

# Mảng trong JavaScript

Tạo đối tượng mảng trong JavaScript, các cách thức khởi tạo mảng và truy cập đến giá trị phần tử mảng

## Mảng JavaScript

Mảng Array là một đối tượng định nghĩa sẵn trong JS, nó là đối tượng chứa danh sách các dữ liệu. Có hai cách tạo ra đối tượng mảng, sử dụng toán tử `new` và khởi tạo bằng ký hiệu `[]`

**Cách 1**, tạo mảng với `new Array()`, ví dụ:

```
var courses = new Array("HTML", "CSS", "JS");
```

**Cách 2**, tạo mảng với khai báo dữ liệu trong `[]`

```
var courses = ["HTML", "CSS", "JS"];
```

Mỗi phần tử có một chỉ số trong mảng, phần tử thứ nhất chỉ số là `0`. Để truy cập tới phần tử mảng dùng ký hiệu `tên-mảng[chỉ-số-phần-tử]`

```
var cacmonhoc = new Array("HTML", "CSS", "JS");  
var monhoc = courses[0]; // HTML  
cacmonhoc[1] = "C++"; //đổi phần tử thứ 2 thành C++
```

Khi cố gắng truy cập phần tử mảng, mà phần tử đó không tồn tại thì nó trả về giá trị bằng `undefined`

```
var courses = new Array("HTML", "CSS", "JS");  
document.write(courses[10]);  
//Xuất ra "undefined" vì phần tử thứ 10 không có
```

## Tạo mảng bằng cách chỉ định số lượng phần tử

Ở cách tạo thứ 2 này, các phần tử không được gán giá trị ngay, mảng sẽ tạo ra số lượng phần tử yêu cầu sau đó giá trị từng phần tử được gán sau.

```
var courses = new Array(3);  
courses[0] = "HTML";  
courses[1] = "CSS";  
courses[2] = "JS";
```

## Một số thuộc tính và phương thức

Một số thuộc tính và phương thức định nghĩa sẵn cho các đối tượng mảng

## Số lượng phần tử trong mảng

Thuộc tính `length` cho biết số lượng phần tử trong mảng

```
var courses = ["HTML", "CSS", "JS"];
document.write(courses.length);
//Outputs 3
```

## Kết hợp 2 mảng

Để kết hợp 2 mảng thành một mảng, dùng phương thức `concat`

```
var c1 = ["HTML", "CSS"];
var c2 = ["JS", "C++"];
var courses = c1.concat(c2);
```

Ví dụ trên có được mảng `courses` bằng cách nối `c2` vào `c1`. Vậy `courses` có bốn phần tử: HTML, CSS, JS, C++. Nhờ là `concat` không tác động gì trên mảng gốc, mà nó chỉ sử dụng các mảng gốc để tạo ra mảng kết hợp mới. Vậy mảng `c1`, `c2` vẫn không có sự thay đổi nào

## Duyệt qua các phần tử mảng bằng `forEach`

```
courses.forEach(function(item, index, array) {
    console.log(item, index);
});

//HTML 0
//CSS 1
//JS 2
```

## Thêm một phần tử vào cuối bằng `push`

```
courses.push("GO");
```

## Thêm một phần tử vào đầu bằng `unshift`

```
courses.unshift("C++");
```

## Xóa phần tử cuối của mảng bằng `pop`

```
courses.pop();
```

## Xóa phần tử đầu của mảng bằng `shift`

```
courses.shift();
```

## Xóa các phần tử theo chỉ số `splice`

```
//Xóa 2 phần tử bắt đầu phần tử ở vị trí index
courses.splice(index, 2);
```

## Nhân bản một mảng với slice

```
var courses2 = courses.slice();
```

## Mảng kết hợp

Nhiều ngôn ngữ lập trình có hỗ trợ đánh chỉ số không chỉ là bằng số mà có thể bằng text, đây là các mảng kết hợp. Rất tiếc là JavaScript không hỗ trợ loại mảng này.

Tuy nhiên, bạn nhớ lại về đối tượng, truy cập thuộc tính đối tượng hoạt động khá giống với mảng kết hợp.

```
var person = []; //mảng rỗng
person["name"] = "John";
person["age"] = 46;

document.write(person["age"]);
//Outputs "46"
```

Bạn thấy `person` hoạt động giống như một đối tượng. Các chỉ số `name`, `age` không phải là chỉ số mảng mà là thuộc tính của đối tượng `person`. Do đó nếu bạn lấy số phần tử mảng `person.length` sẽ trả về 0

## Cách đơn giản để duyệt qua các phần tử mảng

```
var index;
var a = ["a", "b", "c"];
for (index = 0; index < a.length; ++index) {
    console.log(a[index]);
}
```