Amazing arrays –4 (37 coding exercises)

“The best yardstick for our progress is not other people, but ourselves.” — *C. Matakas*

4.1 Arrays in JavaScript

Exercises

1. What is the fundamental difference between an array and an object?

Answers

1. In an array items are ordered by indices. Whereas , objects represent a group of items in **key : value** pairs

4.2 Array declaration

4.2.1 Array literal notation

Exercises

1. What are the three ways to create arrays?

2. Create an empty array called `items` and initialize it with values of 5 different data types. What is the length of the array?

3. What method can be used to determine whether a given object is an array?

Answers

1. Array literal syntax, Array() constructur and Array.of() method

2. `items` array:

   ```
   let items = ['hello', {id: 1}, true, null, 10];
   ```

   The length of the array is 5 since it has has 5 items.

3. Determine whether a given object is an array:

```
Array.isArray(value)
```

4.2.2 Array() constructor

Exercises

1. Answer the following using the array constructor method:

   - Using an array constructor initialize an array called `employeeList` which will have `5` values
   - Add an item with the string value `‘Reno’` in the `employeeList` array at index 0
   - What values does the `employeeList` array contain? Try to guess without console logging

Answers

1. `employeeList`

```
let employeeList = new Array(5);
employeeList[0] = 'Reno';
["Reno", undefined, undefined, undefined, undefined]
```

4.2.3 Array.of()

Exercises

1. What is the difference between these two ways of creating arrays:

```
let array1 = Array.of(1, 2, 3);
let array2 = new Array(3);
```

Answers

1. The `array.of()` method creates a new array with the arguments passed to it as the new array’s values. Whereas the `Array()` constructor method will create a new array with 3 indices of undefined values:

```
console.log(array1);//[1, 2, 3]
console.log(array2);//[undefined, undefined, undefined]
```

4.2.3 Array values

Exercises

1. Consider the following array of objects:

```
let arrayObj = [{id:1, enrolled:true}, {id:2,
enrolled:true}, {id:3, enrolled:false}, {id:4,
enrolled:true}, {id:5, enrolled:false}];
```

    - Access the value of the `enrolled` property of the item at index `4`in the `arrayObj` array
    - Change its value to `true`

Answers

1.

```
arrayObj[4].enrolled

arrayObj[4].enrolled = true;
```

4.3 Array properties

4.3.1 Length property of an array

Exercises

1. What is the length of this array:

```
let employeeList = new Array(5);
```

2. Loop over the following array and log to the console the value of the array item at each index:

```
let arrayObj = [{id:1, enrolled:true}, {id:2,
enrolled:true}, {id:3, enrolled:false}, {id:4,
enrolled:true}, {id:5, enrolled:false}];
```

Answers

1. 5

2.

```
for(let i =0; i < arrayObj.length; i++){
  console.log(arrayObj[i])
}
```

4.3.2 Constructor property of an array

Exercises

1. How will you query the constructor of the following array?

```
let myArr =['map()', 'reduce()', 'filter()','forEach()'];
```

Answers

1.

```
console.log(myArr.constructor);
```

4.3.3 Prototype property of an array

Exercises

1. Is it good practice to add methods and properties to the prototype Array object? If not, why?

Answers

1. It is not advisable to add methods and properties to the prototype Array object. As this leaves open the possibility that a modification on the prototype might cause conflict with code from other JavaScript libraries or frameworks.

4.4 Array Methods

Exercises

1. Consider the following array and answer the question:

```
let myAlphabet = ['A', 'B', 'C', 'D','E','F', 'G'];
```

- What is the length of the array?

- Write a function called `myAlphabetLength` which `console.logs` the length of the array
- Within the function also use an `if-conditional` statement that checks if the number of items within the array is less than 4

2. Declare and initialize an array called `'Planets'` with 5 string values
   - `console.log` each item in the array
   - Also `console.log` the index in each iteration
3. Declare and initialize an array called `wowDataTypes`
   - The array must have 5 different data types
   - Iterate over the array and console.log each item in the array, its index and data type in the array
4. `console.log` each item in this array WITHOUT using a for loop

```
let myArr = [ 1, 2, 'One', true];
```

5. Loop over the 2 arrays and if there are any common occurrences/elements. Iif so `console.log` them

```
let student1Courses = ['Math', 'English', 'Programming'];

let student2Courses = ['Geography', 'Spanish',
'Programming'];
```

6. Compare the 2 arrays and find common food if any:

```
let food = ['Noodle', 'Pasta', 'Ice-cream'];

let food1 = ['Fries', 'Ice-cream', 'Pizza'];
```

7. Compare the 3 arrays and find any common elements:

```
let values1= ['Apple', 1, false];

let values2 = ['Fries', 2 ,true];

let values3 = ['Mars', 9, 'Apple'];
```

8. For each item in this array `console.log` the letters in each item

```
let furniture = ['Table', 'Chairs','Couch'];
```

9. Does the `push()` method modify/mutate the original array?

10. What will be logged?

```
let villagers =
['Coco','Merengue','Drago','Flip','Hazel','Rocket'];

villagers[2] = 'Poppy';

console.log(villagers);
```

11. Remove the letter `e` from the string `'icecream'`

12. `pop()` will mutate the original array’s length and return the array. True or False?

13. Using the `concat()` method, concat these 2 arrays and remove any extra occurrences of an element

```
let arr = [1,2,3];

let arr1 = [1,5,6];
```

14. Consider the following array of objects for a small indie bakery:

```
let bakery = [

  {

    cookie: 'oreo',

    calories:350

  },

  {

    cookie: 'fudge',

    calories:450

  },

  {

    cookie: 'butter',

    calories:700

  },

];
```

- Write some code so that when the `bakery` array is queried or console.logged it is now:

```
[[object Object] {
  calories: 350,
  cookie: "oreo",
  twoHelpings: 700
}, [object Object] {
  calories: 450,
  cookie: "fudge",
  twoHelpings: 900
}, [object Object] {
  calories: 700,
  cookie: "butter",
  twoHelpings: 1400
}]
```

15. Remove the first element from this array:

```
let amazingArray = [ {}, null, undefined, '' ];
```

16. What is the difference between `splice()` and `slice()`?

17. Splice the following array and remove the item `'new code'` from the `companies` array:

```
let companies = ['cultivating coders', 'purple moon',
'enterprise skills', 'new code', 'soft code'];
```

    - Slice the item `'cultivating coders'` and assign it to a variable declared as `techX`

    - Convert `techX` into an object (hint you will need the spread operator). This object is referenced by a variable called `techZ`

    - Assign a `students` property to `techZ` whose value is the number `8200`

    - Console.log the `techZ` object it should look like this:

```
Object {

  0: "cultivating coders",

  students: 8200

}
```

18. Have a look at the following code for `insects` and find if the `insects` object has an insect called `'Diptera'`

```
let insects =[
  {
  taxanomy: 'insecta',
  name: 'Archaeognatha',
  species: 513,
  exoskeleton: true
  },
    {
  taxanomy: 'insecta',
  name: 'Plecoptera',
  species: 3743,
  exoskeleton: true
  },
    {
  taxanomy: 'insecta',
  name: 'Thysanoptera',
  species: 5864,
  exoskeleton: true
  },
    {
  taxanomy: 'insecta',
  name: 'Trichoptera',
  species: 14391,
  exoskeleton: true
  },
    {
  taxanomy: 'insecta',
  name: 'Diptera',
  species:155477,
  exoskeleton: true
  },
];
```

19. Add the follow object to the `insects` object in question **#18**:

```
  taxanomy: 'insecta',
  name: 'Mantodea',
  species:2400 ,
  exoskeleton: true
```

20. Carrying on from question #19, the `insects` object which is as below. Query whether the `species` property in each object of the insect array has a value of greater than `1000`

```
let insects =[
  {
  taxanomy: 'insecta',
  name: 'Archaeognatha',
  species: 513,
  exoskeleton: true
  },
    {
  taxanomy: 'insecta',
  name: 'Plecoptera',
  species: 3743,
  exoskeleton: true
  },
    {
  taxanomy: 'insecta',
  name: 'Thysanoptera',
  species: 5864,
  exoskeleton: true
  },
    {
   taxanomy: 'insecta',
  name: 'Trichoptera',
  species: 14391,
  exoskeleton: true

  },
    {
  taxanomy: 'insecta',
  name: 'Diptera',
  species:155477,
  exoskeleton: true
  },
  {
taxanomy: 'insecta',
  name: 'Mantodea',
  species:2400 ,
  exoskeleton: true
  },
];
```

21. Consider the following array referenced by `let eshoppe` which is an array of objects for a single page shopping cart application being built:

```
let eshoppe = [{
    name: 'Pens',
    units: 403,
    price: '$1.99'
  },
    {
    name: 'Cotton socks',
    units: 432,
    price: '$3.99'
  },
    {
    name: 'Shirts',
    units: 1010,
    price: '$12.99'
  },
    {
    name: 'Stickers',
    units: 8200,
    price: '$1.99'
  },
    {
    name: 'Coffee mug',
    units: 2140,
    price: '$10.99'
  }];
```

- Sort the `eshoppe` array by price in ascending order (from the lowest to the highest price) as some users would like to see items sorted by price.

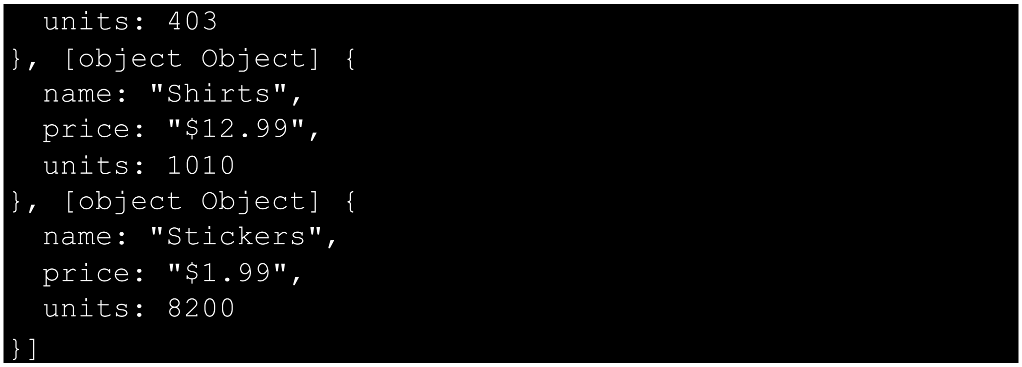
- Console.log the `eshoppe` array, you should see that the objects in the array are sorted by ascending order:

```
[[object Object] {
  name: "Pens",
  price: 1.99,
  units: 403
}, [object Object] {
  name: "Stickers",
  price: 1.99,
  units: 8200
}, [object Object] {
  name: "Cotton socks",
  price: 3.99,
  units: 432
}, [object Object] {
  name: "Coffee mug",
  price: 10.99,
  units: 2140
}, [object Object] {
  name: "Shirts",
  price: 12.99,
  units: 1010
}]
```

- Now sort the `eshoppe` array by alphabetical order (from A - Z). As sometimes users like to see items in alphabetical order.

- Console.log the `eshoppe` array, you should see that the objects in alphabetical order, like so:

```
[[object Object] {
  name: "Coffee mug",
  price: "$10.99",
  units: 2140
}, [object Object] {
  name: "Cotton socks",
  price: "$3.99",
  units: 432
}, [object Object] {
  name: "Pens",
  price: "$1.99",
```



```
  units: 403
}, [object Object] {
  name: "Shirts",
  price: "$12.99",
  units: 1010
}, [object Object] {
  name: "Stickers",
  price: "$1.99",
  units: 8200
}]
```

22. For the following array use the `fill()` method such that the array returned is `[199.99, 89.75, 10, 10, 8200.99, 79.95]`

```
let prices = [199.99, 89.75, 62.25, 13.99, 8200.99,
79.95];
```

23. For the same prices array in #22, find the index of the `8200.99` using the `findIndex()` method:

```
let prices = [199.99, 89.75, 62.25, 13.99, 8200.99,
79.95];
```

- Once you find the index, replace the value `8200.99` with the value 9900

24. First `sort()` and then `reverse()` the following array:

```
let items = ['Calculator', 'Laptop',
'Console','USB','Keyboard'];
```

25. Consider the following question from Toptal and try to work out what will be console.logged:

```
var arr1 = "john".split('');

var arr2 = arr1.reverse();

var arr3 = "jones".split('');

arr2.push(arr3);

console.log("array 1: length= " + arr1.length + " last="
+ arr1.slice(-1));

console.log("array 2: length=" + arr2.length + " last=" +
arr2.slice(-1));
```

26. The `reverse()` method reverses the contents of an array , thereby mutating the original array. Can you come up with a way of reversing the contents of the original array called `num1` without mutating it?

```
let num1 = [100, 818, 319000, 79];
```

Answers

1.

- The length of the `myAlphabet` array is 7, as there are 7 items in the array.
- Function to log the length of the array:
- If conditional to check if the number of items in the array is less than 4

```
function myAlphabetLength(arr) {
  console.log(arr.length);
}
myAlphabetLength(myAlphabet); //7
```

```
function myAlphabetLength(arr) {
  console.log(arr.length);
  if(arr.length < 4) {
    console.log('less than 4 items')
  } else {
    console.log('more than 4 items')
  }
}
myAlphabetLength(myAlphabet); //7 "more than 4 items"
```

2.

```
let planets = ['Earth', 'Mars', 'Jupiter', 'Venus',
'Pluto'];


for(let i = 0; i < planets.length; i++) {
  console.log(planets[i] + ' ' + i);
}
```

3.

```
let wowDataTypes = [true, 'a', 100, {greeting: 'hello'}, null];

for(let i = 0; i < wowDataTypes.length; i++) {
  console.log(`${wowDataTypes[i]} ${i} ${typeof(wowDataTypes[i])}`)
}
```

4.

```
console.log(myArr); // [1, 2, "One", true]
```

5.

```
let student1Courses = ['Math', 'English', 'Programming'];
let student2Courses = ['Geography', 'Spanish', 'Programming'];

for(let i = 0; i < student1Courses.length; i++) {
  for(let k = 0; k < student2Courses.length; k++) {
    if(student1Courses[i] === student2Courses[k]){
      console.log(student1Courses[i])
    }
  }
}
```

```
"Programming"
```

6.

```
let food = ['Noodle', 'Pasta', 'Ice-cream'];
let food1 = ['Fries', 'Ice-cream', 'Pizza'];


for(let i = 0; i < food.length; i++) {
  for(let k = 0; k < food1.length; k++) {
    if(food[i] === food1[k]){
      console.log(food[i])
    }
  }
}
```

```
"Ice-cream"
```

7.

```
let combinedValues = [values1, values2, values3].flat();
```

8.

```
for(let i = 0; i < furniture.length; i++){
  for(let k = 0; k < furniture[i].length; k++) {
    console.log(furniture[i][k])
  }
}
```

9. Yes

10. The value at `villagers[2]` is replaced with the new value.

```
["Coco", "Merengue", "Poppy", "Flip", "Hazel", "Rocket"]
```

11.

```
let icecream = "icecream";
let newIcecream = icecream.split('e').join('');
```

```
"iccram"
```

`split()` method will truncate the characters in the string:

```
let newIcecream = icecream.split('e') // ["ic", "cr",
"am"]
```

And then `join()` will join the characters in the array without any separator:

```
let newIcecream = icecream.split('e').join('');
```

12. True
13.

```
let arr2 = arr.concat(arr1);

let noDuplicate =[];

arr2.forEach(function(item){

 if(!noDuplicate.includes(item)){

   noDuplicate.push(item)

 }

});

console.log(noDuplicate);
```

14.

```
bakery.forEach(function(item){

  return(item.twoHelpings = item.calories * 2);

});

console.log(bakery);
```

15.

```
amazingArray.shift();
```

```
[null, undefined, ""]
```

16. The `slice()` method will slice the array values at the specified index and return the sliced values. It does not mutate the original array. Instead it returns a new array containing the sliced values. This method takes two arguments:

- The start index at which to slice from

- The end index at which to end the slice at. The value at the end index will not be included

- The `splice()` method can also be used to delete items from an array. The `splice()` method takes three arguments:

  The first argument specifies the index at which the items should be deleted in the array

  The second argument specifies the number of items that should be removed from the array

  The third argument specifies the elements to be added to the array

17.

```
let companies = ['cultivating coders', 'purple moon',
'enterprise skills', 'new code', 'soft code'];
companies.splice(3, 1);
let techX = companies.slice(0, 1)
let techZ = {...techX}
techZ.students = 8200;
console.log(techZ);
```

18.

```
let diptera = insects.some(function(item){
 return(item.name === 'Diptera')

  });
console.log(diptera); //true
```

19.

```
insects.push({
  taxanomy: 'insecta',
  name: 'Mantodea',
  species:2400 ,
  exoskeleton: true
});
```

20.

```
let speciesNumberCheck = insects.every(function(item){
  return(item.species > 1000);
});
console.log(speciesNumberCheck); //false
```

21. Sort by price

```
eshoppe.forEach(function(item){
  item.price = item.price.substring(1);
  item.price = parseFloat(item.price)
  console.log(item.price)
});
eshoppe.sort(function(a,b){
  return(a.price - b.price)
});
```

Sort by alphabetical order:

```
eshoppe.sort(function(a,b){
  return(a.name > b.name)
});
```

22. `fill()` method:

```
prices.fill(10, 2, 4);
```

23. Find the index of `8200.99`

```
let index = prices.findIndex(function(item){
return(item === 8200.99);
});
console.log(index); //4
```

Replace `8200.99` with `9900`

```
prices[index] = 9900;

console.log(prices);
```

```
[199.99, 89.75, 62.25, 13.99, 8900, 79.95]
```

24.

```
items.sort();

items.reverse();

console.log(items);
```

```
["USB", "Laptop", "Keyboard", "Console", "Calculator"]
```

25.

```
"array 1: length= 5 last=j,o,n,e,s"

"array 2: length=5 last=j,o,n,e,s"
```

https://www.toptal.com/javascript/interview-questions

26.

```
let num2 = num1.slice().reverse();

console.log(num2);

console.log(num1);
```

```
[79, 319000, 818, 100]

[100, 818, 319000, 79]
```