

call, apply, bind

Chỉ có các function mới có thể gọi 3 hàm này.

call

Phương thức call() cho phép bạn gọi function và truyền vào một object và các đối số(argument) cách nhau bởi dấu phẩy (Comma).

Cú pháp của call() như sau:

```
function.call(thisArg, arg_1, arg_2,...);
```

ví dụ:

```
let person = {
  firstName: "Long",
  lastName: "Do",
}

function saySomething(param_1, param_2) {
  console.log(param_1 + " " + param_2 + ", " + this.firstName + " " +
this.lastName);
}

saySomething.call(person, "Welcome to", "Attlab"); // --> Output: Welcome to
Attlab, Long Do
```

apply

Phương thức apply() cũng tương tự như call(), nhưng nó khác ở chỗ nó truyền vào các đối số thông qua mảng, cú pháp của nó như sau:

```
function.apply(thisArg, [argsArray]);
```

ví dụ:

```
let person = {
  firstName: "Long",
  lastName: "Do",
}

function saySomething(param_1, param_2) {
```

```
    console.log(param_1 + " " + param_2 + ", " + this.firstName + " " +  
    this.lastName);  
}  
  
saySomething.apply(person, ["Welcome to", "Attlab"]);  
// --> Output: Welcome to Attlab, Long Do
```

bind

Phương thức `bind()` có chút đặc biệt hơn, nó trả về một function mới, nó cũng cho phép ta truyền vào một objectb và các đối số ngăn cách nhau bởi dấu phẩy, cú pháp của `bind()` như sau.

```
let newFunction = func.bind(thisArg[, arg1[, arg2[, ...]]]);
```

```
let person = {  
    firstName: "Frontend",  
    lastName: "Developer",  
};  
  
function saySomething(param_1, param_2) {  
    console.log(param_1 + " " + param_2 + ", " + this.firstName + " " +  
    this.lastName);  
}  
  
const say = saySomething.bind(person, "Hello", "LongDC");  
say();  
// --> Output: Hello LongDC, Frontend Developer
```

Một lưu ý chung cho cả ba phương thức trên là `this` sẽ đề cập đến object mà ta truyền vào.

Nhìn chung thì cả ba phương thức đều có những nét tương đồng, trong đó `call()` và `apply()` là gần giống nhau nhất, với `bind()` thì hơi khác chút nhưng cả ba đều không khó hiểu như ông thần `this`.