Basic Data Structures

1. Use an Array to Store a Collection of Data

let yourArray = [2, "two", true, 17, "blue"]; // change this line

2. Access an Array’s Contents Using Bracket Notation

let myArray = ["a", "b", "c", "d"];

// change code below this line

myArray[1] = "e";

//change code above this line

console.log(myArray);

3. Add Items to an Array with push() and unshift()

function mixedNumbers(arr) {

// change code below this line

arr.unshift('I', 2, 'three');

arr.push(7, 'VIII', 9);

// change code above this line

return arr;

}

// do not change code below this line

console.log(mixedNumbers(['IV', 5, 'six']));

4. Remove Items from an Array with pop() and shift()

function popShift(arr) {

let popped = arr.pop(); // change this line

let shifted = arr.shift(); // change this line

return [shifted, popped];

}

// do not change code below this line

console.log(popShift(['challenge', 'is', 'not', 'complete']));

5. Remove Items Using splice()

function sumOfTen(arr) {

// change code below this line

arr.splice(2,2);

// change code above this line

return arr.reduce((a, b) => a + b);

}

// do not change code below this line

console.log(sumOfTen([2, 5, 1, 5, 2, 1]));

6. Add Items using splice();

function htmlColorNames(arr) {

// change code below this line

arr.splice(0, 2, "DarkSalmon", "BlanchedAlmond");

// change code above this line

return arr;

}

// do not change code below this line

console.log(htmlColorNames(['DarkGoldenRod', 'WhiteSmoke', 'LavenderBlush', 'PaleTurqoise', 'FireBrick']));

7. Copy Array Items Using slice()

slice()takes only 2 parameters — the first is the index at which to begin extraction, and the second is the index at which to stop extraction (extraction will occur up to, but not including the element at this index).

function forecast(arr) {

// change code below this line

return arr.slice(2,4);

}

// do not change code below this line

console.log(forecast(['cold', 'rainy', 'warm', 'sunny', 'cool', 'thunderstorms']));

8. Copy an Array with the Spread Operator

function copyMachine(arr, num) {

let newArr = [];

while (num >= 1) {

// change code below this line

newArr.push([...arr]);

// change code above this line

num--;

}

return newArr;

}

// change code here to test different cases:

console.log(copyMachine([true, false, true], 2));

9. Combine Arrays with the Spread Operator

function spreadOut() {

let fragment = ['to', 'code'];

let sentence = ['learning', ...fragment, 'is', 'fun']; // change this line

return sentence;

}

// do not change code below this line

console.log(spreadOut());

10. Check For The Presence of an Element With indexOf()

function quickCheck(arr, elem) {

// change code below this line

if(arr.indexOf(elem) === -1) {

return false;

} else {

return true;

}

// change code above this line

}

// change code here to test different cases:

console.log(quickCheck(['squash', 'onions', 'shallots'], 'mushrooms'));

11. Iterate Through All an Array’s Items Using For Loops

function filteredArray(arr, elem) {

let newArr = [];

// change code below this line

for (let i = 0; i < arr.length; i++) {

if (arr[i].indexOf(elem) === -1) {

newArr.push(arr[i]);

}

}

// change code above this line

return newArr;

}

// change code here to test different cases:

console.log(filteredArray([[3, 2, 3], [1, 6, 3], [3, 13, 26], [19, 3, 9]], 3));

12. Create complex multi-dimensional arrays

let myNestedArray = [

// change code below this line

['unshift', false, 1, 2, 3, 'complex', 'nested'],

[['loop', 'shift','deep'], [6, 7, 1000, 'method']],

['concat', false, true, 'spread', 'array'],

[

[

['mutate', 1327.98, 'splice'], ['slice', 'push', 'deeper']

],

[

[

['iterate', 1.3849, 7, '8.4876', 'arbitrary', 'depth','deepest']

]

]

]

// change code above this line

];

13. Add Key-Value Pairs to JavaScript Objects

let foods = {

apples: 25,

oranges: 32,

plums: 28

};

// change code below this line

foods['bananas'] = 13;

foods['grapes'] = 35;

foods['strawberries'] = 27;

// change code above this line

console.log(foods);

14. Modify an Object Nested Within an Object

let userActivity = {

id: 23894201352,

date: 'January 1, 2017',

data: {

totalUsers: 51,

online: 42

}

};

// change code below this line

userActivity.data.online = 45;

// change code above this line

console.log(userActivity);

15. Access Property Names with Bracket Notation

let foods = {

apples: 25,

oranges: 32,

plums: 28,

bananas: 13,

grapes: 35,

strawberries: 27

};

// do not change code above this line

function checkInventory(scannedItem) {

// change code below this line

return foods[scannedItem];

}

// change code below this line to test different cases:

console.log(checkInventory("apples"));

16. Use the delete Keyword to Remove Object Properties

let foods = {

apples: 25,

oranges: 32,

plums: 28,

bananas: 13,

grapes: 35,

strawberries: 27

};

// change code below this line

delete foods.oranges;

delete foods.plums;

delete foods.strawberries;

// change code above this line

console.log(foods);

17. Check if an Object has a Property

let users = {

Alan: {

age: 27,

online: true

},

Jeff: {

age: 32,

online: true

},

Sarah: {

age: 48,

online: true

},

Ryan: {

age: 19,

online: true

}

};

function isEveryoneHere(obj) {

// change code below this line

if(users.hasOwnProperty('Alan') && users.hasOwnProperty('Jeff') && users.hasOwnProperty('Sarah') && users.hasOwnProperty('Ryan')) {

return true;

} else {

return false;

}

// change code above this line

}

console.log(isEveryoneHere(users));

18. Iterate Through the Keys of an Object with a for …in Statement

let users = {

Alan: {

age: 27,

online: false

},

Jeff: {

age: 32,

online: true

},

Sarah: {

age: 48,

online: false

},

Ryan: {

age: 19,

online: true

}

};

function countOnline(obj) {

// change code below this line

let count = 0;

for (let user in obj) {

if (obj[user]['online'] === true) {

count++;

}

}

return count;

// change code above this line

}

console.log(countOnline(users));

19. Generate an Array of All Object Keys with Object.keys()

let users = {

Alan: {

age: 27,

online: false

},

Jeff: {

age: 32,

online: true

},

Sarah: {

age: 48,

online: false

},

Ryan: {

age: 19,

online: true

}

};

function getArrayOfUsers(obj) {

// change code below this line

return Object.keys(obj);

// change code above this line

}

console.log(getArrayOfUsers(users));

20. Modify an Array Stored in an Object

let user = {

name: 'Kenneth',

age: 28,

data: {

username: 'kennethCodesAllDay',

joinDate: 'March 26, 2016',

organization: 'freeCodeCamp',

friends: [

'Sam',

'Kira',

'Tomo'

],

location: {

city: 'San Francisco',

state: 'CA',

country: 'USA'

}

}

};

function addFriend(userObj, friend) {

// change code below this line

userObj.data.friends.push(friend);

return userObj.data.friends; //return the friends array

// change code above this line

}

console.log(addFriend(user, 'Pete'));