**Mảng và các phương thức**

1. **Cú pháp khai báo mảng.**

-Mảng trong js thường được khai báo với **const**;

-Vd: const m = [“m1”, “m2”, “m3”];

-Cách tạo mảng:

VD1:

const cars = [  
   "Saab",  
   "Volvo",  
   "BMW"  
 ];

VD2:

const cars = [];  
 cars[0]= "Saab";  
 cars[1]= "Volvo";  
 cars[2]= "BMW";

VD3:

const cars = new Array("Saab", "Volvo", "BMW");

1. **Các hàm thao tác với mảng.**

Cho mảng: const a = [‘táo’, ’cam’];

Const b = [‘dừa’, ‘xoài’];

|  |  |  |
| --- | --- | --- |
| Hàm | Chức Năng | Ví dụ |
| toString() | Biến một mảng thành chuỗi được ngăn cách bằng dấu phẩy | a.toString();  => táo, cam |
| join() | Chứng năng giống với toString(), nhưng có thể thay đổi dấu phân cách. | a.join(“ / ”);  => táo / cam; |
| pop() | Loại bỏ phần tử cuối cùng ra khỏi mảng. | a.pop();  => táo |
| push() | Thêm một phần tử mới vào cuối mảng. | a.push(“chanh”);  => táo, cam, chanh |
| shift() | Xóa phần tử đầu tiên ra khỏi mảng | a.shift();  => cam |
| unshift() | Thêm một phần tử vào đầu mảng | a.unshift(“mít”);  =>mít, táo , cam |
| length() | Lấy độ dài mảng | a.length  => 2 |
| concat() | Nối hai mảng  Cú pháp: a.concat(b);  Chú ý: Hàm trong ngoặc concat được nối vào hàm trước concat, như ở trên thì b được nối vào a.  -> Mở rộng: a.concat(b,c);  B,c nối vào a; | a.concat(b)  => táo, cam, dừa, xoài. |
| splice() | -Vừa có thể xóa phần tử, vừa có thể nối phần tử.  Cú pháp a.splice(a,b,c);  a: chỉ số phần tử cần thêm và cũng là vị trí cần xóa nếu b khác 0;  b: số lượng phần tử cần xóa  c: phần tử cần thêm vào | \*)a.splice(1,0,”kiwi”);  => táo, kiwi, cam  Giải thích: thêm phần tử kiwi vào vị trí có chỉ số 1 và không xóa phần tử nào   * a: 1 * b: 0 * c: “kiwi”   \*)a.splice(1,1,”kiwi”);  => táo, kiwi  Giải thích: thêm phần tử kiwi vào vị trí có chỉ số 1 và đồng thời xóa 1 phần tử (vì  b = 1 nên chỉ xóa 1 phần tử) đang ở vị trí 1 là cam.  \*) a.splice(0,1);  => cam  -Xóa phần tử có chỉ số 0 ra khỏi mảng |
| -slice() | - Tạo một mảng mới bằng cách cắt mảng cũ mà không làm mất đi giá trị mảng cũ  -Gồm 2 đối số:  Const m = a.slice(b,c);  m: mảng mới  a: mảng ban đầu  b: chỉ số của vị trí cần cắt.  c: độ dài cần cắt (không bao gồm nó) -> không bắt buộc phải có. | VD:  const a = ["Banana", "Orange", "Lemon", "Apple"]  \*) const m = a.slice(1,3)  => "Orange", "Lemon"  Giải thích: Nó cắt từ vị trí có chỉ số 1 trong mảng và cắt đến trước vị trí có chỉ số 3, có thể hiểu giới hạn của nó như sau: 1 <= m < 3, nghĩ là nó lấy phần tử 1 và 2   * Nếu đối số 2 bị bỏ qua thì nó cắt từ vị trí chọn đến cuối mảng thì dừng   \*) const m = a.slice(1)  => "Orange", "Lemon", "Apple" |
| sort() | -Sắp xếp mảng theo thứ tự bảng chữ cái. | VD:  const fruits = ["Banana", "Orange", "Apple", "Mango"];  fruits.sort();   * Apple,Banana,Mango,Orange |
| reverse() | -Đảo ngược các phần tử trong mảng | const fruits = ["Banana", "Orange", "Apple", "Mango"]; fruits.reverse();   * Mango, Apple, Orange, Banana |