

 Cập nhật tháng 8 năm 2024

[Bài Đọc] Function expression

Function Expression là một cách định nghĩa hàm trong JavaScript bằng cách gán một hàm cho một biến, hằng số, hoặc một thuộc tính của đối tượng. Nó khác với **Function Declaration** ở cách nó được khai báo và sử dụng.

1. Cách sử dụng Function expression

- Bạn tạo một hàm và gán nó cho một biến.
- Hàm này có thể được gọi thông qua tên biến.

Ví dụ:

```
const greet1 = function (name) {  
  | return `Hello, ${name}!`;   
};  
  
console.log(...data: greet1(name: 'Alice')); // Output: Hello, Alice!  
  
function greet2(name) {  
  | return `Hello, ${name}!`;   
}  
  
console.log(...data: greet2(name: 'Bob')); // Output: Hello, Bob!
```

2. Đặc điểm của Function expression

- Ẩn danh (Anonymous Function):
 - Function Expression thường được sử dụng với các hàm không tên (hàm ẩn danh).
 - Bạn cũng có thể đặt tên, nhưng tên đó chỉ hữu ích trong ngữ cảnh nội bộ của chính hàm.

```
const anonymousFunction = function() {  
  console.log(...data: "This is an anonymous function.");  
};  
anonymousFunction();
```

- **Không bị hoisting:**

- Khác với Function Declaration, Function Expression không được hoisting.
- Bạn phải khai báo hàm trước khi gọi nó.

```
// Sai: Gây lỗi  
greet(name: "Alice"); // ReferenceError: Cannot access 'greet' before initialization  
const greet = function(name) {  
  return `Hello, ${name}!`;  
};
```

- **Lưu trữ hàm trong biến:**

- Function Expression cho phép lưu trữ hàm trong các biến hoặc truyền chúng làm đối số.

```
const add = function(a, b) {  
  return a + b;  
};  
console.log(...data: add(a: 3, b: 4)); // Output: 7
```

3. Tài liệu tham khảo

<https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Operators/function>

Danh sách các bài học

