

yo future kaven, your shart ass solution are the red the carbons are class. remember that

Arrays

There exists a special data structure named `Array`, to store ordered collections.

Declaration

There are two syntaxes for creating an empty array:

```
let arr = new Array();  
let arr = [];
```

Almost all the time, the second syntax is used. We can supply initial elements in the brackets:

```
let fruits = ["Apple", "Orange", "Plum"];
```

Array elements are numbered, starting with zero.

We can get an element by its number in square brackets:

```
let fruits = ["Apple", "Orange", "Plum"];
```

```
alert( fruits[0] ); // Apple  
alert( fruits[1] ); // Orange  
alert( fruits[2] ); // Plum
```

We can replace an element:

```
fruits[2] = 'Pear'; // now ["Apple", "Orange", "Pear"]
```

...Or add a new one to the array:

```
fruits[3] = 'Lemon'; // now ["Apple", "Orange", "Pear", "Lemon"]
```

The total count of the elements in the array is its `length`:

```
let fruits = ["Apple", "Orange", "Plum"];
```

```
alert( fruits.length ); // 3
```

We can also use `alert` to show the whole array.

```
let fruits = ["Apple", "Orange", "Plum"];
```

```
alert( fruits ); // Apple,Orange,Plum
```

An array can store elements of any type.

Methods pop/push, shift/unshift

A [queue](#) is one of most common uses of an array. In computer science, this means an ordered collection of elements which supports two operations:

- **push** appends an element to the end.
- **shift** get an element from the beginning, advancing the queue, so that the 2nd element becomes the 1st.



Arrays support both operations.

In practice we meet it very often. For example, a queue of messages that need to be shown on-screen.

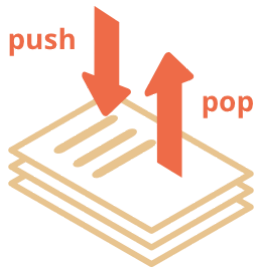
There's another use case for arrays – the data structure named [stack](#).

It supports two operations:

- **push** adds an element to the end.
- **pop** takes an element at the end.

So new elements are added or taken always from the “end”.

A stack is usually illustrated as a pack of cards: new cards are added to the top or taken from the top:



For stacks, the latest pushed item is received first, that's also called LIFO (Last-In-First-Out) principle. For queues, we have FIFO (First-In-First-Out).

Arrays in JavaScript can work both as a queue and as a stack. They allow to add/remove elements both to/from the beginning or the end.

Methods that work with the end of the array:

pop

Extracts the last element of the array and returns it:

```
let fruits = ["Apple", "Orange", "Pear"];

alert( fruits.pop() ); // remove "Pear" and alert it

alert( fruits ); // Apple, Orange
```

push

Append the element to the end of the array:

```
let fruits = ["Apple", "Orange"];

fruits.push("Pear");

alert( fruits ); // Apple, Orange, Pear
```

The call `fruits.push(...)` is equal to `fruits[fruits.length] = ...`

Methods that work with the beginning of the array:

shift

Extracts the first element of the array and returns it:

```
let fruits = ["Apple", "Orange", "Pear"];

alert( fruits.shift() ); // remove Apple and alert it

alert( fruits ); // Orange, Pear
```

unshift

Add the element to the beginning of the array:

```
let fruits = ["Orange", "Pear"];

fruits.unshift('Apple');

alert( fruits ); // Apple, Orange, Pear
```

Methods push and unshift can add multiple elements at once:

```
let fruits = ["Apple"];

fruits.push("Orange", "Peach");
fruits.unshift("Pineapple", "Lemon");

// ["Pineapple", "Lemon", "Apple", "Orange", "Peach"]
alert( fruits );
```

Loops

One of the oldest ways to cycle array items is the for loop over indexes:

```
let arr = ["Apple", "Orange", "Pear"];

for (let i = 0; i < arr.length; i++) {
    alert( arr[i] );
}
```

A word about “length”

The length property automatically updates when we modify the array. To be precise, it is actually not the count of values in the array, but the greatest numeric index plus one.

For instance, a single element with a large index gives a big length:

```
let fruits = [];
fruits[123] = "Apple";

alert( fruits.length ); // 124
```

Note that we usually don't use arrays like that.

Another interesting thing about the `length` property is that it's writable.

If we increase it manually, nothing interesting happens. But if we decrease it, the array is truncated. The process is irreversible, here's the example:

```
let arr = [1, 2, 3, 4, 5];

arr.length = 2; // truncate to 2 elements
alert( arr ); // [1, 2]

arr.length = 5; // return length back
alert( arr[3] ); // undefined: the values do not return
```

So, the simplest way to clear the array is: `arr.length=0`.

toString

Arrays have their own implementation of `toString` method that returns a comma-separated list of elements.

For instance:

```
let arr = [1, 2, 3];

alert( arr ); // 1,2,3
alert( String(arr) === '1,2,3' ); // true
```

Summary

Array is a special kind of objects, suited to store and manage ordered data items.

- The declaration:

```
// square brackets (usual)
let arr = [item1, item2...];
```
- The `length` property is the array length or, to be precise, its last numeric index plus one. It is auto-adjusted by array methods.
- If we shorten `length` manually, the array is truncated.
We can use an array as a deque with the following operations:
- `push(...items)` adds `items` to the end.
- `pop()` removes the element from the end and returns it.
- `shift()` removes the element from the beginning and returns it.
- `unshift(...items)` adds `items` to the beginning.
- To loop over the elements of the array:
`for(let i=0; i<arr.length; i++)` – works fastest, old-browser-compatible.

Questions and Exercise

1. What is this code going to show?

```
let fruits = ["Apples", "Pear", "Orange"];

// push a new value into the "copy"
```

```
let shoppingCart = fruits;
shoppingCart.push("Banana");

// what's in fruits?
alert( fruits.length ); // ?
```

2. Create an array `styles` with items "Jazz" and "Blues".

Append "Rock-n-Roll" to the end.

Replace the value in the middle by "Classics". Your code for finding the middle value should work for any arrays with odd length.

Strip off the first value of the array and show it.

Prepend Rap and Reggie to the array.

The array in the process:

Jazz, Blues

Jazz, Bues, Rock-n-Roll

Jazz, Classics, Rock-n-Roll

Classics, Rock-n-Roll

Rap, Reggie, Classics, Rock-n-Roll

```
let music=["Jazz","Blues"]

    console.log(music);
    music.push("Rock-n-Roll");
    console.log(music);
    music.splice(1,1);
    music.splice(1,0,"Classics");
    console.log(music);
    music.shift();
    console.log(music);
    music.unshift("Reggie");
    music.unshift("Rap");
    console.log(music);
```

3. Create a program that fills a 10 element array with 10 random numbers between 1 and 100. Display these numbers and then display the largest and smallest.

```
let numbers=[];
let number=0;
for(let x=0;x<=10;x++)
{
    number=Math.round(Math.random()*100);

    numbers.push(number);
}
console.log("List of random
numbers");
console.log(numbers);
let list=numbers.sort(function(a, b)
{return a-b});
console.log(list);
```

```
let numbers=[];
let number=0;
for(let x=0;x<=10;x++)
{
    number=Math.round(Math.random()*100);
    numbers.push(number);
}
console.log("List of random numbers");
console.log(numbers);
let list=numbers.sort(function(a, b){return a-b});
console.log(list);
```

4. Create a program that fills a 100 element array with the squares of the numbers from 1 to 100. Display the index of each element alongside it's value.

```
let numbers1=[];
let number1=0;
for(let x=0;x<=100;x++)
{
    number1=Math.round(Math.random()*100);
    number1=number1**2;
    numbers1.push(number1);
}
```

```
console.log("List of random numbers");  
console.log(numbers1);
```



```
let squaresArr = [];  
for(let i = 1; i <= 100; i++){  
    squaresArr.push("The index: " + i + "\tValue: " + i**2);  
}  
for(let x = 0; x < 100; x++){  
    console.log(squaresArr[x]);  
}
```

5. Randomly fill a 1000 element array with random numbers between 1 and 100. Use the array to calculate the average of all those numbers.

```
let numbers2=[];  
let number2=0;  
let averageT=0;  
for(let x=0;x<1000;x++){  
    {  
        number2=Math.round(Math.random()*100);  
        averageT+=number2;  
        numbers2.push(number2);  
    }  
}  
averageT=averageT/1000;  
console.log("List of random numbers");  
console.log(numbers2);  
console.log("The total average is "+averageT);
```

```

let numbers2=[];
  let number2=0;
  let averageT=0;
  for(let x=0;x<1000;x++)
  {

    number2=Math.round(Math.random()*100);
    averageT+=number2;
    numbers2.push(number2);
  }
  averageT=averageT/1000;
  console.log("List of random
numbers");
  console.log(numbers2);
  console.log("The total average is
"+averageT);

```

6. Fill an array with 100 random numbers between 1 and 100. Ask a user for any number between 1 and 100. Find out if and where the number first appears in the array.

```

let numbers3=[];
let number3=0;
let userInputedNumber=+prompt("Please enter a number to find where it first appears")
for(let x=0;x<=100;x++)
{
  number3=Math.round(Math.random()*100);
  numbers3.push(number3);
}
let position=numbers3.indexOf(userInputedNumber);
if (position<0)
{
  console.log("your number is not included in the array")
}
else
{
  console.log("Your number first appears in position "+position)
}

```



```

let array = [];
let randNum = 0;
for(let i = 0; i < 100; i++){
    randNum = Math.round(Math.random()*100);
    array.push(randNum);
}

console.log(array);

let userInput = +prompt('Pick any number between 1 and 100');

console.log(userPick(userInput));

function userPick(input){
    for(let x = 0; x < array.length; x++){
        if(array[x] == input){
            return "Your number is first found at index: " +
array.indexOf(input);
        }
    }
    return 'Your number is not in the array';
}

```

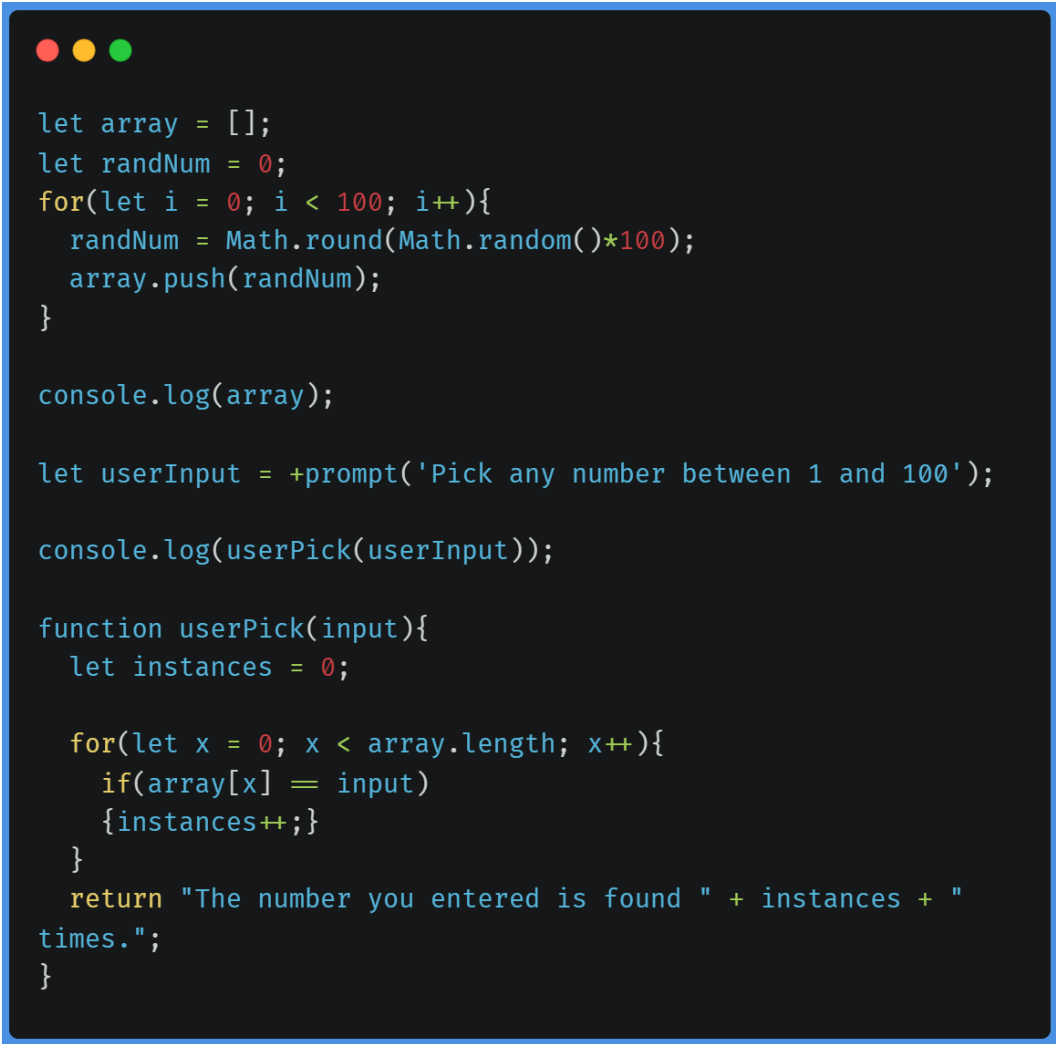
7. Do the same as #6 but this time display how many instances of the number appears in the array.

```

let numbers4=[];
let number4=0;
let instances=0;
let userInputedNumber1=+prompt("Please enter a number to find where it first appears")
for(let x=0;x<=100;x++)
{
    number4=Math.round(Math.random()*100);
    numbers4.push(number4);
}
let position1=numbers3.indexOf(userInputedNumber1);
if (position<0)
{
    console.log("your number is not included in the array")
}
else
{
    for (let y=0;y<=numbers4.length;y++)
    {
        if (numbers4[y]==userInputedNumber1)
        {
            instances++
        }
    }
}

```

```
    console.log("The amount of instances is "+instances);  
  }  
  console.log(numbers4);
```



```
let array = [];  
let randNum = 0;  
for(let i = 0; i < 100; i++){  
  randNum = Math.round(Math.random()*100);  
  array.push(randNum);  
}  
  
console.log(array);  
  
let userInput = +prompt('Pick any number between 1 and 100');  
console.log(userPick(userInput));  
  
function userPick(input){  
  let instances = 0;  
  
  for(let x = 0; x < array.length; x++){  
    if(array[x] === input)  
      {instances++;}  
  }  
  return "The number you entered is found " + instances + "  
times.";  
}
```

8. Create an array of 10 different animals. Determine how many of them have the vowels 'a' and 'e' in them.

```

let randAnims = ['antelope', 'aardvark', 'lion', 'tiger',
  'giraffe', 'hyena', 'rhino', 'elephant', 'cobra', 'monkey'];
let count = 0;

for(let i = 0; i < randAnims.length; i++){
  if(randAnims[i].includes('a') || randAnims[i].includes('e'))
    {count++;}
}
console.log(count);

```

9. Using the same array from the last example, print the list of animals. Switch the first and last animals in the array and then print the list again.

```

let randAnims = ['antelope', 'aardvark', 'lion', 'tiger',
  'giraffe', 'hyena', 'rhino', 'elephant', 'cobra', 'monkey'];

switchFirstLast();

function switchFirstLast(){
  let firstWord = randAnims.shift(randAnims[0]);
  let lastWord = randAnims.pop(randAnims[length - 1]);
  randAnims.unshift(lastWord);
  randAnims.push(firstWord);
  console.log(randAnims);
}

```

10. Fill an array with 100 random numbers between 1 and 100. Write a program to read **n** number of values in an array and display it in reverse order.



```


let arr = [];
let num = 0;

for(let i = 0; i < 100; i++){
    num = Math.round(Math.random()*100);
    arr.push(num);
}

let userAmount = +prompt("How many values do you need?");
let newArr = arr.slice(0, userAmount);
console.log(newArr.reverse());

```

11. Fill an array with 100 random numbers between 1 and 100. Write a program to count a total number of duplicate elements in an array.



```

let array = [];
let num = 0;
let count = 0;

for(let i = 0; i < 100; i++){
    num = 1+Math.round(Math.random()*99);
    array.push(num);
}

array.sort(function (a, b){return a - b;});

for(let i = 0; i < array.length; i++){
    if(array[i] == array[i + 1]){
        count++;
    }
}

console.log('There are ' + count + ' duplicates!');

```

12. Fill an array with 100 random numbers between 1 and 100. Write a program to print all unique elements in an array.

```

let array = [];
let num = 0;
let count = 0;

for(let i = 0; i < 100; i++){
    num = 1+Math.round(Math.random()*99);
    array.push(num);
}

array.sort(function (a, b){return a - b;});

for(let i = 0; i < array.length; i++){
    if(array[i] != array[i + 1]){
        count++;
    }
}

console.log('There are ' + count + ' unique numbers!');

```

//Note: Shouldn't be a space between the != on the 5th last line. creates some weird symbol when they are right beside each other in carbon.

13. Fill an array with 100 random numbers between 1 and 100. Write a program to count the frequency of each element of an array.

```

let numbsGenerated=[];
let theGenerator=0;

for (let x=0;x<100;x++)
{
    theGenerator=Math.round(Math.random()*100);
    numbsGenerated.push(theGenerator);
}

numbsGenerated.sort(function(e, f){return e-f});
let frequency=0;
let freqDisplay="";

for (let x=0;x<numbsGenerated.length;x++)
{
    if (numbsGenerated[x]==numbsGenerated[x+1])
    {
        frequency++;
    }
    else if (numbsGenerated[x]!=numbsGenerated[x+1])
    {
        if (numbsGenerated[x]==numbsGenerated[x-1])
        {
            frequency++;
            freqDisplay+=numbsGenerated[x]+" --> "+frequency+" \n";
            frequency=0;
        }
        else
        {
            frequency=1;
            freqDisplay+=numbsGenerated[x]+" --> "+frequency+" \n";
            frequency=0;
        }
    }
}

console.log(numbsGenerated);
console.log(freqDisplay);

```

14. Fill an array with 100 random numbers between 1 and 100. Write a program to display odd and even integers in an array.



```
let array = [];  
let num = 0;  
let count = 0;  
  
for(let i = 0; i < 100; i++){  
    num = 1+Math.round(Math.random()*99);  
    array.push(num);  
}  
  
let even = [];  
let odd = [];  
  
for(let i = 0; i < array.length; i++){  
    if(array[i]%2 == 0){  
        even.push(array[i]);  
    } else {  
        odd.push(array[i]);  
    }  
}  
  
console.log(even.sort(function (a,b){return a-b;}));  
console.log(odd.sort(function (a,b){return a-b;}));
```