

closure)

Array → substr
substring shift
push pop

Hof
callback

Some more only Array Methods

method
↓
automatically

✓

for Each(cb) → method

✓ ✓ ✓ cb

① it accepts a callback func ✓

Array ✓

for each (function(i) {

clg()

iterator

Sachin ✓

Manish ✓

Sufga ✓

akshay ✓

sab pane

Let arr = [10, 20, 30, 40, 50]

```
for (let item of arr) {  
    log(item);  
}
```

poora
way

Kaam
tark

A handwritten diagram illustrating the relationship between an array and a function. The word 'arr' is written on the left. A blue line extends from 'arr' to the word 'forEach', which is circled in green. Above 'forEach' is the text 'Kisi ne likha' with an arrow pointing to 'forEach'. Below 'forEach' is the text 'arr ko kaise X' with an arrow pointing to 'forEach'. To the right of 'forEach' is the word 'function' with a green checkmark below it. A green line connects 'forEach' to 'function'. On the far right, there is a vertical line with 'dbl' written next to it and a green checkmark below it.

→ array

① forEach → return X
 ② map() → return ✓

↪ # map() ↪ (automatic cell)

- ① it accepts a callback funcⁿ.
- * ② it returns a new array.

React

✓ # filter()

- ① it accepts a callback (cb) ✓
 - ② it returns a new array. ✓
 - * ③ it only sends the truthful value to the new Array.
- (condition) → Sach (T) ✓ add (value)
 → Jhuth (F) ✗ nahi

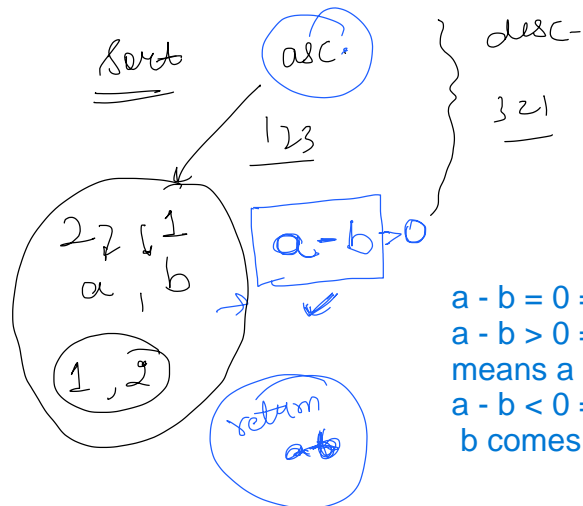
[✓|✗]

sort

sort the value lexicographically.

111, 11 , {2
Lexicographically

homework
 Comparators
 5-7 mins
 youtube



a - b = 0 => No changes
 a - b > 0 => a is greater than b means a comes after b.
 a - b < 0 => a is less than b means b comes after a.

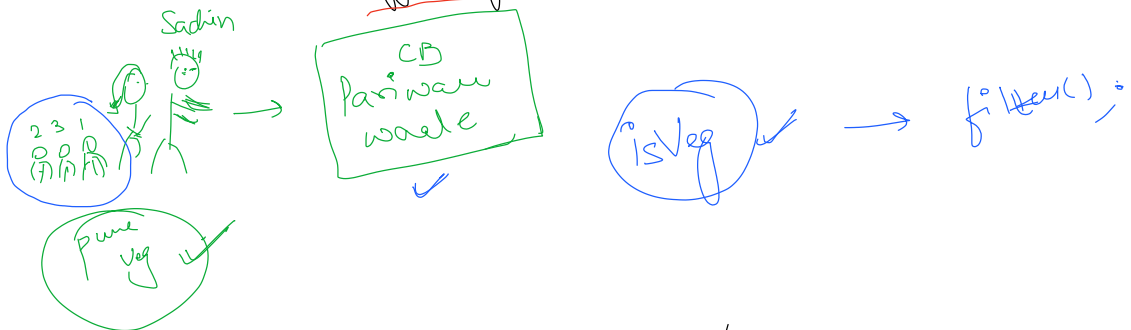
for P . . . a = b 7 7 7 7 7 7

Compare → INC → a & b { Keaton }
 → disc → b-a

_____ X _____ X _____ X -

Restaurant

Menu = ['paneer B.M.', 'pizza', 'chicken & pizza', 'd',
 'palak paneer', 'chicken B.', 'mix veg',
 'egg buriy', 'Soda', 'hoodles'] ;



_____ X _____ X _____ Closure

closures

A function bundled together with
 reference to its surrounding
 state or lexical environment
 that is called closure.

Jodd → closure

jab bhi hum koi function return karte hai toh simple function return nahi hota woh apne sath sath un
 variables ko bhi lata hai (kon se variable ?) jinka istemaal uss function ke ander kahi na kahi ho rakha
 hai (lexical environment m se variables ko lekar jata hai agar woh variables uske ander present ya
 declared hai toh taki woh unhe baad m istemaal kar sake.

A closure is a function that has access to its outer function scope even after the function has returned. Meaning, A
 closure can remember and access variables and arguments reference of its outer function even after the function has
 returned.

A closure is the combination of a function bundled together (enclosed) with references to its surrounding state (the lexical environment). In other words, a closure gives a function access to its outer scope. In JavaScript, closures are created every time a function is created, at function creation time.

```
function init() {  
  
    var name = "Mozilla"; // name is a local variable created by init  
  
    function displayName() {  
  
        // displayName() is the inner function, that forms a closure  
        console.log(name); // use variable declared in the parent function  
  
    }  
  
    displayName();  
}  
init();
```

init() creates a local variable called name and a function called displayName(). The displayName() function is an inner function that is defined inside init() and is available only within the body of the init() function. Note that the displayName() function has no local variables of its own. However, since inner functions have access to the variables of outer scopes, displayName() can access the variable name declared in the parent function, init().

Closure Detailed explanation:

<https://chatgpt.com/c/30da716d-c4ea-42cf-a2d0-43f2a0653ff2>