

Đầu tiên xem qua ví dụ sau

```
let globalFunction = []
class Hello {
  constructor(){
    this.a = 10;
    this.b = 10;
    globalFunction.push(this.printSumAB)
  }
  printSumAB() {
    console.log(this.a + this.b);
  }
}

// ở một nơi nào khác nữa
let checking = new Hello();

// ở một nơi nào đó
globalFunction[0]();
```

Đoạn chương trình trên chứa một mảng globalFunction được gọi tại mọi nơi trong hệ thống. Một class Hello, khi được khởi tạo một instance sẽ đăng kí một hàm printSumAB và globalFunction. Và sau đó một nơi nào đó sẽ gọi hàm printSumAB của instance Hello được khởi tạo kia. Và yêu cầu của chúng ta khi gọi hàm kia là phải in ra được số 20. Nhưng khi chạy đoạn chương trình trên thì nó lại in ra NaN. Vậy thì cái gì xảy ra ở đây, tại sao không thể in ra 20 dù đã đăng kí hàm rồi???

- Instance của lớp Hello được lưu ở một vùng nhớ khác
- globalFunction lại ở vùng nhớ khác
 - > khi globalFunction[0] được gọi hàm thông qua () thì cơ chế của JS là sẽ thực hiện compile đoạn code đó, (do khi đăng kí bằng lệnh push thì nó chỉ đẩy đoạn code đó vào mảng, chứ không thực hiện chạy)

```

let globalFunction = []
class Hello {
  constructor(){
    this.a = 10;
    this.b = 10;
    globalFunction.push(this.printSumAB.bind(this))
  }
  printSumAB() {
    console.log(this.a + this.b);
  }
}

let checking = new Hello();
console.log(globalFunction[0])

globalFunction[0]();

```

Kết quả khi chạy

```

f printSumAB() {
  console.log(this.a + this.b);
}

```

Vậy nên khi được gọi tới, JS compile, nó thấy `this.a` và `this.b` tương đương với hai biến `a` và `b` của đoạn code được đăng kí, và nó cộng lại với nhau -> NaN (do *this* trong JS nó bị ngu, nên lúc này *this* được hiểu là function mình đăng kí chạy)

Vậy muốn gọi đúng ra `a` và `b` của đối tượng `Hello` kia thì phải làm sao -> sử dụng `bind`

```

let globalFunction = []
class Hello {
  constructor(){
    this.a = 10;
    this.b = 10;
    globalFunction.push(this.printSumAB.bind(this))
  }
  printSumAB() {
    console.log(this.a + this.b);
  }
}

```

```
    }  
    printSumAB() {  
        console.log(this.a + this.b);  
    }  
}  
  
// ở một nơi nào khác nữa  
let checking = new Hello();  
  
// ở một nơi nào đó  
globalFunction[0]();
```

Khi đi kèm bind, mình đã chỉ định cho thằng function kia là *this* trong đoạn code trên là của đối tượng Hello mình vừa khởi tạo và khi chạy nó sẽ sử dụng instance của biến Hello để lấy