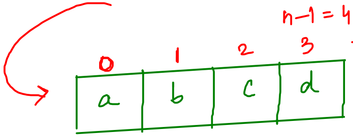
★ Arrays :

① const friends = ["a", "b", "c", "d"] ; // creation



$n-1 = 4-1 \rightarrow$ (indices of an array)

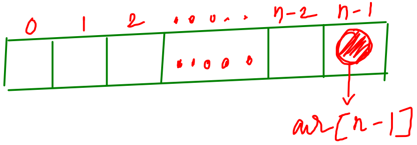
$n=4$

② friends [0] → "a"
friends [2] → "c"
} accessing elements (through indexing)

③ friends.length → 4

④ If there are n elements, what is the last element ?

const arr =

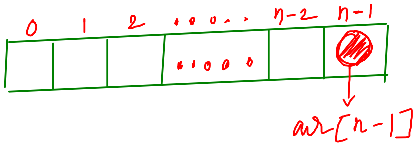


n → total elements

n = arr.length

5. If there are n elements, how to print all elements?

const arr =



n → total elements

n = arr.length

console.log(arr[0])
console.log(arr[1])
console.log(arr[2])
⋮
console.log(arr[n-2])
console.log(arr[n-1])

use a loop $\Longrightarrow$

★ iterate on indices,

```
for (let i = 0; i < n; i++) {
    console.log(arr[i]);
}
```

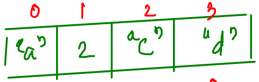
1. i = 0 → arr[0]
2. i = 1 → arr[1]
3. i = 2 → arr[2]

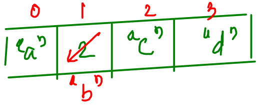
⋮

n. i = n-1 → arr[n-1]

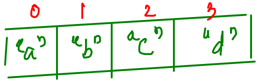
n+1. i = n , i < n ✗

⑥ how to change a box value/element?

arr =


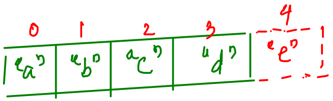
arr[1] = "b"


⑦

arr =


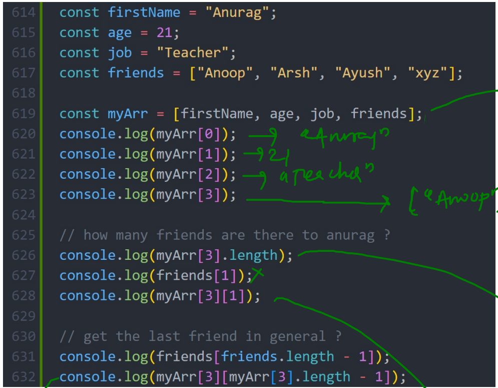
add a new element "e" at the end

arr.push("e")

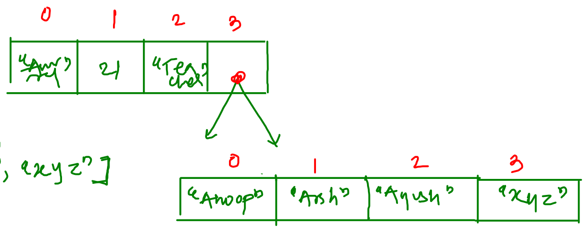
arr =


⑧ Array can store anything (literally an array itself as well)

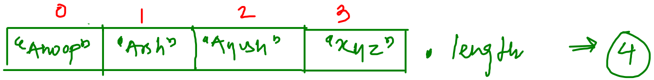
$\boxed{\text{9:25 - 9:40 pm BREAK}}$



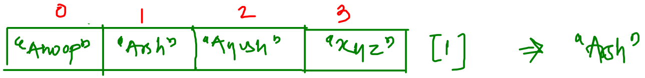
```
const firstName = "Anurag";
const age = 21;
const job = "Teacher";
const friends = ["Anoop", "Arsh", "Ayush", "xyz"];

const myArr = [firstName, age, job, friends];
console.log(myArr[0]);   → "Anurag"
console.log(myArr[1]);   → 21
console.log(myArr[2]);   → "Teacher"
console.log(myArr[3]);   → ["Anoop", "Arsh", "Ayush", "xyz"]

// how many friends are there to anurag ?
console.log(myArr[3].length);
console.log(friends[1]); ✗
console.log(myArr[3][1]);

// get the last friend in general ?
console.log(friends[friends.length - 1]);
console.log(myArr[3][myArr[3].length - 1]);
```



myArr[3].length

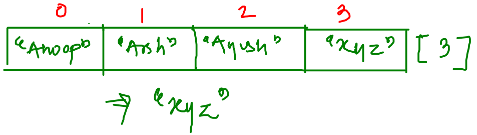


myArr[3] [1]



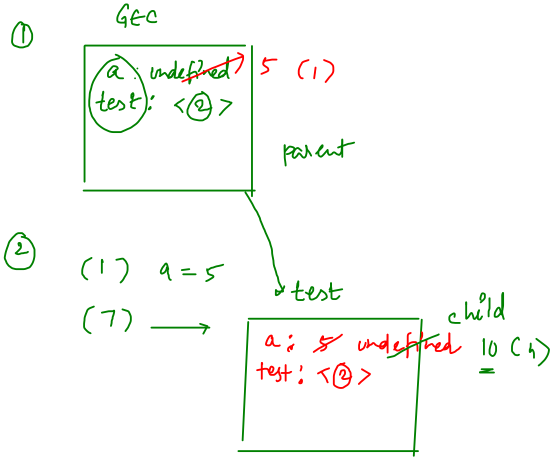
4 - 1 = 3

myArr[3] [myArr[3] . length - 1]



Doubt in Break

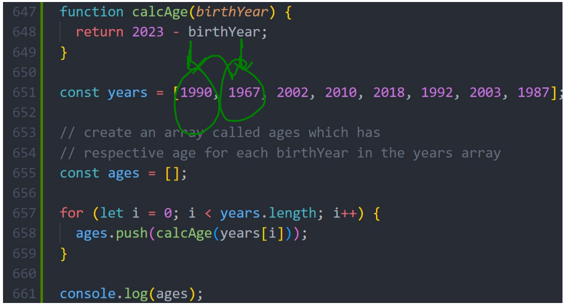
```
① var a = 5;
② function test() {
③     console.log(a)
④     var a = 10;
⑤     console.log(a);
⑥ }
⑦ test();
```



(3) → undefined

(4) → a = 10

(5) → 10



```
647  function calcAge(birthYear) {
648    return 2023 - birthYear;
649  }
650
651  const years = [1990, 1967, 2002, 2010, 2018, 1992, 2003, 1987];
652
653  // create an array called ages which has
654  // respective age for each birthYear in the years array
655  const ages = [];
656
657  for (let i = 0; i < years.length; i++) {
658    ages.push(calcAge(years[i]));
659  }
660
661  console.log(ages);
```

① i = 0

33

ages.push(calcAge(years[0]))

1990

(647)

(647) return 2023 − 1990

ages.push(33)

56

② ages. push((calcAge(years[1])))

1967

(647)

(648) return 2023 − 1967

ages: []

: [33]

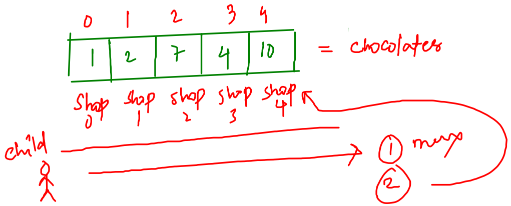
: [33, 56]

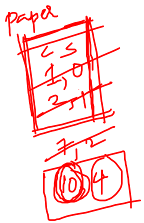
* Max Ele and its Index :

eg :

$$\begin{array}{ccccc} 0 & 1 & 2 & 3 & 4 \\ \end{array}$$
[1, 2, 7, 4, 10]

op : 10 4





① he will go to shop 0 → ①

② shop-1 → ②

③ shop-2 → ⑦

④ shop-3 → ④

⑤ shop-4 → ⑩

```
n = arr.length;
maxEle = 1;
maxEleIdx = 0;          5<4 ✗
for(let i = 1; i<n; i++) {
    if (arr[i] > maxEle) {
        maxEle = arr[i];
        maxEleIdx = i;
    }
}
```

maxEle = ~~1~~ ~~2~~ ~~7~~ 10

maxEleIdx = ~~0~~ ~~1~~ ~~2~~ 4

① i = 1,
arr[1] > maxEle
2 > 1 ✓

② i = 2
arr[2] > maxEle
7 > 2 ✓

③ i = 3
arr[3] > maxEle
4 > 7 ✗

④ i = 4
arr[4] > maxEle
10 > 7 ✓