**Tên nhóm: Nhóm 3**

Thành viên

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Họ và tên** | **Email** | **SDT** |
| 1 | Trần Thị Xuân Hoàng | xuanhoang2937@gmail.com | 0978532509 |
| 2 | Ngô Xuân Chiến | xuanchienhut@gmail.com | 0355062707 |

Tìm hiểu :

1. Tìm hiểu array, map, set

* Array

**Khởi tạo**

**let** alphas = ["1","2","3","4"];

Lấy giá trị cơ bản

console.log(alphas[0]);  
console.log(alphas[1]);

Kết quả: 1, 2

**for**(**var** i = 0;i<alphas.length;i++) {  
 console.log(alphas[i])  
}

Kết quả: 1, 2, 3, 4

**Hàm concat (gộp mảng)**

**let** alphas = ["1","2","3","4"];

**var** names = **new** Array("Mary","Tom","Jack","Jill")  
console.log(alphas.concat(names))

Kết quả

[ '1', '2', '3', '4', 'Mary', 'Tom', 'Jack', 'Jill' ]

//Array.prototype.find  
**var** numbers = [1, 2, 3];  
**var** oddNumber = numbers.find((x) => x % 2 == 1);  
console.log(oddNumber); // value: 1

Trả về giá trị đầu tiên thõa điều kiện, nếu không tồn tại trả về “null”

//Array.prototype.findIndex  
**var** numbers = [1, 2, 3];  
**var** oddNumber = numbers.findIndex((x) => x % 2 == 1);  
console.log(oddNumber); //value = 0

Trả về vị trí đầu tiên thõa điều kiện, nếu không tồn tại trả về -1 “false”

//Array.prototype.entries  
**var** numbers = [1, 2, 3];  
**var** val = numbers.entries();  
console.log(val.next().value);  
console.log(val.next().value);  
console.log(val.next().value);  
console.log(val.next().value);

//Value:

[ 0, 1 ]

[ 1, 2 ]

[ 2, 3 ]

undefined

Trả về các mảng trong đó chứa cặp giá trị trả về [vị trí, giá trị], ngoài mảng trả về “null”

//Array.from  
**for** (**let** i **of** Array.from('hello')) {  
 console.log(i)  
}

//value:

h

e

l

l

o

Trả về 1 mảng chứa các ký tự từ chuỗi (Tạo mảng từ 1 đối tượng)

//Array.prototype.keys ()  
console.log(Array.from(['a', 'b'].keys()))

//value:

[ 0, 1 ]

Trả về vị trí các phần tử

* Map

Khởi tạo

**var** map = **new** Map([[1,2],[2,3]]);  
console.log(map)

//value:

Map { 1 => 2, 2 => 3 }

**var** map = **new** Map([[1,2],[2,3]]);  
console.log(map)

//value:

Map { 2 => 3 }

Tạo 1 mảng với từng cặp key-value

Nếu trùng key thì sẽ bị set lại

Duyệt:

**var** map = **new** Map([[1,2],[2,3]]);  
**for** (**const** item **of** map){  
 console.log(item);  
}

//Value:

[ 1, 2 ]

[ 2, 3 ]

**for** (**const** [key,value] **of** map){  
 console.log(`key: ${key}, value: ${value}`);  
}

//value:

key: 1, value: 2

key: 2, value: 3

**//get**

**var** map = **new** Map([[2,2],[2,3]]);  
console.log(map.get(2))

//value: 3

console.log(map.get(1))

//value: undefined

Lấy value theo key, không tồn tại trả về null

**//has**

**var** map = **new** Map([[2,2],[2,3]]);  
console.log(map.get(1)) //value: false  
console.log(map.has(2)) //value: true

Kiểm tra key tồn tại hay không, có trả về true, không trả về false

**//add element (set)**

**var** map = **new** Map([[1,2],[2,3]]);  
map.set(4,5);  
console.log(map)

//value: Map { 1 => 2, 2 => 3, 4 => 5 }

**var** map = **new** Map([[1,2],[2,3]]);  
map.set(4,5);  
console.log(map) //value: Map { 1 => 2, 2 => 3, 4 => 5 }  
console.log(map.delete(1)) //value: true  
console.log(map) //value: Map { 2 => 3, 4 => 5 }  
console.log(map.delete(1)) //value: false  
console.log(map) //value: Map { 2 => 3, 4 => 5 }

Xóa phần từ theo key, key tồn tại trả về true, key không tồn tại trả về false

**var** map = **new** Map([[1,2],[2,3]]);  
console.log(map) //value: Map { 1 => 2, 2 => 3 }  
map.clear();  
console.log(map) //value: Map { }

clear: xóa toàn bộ key-value trong map

* Set

Khởi tạo

**var** emptySet = **new** Set();  
console.log(emptySet) //value: Set { }  
**var** exampleSet = **new** Set([1,2,3]);  
console.log(exampleSet) //value: Set { 1, 2, 3 }

**var** exampleSet = **new** Set([1,2,3]);  
console.log(exampleSet[2]); // value: undefined

Set không chứa key, chỉ chứa value nên khi lấy giá trị tại vị trí index = 2 => “undefined”

//has  
**var** exampleSet = **new** Set([1,2,3]);  
console.log(exampleSet.has(0)); //value: false  
console.log(exampleSet.has(1)); //value: true

Kiểm tra sự tồn tại 1 giá trị, tồn tại trả về true, không tồn tại trả về false

**//add element**

**var** exampleSet = **new** Set([1,2,3]);  
exampleSet.add(2)  
console.log(exampleSet); //value: Set { 1, 2, 3 }  
exampleSet.add(4)  
console.log(exampleSet); //value: Set { 1, 2, 3, 4 }

Thêm giá trị vào set, chỉ có thể thêm những giá trị chưa tồn tại, giá trị đã tồn tại sẽ không được thêm lại

**//delete element**

**var** exampleSet = **new** Set([1,2,3]);  
console.log(exampleSet.delete(2)); //value: true  
console.log(exampleSet); //value: Set { 1, 3 }  
console.log(exampleSet.delete(2)); //value: false  
console.log(exampleSet); //value: Set { 1, 3 }

Xóa giá trị trong set: xóa giá trị tồn tại trong set trả về true, xóa giá trị không tồn tại trong set trả về false

**//clear**

**var** exampleSet = **new** Set([1,2,3]);  
exampleSet.clear();  
console.log(exampleSet); //value: Set { }

clear: Xóa toàn bộ giá trị trong set

1. Giao diện quản lý kho hàng



