CHEAT SHEET FOR ARRAY

➢ Accessing array element :

We can acces element of array using it’s index value.

Ex:
```
let arr = [1, 2, 3, 4, 5];
let ele = arr[2];
console.log(ele);
```

O/P: 3

➢ Change an array element :

We can change value in array usnig it’s index value.

Ex:
```
let arr = [1, 2, 3, 4, 5];
arr[2] = 20;
console.log(arr[2]);
```

O/P : 20

➢ .length property :

The length property of array returns length of an array (the numbers of array elements).

Ex :
```
let arr = [1, 2, 3, 4, 5];
let arrLen = arr.length;
console.log(arrLen);
```

O/P: 5

➢ .push() :

The push method of array method used to push element in to the array. We can push more than one element at a time using push method.

Ex :
```
let arr = [1, 2, 3, 4, 5];
arr.push(6, 7, 8)
console.log(arr);
```

O/P: [1, 2, 3, 4, 5, 6, 7, 8]

➢ .pop() :

The pop method of array method used to pop last entered element in to the array. We can push more than one element at a time using push method.

Ex:
```
let arr = [1, 2, 3, 4, 5];
arr.pop()
console.log(arr);
```

O/P: [1, 2, 3, 4]

➢ .toString() :

The toString() method of array converts whole array into string (separated by comma).

Ex:
```
let arr = [“One”, “Two”, “Three”, “Four”, “Five”];
let arrToString = arr.toString();
console.log(arrToString);
```

O/P : One,Two,Third,Four,Five

➢ .join() :

The join() method also behave like toString() method. The only difference beetween join() and toString() is that join() method specified seprator.

Ex:
```
let arr = [“One”, “Two”, “Three”, “Four”, “Five”];
let arrJoin = arr.join(“:”);
console.log(arrJoin);
```

O/P: One:Two:Three:Four:Five

➢ .concat() :

The concat() method of array is used to merge array.

Ex:
```
let arr1 = [1, 2, 3, 4, 5];
let arr2 = [6, 7, 8, 9, 10];
let finalArr = arr1.concat(arr2);
console.log(finalArr);
```

O/P: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

.splice() :

The splice() method can be used to add new element as well ass to add remove existing element from the array. Defaultly it's accept three arguments first parameter defines position where new element should be added, second parameter defines how many element you wan to delete, and rest of the parameter defines element to be insert in array.

```
Ex:   let arr = [“One”, “Two”, “Three”, “Four”, “Five”];
      arr.splice(2, 0, “Six”, “Seven”);
      console.log(arr);
```

O/P: [“One”, “Two”, “Six”, “Seven”, “Three”, “Four”, “Five”]

.sort() :

The sort() method sort elements of array in ascending order. It overwrites the original array.

```
Ex:   let arr = [4, 3, 5, 2, 1];
      arr.sort();
      console.log(arr);
```

O/P: [1, 2, 3, 4, 5];

.reverse() :

The reverse() method reverse order of all elements in array, It overwrites the original array.

```
Ex:   let arr = [4, 3, 5, 2, 1];
      arr.reverse();
      console.log(arr);
```

O/P: [1, 2, 5, 3, 4];

.forEach() :

The forEach() method execute function for each elements of the array and perofm some specific tasks. The .forEach() method can not be applied to empty empty array element.

```
Ex:   let arr = [1, 2, 3, 4, 5];
      arr.forEach( ele => {
            console.log(“This is : “ + ele);
      });
```

O/P: This is 1
This is 2
This is 3
This is 4
This is 5

➢ .map() :

The map() method map all element of array, perform specific operation on that element and insert that element in to newly created array. So, we can say that map() always return new array.

```
Ex:   let arr = [1, 2, 3, 4, 5];
      let squareArr = arr.map( ele => {
            retuen ele*ele;
      });
      console.log(squareArr);
```

O/P: [1, 4, 9, 16, 25]

➢ .filter() :

The filter() method apply condition on each element of array and check if element satisfied condition then it add to te new array else it will skip that element. So, we can say that filter() method always return new array.

```
Ex:   let trainee = [“Manish”, “Krushit”, “Vinayak”, “Lav”, “Jay”];
      let filteredArr = trainee.filter( ele => {
            return ele.length > 5;
      });
      console.log(filteredArr);
```

O/P: [“Manish”, “Krushit”, “Vinayak”]

➢ .reduce() :

The reduce() method perform some operation on each element of array and return single value. Unlike map(), filter() method reduce() does not return new array.

The reduce() method accept tho parameter. First parameter is callback function and second parameter is default value. Whatever value we return in call back function it will be stored in to accumlator.

Ex:
```
let trainee = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10];
let reduceArr = trainee.reduce( (ac, curr) => {
        return acc + curr;
}, 0);
console.log(reduceArr);
```

O/P: 55