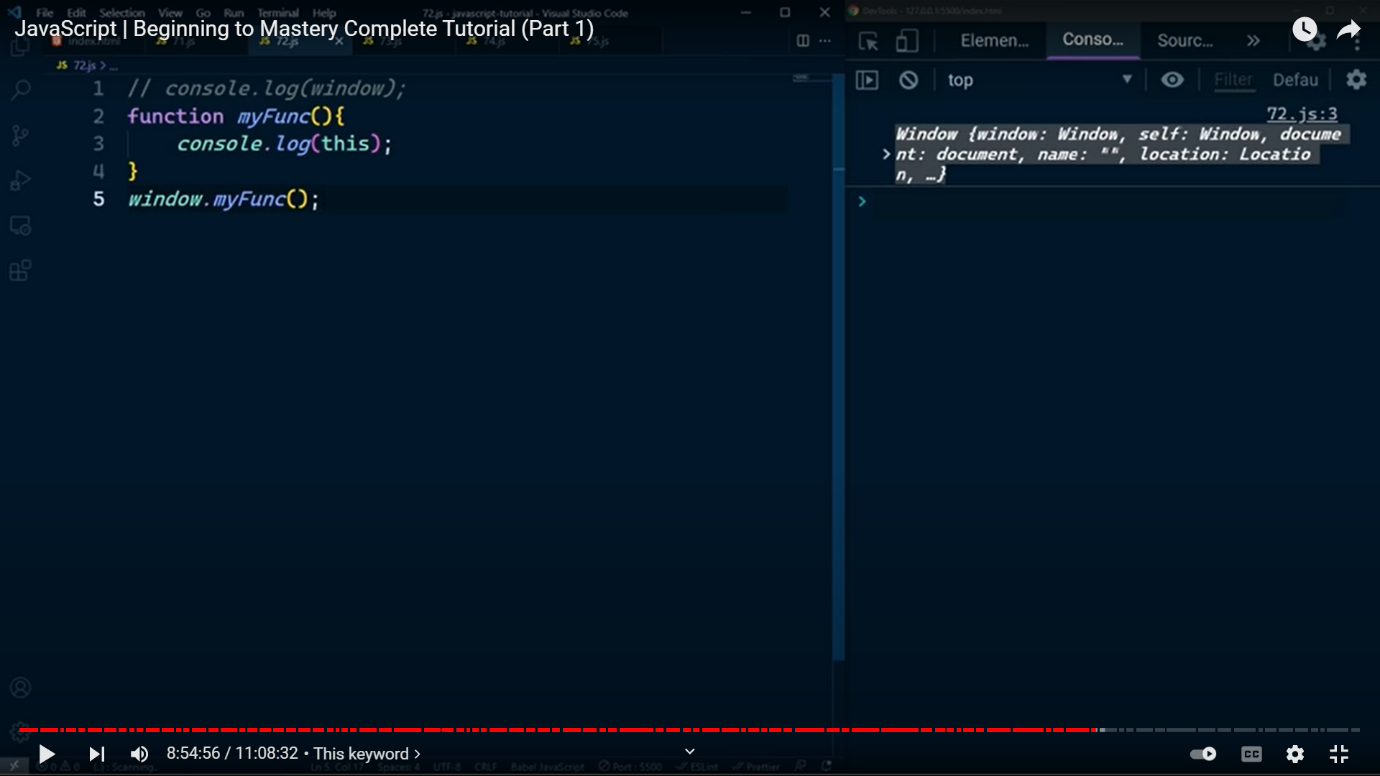
    console.log(typeof(i));

}

Note : 99.99% of the case we will have string as data type s

Map : we can have key of different data types (we can a have a key of number or string or of any other data types )

* IMPORTANCE OF USER STRICT MODE :



* REMOVE DUPLICATE FROM ARRAYS :





Array's length property is mutable whereas String's length property is immutable. String is immutable, hence its length property is also immutable.

Changing the Array's length property will modify the array itself by adding some space or, removing elements at the end.

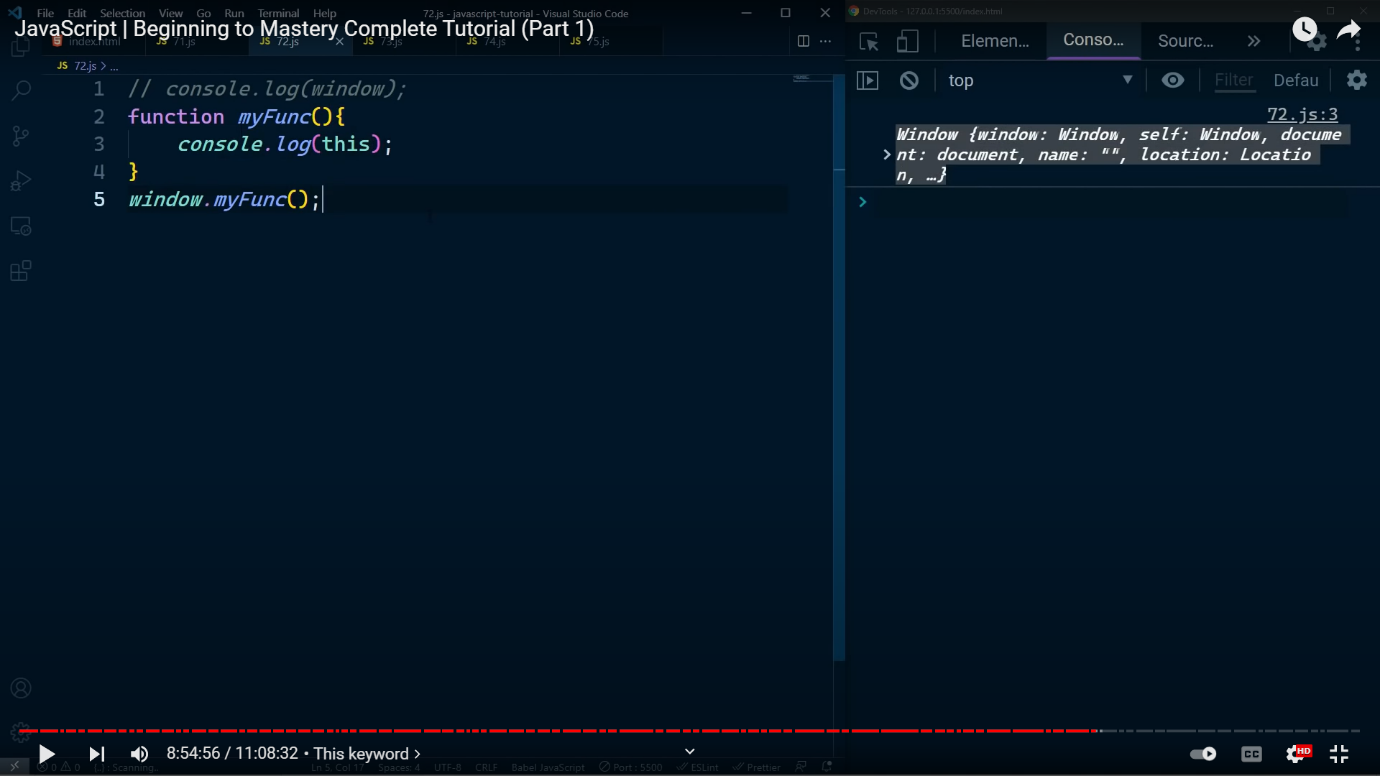
Changing String's length property won't result into any Error. The operation is just ignored.

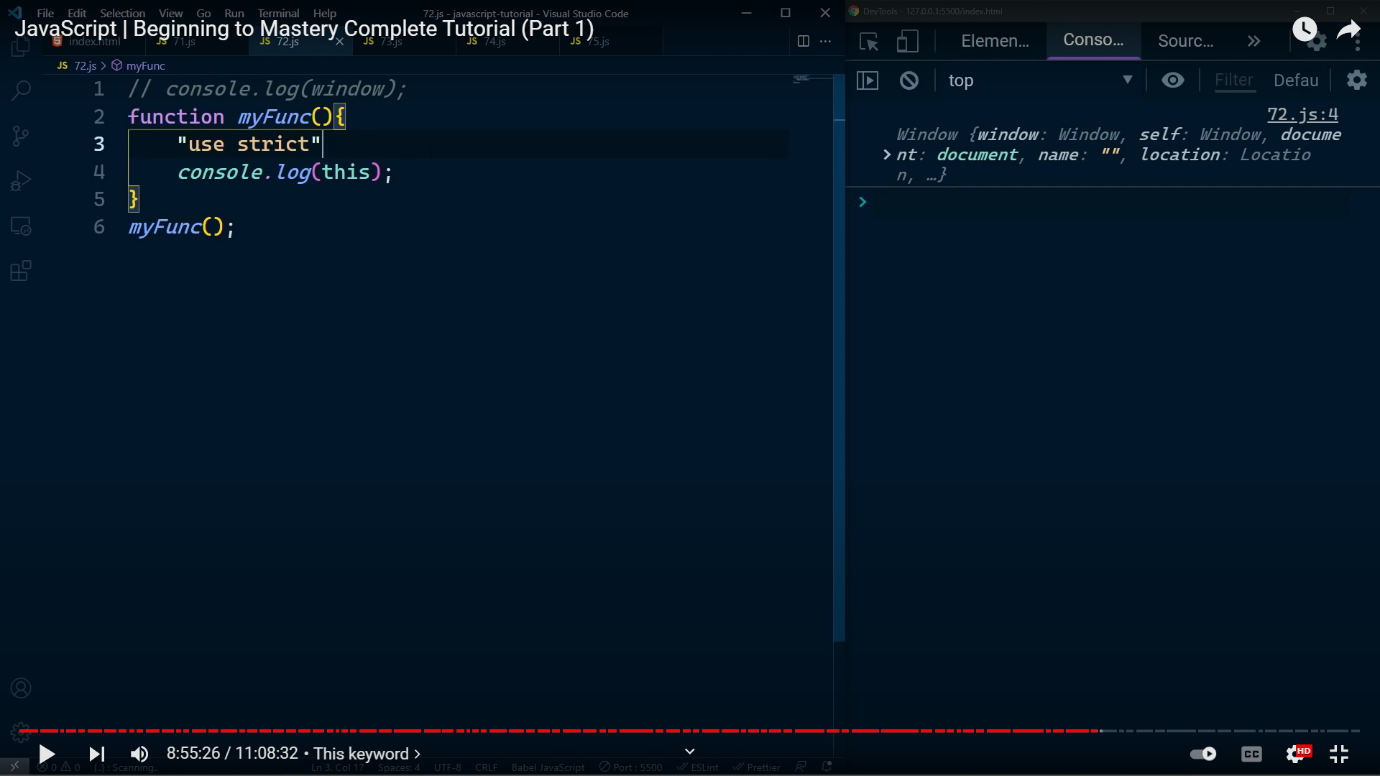
* WAY OF CONVERTING ANYTHING INTO ARRAYS



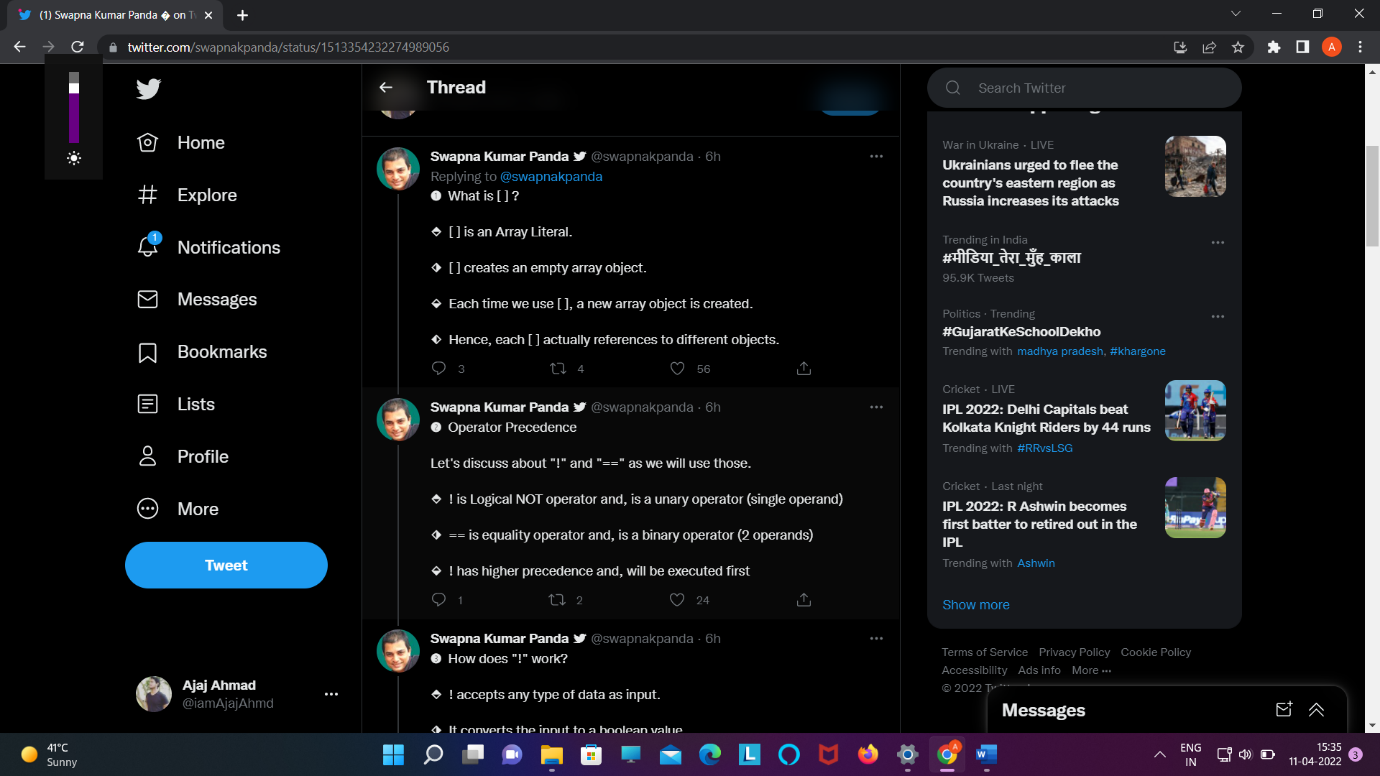
Note :it will only going to work on the iterable object or array like object

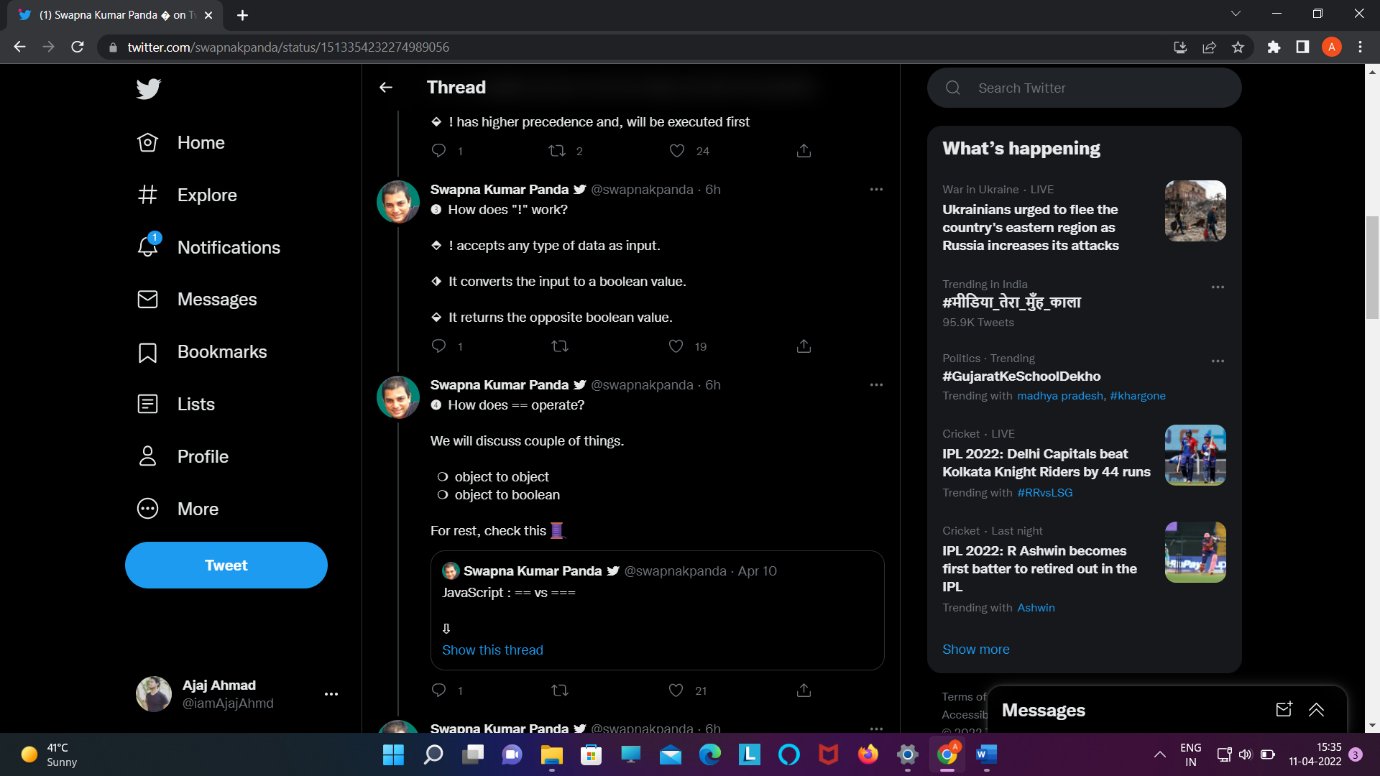
Difference of of using strict mode or not using strict mode :

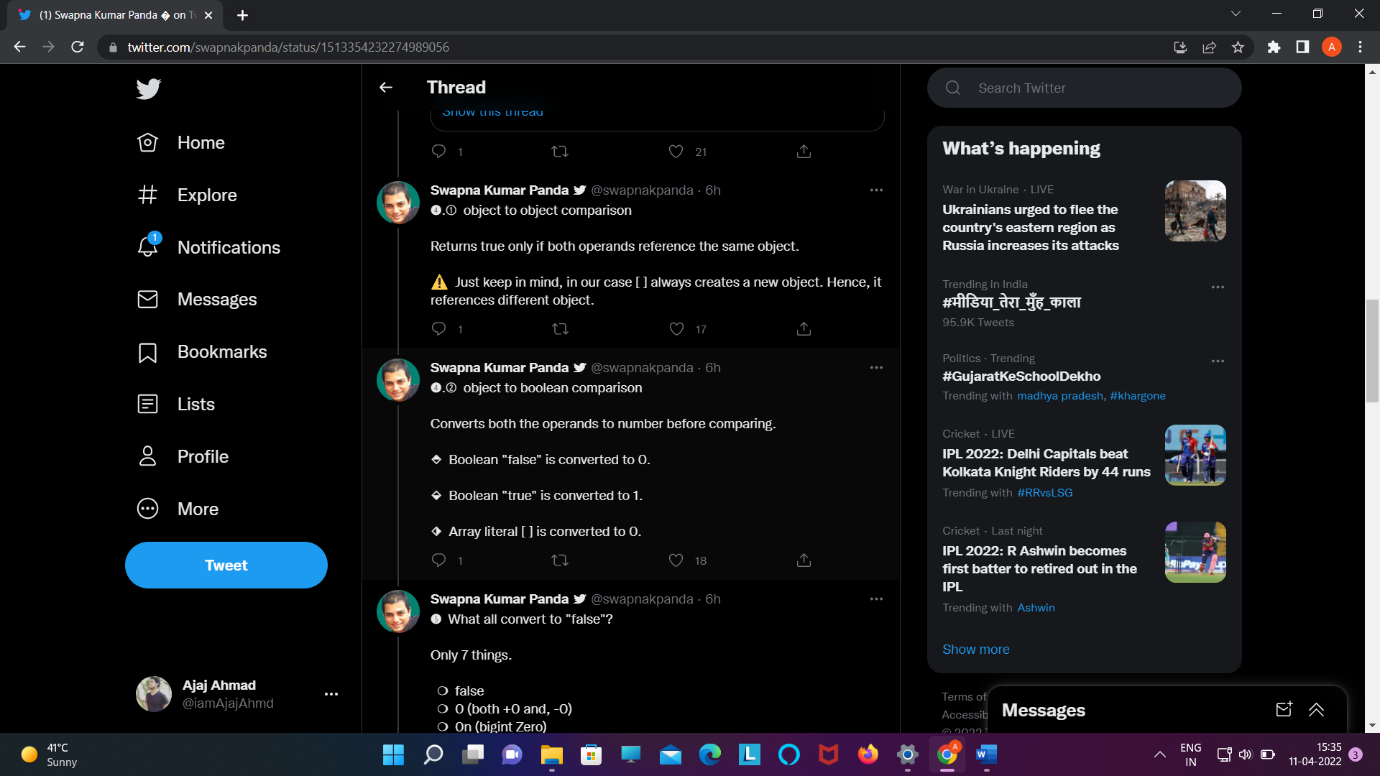


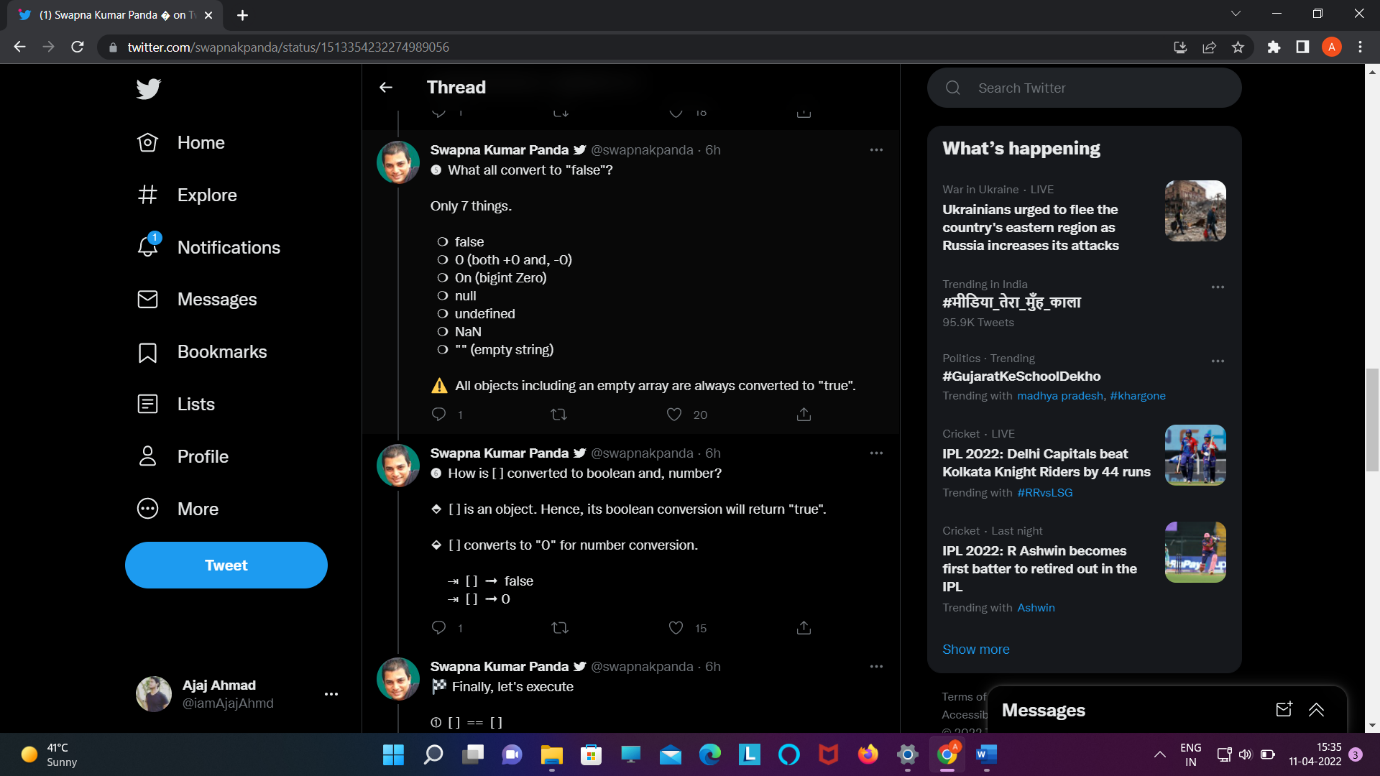


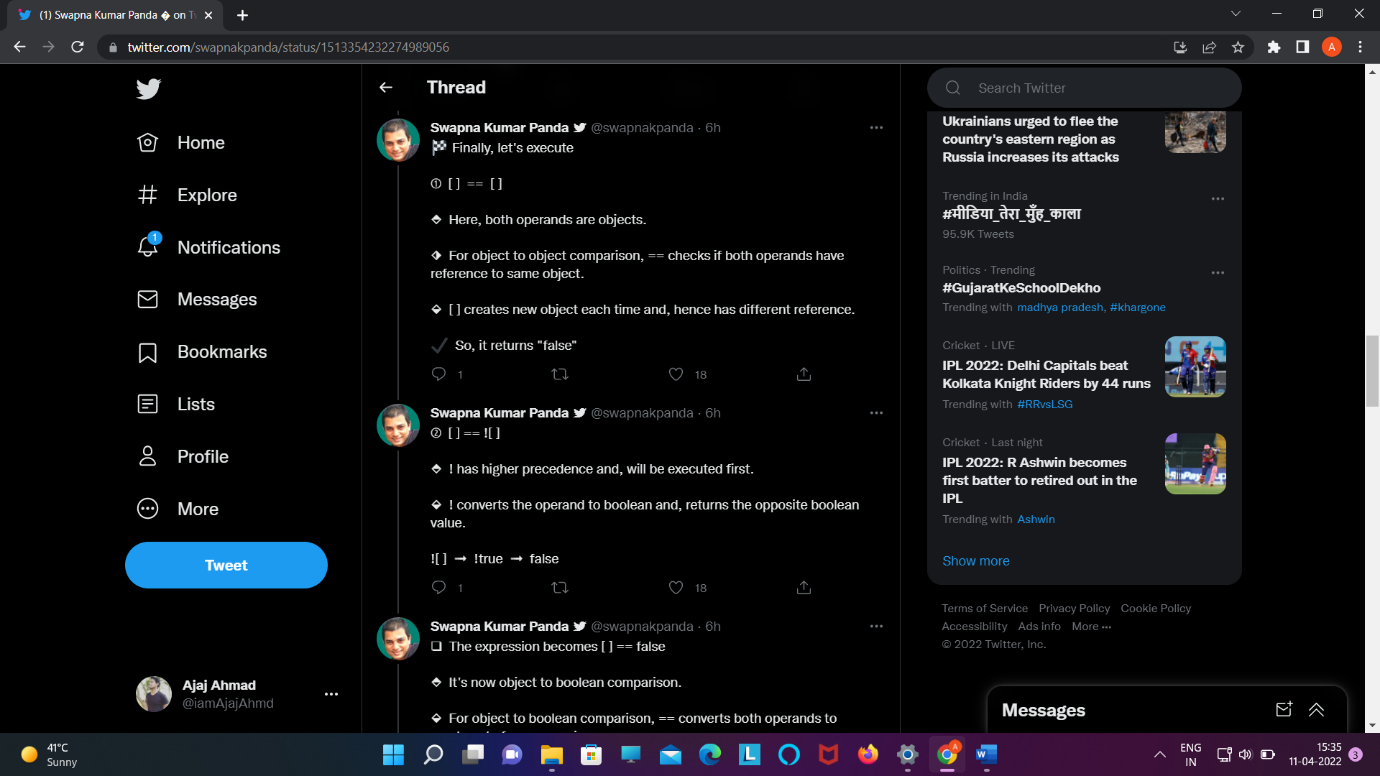
Q . In JavaScript, [ ] == [ ] ➟ false But, [ ] == ![ ] ➟ true Do you know why?













Q . JavaScript : == vs ===

