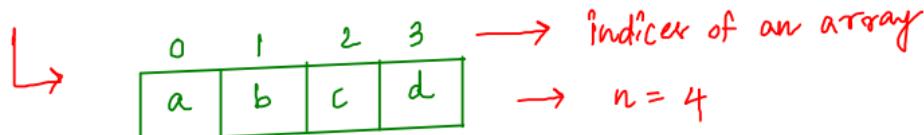


## \* Arrays:

0      1      2      3

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



diagrammatic representation

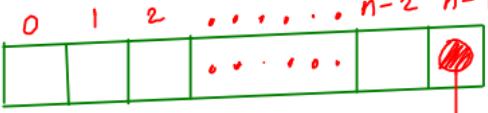
② Accessing array elements through index,

friends[0] → "a"

friends[2] → "c"

③ friends.length → 4

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

const arr = 

⇒ friends[n-1]

friends[4-1]

friends[3] ⇒ "d"

arr[n-1]

⑤ If there are  $n$  elements, How to print each element on a new line?

Const arr =

0	1	2	.....	$n-2$	$n-1$

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

\* iterate on indices

for(let i=0; i<n; i++) {

    console.log(arr[i]);

}

①  $i=0 \rightarrow arr[0]$

②  $i=1 \rightarrow arr[1]$

③  $i=2 \rightarrow arr[2]$

:

:

(n-1)  $i=n-2 \rightarrow arr[n-2]$

(n)  $i=n-1 \rightarrow arr[n-1]$

(n+1)  $i=n$  ( $i < n$ ) ✗

⑥ Can I change a box value / element inside an array ?

const arr =

0	1	2	3
"a"	"b"	"c"	"d"

arr[1] = "e"

0	1	2	3
"a"	"b"	"c"	"d"

↓  
"e"

⑦

const arr =

0	1	2	3
"a"	"b"	"c"	"d"

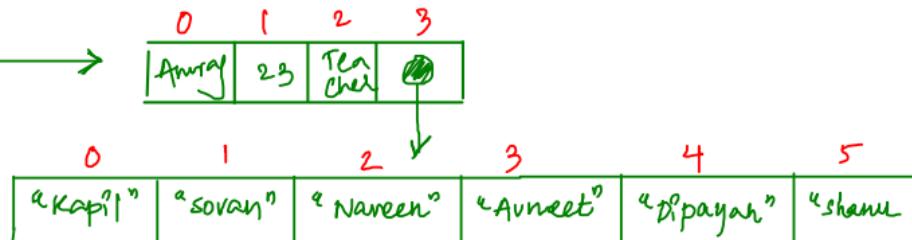
add a new element "e" at the end ,

arr.push("e")

0	1	2	3	4
"a"	"b"	"c"	"d"	"e"

## ⑥ Arrays can store any datatype (literally an entire array "itself")

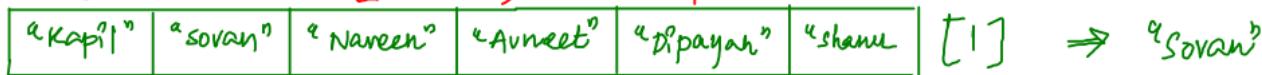
```
766 const firstName = "Anurag";
767 const age = 23;
768 const job = "Teacher";
769 const friends = ["Kapil", "Sovan", "Naveen", "Avneet", "Dipayan", "Shanu"];
770
771 const myArr = [firstName, age, job, friends];
772 console.log(myArr);
773 console.log(myArr[0]);
774 console.log(myArr[1]);
775 console.log(myArr[2]);
776 console.log(myArr[3]);
777
778 // given myArr, print the number of friends he has
779 console.log(myArr[3].length);
780
781 // given myArr, print the 2nd friend
782 console.log(myArr[3][1]);
783
784 // given myArr, print the last friend
785 const numFriends = myArr[3].length;
786 console.log(myArr[3][numFriends - 1]);
```



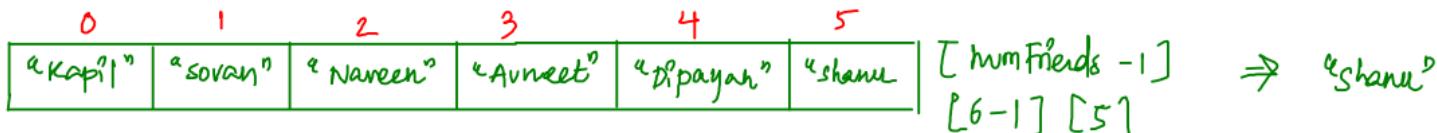
myArr[3] = length  
↓



⇒ ⑥



⇒ "sovan"



⇒ "shanu"

## # Example :

```
788 // Practice Question  
789 function calcAge(birthYear) {  
790 | return 2024 - birthYear;  
791 }  
792  
793 const years = [1990, 1997, 1968, 2002, 2010, 1992, 2003, 1998];  
794  
795 // Create an array called ages which has  
796 // repetitive age for each birthYear in the years array  
797 const ages = [];  
798  
799 // write your code  
800 for (let i = 0; i < years.length; i++) {  
801 | ages.push(calcAge(years[i]));  
802 }  
803  
804 console.log(ages);
```

ages :   
[ ] ;  
; [34]  
: [34, 21]

①  $i = 0$

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

↓  
(789) birthYear = 1990

(790) return 2024 - 1990

ages.push(34);

②  $i = 1$

ages.push(calcAge(years[1]));

↓  
(789) birthYear = 1997

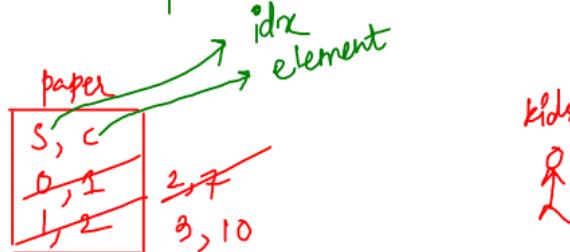
(790) return 2024 - 1997

ages.push(21);

\* Max Ele and its Index :

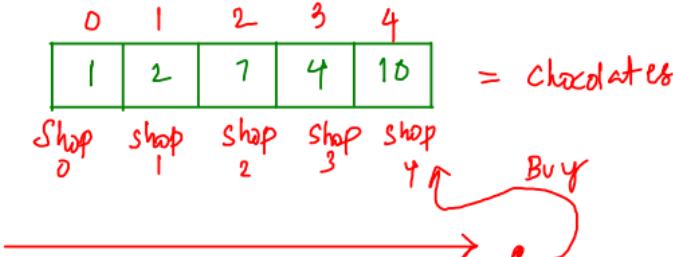
Eg: [1, 2, 7, 4, 10]  
       $\begin{matrix} 0 & 1 & 2 & 3 & 4 \end{matrix}$

op: 10 4



- ① he/she will go to shop - 0  $\rightarrow$  ①
- ② shop - 1  $\rightarrow$  ② Is this shop better ?
- ③ shop - 2  $\rightarrow$  ③ Is this shop better ?
- ④ shop - 3  $\rightarrow$  ④ Is this shop better ?
- ⑤ shop - 4  $\rightarrow$  ⑩ Is this shop better ?

[ 1st class  
what is an algorithm?  
Already discussed ]



as low as possible

```

maxEle = -∞ ; } initial
maxIdx = -1; } state
for (let i = 0; i < n; i++) {
    if (arr[i] > maxEle) {
        maxEle = arr[i];
        maxIdx = i;
    }
}

```

```

568 function ArrayProblem1(n, arr) {
569     // Write code here
570     let maxEle = -Infinity;
571     let maxIdx = -1;
572     for (let i = 0; i < n; i++) {
573         if (arr[i] > maxEle) {
574             maxEle = arr[i];
575             maxIdx = i;
576         }
577     }
578     return maxIdx;
579 }
580

```

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

~~maxEle =  $\infty$~~   
~~maxIdx =  $\infty$~~

①  $i = 0$

$arr[0] > -\infty$

$\Rightarrow 1 > -\infty$  (True)

③  $i = 2$

$arr[2] > maxEle$

$\Rightarrow 7 > 1$  (True)

②  $i = 1$

$arr[1] > 1$

$\Rightarrow 2 > 1$  (True)

④  $i = 3$

$arr[3] > maxEle$

$\Rightarrow 4 > 1$  (No)

⑤  $i = 4$

$arr[4] > maxEle$

$\Rightarrow 10 > 1$  (True)

⑥  $i = 5$

$5 < n \Rightarrow 5 < 5(\times)$

\* Max difference b/w any 2 elements :

fg: [16, 24, 89, 35]

$$\begin{array}{lll}
 8 \leftarrow (16, 24) & (24, 89) \xrightarrow{65} (89, 35) \\
 19 \leftarrow (16, 89) & (24, 35) \rightsquigarrow 11 \downarrow 54 \\
 19 \leftarrow (16, 35)
 \end{array}$$

- ( $\overset{\circ}{1}, \overset{\bullet}{1}$ )
- (0, 1)    (1, 2)    (2, 3)
- (0, 2)    (1, 3)
- (0, 3)

→ How to generate pair of any 2 elements?

```
for(let i = 0; i < n; i++) {
```

```
for (let j = i+1; j < n; j++) {
```

```
console.log([arr[i], arr[j]]);
```

3

①  $j = 0$

$\rightarrow j = i + 1 = 0 + 1 = 1 \rightarrow$  and [0], and [1]

$\rightarrow j=2 \rightarrow \text{and}[0], \text{and}[2]$

$\rightarrow j = 9 \rightarrow$  and [o], and [s]

$\rightarrow j = 4 \quad (4 \times 4) \times$

(2)  $i = 1$

$\rightarrow j = i+1 = 1+1 = 2 \rightarrow \text{arr}[1], \text{arr}[2]$

$\rightarrow j = 3 \rightarrow \text{and } [1], \text{ and } [3]$

$\rightarrow j = 4 \quad (4 \times 4) \times$

(generate pairs + findMax)

```
617 function ArrayProblem(arr) {  
618     // Write your code here  
619     let maxDiff = -Infinity;  
620     for (let i = 0; i < n; i++) {  
621         for (let j = i + 1; j < n; j++) {  
622             const diff = Math.abs(arr[i] - arr[j]);  
623             if (diff > maxDiff) {  
624                 maxDiff = diff;  
625             }  
626         }  
627     }  
628     console.log(maxDiff);  
629 }  
630 }
```

②  $i=1$

$\rightarrow j=2$

$$\rightarrow arr[1] - arr[2] = 24 - 89 \\ = 65$$

$$\rightarrow 65 > 73 \text{ (No)}$$

$\rightarrow j=3$

$$\rightarrow arr[1] - arr[3] = 24 - 35 \\ = 11$$

$$\rightarrow 11 > 73 \text{ (No)}$$

0 1 2 3  
[16, 24, 89, 35]

$$\text{maxDiff} = \cancel{-\infty} \quad 8 \quad 73$$

①  $i=0$

$$\rightarrow j = i+1 = 0+1 = 1$$

$$\rightarrow arr[0] - arr[1] = 16 - 24 = 8$$

$$\rightarrow 8 > \text{maxDiff} \rightarrow 8 > -\infty \text{ (Yes)}$$

$$\rightarrow j=2$$

$$\rightarrow arr[0] - arr[2] = 16 - 89 = 73$$

$$\rightarrow 73 > \text{maxDiff} \rightarrow 73 > 8 \text{ (Yes)}$$

$$\rightarrow j=3$$

$$\rightarrow arr[0] - arr[3] = 16 - 35 = 19$$

$$\rightarrow 19 > \text{maxDiff} \rightarrow 19 > 73 \text{ (No)}$$

\*  $\text{Math.abs}(+ve) \rightarrow +ve$   
 $\text{Math.abs}(-ve) \rightarrow +ve$   
 $(-16) \quad (16)$