

Buổi 13

JS Object

I. Tổng quan về Object

Khi giải quyết 1 bài toán, đôi khi chúng ta cần lưu trữ và thao tác với các kiểu dữ liệu có cấu trúc phức tạp, ví dụ:

- Phân số có tử số và mẫu số.
- Điểm trong không gian 2 chiều có hoành độ và tung độ.
- Tam giác có 3 đỉnh.
- Sinh viên có mã SV, họ tên, ngày sinh, điểm trung bình môn.
- ...

Với các bài toán này, nếu chỉ dùng cách lưu dữ liệu vào các biến cơ bản thì sẽ rất tốn thời gian và tốn rất nhiều biến, ví dụ với 1 phân số cần 2 biến để lưu giá trị, với tam giác cần 6 biến...

JS cung cấp khả năng tạo ra 1 đối tượng (*object*). Đối tượng cũng là 1 biến, nhưng nó có thể chứa số lượng thông tin, dữ liệu tùy ý.

Ví dụ:

- Tạo đối tượng phân số:
`var phanSo = { tuSo: 1, mauSo: 2 };`
- Tạo đối tượng điểm trong không gian 2 chiều:
`var diem = { x: 4, y: 2 };`
- Tạo đối tượng sinh viên:
`var sv = {
 maSV: "0175",
 hoTen: "Dương Hữu Phước",
 namSinh: 1991,
 dtb: 9.5
};`

II. Thuộc tính và phương thức

1. Thuộc tính

Khi tạo Object, những cặp *name:value* được gọi là thuộc tính (*property*). Object là tập hợp các cặp thuộc tính không theo thứ tự, mỗi cặp gồm có tên thuộc tính và giá trị của thuộc tính đó.

Các thao tác trên thuộc tính của Object:

- Truy cập thuộc tính: Dùng 1 trong 2 cách sau:
`alert("Họ tên SV là: " + sv.hoTen);`
`document.write("ĐTB: " + sv["dtb"]);`
- Thêm thuộc tính:
`sv.sdt = "0905 939 947";`
- Sửa thuộc tính:
`sv.dtb = 10;`
- Xóa thuộc tính:
`delete sv.ngaySinh;`

Để truy cập toàn bộ thông tin của 1 Object, chúng ta có thể dùng vòng lặp for/in:

```
for (x in sv) {  
    document.write(sv[x] + "<br>");  
}
```

2. Phương thức

Khi tạo Object, chúng ta có thể cài đặt các hàm để thực hiện các tác vụ trên Object đó. Những hàm này được gọi là phương thức (*method*).

Cú pháp cài đặt 1 phương thức:

```
var sv = {  
    .....  
    .....  
    .....  
    XepLoai: function () {  
        if (this.dtb >= 5) {  
            return "Đậu";  
        }  
        return "Rớt";  
    }  
};
```

Để gọi thực hiện 1 phương thức, chúng ta dùng câu lệnh sau:

```
sv.XepLoai();
```

Lưu ý: Câu lệnh sv.XepLoai không có dấu ngoặc đơn () sẽ chỉ trả về phần mã nguồn của phương thức này.

Khi cài đặt phương thức, nếu cần truy cập đến thuộc tính của đối tượng hiện tại, chúng ta truy cập thông qua từ khóa this. Từ khóa this đại diện cho đối tượng đang gọi phương thức này.

Cú pháp để thêm/xóa 1 phương thức tương tự như thêm/xóa 1 thuộc tính:

```
sv.TinhTuoi = function () {  
    var d = new Date();  
    return d.getFullYear() - this.namSinh;  
}
```

```
delete sv.XepLoai;
```

III. Phương thức khởi tạo

Những câu lệnh để tạo Object ở mục I và II chỉ dùng để tạo 1 Object đơn lẻ. Nếu có nhu cầu tạo nhiều Object cùng loại (có thông tin giống nhau), chúng ta cần 1 hàm khởi tạo đối tượng.

Trong ví dụ dưới đây, chúng ta sẽ cài đặt 1 hàm khởi tạo đối tượng cho 1 Object Smartphone:

```
function Smartphone(name, brand, provider, price) {  
    this.name = name;  
    this.brand = brand;  
    this.provider = provider;  
    this.price = price;  
    this.TotalPrice = function () {  
        return Math.round(this.price * 1.1);  
    }  
}
```

Sau đó, nếu cần tạo 1 Object kiểu Smartphone, chúng ta dùng từ khóa new theo cú pháp sau:

```
var phone1 = new Smartphone("iPhone Xs Max", "Apple", "AT&T", 499);  
var phone2 = new Smartphone("Galaxy Note 10", "Samsung", "Verizon", 399);
```

Khi đó, mỗi đối tượng thuộc kiểu Smartphone sẽ có những thuộc tính name, brand, provider, price và phương thức TotalPrice() của riêng nó.

Cách thêm thuộc tính, phương thức vào đối tượng được tạo theo cách này hoàn toàn tương tự như mục II. Tuy nhiên, cần chú ý rằng thuộc tính, phương thức được thêm vào đối tượng nào thì chỉ có đối tượng đó mới có thuộc tính, phương thức đó.

Ví dụ: Thêm thuộc tính releasedDate vào đối tượng phone1. Đối tượng phone2 sẽ không có thuộc tính này:

```
phone1.releasedDate = "15/06/2019";
```

IV. Phương thức truy cập dữ liệu

Giả sử có 1 Object Giáo viên như sau:

```
var gv = {  
    ho: "Dương",  
    tenLot: "Hữu",  
    ten: "Phước",  
    queQuan: "Nha Trang"  
};
```

1. Getter

Để hiển thị đầy đủ họ và tên của GV này, cần bổ sung 1 phương thức fullName() như sau:

```
var gv = {  
    .....,  
    showFullName: function () {  
        return this.ho + " " + this.tenLot + " " + this.ten;  
    }  
};  
document.write(gv.showFullName());
```

Ngoài cách viết như trên, JS hỗ trợ 1 phương thức lấy dữ liệu của 1 đối tượng. Phương thức này được gọi là *getter* (hoặc *accessor*):

```
var gv = {  
    .....,  
    get fullName() {  
        return this.ho + " " + this.tenLot + " " + this.ten;  
    }  
};  
document.write(gv.fullName);
```

Lưu ý: Khi gọi getter, không gọi kèm dấu ngoặc đơn () như khi gọi phương thức.

2. Setter

Để thay đổi quê quán của GV này, cần bổ sung 1 phương thức changeHometown() như sau:

```
var gv = {  
    .....,  
    changeHometown: function (value) {  
        this.queQuan = value;  
    }  
};
```

```
gv.changeHometown("Tp.HCM");
```

Ngoài cách viết như trên, JS hỗ trợ 1 phương thức thay đổi dữ liệu của 1 đối tượng. Phương thức này được gọi là *setter* (hoặc *mutator*):

```
var gv = {  
    .....,  
    set hometown(value) {  
        this.queQuan = value;  
    }  
};  
gv.hometown = "Tp.HCM";
```

Lưu ý: Khi gọi setter, không gọi kèm dấu ngoặc đơn () như khi gọi phương thức. Setter chỉ dùng để thay đổi giá trị của 1 thuộc tính.

V. Bài tập

Thực hiện các bài tập sau đây, mỗi bài tập nằm trong 1 trang web riêng.

1. Cho phép người dùng nhập vào 2 phân số. Tính tổng, hiệu, tích và thương của 2 phân số đó.
2. Cho phép người dùng nhập vào tọa độ 2 điểm trong không gian 2 chiều. Tính khoảng cách giữa 2 điểm đó.
3. Cho phép người dùng nhập vào tọa độ 3 đỉnh của 1 tam giác. Tính chu vi và diện tích của tam giác đó.

Diện tích tam giác tính theo công thức Heron như sau:

$$S_{\Delta ABC} = \sqrt{p(p-a)(p-b)(p-c)}$$

Trong đó:

- p là nửa chu vi
 - a, b, c là độ dài 3 cạnh
4. Cho phép người dùng nhập thông tin của SV gồm mã SV, họ, tên lót, tên, điểm Toán, điểm Lý, Điểm Hóa. Với mỗi sinh viên nhập vào, đưa thông tin sinh viên đó vào 1 mảng và hiển thị thông tin từng sinh viên theo định dạng sau:

SV #1:

Mã SV:

Họ tên:

Điểm Toán:

Điểm Toán:

Điểm Lý:

Điểm Hóa:

Điểm TB:

Xếp loại:

Trong đó:

- Điểm trung bình là trung bình cộng của điểm Toán, Lý và Hóa.
- Nếu Điểm TB ≥ 5 thì xếp loại là "Đạt", ngược lại xếp loại là "Thi lại".