JS Day 05 - Lokesh

15 January 2024 13:58

For each loop:

↓

Used when you want to apply a function to each & every element of an array (same like apply function in python)

→ { You can perform similar tasks with for of loop }

numbers = [1,2,3,4,5] → i want to multiply each element by 2

function mul2 (element) {

return `Value multiply by 2 is ${element * 2}`

console.log(numbers.forEach(mul2))

Mastering Map, filter, Reduce,

Map ↓

Use fun & returns a list

filter ↓

Use function that returns true or false & later keep only values in the array that specify certain true or false condition represent on the function

Reduce ↘ same like

acc = 0

{ acc = acc + num }

take 2 or more values & returns a single computational value.

Sort → Sort array or objects based on the keys or values

Map → const array = [1,2,3,4,5]

const output = array.map ((num) => {

return num*2 })

console.out (output)

Filter ↓

const array = [1,2,3,4,5]

const output = array.filter

((num) => { return

(num%2==0)

console.out (output)

Reduce ↓ const cart = [{ Name: "Laptop", price: 100000 },

{ Name: "Mobile", price: 30000 },

{ Name: "guitar", price: 3000 },]

const total_amount = cart.reduce ((sum, item) => {

return sum + item.price 30)

return sum + item.price 30)

Consol. log (total_amount)

Sort → array = [2, 1, 1, 3, 5] → for numbers (since JS treat every no as a str (code).

Console.log(array. sort ((a,b) => { return a-b }))

array = ["a", "d", "c", "e", "b"]

Console.log(array.sort()) → for string

b-a for descending.

array fun:

find → only finds 1 no value

every → Check (returns false if anything is false)

some → Check (returns true if any is true) → Take example flipKart Cart

Fill → replace anything from value inside a string

value, start, end → index

Splice → start, delete, insert

start → index

delete → no

insert → string

✓✓ Can also perform deletion

Iterable → whatever you can use for of loop.

array like → whatever you can use len, ar[0] methods. eg string, sets

objects

(Function will also return its operation)

Set → every element will not repeat itself

Unordered, unable to access

[new Keyword] → bord

Array Methods → find → returns only 1 match

every → returns false if anything is false

some → returns true if anything is true

array = [1, 2, 3, 4, 5] array obj with {target}

let a = array.find (name => {

name === 'Target N'}

Console.log (a)

Can be every & some.

fill → array. fill (0, 1, 3) → fill the array for loc 1 to 3 to 0.

splice → array. splice (0, 5, "Hello") → fill the location from 0 to 5 with "Hello"

Unordered (no order of elements)

Sets :→

Unordered (no order of elements)

iterable (for of (can be used)

immutable (Unable to make any changes)

Uniqueness (will only contain unique values)

Functions → fill the location from 0 to 5 with "Hello"

```
let Aset = new Set ( )
Aset.add (5)
```

Only way to find length of Set ↓

```
result = 0
for (elemn of Aset){
result++ }
Console.log (result)
```