JavaScript Array Methods

chat.openai.com/chat

how many method of arrays in javascript?

JavaScript provides many methods to manipulate arrays, some of the most commonly used methods are:

1. push() - Adds one or more elements to the end of an array and returns the new length of the array.
2. pop() - Removes the last element from an array and returns that element.
3. shift() - Removes the first element from an array and returns that element.
4. unshift() - Adds one or more elements to the beginning of an array and returns the new length of the array.
5. splice() - Adds or removes elements from an array at a specific index.
6. slice() - Returns a shallow copy of a portion of an array into a new array object selected from begin to end (end not included).
7. concat() - Joins two or more arrays and returns a new array that contains all the elements.
8. indexOf() - Returns the first index at which a given element can be found in the array, or -1 if it is not present.
9. lastIndexOf() - Returns the last index at which a given element can be found in the array, or -1 if it is not present.
10. forEach() - Executes a provided function once for each array element.
11. map() - Creates a new array with the results of calling a provided function on every element in the calling array.
12. filter() - Creates a new array with all elements that pass the test implemented by the provided function.
13. reduce() - Executes a reducer function on each element of the array, resulting in a single output value.
14. sort() - Sorts the elements of an array in place and returns the sorted array.
15. reverse() - Reverses the order of the elements in an array in place.

These are just some of the most commonly used methods for manipulating arrays in JavaScript. There are many other methods available, and you can also create your own custom methods to manipulate arrays in JavaScript.

what is method of reverse() in javascript? explain in roman urdu `reverse()` JavaScript ka ek method hai jo ek array ke elements ki tarteeb ulta kar deta hai. Yaani ke agar kisi array mein 1, 2, 3, 4 elements hain to `reverse()` method ke istemaal se yeh 4, 3, 2, 1 ban jayenge.

Is method ko istemaal karne ke liye, array variable ke baad dot (.) laga kar `reverse()` likhna hota hai. Jaise:

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

Iss code mein `reverse()` method ne `arr` array ke elements ki tarteeb ko ulta kar diya aur uss mein 4, 3, 2, 1 ko bana diya.

what is method of sort() in javascript? explain in roman urdu

`sort()` JavaScript ka ek method hai jo ek array ke elements ko tarteeb mein tarteeb mein laga deta hai. Is method ke istemaal se array ke elements chote se bara ya bara se chota kar diye jaate hain.

Iss method ko istemaal karne ke liye, array variable ke baad dot (.) laga kar `sort()` likhna hota hai. Jaise:

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

Iss code mein `sort()` method ne `arr` array ke elements ko chote se bara mein tarteeb mein laga diya hai. Iss tarah se `arr` mein 1, 2, 3, 4 ki tarteeb ban gayi hai.

`sort()` method ko chote se bara ya bara se chota karne ke liye, hum apne function mein compare kar sakte hain. Jaise ke agar hum `arr` ko bara se chota mein sort karna chahte hain to hum iss tarah ka function likh sakte hain:

```
let arr = [3, 1, 4, 2];
arr.sort(function(a, b) {
  return a - b;
});
console.log(arr); // [1, 2, 3, 4]
```

Iss code mein humne `sort()` method mein ek function pass kiya hai, jismein humne `a - b` likha hai. Iss se yeh pata chalta hai ke humne `arr` ko bara se chota mein sort karna chahte hain.

what is method of reduce() in javascript? explain in roman urdu

`reduce()` JavaScript ka ek method hai jo ek array ke elements ko reduce (kam) karke ek single value ko produce karta hai. Iss method ko array ke har element ke liye call kiya jata hai aur har call ke baad ek naya value produce kiya jata hai.

Iss method ko istemaal karne ke liye, array variable ke baad dot (.) laga kar `reduce()` likhna hota hai. Jaise:

```javascript
let arr = [1, 2, 3, 4];
let sum = arr.reduce(function(total, current) {
  return total + current;
}, 0);
console.log(sum); // 10
```

Iss code mein `reduce()` method ne `arr` ke sabhi elements ko add karke `sum` ko produce kiya hai. `reduce()` method ke pehle argument mein humne `total` aur `current` ko add karke naya value produce karne ka rule likha hai. Dusra argument 0 hai jo initial value hai.

`reduce()` method ke pehle argument mein hum apni marzi ke mutabiq function likh sakte hain jis mein hum `total` aur `current` ko add, multiply, divide, concatenate ya koi bhi operation kar sakte hain.

what is method of filter() in javascript? explain in roman urdu

`filter()` JavaScript ka ek method hai jo ek array ke elements mein se kuch specific elements ko filter (alag) karke naya ek array produce karta hai. Is method ke istemaal se array ke elements ko check kiya jata hai aur jin elements ki conditions ko pura karte hain woh naya array mein shamil kiye jate hain.

Iss method ko istemaal karne ke liye, array variable ke baad dot (.) laga kar `filter()` likhna hota hai. Jaise:

```javascript
let arr = [1, 2, 3, 4, 5];
let filteredArr = arr.filter(function(num) {
  return num % 2 === 0;
});
console.log(filteredArr); // [2, 4]
```

Iss code mein `filter()` method ne `arr` ke elements mein se jinhe 2 se divide karne par remainder 0 hai, unhe naya `filteredArr` array mein shamil kiya hai. Iss tarah se `filteredArr` mein sirf 2 aur 4 shamil hain.

`filter()` method ke argument mein hum apni marzi ke mutabiq function likh sakte hain jis mein hum conditions likh sakte hain jaise ke sirf even numbers ko select karna, sirf odd numbers ko select karna, strings mein se sirf capital letters ko select karna, etc.

what is method of map() in javascript? explain in roman urdu

`map()` JavaScript ka ek method hai jo ek array ke elements ko modify karke naya ek array produce karta hai. Is method ke istemaal se array ke elements par specific operations apply kiye jate hain aur phir naye modified elements ko naya array mein shamil kiya jata hai.

Iss method ko istemaal karne ke liye, array variable ke baad dot (.) laga kar `map()` likhna hota hai. Jaise:

```javascript
let arr = [1, 2, 3, 4, 5];
let doubledArr = arr.map(function(num) {
  return num * 2;
});
console.log(doubledArr); // [2, 4, 6, 8, 10]
```

Iss code mein `map()` method ne `arr` ke har element ko 2 se multiply karke naya `doubledArr` array produce kiya hai. Iss tarah se `doubledArr` mein har element ko 2 se multiply karke modify kiya gaya hai.

`map()` method ke argument mein hum apni marzi ke mutabiq function likh sakte hain jis mein hum array ke elements ko modify karne ke liye koi bhi operation kar sakte hain jaise ke elements ko add karna, subtract karna, divide karna, concatenate karna, etc.

what is method of forEach() in javascript? explain in roman urdu

`forEach()` JavaScript ka ek method hai jo ek array ke har element par specific operations apply karta hai. Is method ke istemaal se array ke elements ko iterate (dohrana) kiya jata hai aur har element ke liye koi bhi operations perform kiye jate hain.

Iss method ko istemaal karne ke liye, array variable ke baad dot (.) laga kar `forEach()` likhna hota hai. Jaise:

```javascript
let arr = [1, 2, 3, 4, 5];
arr.forEach(function(num) {
  console.log(num * 2);
});
```

Iss code mein `forEach()` method ne `arr` ke har element ko 2 se multiply karke console par print kiya hai. Iss tarah se `forEach()` method ke istemaal se hum array ke har element par operations perform kar sakte hain.

`forEach()` method ke argument mein hum apni marzi ke mutabiq function likh sakte hain jis mein hum har element par koi bhi operation kar sakte hain jaise ke console mein print karna, alert karna, kisi variable mein store karna, etc.

what is method of lastIndexOf() in javascript? explain in roman urdu

`lastIndexOf()` JavaScript ka ek method hai jo ek array mein se specific element ka last index (aakhri index) dhoondhne mein madad karta hai. Agar specified element array mein nahi milta hai toh -1 return karta hai.

Iss method ko istemaal karne ke liye, array variable ke baad dot (.) laga kar `lastIndexOf()` likhna hota hai. Jaise:

```javascript
let arr = [1, 2, 3, 4, 5, 4];
let lastIndex = arr.lastIndexOf(4);
console.log(lastIndex); // 5
```

Iss code mein `lastIndexOf()` method ne `arr` array mein se 4 ka last index dhoondha hai aur phir usko `lastIndex` variable mein store kiya hai. Iss tarah se `lastIndex` mein 5 store hoga kyunki 4 ka last index 5 hai.

`lastIndexOf()` method ke argument mein hum apni marzi ke mutabiq element likh sakte hain jisko hum dhoondhna chahte hain. Agar yeh element array mein nahi milta hai toh method `-1` return karega.

what is method of IndexOf() in javascript? explain in roman urdu

`indexOf()` JavaScript ka ek method hai jo ek array mein se specific element ka index dhoondhne mein madad karta hai. Agar specified element array mein nahi milta hai toh -1 return karta hai.

Iss method ko istemaal karne ke liye, array variable ke baad dot (.) laga kar `indexOf()` likhna hota hai. Jaise:

```javascript
let arr = [1, 2, 3, 4, 5];
let index = arr.indexOf(4);
console.log(index); // 3
```

Iss code mein `indexOf()` method ne `arr` array mein se 4 ka index dhoondha hai aur phir usko `index` variable mein store kiya hai. Iss tarah se `index` mein 3 store hoga kyunki 4 ka index 3 hai.

`indexOf()` method ke argument mein hum apni marzi ke mutabiq element likh sakte hain jisko hum dhoondhna chahte hain. Agar yeh element array mein nahi milta hai toh method `-1` return karega.

what is method of concat() in javascript? explain in roman urdu

`concat()` JavaScript ka ek method hai jo do ya zyada arrays ko combine kar ke naye array mein store karta hai. Yeh method naye array ko return karta hai aur existing arrays ko modify nahi karta hai.

Iss method ko istemaal karne ke liye, pehle array variable ke baad dot (.) laga kar `concat()` likhna hota hai, phir uske baad ek ya zyada arrays comma separated form mein likhne hote hain. Jaise:

```javascript
let arr1 = [1, 2, 3];
let arr2 = [4, 5, 6];
let newArr = arr1.concat(arr2);
console.log(newArr); // [1, 2, 3, 4, 5, 6]
```

Iss code mein `concat()` method ne pehle `arr1` array ko `arr2` array ke saath combine karke `newArr` array mein store kiya hai. Iss tarah se `newArr` mein `[1, 2, 3, 4, 5, 6]` store hoga.

`concat()` method ke argument mein hum apni marzi ke mutabiq ek ya zyada arrays comma separated form mein likh sakte hain jinko hum combine karna chahte hain. Yeh method in arrays ko combine karke ek naye array ko return karta hai aur existing arrays ko modify nahi karta hai.

what is method of slice() in javascript? explain in roman urdu

`slice()` JavaScript ka ek method hai jo ek array ke specific portion (slice) ko naya array mein store karta hai. Yeh method naye array ko return karta hai aur existing array ko modify nahi karta hai.

Iss method ko istemaal karne ke liye, array variable ke baad dot (.) laga kar `slice()` likhna hota hai. `slice()` method ke argument mein hum specify kar sakte hain ke humein kis index se lekar kis index tak ke elements ko naye array mein store karna hai. Jaise:

```javascript
let arr = [1, 2, 3, 4, 5];
let slicedArr = arr.slice(2, 4);
console.log(slicedArr); // [3, 4]
```

Iss code mein `slice()` method ne `arr` array ke index 2 se lekar index 4 tak ke elements ko naye array `slicedArr` mein store kiya hai. Iss tarah se `slicedArr` mein `[3, 4]` store hoga.

Agar `slice()` method ke sirf ek argument diya jaaye, jaise `arr.slice(2)` toh yeh uss index se lekar array ke end tak ke elements ko naye array mein store karega.

```javascript
let arr = [1, 2, 3, 4, 5];
let slicedArr = arr.slice(2);
console.log(slicedArr); // [3, 4, 5]
```

Iss code mein `slice()` method ne `arr` array ke index 2 se lekar array ke end tak ke elements ko naye array `slicedArr` mein store kiya hai. Iss tarah se `slicedArr` mein `[3, 4, 5]` store hoga.

what is method of splice() in javascript? explain in roman urdu

`splice()` JavaScript ka ek method hai jo ek array se kisi portion ko remove karke usmein naye elements insert karta hai. Yeh method existing array ko modify karta hai aur modify kiya hua array ko return karta hai.

Iss method ko istemaal karne ke liye, array variable ke baad dot (.) laga kar `splice()` likhna hota hai. `splice()` method ke pehle argument mein hum specify kar sakte hain ke humein kis index se elements remove karna hai. Dusra argument yeh batata hai ke hum kitne elements remove karna chahte hain. Teesra aur agla argument koi bhi ho sakte hain aur yeh batata hai ke hum uss index pe kis naye elements ko insert karna chahte hain. Jaise:

```bash
let arr = [1, 2, 3, 4, 5];
arr.splice(2, 2, "apple", "banana");
console.log(arr); // [1, 2, "apple", "banana", 5]
```

Iss code mein `splice()` method ne `arr` array ke index 2 se lekar 2 elements ko remove karke usmein "apple" aur "banana" elements insert kiya hai. Iss tarah se `arr` mein `[1, 2, "apple", "banana", 5]` store hoga.

Agar `splice()` method ke sirf ek argument diya jaaye, jaise `arr.splice(2)` toh yeh uss index se lekar array ke end tak ke elements ko remove karega.

```scss
let arr = [1, 2, 3, 4, 5];
arr.splice(2);
console.log(arr); // [1, 2]
```

Iss code mein `splice()` method ne `arr` array ke index 2 se lekar array ke end tak ke elements ko remove kar diya hai. Iss tarah se `arr` mein `[1, 2]` store hoga.

what is method of unshift() in javascript? explain in roman urdu

`unshift()` JavaScript ka ek method hai jo ek array ke shuruaat mein ek ya zyada elements insert karta hai. Yeh method existing array ko modify karta hai aur modify kiya hua array ka naya length return karta hai.

Iss method ko istemaal karne ke liye, array variable ke baad dot (.) laga kar `unshift()` likhna hota hai. `unshift()` method ke argument mein hum specify kar sakte hain ke humein kis elements ko array ke shuruaat mein insert karna hai. Jaise:

```
javascript
```

```javascript
let arr = [1, 2, 3];
let length = arr.unshift(4, 5);
console.log(arr); // [4, 5, 1, 2, 3]
console.log(length); // 5
```

Iss code mein `unshift()` method ne `arr` array ke shuruaat mein 4 aur 5 elements insert kiya hai. Iss tarah se `arr` mein `[4, 5, 1, 2, 3]` store hoga. Aur `unshift()` method ka return value hai 5 kyunki `arr` array mein ab 5 elements hain.

`unshift()` method se hum ek se zyada elements ko bhi array ke shuruaat mein insert kar sakte hain.

```
javascript
```

```javascript
let arr = [1, 2, 3];
let length = arr.unshift(4, 5, 6, 7);
console.log(arr); // [4, 5, 6, 7, 1, 2, 3]
console.log(length); // 7
```

Iss code mein `unshift()` method ne `arr` array ke shuruaat mein 4, 5, 6 aur 7 elements insert kiya hai. Iss tarah se `arr` mein `[4, 5, 6, 7, 1, 2, 3]` store hoga. Aur `unshift()` method ka return value hai 7 kyunki `arr` array mein ab 7 elements hain.

what is method of shift() in javascript? explain in roman urdu

`shift()` JavaScript ka ek method hai jo array ke shuruaat se ek element remove karta hai. Yeh method existing array ko modify karta hai aur remove kiya hua element return karta hai.

Iss method ko istemaal karne ke liye, array variable ke baad dot (.) laga kar `shift()` likhna hota hai. `shift()` method ka koi argument nahi hota hai. Yeh hamesha array ke shuruaat se pehle wala element remove karta hai. Jaise:

```
javascript



let arr = [1, 2, 3];
let removedElement = arr.shift();
console.log(arr); // [2, 3]
console.log(removedElement); // 1
```

Iss code mein `shift()` method ne `arr` array ke shuruaat se 1 element remove kiya hai. Iss tarah se `arr` mein `[2, 3]` store hoga. Aur `shift()` method ka return value hai 1 kyunki `arr` array se 1 element remove hua hai.

Agar `shift()` method ko empty array par istemaal kiya jaaye toh yeh `undefined` return karega. Jaise:

```
javascript



let arr = [];
let removedElement = arr.shift();
console.log(arr); // []
console.log(removedElement); // undefined
```

Iss code mein `arr` array khali hai, isliye `shift()` method ka return value `undefined` hai.

what is method of push() in javascript? explain in roman urdu

`push()` JavaScript ka ek method hai jo ek array ke end mein ek ya zyada elements add karta hai. Yeh method existing array ko modify karta hai aur modify kiya hua array ka naya length return karta hai.

Iss method ko istemaal karne ke liye, array variable ke baad dot (.) laga kar `push()` likhna hota hai. `push()` method ke argument mein hum specify kar sakte hain ke humein kis elements ko array ke end mein add karna hai. Jaise:

```
javascript



let arr = [1, 2, 3];
let length = arr.push(4, 5);
console.log(arr); // [1, 2, 3, 4, 5]
console.log(length); // 5
```

Iss code mein `push()` method ne `arr` array ke end mein 4 aur 5 elements add kiya hai. Iss tarah se `arr` mein `[1, 2, 3, 4, 5]` store hoga. Aur `push()` method ka return value hai 5 kyunki `arr` array mein ab 5 elements hain.

`push()` method se hum ek se zyada elements ko bhi array ke end mein add kar sakte hain.

```
javascript



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

Iss code mein `push()` method ne `arr` array ke end mein 4, 5, 6 aur 7 elements add kiya hai. Iss tarah se `arr` mein `[1, 2, 3, 4, 5, 6, 7]` store hoga. Aur `push()` method ka return value hai 7 kyunki `arr` array mein ab 7 elements hain.

what is method of pop() in javascript? explain in roman urdu

`pop()` JavaScript ka ek method hai jo ek array ke end se ek element remove karta hai. Yeh method existing array ko modify karta hai aur remove kiya hua element return karta hai.

Iss method ko istemaal karne ke liye, array variable ke baad dot (.) laga kar `pop()` likhna hota hai. `pop()` method ka koi argument nahi hota hai. Yeh hamesha array ke end se last wala element remove karta hai. Jaise:

```javascript
let arr = [1, 2, 3];
let removedElement = arr.pop();
console.log(arr); // [1, 2]
console.log(removedElement); // 3
```

Iss code mein `pop()` method ne `arr` array ke end se 1 element remove kiya hai. Iss tarah se `arr` mein `[1, 2]` store hoga. Aur `pop()` method ka return value hai 3 kyunki `arr` array se 3 element remove hua hai.

Agar `pop()` method ko empty array par istemaal kiya jaaye toh yeh `undefined` return karega. Jaise:

```javascript
let arr = [];
let removedElement = arr.pop();
console.log(arr); // []
console.log(removedElement); // undefined
```

Iss code mein `arr` array khali hai, isliye `pop()` method ka return value `undefined` hai.