ARRAYS SOME

Array.prototype.some()

The some() method tests whether at least one element in the array passes the test implemented by the provided function. If returns true if in the array, it finds an elemeNt for which the provided function returns true; otherwise it returns false. It doesn't modify the array.

```
1
2   // some returns true if someone in the array fullfills the criteria
3   let arr = [3, 9, 15, 17, 21];
4   let isThereAnyEvenElement = arr.some(function(v, i){
5       console.log(v + " - " + i);
6       if(v % 2 == 0){
7           return true;
8       }else {
9           return false;
10      }
11  })
12  console.log(isThereAnyEvenElement);
13
14  let arr2 = [3, 9, 16, 17, 21];
15  let isThereAnyEvenElement2 = arr2.some(function(v, i){
16      console.log(v + " - " + i);
17      if(v % 2 == 0){
18          return true;
19      }else {
20          return false;
21      }
22  })
23
24  console.log(isThereAnyEvenElement2);
```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

```
→  Lecture_36 git:(main) ✗ node 2_ArraySomeDemo.js
3 - 0
9 - 1
15 - 2
17 - 3
21 - 4
false
3 - 0
9 - 1
16 - 2
true
```

Syntax

array.some(function(value, index, arr), this)

Parameters

Parameter	Description
function	Required. A function to run for each array element.

Function parameters:

value	Required. The value of the current element.
index	Optional. The index of the current element.
arr	Optional. The array the current element belongs to.
this	Optional. Default undefined. A value passed to the function to be used as its "this" value.

Return Value

Type	Description
A boolean	true if any of the array elements pass the test, otherwise false.

https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array/some

CUSTOM SOME

```
Array.prototype.mySome = function(cb){
    let oarr = this;
    for(let i = 0; i < oarr.length; i++){
        let v = oarr[i];
        let rv = cb(v, i, oarr);
        if(rv == true){
            return true;
        }
    }
    return false;
}

let arr = [
    {name: "A", age: 14, gender: "M"},
    {name: "B", age: 34, gender: "M"},
    {name: "C", age: 34, gender: "F"},
    {name: "D", age: 44, gender: "F"},
    {name: "E", age: 44, gender: "M"},
    {name: "I", age: 28, gender: "F"},
    {name: "G", age: 36, gender: "M"},
    {name: "H", age: 47, gender: "F"}
];

// some takes each value 1 by 1 nd returns a true if any cb returns true
// return false only if all cb return false

// Is there a valid candidate (F and between 20 and 30)
let isThereAnyValidCandidate = arr.mySome(function(v, i, oarr){
    if(v.gender == 'F' && v.age >= 20 && v.age <= 30){
        return true;
    } else {
        return false;
    }
})
```

```
 →  Lecture_36 git:(main) ✗ node 1_ArraySomeDemo.js
true
 →  Lecture_36 git:(main) ✗
```

ARRAYS EVERY

Array.prototype.every()

The every() method tests whether all elements in the array pass the test implemented by the provided function. It return a Boolean value.

https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array/ever

Syntax

```
array.every(function(currentValue, index, arr), thisValue)
```

Parameters

Parameter	Description
function()	Required. A function to be run for each element in the array.
currentValue	Required. The value of the current element.
index	Optional. The index of the current element.
arr	Optional. The array of the current element.
thisValue	Optional. Default undefined. A value passed to the function as its this value.

Return Value

Type	Description
A boolean	true if all elements pass the test, otherwise false.

Array every demo

```
const isBelowThreshold = (currentValue) => currentValue < 40;

const array1 = [1, 30, 39, 29, 10, 13];

console.log(array1.every(isBelowThreshold));
// expected output: true
```

CUSTOM EVERY

```
Array.prototype.myEvery = function(cb){
    let oarr = this;

    for(let i = 0; i < oarr.length; i++){
        let v = oarr[i];
        let rv = cb(v, i, oarr);

        if(rv == false){
            return false;
        }
    }

    return true;
}

let arr = [
    {name: "A", age: 14, gender: "M"},
    {name: "B", age: 34, gender: "M"},
    {name: "C", age: 24, gender: "F"},
    {name: "D", age: 24, gender: "F"},
    {name: "E", age: 44, gender: "M"},
    {name: "I", age: 28, gender: "F"},
    {name: "G", age: 36, gender: "M"},
    {name: "H", age: 27, gender: "F"}
];


// if every callback return true then "Every" return true otherwise false
// Are all Female Candidates Valid
let allFemaleCandidatesValid = arr.filter(c => c.gender == 'F').myEvery(fc => fc.age >= 20 && fc.age <= 30);
console.log(allFemaleCandidatesValid);
```

```
PROBLEMS  OUTPUT  TERMINAL  DEBUG CONSOLE                zsh - Lecture_36

→  Lecture_36 git:(main) ✗ node 3_ArrayEveryDemo.js
true
```

ARRAY FIND

Array.prototype.find()

The find() method returns the value of the first element in the provided array that satisfies the provided testing function. If no values satisfy the testing function, undefined is returned.

Syntax

```
array.find(function(currentValue, index, arr),thisValue)
```

Parameters

function()	Required. A function to run for each array element.
currentValue	Required. The value of the current element.
index	Optional. The index of the current element.
arr	Optional. The array of the current element.
thisValue	Optional. Default undefined. A value passed to the function as its this value.

Return Value

Type	Description
A value	The value of the first element that pass the test.

Otherwise it returns undefined.

Array find demo

```
const array1 = [5, 12, 8, 130, 44];

const found = array1.find(element => element > 10);

console.log(found);
// expected output: 12
```

CUSTOM FIND

```
Array.prototype.myFind = function(cb){
    let oarr = this;
    for(let i = 0; i < oarr.length; i++){
        let v = oarr[i];
        let rv = cb(v, i, oarr);

        if(rv == true){
            return v;
        }
    }
    return undefined;
}
let arr = [
    {name: "A", age: 14, gender: "M"},
    {name: "B", age: 34, gender: "M"},
    {name: "C", age: 34, gender: "F"},
    {name: "D", age: 34, gender: "F"},
    {name: "E", age: 44, gender: "M"},
    {name: "I", age: 38, gender: "F"},
    {name: "G", age: 36, gender: "M"},
    {name: "H", age: 47, gender: "F"}
];
// find gives value against first true, if there is no true then undefined
// First valid candidate (F and between 20 and 30)
let fvc = arr.myFind(function(v, i, oarr){
    if(v.gender == 'F' && v.age >= 20 && v.age <= 30){
        return true;
    } else {
        return false;
    }
})
if(fvc != undefined){
    console.log(fvc.name + "@" + fvc.age + "#" + fvc.gender);
} else {
    console.log("Not Found")
}
```

ARRAY FINDINDEX

Array.prototype.findIndex()

The findIndex() method returns the index of the first element in the array that satisfies the provided testing function. Otherwise, it returns -1, indicating that no element passed the test.

Syntax

```
array.findIndex(function(currentValue, index, arr), thisValue)
```

Parameters

Parameter	Description
function()	Required. A function to be run for each array element.
currentValue	Required. The value of the current element.
index	Optional. The index of the current element.
arr	Optional. The array of the current element.
thisValue	Optional. Default undefined. A value passed to the function as its this value.

Return Value

Type	Description
A number	The index of the first element that passes the test.
Otherwise -1	

Array findIndex demo

```
const array1 = [5, 12, 8, 130, 44];

const isLargeNumber = (element) => element > 13;

console.log(array1.findIndex(isLargeNumber));
// expected output: 3
```

CUSTOM FINDINDEX

```
Array.prototype.myFindIndex = function(cb){
    let oarr = this;
    for(let i = 0; i < oarr.length; i++){
        let v = oarr[i];
        let rv = cb(v, i, oarr);
        if(rv == true){
            return i;
        }
    }
    return -1;
}
let arr = [
    {name: "A", age: 14, gender: "M"},
    {name: "B", age: 34, gender: "M"},
    {name: "C", age: 24, gender: "F"},
    {name: "D", age: 34, gender: "F"},
    {name: "E", age: 44, gender: "M"},
    {name: "I", age: 38, gender: "F"},
    {name: "G", age: 36, gender: "M"},
    {name: "H", age: 47, gender: "F"}
];

// find gives value against first true, if there is no true then undefined
// First valid candidate (F and between 20 and 30)
let fvci = arr.myFindIndex(function(v, i, oarr){
    if(v.gender == 'F' && v.age >= 20 && v.age <= 30){
        return true;
    } else {
        return false;
    }
})

console.log(fvci);
```

```
→ Lecture_36 git:(main) ✗ node 5_ArrayFindIndexDemo.js
2
→ Lecture_36 git:(main) ✗
```