JS-Essentials

feb|15|26

- **Array:**

1. [] ; Array(value) → dont used this, try to avoid.
2. array are 0 based
3. Mutating methods

push	unshift
pop	splice
shift	

4. Non-mutating methods

concat	flatmap
slice	
flat	

5. Searching:
   includes

Array(value)
vs
Array.of(value)

Easy things (map, filter, foreach)

- Let K = array of objects
- K = [{ }, { }, { }, { }]
- K.forEach(call back fn) → iterates over values
- K.map((curr, idx) => return_value) → iterates and returns array
- K.filter((t) => return_value) → iterates and returns array if callback return is truthy.
  - t: curr item

- Reduce : SYNTAX

```
const totalrevenue = order.reduce((sum, order) => {
        return_value
}, initial_value);
```

JAVA SCRIPT IS CHEMISTRY OF LANGUAGES
cuz of exceptions LOL

The sort gotcha

```
const tkt = [100, 1, 30, 20, 60]
const sorted = [...tkt].sort()          // 100, 1, 30, 20, 60
const sorted_W = [...tkt].sort((a,b) => a-b)   // 1, 20, 30, 60, 100
```

There's no way to stop or break a forEach(), ... interview question.

Objects:

- delete name.property to delete a property
- rest is basics.. study on code editor
- object inherits..

So, use .hasOwnProperty("name") to check property existence in obj defined.

.key , .entries , .entries , .fromEntries

RUN IT

Object.freeze()

- Completely no change

Object.seal()

- structural freeze
- existing properties can be edited

- Object.defineProperty (name, property {

  value : "value",
  writable : false,
  enumerable : true,
  configurable : false,

});

by default all are true..

Object.getOwnPropertyDescriptor (name, property)

Loops

→ for : most optimized
→ while :
→ do while : > easy ---
→ for... in : avoid in arrays.. , try in objects
→ for... of : सर्वोत्तम

Functions

- function name (a,b) {

}

Type 1

- const a = function name (a,b) {

}

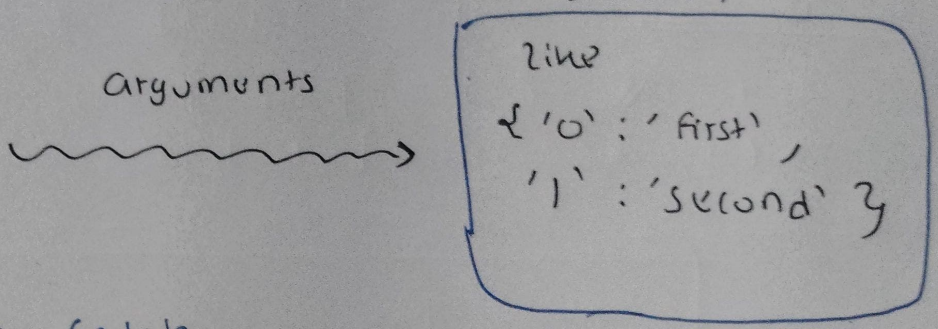
Type 2

- `const arrow = (t) => {` // arrow function

  `};`

no own "this", no 'arguments' object..

↳ its object that stores arguments passed in a function



- Try - catch

~~console.~~

```
try{
   //code block
} catch (e) {
     // error handling
}
```

Closures // To be continued

```
function (a,b) {
   let p = "Mozilla";
   function displayName() {
        console.log(p);
   }
   return displayName;
}
```

called..

→ ghar se koi phone leke aya hai
→ can call P
→ fn returned