README.md

JavaScript-Array-Interview-Practice

#Question1 ##Create an Array object.

```
Method 1

var fruits = ['Apple', 'Banana'];
console.log(fruits) // [ 'Apple', 'Banana' ]



Method 2

var msgArray = [];
msgArray[0] = 'Hello';
console.log(msgArray) //  [ 'Hello' ]



Method 3

var array = new Array('Hello');
console.log(array) // [ 'Hello' ]



Method 4
```

```
var another = Array.of(1, 2, 3);
console.log(another) // [ 1, 2, 3 ]



Method 5

var b = arrayMaker({7: 1}, {2: 3});

function arrayMaker(n) {
  console.log(n);
  if (n !== typeof Array) {
    return Array.prototype.slice.call(arguments);
  }
}

console.log(b) //  [ { '7': 1 }, { '2': 3 } ]
```

#Question 2

##Take this array var array = [1,2,3,4,5] and copy it using

> the slice method and the for loop method

```
Method 1 - The slice method

var array = [1,2,3,4,5,6];

var result = array.slice();  // to copy an array to new array

console.log(array);  // [1,2,3,4,5,6]
console.log(result); // [1,2,3,4,5,6]



Method 2 - The for loop method

var array = [1, 2, 3, 4, 5, 6];
Var array2 = [ ];

for (var i = 0; i < array.length; ++i) {

  array2[i] = array[i];
}

console.log (array2); // [ 1, 2, 3, 4, 5, 6 ]
```

#Question 3

##Empty this array var array = [1,2,3,4,5]

```
Method 1

Var array = [1,2,3,4,5];

Array = [ ];
```

N.B

> This is only recommended if you don't have any other references to this array because it will actually create a new empty array and the other reference will still be available to others in memory.

```
EXAMPLE
var array = [1,2,3,4,5];
var array2 = array;

array = [ ];

console.log(array);  // [ ];
console.log(array2); // [ 1, 2, 3, 4, 5 ]
```

```
Method 2
var array3 = [1,2,3,4,5];
array3.length = 0
console.log(array3);  // [ ];
```

NB

> This even empties to referenced arrays

```
var array3 = [1,2,3,4,5];
var array4 = array3;
array3.length = 0;
console.log(array3);  // [ ];
console.log(array4);  // [ ];
```

```
Method 3
var array5 = [1,2,3,4,5];
array5.splice(0,array5.length);
console.log(array5);  // [ ];
```

```
Method 4
var array6 = [1,2,3,4,5];
console.log(array6);  // [1,2,3,4,5]
```

```
function emptyArray(array){
  'use strict';
    while(array.length){
     array6.pop();
}
}

emptyArray(array6);  // call function
console.log(array6);  // [ ] ; now empty
```

#Question 4 ##What type is an Array set to?

```
Var array3 = [1,2,3,4,5];
console.log(typeof(array3));  // Object
```

#Question 5 ##How can you check if something is an Array?

```
Method 1

var check = [1, 2, 3];
var a = Array.isArray([1, 2, 3]);
var b = Array.isArray({
  foo: 123
});
var c = Array.isArray('foobar');
var d = Array.isArray(undefined);
var e = Array.isArray(check);

console.log(a); // true
console.log(b); // false
console.log(c); // false
console.log(d); // false
console.log(e); // true



Method 2

function checkIfArray(array) {
  'use strict';

  if (Object.prototype.toString.call(array) === '[object Array]') {
    console.log('array it is  ');
  } else {
    console.log('array it is Not ');
  }
}

var array2 = 'testing';
```

```
checkIfArray(array2);  // array it is Not
var array3 = [1,2,3,4,5];
checkIfArray(array3); //array it is



Method 3

var array = [1, 2, 3, 4, 5];

function checkIfArray(object) {
  'use strict';
  if (typeof object === 'string') {
    console.log('array it is NOT ');
  } else {
    console.log('array it is ');
  }
}

checkIfArray(array);  //array it is
```

#Question 6 ##Add an item to the end of an array.

```
Method 1
var array = ['a','b','c'];

array.push('d');
console.log(array); // [ 'a', 'b', 'c', 'd' ]



Method 2
array[array.length] = 'e';
console.log(array); // [ 'a', 'b', 'c', 'd', 'e' ]
```

#Question 7 ##Find the index position of d in this array var arr= ['a','b','c','d'];

`Answer : console.log(arr.indexOf('d')); // 3`

#Question 8 ##What will be returned if you look for the index of something that does not exist?

`var arr= ['a','b','c','d']; console.log(arr.indexOf(7)); // -1 === does not exist`

#Question 9 ##Write a function to check if milk exists in your array var items = ['milk', 'bread', 'sugar'];

Answer

```
var items = ['milk', 'bread', 'sugar'];

function checkForProduct(item){

    if (items.indexOf(item) === -1) {

    console.log('item does not exist');
} else {

    console.log('item is in your list');

}
}

checkForProduct('socks'); //item does not exist
checkForProduct('milk'); //item is in your list
```

#Question 10 ##Now you've found milk exists add some code to find the index of milk and remove that item.

```
var items = ['milk', 'bread', 'sugar'];

//find index of item if it exists
var a = items.indexOf('milk');
console.log(a); // 0

//remove that index from array
items.splice(0,1);
console.log(items); // [ 'bread', 'sugar']
```

#Question 11 ##List the ways to loop over an array.

> For Each

> For in

> For loop

#Question 12 ##Write some code to put these numbers in order var numbers = [1, 12, 2 ,23,77,7,33,5,99,234,];

```
var numbers2 = [1, 12, 2 ,23,77,7,33,5,99,234];
var numbers3 = numbers2.sort((a, b) => {
   return a - b;
});

console.log(numbers3); // [ 1, 2, 5, 7, 12, 23, 33, 77, 99, 234 ]
```

#Question 13 ##Write some code to place this list in alphabetical order var p = ['a','z','e','y'];

```
var p = ['a','z','e','y'];
p.sort();
console.log(p); // [ 'a', 'e', 'y', 'z' ]
```

#Question 14 ##What is the length of these arrays

```
A. var arr1 = [,,,];

B. var arr2 = new Array(3)

C. var arr3 = [1,2,3,4,5]

D. var array = [ [1,2,3], [4,5,6]  ];

E. var array[0].length = [ [1,2,3], [4,5,6]  ];


Results

A. arr1.length = 3
B. arr2.length = 3
C. arr3.length = 5
D. array.length = 2    counts the number of internal array
E. array[0].length = 3 first internal array within the outer array
```

#Question 15 ##What are the results of these splice and slice methods

```
var a = ['zero', 'one', 'two', 'three'];
var names = ['jason', 'john', 'peter', 'karen'];

var sliced = a.slice(1, 3);
var spliced = names.splice(1,3);


The slice() method returns a shallow copy of a portion of an array into a new
array object selected from begin to end (end not included). The original array
will not be modified.

console.log(sliced); // creates a new array ['one', 'two']
console.log(a); // main array remains untouched

The splice() method changes the content of an array by removing existing elements
and/or adding new elements.

console.log(spliced); // it returns  [ 'john', 'peter', 'karen' ]
console.log(names); // however the array only contains jason now
```

#Question 16 ##What are the console logs of these shift and unshift methods

```
Var a = [ ] ;

We take an empty array and

a.unshift(1);
var a = console.log(a)
a.unshift(22);
var b = console.log(a)
a.shift();
var c = console.log(a)
a.unshift(3,[4,5]);
var d = console.log(a)
a.shift();
var e = console.log(a)
a.shift();
var f = console.log(a)
a.shift();
var g = console.log(a)
Results

Var a = [ 1 ]       // we a.unshift(1) so added 1 to front

Var b = [ 22, 1 ]   // we a.unshift(22) so added 22 to front

Var c = [ 1 ]       // we a.shift() so removed the first element

Var d = [ 3, [ 4, 5 ], 1 ]   // we a.unshift(3,[4,5]) so added
                                these to front
Var e = [ [ 4, 5 ], 1 ]   // we a.shift() so remove first element

Var f = [ 1 ]   // we a.shift() so remove first element

Var g = [ ]   // we a.shift() so remove first element leaving it
                 empty
```

#Question 17

##Using reduce add all these numbers var numbers = [1, 2, 3, 4, 5, 6];

```
var numbers = [1, 2, 3, 4, 5, 6];

var total = numbers.reduce((a, b) => {
  return a + b;
});

console.log(total); // Total returned is : 21
```

#Question 18 ##Flatten this array to one single array using reduce Var array = [[0, 1], [2, 3], [4, 5]];

```
Var array =  [[0, 1], [2, 3], [4, 5]];

var flattened = array.reduce(function(a, b) {
    return a.concat(b);
},[ ]);

console.log(flattened); // [ 0, 1, 2, 3, 4, 5 ]
```

#Question 19 ##Filter this array to return just the dogs

```
var animals = [
    { name: "Jason", species:"rabbit"},
    { name: "Jessica", species:"dog"},
    { name: "Jacky", species:"owl"},
    { name: "Luke", species:"fish"},
    { name: "Junior", species:"rat"},
    { name: "Thomas", species:"cat"}
]
Answer

/*******************************************
   filter method with callback function
*******************************************/

var dogs = animals.filter(function(animals){
    return animals.species === "dog";
});

console.log(dogs);

Returns

[ { name: 'Jessica', species: 'dog' }]

The filter() method creates a new array with all elements that pass the test
implemented by the provided function.
```

#Question 20 ##Using array in question 19 use map function to return all the species

```
var types = animals.map(function(animals){
    return animals.species;
});
console.log(types); // [ 'rabbit', 'dog', 'owl', 'fish', 'rat', 'cat' ]
```

Releases

No releases published

Packages

No packages published