Arrays and Collections

Array Extensions

ArrayBuffer and Typed Arrays

DataView and Endianness

Map and WeakMap

Set and WeakSet

Subclassing

Array Extensions

```
let salaries = Array(90000);
console.log(salaries.length);
```

Question

What shows in the console?

Answer

90000

```
let salaries = Array.of(90000);
console.log(salaries.length);
```

Question

What shows in the console?

Answer

1

```
let amounts = [800, 810, 820];
let salaries = Array.from(amounts, v => v+100 );
console.log(salaries);
```

Q What shows in the console?

A [900,910,920]

```
let amounts = [800, 810, 820];
let salaries = Array.from(amounts, function (v) {
    return v + this.adjustment;
}, { adjustment: 50 });
console.log(salaries);
```

Q What shows in the console?

A [850,860,870]

```
let amounts = [800, 810, 820];
let salaries = Array.from(amounts, v => v + this.adjustment,
    { adjustment: 50 });
console.log(salaries);
```

Q What shows in the console?

A [NaN, NaN, NaN]

```
let salaries = [600, 700, 800];
salaries.fill(900);
console.log(salaries);
```

Question

What shows in the console?

Answer

[900, 900, 900]

```
let salaries = [600, 700, 800];
salaries.fill(900, 1);
console.log(salaries);
```

Question

What shows in the console?

Answer

[600, 900, 900]

```
let salaries = [600, 700, 800];
salaries.fill(900, 1, 2);
console.log(salaries);
```

Question

What shows in the console?

Answer

[600, 900, 800]

```
let salaries = [600, 700, 800];
salaries.fill(900, -1);
console.log(salaries);
```

Question

What shows in the console?

Answer

[600, 700, 900]

```
let salaries = [600, 700, 800];
let result = salaries.find(value => value >= 750);
console.log(result);
```

Q What shows in the console?

A 800

```
let salaries = [600, 700, 800];
let result = salaries.find(value => value >= 650);
console.log(result);
```

Q What shows in the console?

A 700

```
let salaries = [600, 700, 800];
let result = salaries.findIndex(function (value, index, array) {
    return value == this;
}, 700);
console.log(result);
```

Q What shows in the console?

A 1

```
let salaries = [600, 700, 800];
salaries.copyWithin(2, 0);
console.log(salaries);
```

Question

What shows in the console?

Answer

[600, 700, 600]

```
let ids = [1, 2, 3, 4, 5];
ids.copyWithin(0, 1);
console.log(ids);
```

Question

What shows in the console?

Answer

[2, 3, 4, 5, 5]

```
let ids = [1, 2, 3, 4, 5];
ids.copyWithin(3, 0, 2);
console.log(ids);
```

Question

What shows in the console?

Answer

[1, 2, 3, 1, 2]

```
let ids = ['A', 'B', 'C'];
console.log(...ids.entries());
```

Question

What shows in the console?

Answer

[0,"A"], [1,"B"], [2,"C"]

```
let ids = ['A', 'B', 'C'];
console.log(...ids.keys());
```

Question

What shows in the console?

Answer

0 1 2

```
let ids = ['A', 'B', 'C'];
console.log(...ids.values());
```

Question

What shows in the console?

Answer

A B C

ArrayBuffer and Typed Arrays

```
let buffer = new ArrayBuffer(1024);
console.log(buffer.byteLength);
```

Question

What shows in the console?

Answer

1024

```
let buffer = new ArrayBuffer(1024);
buffer[0] = 0xff;
console.log(buffer[0]);
```

Question

What shows in the console?

Answer

255

Typed Arrays

Int8Array()
Uint8Array()
Uint8ClampedArray()

Int16Array()
Uint16Array()

Int32Array()
Uint32Array()

Float32Array()
Float64Array()

```
let buffer = new ArrayBuffer(1024);
let a = new Int8Array(buffer);
a[0] = 0xff;
console.log(a[0]);
```

Question

What shows in the console?

Answer

-1

```
let buffer = new ArrayBuffer(1024);
let a = new Uint8Array(buffer);
a[0] = 0xff;
console.log(a[0]);
```

Question

What shows in the console?

Answer

255

```
let buffer = new ArrayBuffer(1024);
let a = new Uint8ClampedArray(buffer);
a[0] = -12;
console.log(a[0]);
```

Question

What shows in the console?

Answer

0

```
let buffer = new ArrayBuffer(1024);
let a = new Uint8Array(buffer);
let b = new Uint16Array(buffer);
a[0] = 1;
console.log(b[0]);
```

Question

What shows in the console?

Answer

1

```
let buffer = new ArrayBuffer(1024);
let a = new Uint8Array(buffer);
let b = new Uint16Array(buffer);
a[1] = 1;
console.log(b[0]);
```

Question

What shows in the console?

Answer

256

DataView and Endianness

```
let buffer = new ArrayBuffer(1024);
let dv = new DataView(buffer);
console.log(dv.byteLength);
```

Question

What shows in the console?

Answer

1024

```
let buffer = new ArrayBuffer(1024);
let dv = new DataView(buffer, 0, 32);
console.log(dv.byteLength);
```

Question

What shows in the console?

Answer

32

```
let buffer = new ArrayBuffer(1024);
let dv = new DataView(buffer);
dv.setUint8(0, 1);
console.log(dv.getUint16(0));
```

Question

What shows in the console?

Answer

256

```
let buffer = new ArrayBuffer(1024);
let dv = new DataView(buffer);
dv.setUint8(0, 1);
console.log(dv.getUint16(0, true));
```

Question

What shows in the console?

Answer

1

Map and WeakMap

```
let employee1 = { name: 'Jake' };
let employee2 = { name: 'Janet' };

let employees = new Map();
employees.set(employee1, 'ABC');
employees.set(employee2, '123');

console.log(employees.get(employee1));
```

Question

What shows in the console?

Answer

ABC

```
let employee1 = { name: 'Jake' };
let employee2 = { name: 'Janet' };

let employees = new Map();
employees.set(employee1, 'ABC');
employees.set(employee2, '123');

console.log(employees.size);
```

Question

What shows in the console?

Answer

2

```
let employee1 = { name: 'Jake' };
let employee2 = { name: 'Janet' };

let employees = new Map();
employees.set(employee1, 'ABC');
employees.set(employee2, '123');

employees.delete(employee2);
console.log(employees.size);
```

Question

What shows in the console?

Answer

1

```
let employee1 = { name: 'Jake' };
let employee2 = { name: 'Janet' };

let employees = new Map();
employees.set(employee1, 'ABC');
employees.set(employee2, '123');

employees.clear();
console.log(employees.size);
```

Question

What shows in the console?

Answer

0

```
let employee1 = { name: 'Jake' };
let employee2 = { name: 'Janet' };
let arr = [
    [employee1, 'ABC'],
    [employee2, '123']
];

let employees = new Map(arr);

console.log(employees.size);
```

Question

What shows in the console?

Answer

2

```
let employee1 = { name: 'Jake' };
let employee2 = { name: 'Janet' };
let arr = [
    [employee1, 'ABC'],
    [employee2, '123']
];

let employees = new Map(arr);

console.log(employees.has(employee2));
```

Question

What shows in the console?

Answer

true

```
let employee1 = { name: 'Jake' };
let employee2 = { name: 'Janet' };
let arr = [
    [employee1, 'ABC'],
    [employee2, '123']
];

let employees = new Map(arr);

let list = [...employees.values()];
console.log(list);
```

Question

What shows in the console?

Answer

['ABC', '123']

```
let employee1 = { name: 'Jake' };
let employee2 = { name: 'Janet' };
let arr = [
    [employee1, 'ABC'],
    [employee2, '123']
];

let employees = new Map(arr);

let list = [...employees.entries()];
console.log(list[0][1]);
```

Question

What shows in the console?

Answer

ABC

```
let employee1 = { name: 'Jake' };
let employee2 = { name: 'Janet' };

let employees = new WeakMap([
    [employee1, 'ABC'],
    [employee2, '123']
]);

employee1 = null;
// wait for GC cycle

console.log(employees.size);
```

Question

What shows in the console?

Answer

undefined
(but the size is probably 1)

Set and WeakSet

```
let perks = new Set();

perks.add('Car');
perks.add('Super Long Vacation');

console.log(perks.size);
```

Question

What shows in the console?

Answer

2

```
let perks = new Set();

perks.add('Car');
perks.add('Super Long Vacation');
perks.add('Car');

console.log(perks.size);
```

Question

What shows in the console?

Answer

2

```
let perks = new Set([
        'Car',
    '10 Weeks Vacation',
    'Jet'
]);

console.log(perks.size);
```

Question

What shows in the console?

Answer

3

```
let perks = new Set([
        'Car',
    '10 Weeks Vacation',
    'Jet'
]);

let newPerks = new Set(perks);
console.log(newPerks.size);
```

Question

What shows in the console?

Answer

3

```
let perks = new Set([
        'Car',
    '10 Weeks Vacation',
    'Jet'
]);

console.log(perks.has('Jet'));
console.log(perks.has('Cool Hat'));
```

Question

What shows in the console?

Answer

true
false

```
let perks = new Set(['Car', 'Jet']);

console.log(...perks.keys());
console.log(...perks.values());
console.log(...perks.entries());
```

Question

What shows in the console?

Answer

Car Jet
Car Jet
Car,Car Jet,Jet

```
let perks = new Set([
    { id: 800 },
    { id: 800 }
]);

console.log(perks.size);
```

Question

What shows in the console?

Answer

2

```
let perks = new Set([
    1,
    '1'
]);

console.log(perks.size);
```

Question

What shows in the console?

Answer

2

WeakSet

```
let perks = new WeakSet([1, 2, 3]);

console.log(perks.size);
```

Question

What shows in the console?

Answer

Runtime Error: WeakSet.prototype.add: 'key' is not an object

```
let p1 = { name: 'Car' };
let p2 = { name: 'Jet' };
let perks = new WeakSet([p1, p2]);

console.log(perks.size);
```

Question

What shows in the console?

Answer

undefined

```
let p1 = { name: 'Car' };
let p2 = { name: 'Jet' };
let perks = new WeakSet([p1, p2]);

console.log(perks.has(p1));
```

Question

What shows in the console?

Answer

true

```
let p1 = { name: 'Car' };
let p2 = { name: 'Jet' };
let perks = new WeakSet([p1, p2]);

p1 = null;

console.log(perks.has(p1));
```

Question

What shows in the console?

Answer

false

Subclassing

```
class Perks extends Array {
}

let a = Perks.from([5, 10, 15]);

console.log(a instanceof Perks);
```

Question

What shows in the console?

Answer

true

```
class Perks extends Array {
}

let a = Perks.from([5, 10, 15]);

console.log(a.length);
```

Question

What shows in the console?

Answer

3

```
class Perks extends Array {

}

let a = Perks.from([5, 10, 15]);
let newArray = a.reverse();

console.log(newArray instanceof Perks);
```

Question

What shows in the console?

Answer

true

```
class Perks extends Array {

}

let a = Perks.from([5, 10, 15]);
let newArray = a.reverse();

console.log(newArray instanceof Array);
```

Question

What shows in the console?

Answer

true

```
class Perks extends Array {
    sum() {
        let total = 0;
        this.map(v => total += v);
        return total;
    }
}

let a = Perks.from([5, 10, 15]);
console.log(a.sum());
```

Question

What shows in the console?

Answer

30

Summary

Array Extensions

```
let amounts = [800, 810, 820];
let salaries = Array.from(amounts, function (v) {
    return v + this.adjustment;
}, { adjustment: 50 });
console.log(salaries);
```

Summary

ArrayBuffer and Typed Arrays

```
let buffer = new ArrayBuffer(1024);
buffer[0] = 0xff;
console.log(buffer[0]);
```

Summary

DataView and Endianness

```
let buffer = new ArrayBuffer(1024);
let dv = new DataView(buffer);
dv.setUint8(0, 1);
console.log(dv.getUint16(0, true));
```

Summary

Map and WeakMap

```
let employee1 = { name: 'Jake' };
let employee2 = { name: 'Janet' };
let arr = [
    [employee1, 'ABC'],
    [employee2, '123']
];

let employees = new Map(arr);

console.log(employees.size);
```

Summary

Set and WeakSet

```
let perks = new Set([
    'Car',
    '10 Weeks Vacation',
    'Jet'
]);

console.log(perks.has('Jet'));
console.log(perks.has('Cool Hat'));
```

Summary

Subclassing

```
class Perks extends Array {
}

let a = Perks.from([5, 10, 15]);

console.log(a.length);
```