

```
// 1. Print each element of an array of strings:

const strings = ["apple", "banana", "cherry", "date", "elderberry"];
for (let i = 0; i < strings.length; i++) {
  console.log(strings[i]);
}
```

```
// 2. Find the average of an array of numbers:
const numbers = [10, 20, 30, 40, 50];
const average = numbers.reduce((num , sum)=> sum+num ,
0)/numbers.length;
console.log(average);
```

```
//3. Check if an array is empty:
```

```
if(!numbers.length === 0){
return numbers;
```

```
} else{
  return console.log("empty");
}
```

```
//4. Print the square of each element in an array
// const integers = [1, 2, 3, 4, 5];
const sqOfElement = integers.forEach(num => num*num);
console.log(sqOfElement);
```

```
// 5. Find the length of an array without using .length:
const integers = [1, 2, 3, 4, 5];
let count = 0;
for(let i in integers){
  count = + count;
}
console.log(count);
```

```
// 6. Print even-indexed elements:
const numbers1 = [10, 20, 30, 40, 50];
```

```
for( let i =0; i < numbers1.length; i++){
  if( i % 2 === 0){
    console.log(numbers1[i]);
  }
}
```

```
// 7. Check if a specific value exists in an array:
const include = numbers.includes(20);
```

```
//8. Add a new element to the end of an array without using push():
function addEle(arr, ele){
```

```
    arr[arr.length] = ele;
}
```

```
const arr = [1,2,3 ]
addEle(arr, 4)
console.log(arr); // [1, 2, 3, 4]
```

```
//9. Remove the first element without using shift():
```

```
function removeFirstElement(arr) {
    for (let i = 0; i < arr.length - 1; i++) {
        arr[i] = arr[i + 1];
    }
    arr.length -= 1;
}

const arr = [1, 2, 3];
removeFirstElement(arr);
console.log(arr); // [2, 3]
```

```
// 10. Return the first n elements of an array:
```

```
function getFirstNElements(arr, n) {
    return arr.slice(0, n);
}

// Example usage:
console.log(getFirstNElements([1, 2, 3, 4, 5], 3)); // [1, 2, 3]
```

```
// 11. Find the difference between the largest and smallest numbers:
```

```
function findDifference(arr) {
    const max = Math.max(...arr);
    const min = Math.min(...arr);
    return max - min;
}

// Example usage:
console.log(findDifference([10, 20, 30, 5, 15])); // 25
```

```
// 12. Count occurrences of a number in an array:
```

```
function countOccurrences(arr, num) {
    return arr.filter(element => element === num).length;
}

// Example usage:
console.log(countOccurrences([1, 2, 3, 1, 2, 1], 1)); // 3

// 13. Remove duplicates from an array:
function removeDuplicates(arr) {
```

```
    return [...new Set(arr)];
  }

  // Example usage:
  console.log(removeDuplicates([1, 2, 2, 3, 3, 3])); // [1, 2, 3]
  // 14. Split an array into chunks:
```

```
function chunkArray(arr, size) {
  const chunks = [];
  for (let i = 0; i < arr.length; i += size) {
    chunks.push(arr.slice(i, i + size));
  }
  return chunks;
}

// Example usage:
console.log(chunkArray([1, 2, 3, 4, 5], 2)); // [[1, 2], [3, 4], [5]]
```

```
//15. Return all elements greater than a given number:
function filterGreaterThan(arr, num) {
  return arr.filter(element => element > num);
}

// Example usage:
console.log(filterGreaterThan([10, 20, 30, 40], 25)); // [30, 40]

// 16. Sum elements at odd indices:
function sumOddIndices(arr) {
  return arr.reduce((sum, num, index) => (index % 2 !== 0 ? sum +
num : sum), 0);
}

// Example usage:
console.log(sumOddIndices([1, 2, 3, 4, 5])); // 6
```

```
//17. Insert an element at a specific index:
function insertAt(arr, index, value) {
  arr.splice(index, 0, value);
}

// Example usage:
const arr = [1, 2, 4];
insertAt(arr, 2, 3);
console.log(arr); // [1, 2, 3, 4]

// 18. Find the product of all elements:
function productOfElements(arr) {
  return arr.reduce((product, num) => product * num, 1);
```

```
}

// Example usage:
console.log(productOfElements([1, 2, 3, 4])); // 24
```

```
//19. Return the cumulative sum:
function cumulativeSum(arr) {
  const result = [];
  arr.reduce((sum, num, index) => (result[index] = sum + num), 0);
  return result;
}

// Example usage:
console.log(cumulativeSum([1, 2, 3])); // [1, 3, 6]
```

```
/// 20. Check if an array is sorted in ascending order:
function isSortedAscending(arr) {
  for (let i = 0; i < arr.length - 1; i++) {
    if (arr[i] > arr[i + 1]) return false;
  }
  return true;
}

// Example usage:
console.log(isSortedAscending([1, 2, 3])); // true
console.log(isSortedAscending([3, 2, 1])); // false
```