arrays-interview-questions.js

```
/*

======================================================================

                Arrays Interview Questions

======================================================================

*/

/*

1. Add "December" to the end of the array

*/

const months = ["January", "February", "March", "April", "May", "June", "July", "August", "September", "October", "November"];

months.splice(months.length, 0, "December");

console.log(months);

/*

2. Update the value of "March" to "march"

*/

months.splice(2, 1, "march");

console.log(months);

/*

3. Delete "April" from the array

```

```
*/

months.splice(3, 1);

console.log(months);

/*

4. Given an array of products where each product has a name and a price. Write a function that uses filter to return only the products that have a price less than 500.

*/

const products = [

    {
        name: "Laptop",
        price: 1200
    },

    {
        name: "Phone",
        price: 800
    },

    {
        name: "Tablet",
        price: 300
    },

    {
        name: "Smart Watch",
        price: 150
    }
]

const filteredProducts = products.filter((currentProduct) => { currentProduct.price < 500; })

```

```
// currentProduct Value is 1200 < 500 = false

// currentProduct Value is 800 < 500 = true

// currentProduct Value is 300 < 500 = true

// currentProduct Value is 150 < 500 = true

console.log(filteredProducts);

/*

5. Filter unique values from an array

*/

const numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10];

const uniqueValues = numbers.filter((currentNumber, index) => {

    return numbers.indexOf(currentNumber) === index;
})

console.log(uniqueValues);

// Using Set Method: It will remove all duplicate values, includes only unique values and returns a new array.

const uniqueValues2 = [...new Set(numbers)];

console.log(uniqueValues2);

/*

6. Use map() to add 5 to each element in the array

*/

const newNumbers = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10];
```

```
112
113 const addNumbers = newNumbers.map((currentNumber) => {
114
115     return currentNumber + 5;
116 });
117
118
119 console.log(addNumbers);
120
121 // Current value of currentNumber is 1 + 5 = 6
122
123 // Current value of currentNumber is 2 + 5 = 7
124
125 // Current value of currentNumber is 3 + 5 = 8
126
127 // Current value of currentNumber is 4 + 5 = 9
128
129 // Current value of currentNumber is 5 + 5 = 10
130
131 // Current value of currentNumber is 6 + 5 = 11
132
133 // Current value of currentNumber is 7 + 5 = 12
134
135 // Current value of currentNumber is 8 + 5 = 13
136
137 // Current value of currentNumber is 9 + 5 = 14
138
139 // Current value of currentNumber is 10 + 5 = 15
140
141 /*
142
143 7. Use reduce() to sum up all the elements in the array
144
145 */
146
147 const sumArray = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10];
148
149 const sum = sumArray.reduce((accumulator, currentElement) => {
```

```
150
151     return accumulator + currentElement;
152
153 }, 0);
154
155 // Initial value of accumulator is 0 + 1 = 1
156
157 // Initial value of accumulator is 1 + 2 = 3
158
159 // Initial value of accumulator is 3 + 3 = 6
160
161 // Initial value of accumulator is 6 + 4 = 10
162
163 // Initial value of accumulator is 10 + 5 = 15
164
165 // Initial value of accumulator is 15 + 6 = 21
166
167 // Initial value of accumulator is 21 + 7 = 28
168
169 // Initial value of accumulator is 28 + 8 = 36
170
171 // Initial value of accumulator is 36 + 9 = 45
172
173 // Initial value of accumulator is 45 + 10 = 55
174
175 console.log(sum);
```