**\NỘI DUNG ĐỊNH HƯỚNG VỀ MẢNG**

**I. Hiểu bản chất và thao tác cơ bản với mảng**

1. **Mảng là gì? Tại sao lại cần dùng mảng trong lập trình?**

Mảng là một nơi chứa tập các biến. Ta dùng mảng để có thể gọi hoặc sắp xếp các biến dễ dàng.

**2. Làm thế nào để khai báo một mảng trong JavaScript?\**

Có 2 cách khai báo mảng. Dùng ngoặc vuông

Let x = […,…]

Let x = new Array(…,…,…)

**3. Làm sao để truy cập phần tử đầu tiên, phần tử cuối cùng, hoặc phần tử ở vị trí bất kỳ?**

Phần tử đầu tiền x[0]

Phần tử cuối x[ x.length -1]

Phần tử thứ n x[ n-1]

**4. Làm sao để biết được mảng có bao nhiêu phần tử?**

Console.log(x.length)= n

**5. Điều gì xảy ra nếu truy cập phần tử ở chỉ số không tồn tại?**

Undifined

**6. Có thể lưu các kiểu dữ liệu khác nhau trong cùng một mảng không?**

Có ví dụ x = [1,2,”Táo”]

**7. So sánh giữa let arr = [] và let arr = new Array() – nên dùng cách nào?**

Let arr dễ dàng tạo khi có thông tin các phần tử

New Array() tạo khi muốn mảng có phần tử xác định ví dụ: newArray (5). Tạo 5 phần tử rỗng

**II. Duyệt mảng và xử lý từng phần tử**

**8. Những cách nào có thể dùng để duyệt qua mảng? (gợi ý: for, for...of, forEach, …)**

- for (i=0; i< x.length ; i++)

- for ( let n of x){

console.log(n)}

x.forEach(function(n,index){

console.log(index+ “:”+n)}

**9. Khi nào nên dùng for, khi nào nên dùng forEach hay map()?**

- Dùng for khi cần dùng chỉ số i

- Dùng forEach khi muốn thao tác với từng phần tử rõ rang

- Dùng map() khi muốn tạo một mảng mới từ mảng cũ

**10. Làm sao để tính tổng các phần tử trong một mảng số?**

Let sum = 0

For (i=0; i<x.length ; i++){

Sum+= number[i]

**11. Làm sao để đếm số phần tử thỏa mãn một điều kiện trong mảng?**

For (i=0; i<x.length ; i++){

If (..){

Count++;}

}

Console.log(count)

**12. Làm sao để tìm phần tử lớn nhất hoặc nhỏ nhất trong mảng?**

Let max = x[0]

For (i=0; i<x.length;i++){

If( x[i]>max){

Max=x[i];}

}

Console.log(max)

Let min = x[0]

For (ì=0; i<x.length;i++){

If (x[i]<min){

Min=x[i];}

}

Console.log(min)

**III. Thao tác với phần tử trong mảng**

**13. Làm sao để thêm phần tử vào đầu hoặc cuối mảng?**

Dùng x.push()

**14. Làm sao để xóa phần tử ở đầu hoặc cuối mảng?**

Dùng x.pop()

**15. splice() khác gì với slice()? Dùng splice() để xóa/thêm/thay thế như thế nào?**

Splice() thay đổi mảng

array.splice(start, deleteCount, item1, item2, ..., itemN)

start: Vị trí bắt đầu (chỉ số) từ đó thực hiện thao tác.

deleteCount: Số lượng phần tử cần xóa (nếu có).

item1, item2, ...: Các phần tử mới cần thêm vào mảng (nếu có).

Slice() trích xuất từ một mảng. không thay đổi mảng cũ

**16. Làm sao để xóa phần tử ở vị trí cụ thể trong mảng?**

x.splice(“vị trí cụ thể”, “số phần tử xóa”,..)

**17. Làm sao để cập nhật giá trị phần tử tại vị trí nào đó?**

X[n] = …

**IV. Tìm kiếm, lọc và biến đổi mảng**

**18. Làm sao để kiểm tra một phần tử có tồn tại trong mảng hay không?**

Array.includes(valuetofind, “chỉ số bắt đầu”)

**19. Làm sao để tìm vị trí (chỉ số) của một phần tử trong mảng?**

array.indexOf(valueToFind, fromIndex)

**20. Dùng phương thức filter() để lọc những phần tử thỏa mãn điều kiện như thế nào?**

array.filter( callback (element, index, array), thisArg )

callback: Hàm callback được gọi cho mỗi phần tử trong mảng.

element: Phần tử hiện tại trong mảng.

index: Chỉ số của phần tử trong mảng.

array: Mảng gốc.

thisArg: (Tùy chọn) Giá trị này sẽ được sử dụng làm this trong hàm callback

**21. Dùng map() để tạo ra mảng mới từ mảng cũ như thế nào?**

const numbers = [1, 2, 3, 4, 5];

const doubledNumbers = numbers.map(function(number) { return number \* 2; });

console.log(numbers); // Output: [1, 2, 3, 4, 5] (mảng gốc không thay đổi)

console.log(doubledNumbers); // Output: [2, 4, 6, 8, 10] (mảng mới với các phần tử đã nhân đôi

**22. Dùng reduce() để tổng hợp dữ liệu trong mảng như thế nào?**

const results = [8, 5, 9, 4, 7, 6, 3, 8];

const statistics = results.reduce((accumulator, currentResult) => { if (currentResult >= 5) { accumulator.passed = (accumulator.passed || 0) + 1; }

else { accumulator.failed = (accumulator.failed || 0) + 1; }

return accumulator; }, { passed: 0, failed: 0 }); // initialValue là một đối tượng

console.log("Thống kê kết quả:", statistics); // Output: Thống kê kết quả: { passed: 6, failed: 2 }

**V. Một số thao tác nâng cao và tình huống đặc biệt**

**23. Làm sao để sắp xếp mảng số? Làm sao để sắp xếp mảng chuỗi?**

Dùng hàm sort()

Đối với số

const numbers = [5, 2, 8, 1, 9, 4];

numbers.sort(function(a, b) { return b - a; });

console.log(numbers); // Output: [9, 8, 5, 4, 2, 1] // Sử dụng arrow function

numbers.sort((a, b) => b - a);

console.log(numbers); // Output: [9, 8, 5, 4, 2, 1]

**24. Làm sao để đảo ngược thứ tự của mảng?**

Dùng reverse()

Ví dụ .

const strings = ["banana", "apple", "orange", "grape"];

strings.sort().reverse();

console.log(strings);

// Output: ["orange", "grape", "banana", "apple"]

**25. Làm sao để gộp hai mảng thành một?**

Sử dụng concat() hoặc (…)

Ví dụ

Const a =[ array 1]

Const b= [array 2]

1. Const c = a.concat(b)

2. const c = […array 1, …array 2]

**26. Làm sao để loại bỏ các phần tử trùng nhau trong mảng?**

const arrayWithDuplicates = [1, 2, 2, 3, 4, 4, 5];

const uniqueArray = [...new Set(arrayWithDuplicates)];

console.log(uniqueArray); // Output: [1, 2, 3, 4, 5]

// Hoặc có thể tạo Set trực tiếp từ mảng

const uniqueSet = new Set(arrayWithDuplicates);

const uniqueArrayFromSet = Array.from(uniqueSet);

console.log(uniqueArrayFromSet); // Output: [1, 2, 3, 4, 5]

**27. Làm sao để tạo một bản sao (copy) của mảng mà không làm thay đổi mảng gốc?**

Slice() trích xuất từ một mảng. không thay đổi mảng cũ

**VI. Ứng dụng thực tế**

**28. Làm sao để chọn ngẫu nhiên một phần tử trong mảng?**

Chọn ngẫu nhiên 1 số nguyên:

Const array = [..]

Const n = Math.floor.(Mathrandom()\*array.length);

Console.log (array[n])

**29. Làm sao để nhóm các phần tử trong mảng theo điều kiện nào đó?**

const groupedObject = array.reduce((accumulator, currentValue) => {

// Xác định khóa (key) cho nhóm dựa trên điều kiện của currentValue

const key = /\* điều kiện \*/;

// Nếu khóa chưa tồn tại trong accumulator, khởi tạo một mảng rỗng

if (!accumulator[key]) { accumulator[key] = []; }

// Thêm currentValue vào mảng tương ứng với khóa

accumulator[key].push(currentValue);

// Trả về accumulator để tiếp tục cho lần lặp tiếp theo return accumulator; }, {});

// Giá trị khởi tạo của accumulator là một đối tượng rỗng

**30. Làm sao để chuyển một mảng số thành một chuỗi (string)?**

Sử dụng phương thức map()

const numbers = [1, 2, 3, 4, 5];

const strings = numbers.map(number => number.toString());

console.log(strings);// Output: ['1', '2', '3', '4', '5']

console.log(typeof strings[0]); // Output: string

**31. Làm sao xử lý mảng 2 chiều (ma trận) bằng JavaScript?**

Sử dụng vòng lặp for để đi qua từng dữ liệu trong mảng

Ví dụ tính tổng các phần tử

const matrix = [ [1, 2, 3], [4, 5, 6], [7, 8, 9] ];

let sum = 0;

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

{ for (let j = 0; j < matrix[i].length; j++) { sum += matrix[i][j]; } }

console.log("Tổng các phần tử:", sum); // Output: Tổng các phần tử: 45