

Lớp trong Javascript

Khai báo và sử dụng lớp trong JavaScript, tạo ra các đối tượng lớp, kế thừa lớp

Lớp (Class) trong JS

Lớp để tạo ra các đối tượng (thực chất là một cách thức khác tạo ra các [đối tượng Javascript](#) đã biết). Đối tượng trong nó có các thuộc tính, phương thức. Tạo ra đối tượng sử dụng từ khóa **class** và hàm khởi tạo **constructor**

```
//Khai báo một lớp có tên Product
class Product {
  //Hàm khởi tạo
  constructor(name, price) {
    this.name = name;
    this.price = price;
    this.infomation = `${name} - ${price}`;
  }

  //Khai báo một phương thức
  checkStore(storeid) {
    console.log(this.name + ' in store ' + storeid);
  }

  //Hàm getter
  get info() {
    return this.infomation;
  }

  //Hàm setter
  set info(i) {
    this.infomation = i;
  }

  //Phương thức tĩnh
  static convertMoney(m) {
    console.log(m);
    return m + ' đồng';
  }
}

//SỬ DỤNG LỚP

//Tạo một đối tượng từ lớp bằng new
let sanpham = new Product('Iphone', 1000);

//truy cập thuộc tính đối tượng sanpham.name
console.log(sanpham.name);

//gọi một phương thức của đối tượng
sanpham.checkStore(100);

//Gọi setter
sanpham.info = 'Thông tin sản phẩm ...';
```

```
//Gọi getter
console.log(sanpham.info);

//Gọi một hàm tĩnh
Product.convertMoney(100000);
```

Từ đoạn mã trên lưu ý mấy vấn đề

- Lớp khai báo với từ khóa **class**, bắt buộc có hàm khởi tạo **constructor**, hàm này được gọi khi khởi tạo đối tượng lớp **new Product('Iphone', 1000)**
- Các phương thức (checkStore) khai báo trong lớp giống như khai báo hàm nhưng bỏ đi khóa **function**
- Ở trước phương thức có từ khóa **get** thì nó gọi là phương thức getter, nó được gọi như cách truy cập thông tin thuộc tính
- Ở trước phương thức có từ khóa **set** thì nó là setter, nó được gọi khi bị gán (=)
- Phương thức tĩnh có keyword **static**, được gọi qua tên lớp (nó không thể tham khảo đến đối tượng lớp qua **this**)

Tính kế thừa của lớp

Giống như nhiều ngôn ngữ lập trình hướng đối tượng, JS cung cấp tạo ra một lớp mới có kế thừa lại thuộc tính, phương thức của một lớp khác. Để tạo ra một lớp con kế từ lớp cha dùng từ khóa **extends**

```
class Computer extends Product {
  constructor(name, price, store) {
    super(name, price);
    this.store = store;
  }

  set info(i) {
    //super.info(i) - nếu muốn thì hành phương thức của lớp cha
    this.information = name + ':' + i;
  }

  totalInStore() {
    console.log('totalInStore');
  }
}

//Sử dụng
let sanpham = new Computer('Dell', 2000);
console.log(sanpham.name);
sanpham.checkStore(200);

sanpham.info = 'Thông tin sản phẩm ...';

console.log(sanpham.info);
sanpham.totalInStore();
```

Ở ví dụ trên bạn đã tạo ra một lớp mới có tên **Computer** kế thừa lớp cha là **Product**, do đó lớp con có các thuộc tính, phương thức như lớp cha và khai báo thêm những thuộc tính phương thức riêng.

- Nếu trong lớp con khai báo một phương thức trùng tên đã có ở lớp cha thì phương thức lớp con được sử dụng mặc định, nếu muốn gọi đến phương thức của lớp cha dùng từ khóa **super** thay cho **this** để tham khảo đến lớp cha
- Hàm khởi tạo lớp cha có tham số, do vậy bắt buộc phải gọi hàm tạo của lớp cha bằng **super(..)**; trong **constructor** của lớp con