



Instantly share code, notes, and snippets.

ashishkrtewari / **w3resource-javascript-array-exercises**

Last active 5 months ago

☆ Star

<> Code

🔗 Revisions 2

☆ Stars 1

🔗 Forks 1

W3Resource JavaScript array - Exercises, Practice, Solution

🔗 w3resource-javascript-array-exercises

Raw

```
1 // 1. Write a JavaScript function to check whether an `input` is an array or not. Go to the editor
2 // Test Data :
3 // console.log(is_array('w3resource'));
4 // console.log(is_array([1, 2, 4, 0]));
5 // false
6 // true
7
8 // Solution
9 (function () {
10     const isArray = (arr) => {
11         return toString.call(arr) === "[object Array]";
12     };
13 })();
14
15 /*****
16
17 // 2. Write a JavaScript function to clone an array. Go to the editor
18 // Test Data :
19 // console.log(array_Clone([1, 2, 4, 0]));
20 // console.log(array_Clone([1, 2, [4, 0]]));
21 // [1, 2, 4, 0]
22 // [1, 2, [4, 0]]
23
24 // Solution
25 (function () {
26     const testArray = [1, 2, [4, 0]];
27
28     const cloneArr = (arr) => {
29         return arr.map((item) => {
30             if (Array.isArray(item)) {
31                 return cloneArr(item);
32             } else {
```

```

33         return item;
34     }
35     });
36 };
37 })();
38
39 /*****
40
41 // 3. Write a JavaScript function to get the first element of an array. Passing a parameter 'n' wi
42 // Test Data :
43 // console.log(first([7, 9, 0, -2]));
44 // console.log(first([],3));
45 // console.log(first([7, 9, 0, -2],3));
46 // console.log(first([7, 9, 0, -2],6));
47 // console.log(first([7, 9, 0, -2],-3));
48 // Expected Output :
49 // 7
50 // []
51 // [7, 9, 0]
52 // [7, 9, 0, -2]
53 // []
54
55 // Solution
56 (function () {
57     const getElements = (arr, n) => {
58         return arr.filter((item, index) => index < n);
59     };
60 })();
61
62 /*****
63
64 // 4. Write a JavaScript function to get the last element of an array. Passing a parameter 'n' wil
65 // Test Data :
66 // console.log(last([7, 9, 0, -2]));
67 // console.log(last([7, 9, 0, -2],3));
68 // console.log(last([7, 9, 0, -2],6));
69 // Expected Output :
70 // -2
71 // [9, 0, -2]
72 // [7, 9, 0, -2]
73
74 // Solution
75 (function () {
76     const getElements = (arr, n) => {
77         arr.filter((item, index) => index > arr.length - 1 - n);
78     };
79 })();
80
81 /*****
82
83 // 5. Write a simple JavaScript program to join all elements of the following array into a string.
84 // Sample array : myColor = ["Red", "Green", "White", "Black"];

```

```

85 // Expected Output :
86 // "Red,Green,White,Black"
87 // "Red,Green,White,Black"
88 // "Red+Green+White+Black"
89
90 // Solution
91 (() => {
92     let myColorList = ["Red", "Green", "White", "Black"];
93     const arrayToString = (arr) => {
94         return arr.toString();
95     };
96     arrayToString(myColorList);
97 })();
98
99 /*****
100
101 // 6. Write a JavaScript program which accept a number as input and insert dashes (-) between each
102
103 // Solution
104 (() => {
105     let num = 025468;
106     const dashedEvenNumbers = (num) => {
107         numArray = [];
108         num
109             .toString()
110             .match(/\d/gi)
111             .map((item, i) => {
112                 const numItem = parseInt(item);
113                 // If Item is even and the last item of numArray is also even, first push "-" then number
114                 if (numItem % 2 === 0 && numArray[numArray.length - 1] % 2 === 0) {
115                     numArray.push("-");
116                 }
117                 numArray.push(numItem);
118             });
119         return numArray.join("");
120     };
121
122     dashedEvenNumbers(num);
123 })();
124
125 /*****
126
127 // 7. Write a JavaScript program to find the most frequent item of an array. Go to the editor
128 // Sample array : var arr1=[3, 'a', 'a', 'a', 2, 3, 'a', 3, 'a', 2, 4, 9, 3];
129
130 // Solution
131 (() => {
132     var arr = [3, "a", "a", "a", 2, 3, "a", 3, "a", 2, 4, 9, 3];
133     const getMaxCount = (arr) => {
134         const itemCounter = {};
135         let maxCount = null; //Saves the item with maxCount
136         // Loop and save the counts in an Object itemCounter

```

```

137     arr.map((item) => {
138         if (maxCount === null) {
139             maxCount = item;
140         }
141         if (itemCounter[item]) {
142             itemCounter[item] += 1;
143         } else {
144             itemCounter[item] = 1;
145         }
146         // if the maxCount item has count less than current item's count, update maxCount
147         if (itemCounter[maxCount] < itemCounter[item]) {
148             maxCount = item;
149         }
150     });
151     return maxCount;
152 };
153 })();
154
155 /*****
156
157 // 8. Write a JavaScript program which accept a string as input and swap the case of each character
158
159 // Solution
160 (() => {
161     const str = "The Quick Brown Fox";
162
163     const toggleCase = (str) => {
164         // Get The String Array
165         let strArray = str.match(/\w|(\s\w)/gi);
166         // check if the chars lie in the charCode range above 96 which means they are lowercase
167         strArray = strArray.map((item, i) => {
168             if (item && item.charCodeAt(0) > 96) {
169                 return item.toUpperCase();
170             } else if (item) {
171                 return item.toLowerCase();
172             } else {
173                 return item;
174             }
175         });
176         strArray.join("");
177     };
178 })();
179
180 /*****
181
182 //9. Write a JavaScript program to remove duplicate items from an array (ignore case sensitivity)
183
184 // Solution
185 (() => {
186     var arr = [1, 1, 2, 3, 4, 1, 2, 5, 7, 8, 0];
187     const makeItUnique = (arr) => {
188         // Using Sets

```

```

189     let setFromArray = new Set(arr);
190     console.log(Array.from(setFromArray));
191
192     //alt approach from Object
193     const objFromArray = {};
194     for (const i of arr) {
195         if (objFromArray[i] === undefined) {
196             objFromArray[i] = true;
197         }
198     }
199     return Object.keys(objFromArray).map((item) => parseInt(item));
200 };
201 })();
202
203 /*****
204
205 //10. Write a JavaScript function to get a random item from an array.
206
207 // Solution
208 (() => {
209     var arr = [1, 2, 3, 4, 5, 7, 8, 9];
210     const getARandomItem = (arr) => {
211         return arr[Math.floor(Math.random() * (arr.length - 1))];
212     };
213 })();
214
215 /*****
216
217 // 11. Write a JavaScript function to create a specified number of elements with pre-filled numeric
218
219 // Test Data :
220 // console.log(array_filled(6, 0));
221 // [0, 0, 0, 0, 0, 0]
222 // console.log(array_filled(4, 11));
223 // [11, 11, 11, 11]
224
225 // Solution
226 (() => {
227     function renderArray(length, entry) {
228         const arr = new Array(length);
229         return arr.fill(entry);
230     }
231 })();

```

 [w3resource-javascript-array-exercises-practice-solution.markdown](#)

Raw

## W3Resource JavaScript array - Exercises, Practice, Solution

A [Pen](#) by [Ashish Tewari](#) on [CodePen](#).

[License.](#)

Sign up for free

to join this conversation on **GitHub**. Already have an account? [Sign in to comment](#)

[Terms](#) [Privacy](#) [Security](#) [Status](#) [Docs](#) [Contact GitHub](#) [Pricing](#) [API](#) [Training](#) [Blog](#) [About](#)



© 2021 GitHub, Inc.