**Challenge1**

**Index.html**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>ES5.1 Array Helper Methods</title>

</head>

<body>

    <script src="challenge1.js"></script>

</body>

</html>

**Challenge1.js**

const myCities=["London","New York","Singapore"];

const arrayInfo= (city, position) => {

    return city+" is located at the index "+position+" in the myCities array";

}

// for

console.log("Using For loop");

(function(){

    const LEN=myCities.length;

    for(let i=0;i<LEN;i++){

        console.log(arrayInfo(myCities[i],i));

    }

})();

console.log("---------------------------");

// for Each

console.log("Using For Each loop");

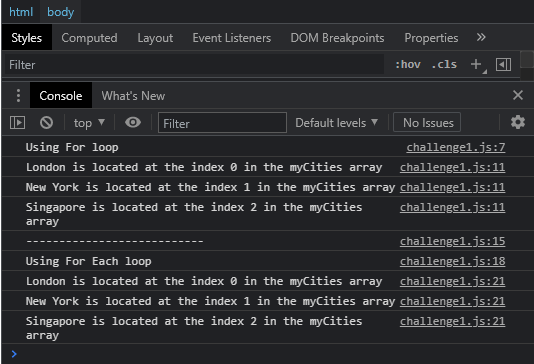
(function(){

    let i=0;

    myCities.forEach(city => console.log(arrayInfo(city,i++)));

})();

**Result**

****

**Challenge2**

**Index.html**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>ES5.1 Array Helper Methods</title>

</head>

<body>

    <script src="challenge2.js"></script>

</body>

</html>

**Challenge2.js**

const postsJSON = [

    '{"postId":1355,"commentsQuantity":5}',

    '{"postId":5131,"commentsQuantity":13}',

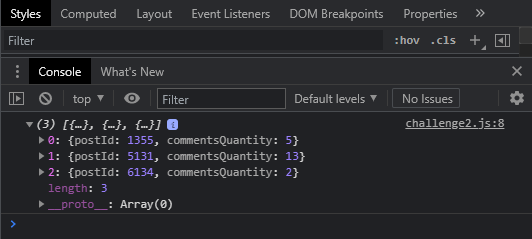
    '{"postId":6134,"commentsQuantity":2}'

  ];

  const postsArray= postsJSON.map(post => JSON.parse(post));

  console.log(postsArray);

**Result**

****

**Challenge3**

**Index.html**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>ES5.1 Array Helper Methods</title>

</head>

<body>

    <script src="challenge3.js"></script>

</body>

</html>

**Challenge3.js**

const posts = [

    { postId: 1355, commentsQuantity: 5 },

    { postId: 5131, commentsQuantity: 13 },

    { postId: 6134, commentsQuantity: 2 }

  ];

  /\* Create a function "findSinglePost" that will have two parameters - "postId" and "posts" and will return object with matched "postId".

  If post wasn't found - return "undefined"

  \*/

const findSinglePost=(findPostId,postsDict) =>{

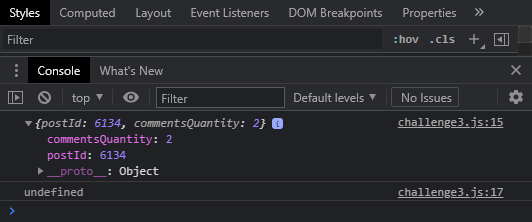
    return postsDict.find(post=> post.postId===(findPostId));

}

  console.log(findSinglePost(6134, posts)); // { postId: 6134, commentsQuantity: 2 }

  console.log(findSinglePost(4511, posts)); // undefined

**Result**

****

**Challenge4**

**Index.html**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>ES5.1 Array Helper Methods</title>

</head>

<body>

    <script src="challenge4.js"></script>

</body>

</html>

**Challenge4.js**

const a = [5, "abc", 10, 1];

const b = [4, 10, 14, 25, 25, 50];

const c = [150, 132, 80, 40];

const d = [15, 26, 10, 23, 85];

const arrayCheck = inputArray=>{

    if(inputArray.some(element => typeof element != "number")){

        return "Some elements are not numbers";

    }

    if(inputArray.every( (element,index,array) => index>0 ? element>=array[index-1] : true)){

        return "Array is sorted is ascending order";

    }

    if(inputArray.every(

        (element,index,array)=>index>0 ? element<=array[index-1] :true

    )){

        return "Array is sorted is descending order";

    }

    return "Array is not sorted";

};

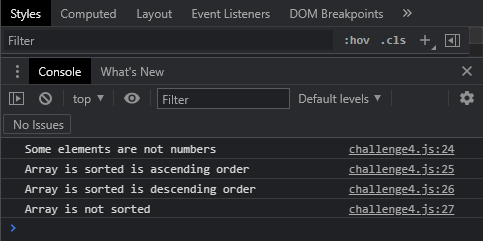
console.log(arrayCheck(a));

console.log(arrayCheck(b));

console.log(arrayCheck(c));

console.log(arrayCheck(d));

**Result**

****

**Challenge5**

**Index.html**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>ES5.1 Array Helper Methods</title>

</head>

<body>

    <script src="challenge5.js"></script>

</body>

</html>

**Challenge5.js**

const a = [1, 2, 3];

const b = [1, 2, 3];

console.log(a === b); // false

const arraysAreEqual = (arrayOne,arrayTwo) =>{

    if(arrayOne.length != arrayTwo.length){

        return false;

    }

    if(arrayOne.some((element,index)=>

            element != arrayTwo[index]

    )){

        return false;

    }

    return true;

}

const c = [2, 1, 3];

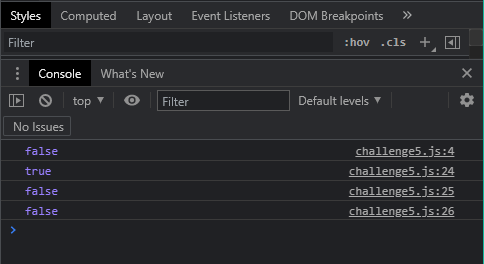
const d = [1, 2, 3, 4];

console.log(arraysAreEqual(a, b)); // true

console.log(arraysAreEqual(a, c)); // false

console.log(arraysAreEqual(a, d)); // false

**Result**

****

**Challenge6**

**Index.html**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>ES5.1 Array Helper Methods</title>

</head>

<body>

    <script src="challenge6.js"></script>

</body>

</html>

**Challenge6.js**

const transport = ["Bus", "Car", "Bicycle", "Airplane"];

const elementFound= (inputArray,searchElement) =>{

    if(inputArray.some(element => element === searchElement)){

        return true;

    }

    return false;

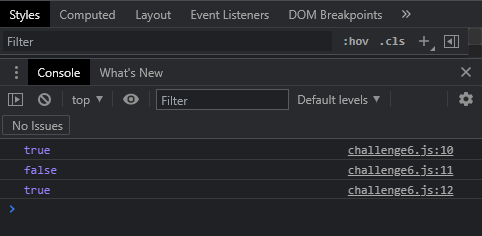
}

console.log(elementFound(transport, "Bus")); // true

console.log(elementFound(transport, "Phone")); // false

console.log(elementFound(transport, "Airplane")); // true

**Result**

****

**Challenge7**

**Index.html**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>ES5.1 Array Helper Methods</title>

</head>

<body>

    <script src="challenge7.js"></script>

</body>

</html>

**Challenge7.js**

const tags = [

    ["javascript", "es6"],

    ["css", "flexbox"],

    ["html", "web-browser"]

  ];

  const fruits = [

    { title: "Orange", quantity: 10 },

    { title: "Banana", quantity: 5 },

    { title: "Apple", quantity: 25 }

  ];

  const primitiveTypesArray = [

    25,

    "x",

    true,

    undefined,

    null

  ];

  const elementIsIncluded=(searchElement,array)=>{

      if(typeof searchElement === "object"){

          searchElement=JSON.stringify(searchElement);

            array.forEach((element,index) => {

              array[index]=JSON.stringify(element);

          });

        //   console.log(searchElement);

        //   console.log(array);

      }

      return (function(){

          return array.includes(searchElement);

      })();

  }

  console.log(elementIsIncluded(["css", "flexbox"], tags)); // true

  console.log(elementIsIncluded(["flexbox", "css"], tags)); // false

  console.log(

    elementIsIncluded(

      { title: "Apple", quantity: 25 },

      fruits

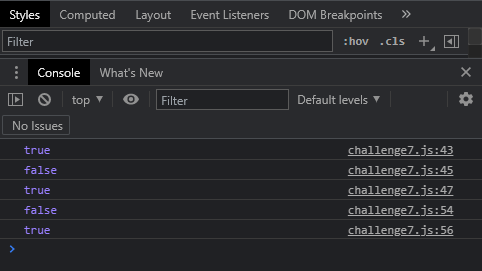
    )

  ); // true

  console.log(elementIsIncluded({ title: "Banana" }, fruits)); // false

  console.log(elementIsIncluded(25, primitiveTypesArray)); // true

**Result**

****

**Challenge8**

**Index.html**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>ES5.1 Array Helper Methods</title>

</head>

<body>

    <script src="challenge8.js"></script>

</body>

</html>

**Challenge8.js**

const myNumbers = [123, 50, 27];

const pushIfUnique =(inputArray,newElement) => {

    // if( inputArray.includes(newElement)){

    //     console.log(newElement +" is already in the array");

    // }

    // else{

    //     inputArray.push(newElement);

    // }

    inputArray.includes(newElement) ? console.log(newElement +" is already in the array") : inputArray.push(newElement);

}

pushIfUnique(myNumbers, 50); // "50 is already in the array"

console.log(myNumbers); // [123, 50, 27]

pushIfUnique(myNumbers, 80);

console.log(myNumbers); // [123, 50, 27, 80]

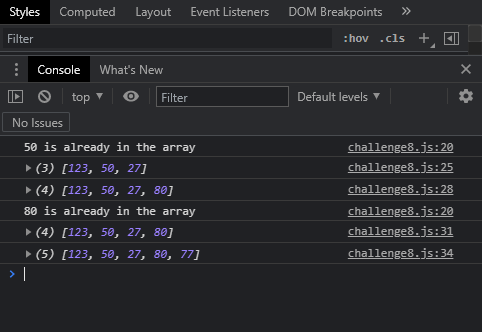
pushIfUnique(myNumbers, 80); // "80 is already in the array"

console.log(myNumbers); // [123, 50, 27, 80]

pushIfUnique(myNumbers, 77);

console.log(myNumbers); // [123, 50, 27, 80, 77]

**Result**

****

**Challenge9**

**Index.html**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>ES5.1 Array Helper Methods</title>

</head>

<body>

    <script src="challenge9.js"></script>

</body>

</html>

**Challenge9.js**

const posts = [

    {

      title: "How to find JavaScript developer job?",

      postId: 3421,

      comments: 25

    },

    {

      title: "Is it hard to learn ES6?",

      postId: 5216,

      comments: 3

    },

    {

      title: "Should I learn React or Angular?",

      postId: 8135,

      comments: 12

    }

  ];

  const minimalComentsQty = 10;

  const popularPostsIds=(posts, minimalComentsQty)=>

      posts.reduce((postIds,post) => {

          if (post.comments>= minimalComentsQty)

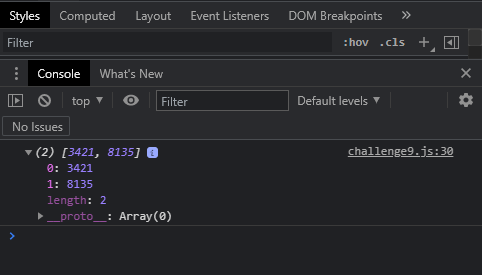
            postIds.push(post.postId);

          return postIds;

      },[]);

  console.log(popularPostsIds(posts, minimalComentsQty)); // [3421, 8135]

**Result**

****

**Challenge10**

**Index.html**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>ES5.1 Array Helper Methods</title>

</head>

<body>

    <script src="challenge10.js"></script>

</body>

</html>

**Challenge10.js**

const products = [

    {

      title: "Phone case",

      price: 23,

      quantity: 2,

      category: "Accessories"

    },

    {

      title: "Android phone",

      price: 150,

      quantity: 1,

      category: "Phones"

    },

    {

      title: "Headphones",

      price: 78,

      quantity: 1,

      category: "Accessories"

    },

    {

      title: "Sport Watch",

      price: 55,

      quantity: 2,

      category: "Watches"

    }

  ];

 const quantityByCategories = products =>

     products.reduce((categories,product) =>{

         const cat=product.category;

         if(categories.hasOwnProperty(cat)){

             categories[cat]+=1;

         }

         else{

             categories[cat]=1;

         }

         return categories;

     }

     ,{});

  console.log(quantityByCategories(products));

  /\* {

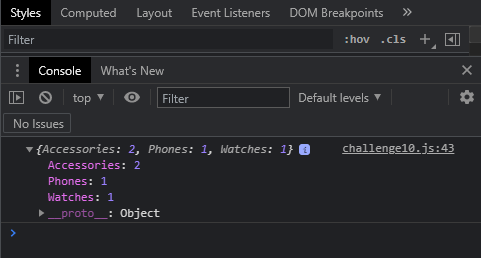
    Accessories: 3,

    Phones: 1,

    Watches: 2

  } \*/

**Result**

****

**Challenge11**

**Index.html**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>ES5.1 Array Helper Methods</title>

</head>

<body>

    <script src="challenge11.js"></script>

</body>

</html>

**Challenge11.js**

const products = [

    {

      title: "Phone case",

      price: 23,

      quantity: 2,

      category: "Accessories"

    },

    {

      title: "Android phone",

      price: 150,

      quantity: 1,

      category: "Phones"

    },

    {

      title: "Headphones",

      price: 78,

      quantity: 1,

      category: "Accessories"

    },

    {

      title: "Sport Watch",

      price: 55,

      quantity: 2,

      category: "Watches"

    }

  ];

  const sortProductsByPrice= products =>{

      products.sort((a,b)=>{

          return a.price - b.price;

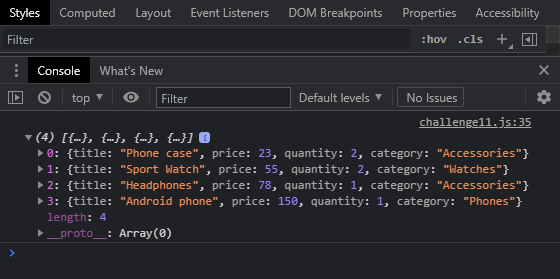
      });

  }

  sortProductsByPrice(products);

  console.log(products);

**Result**

****

**Challenge12**

**Index.html**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>ES5.1 Array Helper Methods</title>

</head>

<body>

    <script src="challenge12.js"></script>

</body>

</html>

**Challenge12.js**

const persons = [

    {

      name: "Andy",

      friendsQty: 10,

      index: 1

    },

    {

      name: "Mike",

      friendsQty: 5,

      index: 2

    },

    {

      name: "Sophia",

      friendsQty: 10,

      index: 3

    },

    {

      name: "Joshua",

      friendsQty: 3,

      index: 4

    },

    {

      name: "John",

      friendsQty: 10,

      index: 5

    },

    {

      name: "Gabriella",

      friendsQty: 8,

      index: 6

    },

    {

      name: "Tyler",

      friendsQty: 7,

      index: 7

    },

    {

      name: "Dylan",

      friendsQty: 2,

      index: 8

    },

    {

      name: "Sarah",

      friendsQty: 5,

      index: 9

    },

    {

      name: "Alexa",

      friendsQty: 10,

      index: 10

    },

    {

      name: "Henry",

      friendsQty: 10,

      index: 11

    },

    {

      name: "Arianna",

      friendsQty: 10,

      index: 12

    }

  ];

 const sortPersonsByFriendsQty= persons => {

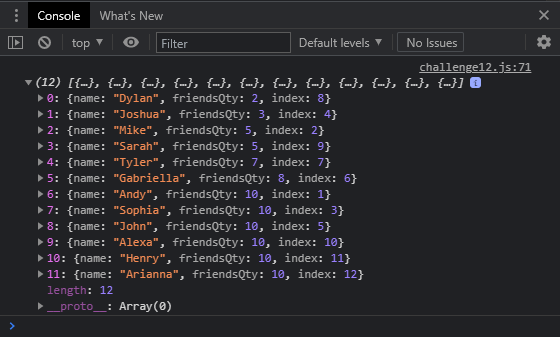
     return persons.sort((a,b)=> a.friendsQty-b.friendsQty);

 }

  sortPersonsByFriendsQty(persons);

  console.log(persons);

**Result**

****