Hàm (function)

Một Javasript function là một đoạn code được thiết kế để làm một nhiệm vụ riêng biệt.

Javascript function được thực thi khi gọi nó.

Hàm trong Javascript chia làm 2 loại

1. Khai báo hàm (Function Declaration)
2. Biểu thức hàm (Function Expression)

Khai báo hàm (Function Declaration)

Với cách này thì function được **Hoisting**. Và Javascript cho phép chúng ta gọi một hàm trước khi hàm đó được khai báo.

```
hoisted() // Output: "This function has been hoisted."
function hoisted() {
  console.log('This function has been hoisted.')
}
```

Biểu thức hàm (Function Expression)

Tuy nhiên với cách khai báo function kiểu này thì sẽ không được hoisting

```
expression() //Output: "TypeError: expression is not a function
var expression = function () {
  console.log('Will this work?')
}
```

Giải thích: Biến `var expression` vẫn được **hoisting** và được đẩy lên trên cùng của scope nhưng chỉ là khai báo mà thôi, nó không được gán cho hàm! Vì thế nó sẽ ném ra lỗi `TypeError`.

IIFE (Immediately Invokable Function Expression)

IIFE là khởi tạo một function và thực thi ngay lập tức sau đó.

```
;(function () {
  let a = 1
  let b = 2
  console.log('a + b = ' + (a + b))
})()
```

Hàm ẩn danh (Anonymous function)

Hàm ẩn danh là hàm không tên. Nếu bạn để ý thì vế bên phải biểu thức hàm là một **anonymous function**, hay **IIFE** cũng thực thi một hàm ẩn danh. Ngoài ra hàm ẩn danh còn xuất hiện ở **callback**

function bên trong `setTimeout` là một hàm ẩn danh

```
setTimeout(function () {
  console.log('Sau 1s thì sẽ in ra dòng này')
}, 1000)
```

Hàm rút gọn (Arrow function)

Hàm rút gọn ngắn hơn biểu thức hàm (function expression) và không phụ thuộc this. Áp dụng tốt cho hàm ẩn danh (anonymous function) nhưng không thể dùng làm hàm khởi tạo

```
const handleClick = () => {
  // thực hiện gì đó
}
```

Lưu ý với arrow function:

- Không có `this`
- Không được gọi với `new`
- Cũng không có super, chúng ta sẽ học về nó trong bài kế thừa class

Phân biệt parameter (tham số) vs argument (đối số)

```
// a, b là tham số
function sum(a, b) {
  return a + b
}
// 1,2 là đối số
sum(1, 2)
```

Tham số mặc định (default parameter)

```
function gx(x, y = x) {
  console.log(x, y)
}
gx(3) // 3 3
gx(3, 5) // 3 5
```

Rest parameter

```
function sum(...theArgs) {
  console.log(theArgs)
}
console.log(sum(1, 2, 3)) // [1, 2, 3]
console.log(sum(1, 2, 3, 4)) // [1, 2, 3, 4]
```