

Bài tập: Javascript cơ bản

Javascript

Bài tập utils function

Tạo file `01-js-{until_name}.js` để thực hiện bài tập sau:

every

```
// Bài 1: Tạo một mảng scores chứa các giá trị điểm của học sinh: 85, 90, 78. Hãy kiểm tra xem tất cả các điểm số đều lớn hơn 70 hay không.

// Bài 2: Tạo một mảng ages chứa các giá trị tuổi: 18, 21, 16, 25. Hãy kiểm tra xem tất cả các tuổi đều lớn hơn 15 hay không.

// Bài 3: Tạo một mảng words chứa các từ: "apple", "banana", "cherry", "date". Hãy kiểm tra xem tất cả các từ đều có độ dài lớn hơn 3 ký tự hay không.
```

filter

```
// Bài 1: Tạo một mảng scores chứa các giá trị điểm của học sinh: 85, 90, 78. Hãy lọc ra các học sinh có điểm trên 80.

// Bài 2: Tạo một mảng ages chứa các giá trị tuổi: 18, 21, 16, 25. Hãy lọc ra các tuổi lớn hơn hoặc bằng 18.

// Bài 3: Tạo một mảng words chứa các từ: "apple", "banana", "cherry", "date". Hãy lọc ra các từ có độ dài lớn hơn 5 ký tự.
```

find

```
// Bài 1: Tạo một mảng scores chứa các giá trị điểm của học sinh: 85, 90, 78. Hãy tìm điểm số đầu tiên lớn hơn 80.

// Bài 2: Tạo một mảng ages chứa các giá trị tuổi: 18, 21, 16, 25. Hãy tìm tuổi đầu tiên lớn hơn 20.

// Bài 3: Tạo một mảng words chứa các từ: "apple", "banana", "cherry", "date". Hãy tìm từ đầu tiên có độ dài lớn hơn 5 ký tự.
```

map

```
// Bài 1: Tạo một mảng scores chứa các giá trị điểm kiểm tra: 85, 90, 78. Hãy sử dụng phương thức map() để tạo ra một mảng mới adjustedScores, trong đó mỗi giá trị là điểm kiểm tra ban đầu được tăng thêm 10% nếu điểm đó dưới 90, và giảm 5% nếu điểm đó từ 90 trở lên.

// Bài 2: Tạo một mảng numbers chứa các số: 1, 2, 3. Hãy chuyển đổi mảng các số thành mảng các chuỗi.

// Bài 3: Tạo một mảng numbers chứa các số: 1, 2, 3. Hãy nhân đôi mỗi giá trị trong mảng.
```

push/pop

```
// Bài 1: Tạo một mảng numbers chứa các số: 1, 2, 3. Hãy thêm số 4 vào cuối mảng.

// Bài 2: Tạo một mảng numbers chứa các số: 1, 2, 3, 4. Hãy loại bỏ số cuối cùng trong mảng.

// Bài 3: Tạo một mảng names chứa các tên: "Alice", "Bob", "Charlie". Hãy thêm tên "David" vào cuối mảng.
```

reduce

```
// Bài 1: Tạo một mảng scores chứa các giá trị điểm của học sinh: 85, 90, 78. Hãy tính tổng các điểm.  
  
// Bài 2: Tạo một mảng numbers chứa các số: 1, 2, 3, 4. Hãy tính tích của các số.  
  
// Bài 3: Tạo một mảng expenses chứa các giá trị chi phí: 50, 100, 150. Hãy tính tổng chi phí.
```

shift/ unshift

```
// Bài 1: Tạo một mảng numbers chứa các số: 1, 2, 3. Hãy loại bỏ phần tử đầu tiên trong mảng.  
  
// Bài 2: Tạo một mảng numbers chứa các số: 2, 3. Hãy thêm số 1 vào đầu mảng.  
  
// Bài 3: Tạo một mảng names chứa các tên: "Bob", "Charlie". Hãy thêm tên "Alice" vào đầu mảng.
```

some

```
// Bài 1: Tạo một mảng scores chứa các giá trị điểm của học sinh: 85, 90, 78. Hãy kiểm tra xem có điểm số nào lớn hơn 80 hay không.  
  
// Bài 2: Tạo một mảng ages chứa các giá trị tuổi: 18, 21, 16, 25. Hãy kiểm tra xem có tuổi nào dưới 18 hay không.  
  
// Bài 3: Tạo một mảng words chứa các từ: "apple", "banana", "cherry", "date". Hãy kiểm tra xem có từ nào có độ dài lớn hơn 5 ký tự hay không.
```

split

```
// Bài 1: Tạo một chuỗi name với giá trị "Nguyễn Văn A". Hãy chia chuỗi thành mảng các từ.

// Bài 2: Tạo một chuỗi emails với giá trị "example1@gmail.com,example2@gmail.com,example3@gmail.com". Hãy chia chuỗi thành mảng các email riêng lẻ.

// Bài 3: Tạo một chuỗi date với giá trị "2024-05-19". Hãy chia chuỗi thành mảng các phần riêng biệt của ngày tháng năm.
```

includes

```
// Bài 1: Tạo một chuỗi name với giá trị "Nguyễn Văn A". Hãy kiểm tra xem chuỗi có chứa "Nguyễn" hay không.

// Bài 2: Tạo một chuỗi email với giá trị "example@gmail.com". Hãy kiểm tra xem chuỗi có chứa ký tự "@" hay không.

// Bài 3: Tạo một chuỗi productName với giá trị "MacBook Pro". Hãy kiểm tra xem chuỗi có chứa từ "Pro" hay không.
```

index of

```
// Bài 1: Tạo một chuỗi name với giá trị "Nguyễn Văn A". Hãy tìm vị trí đầu tiên của ký tự 'a'.

// Bài 2: Tạo một chuỗi email với giá trị "example@gmail.com". Hãy tìm vị trí của ký tự '@'.

// Bài 3: Tạo một chuỗi description với giá trị "Khóa học JavaScript cơ bản". Hãy tìm vị trí của từ "JavaScript".
```

replace

```
// Bài 1: Tạo một chuỗi phoneNumber với giá trị "0123 456 789". Hãy thay thế tất cả khoảng trắng bằng dấu chấm.  
  
// Bài 2: Tạo một chuỗi report với giá trị "Có một lỗi trong hệ thống.". Hãy thay thế từ "lỗi" bằng từ "bug".  
  
// Bài 3: Tạo một chuỗi numbers với giá trị "1,234,567". Hãy thay thế tất cả dấu phẩy bằng dấu chấm.
```

substring

```
// Bài 1: Tạo một chuỗi fullName với giá trị "Nguyễn Văn A". Hãy trích xuất tên họ từ chuỗi.  
  
// Bài 2: Tạo một chuỗi date với giá trị "2024-05-19". Hãy trích xuất phần năm từ chuỗi.  
  
// Bài 3: Tạo một chuỗi email với giá trị "example@gmail.com". Hãy trích xuất phần tên miền từ chuỗi email.
```

trim

```
// Bài 1: Tạo một chuỗi name với giá trị " Nguyễn Văn A ". Hãy loại bỏ khoảng trắng ở đầu và cuối chuỗi.  
  
// Bài 2: Tạo một chuỗi userInput với giá trị " 12345 ". Hãy loại bỏ khoảng trắng ở cuối chuỗi.  
  
// Bài 2: Tạo một chuỗi email với giá trị " example@gmail.com ". Hãy loại bỏ khoảng trắng đầu chuỗi email.
```

Class

1. Tạo file `02-js-class-crm.ts` để thực hiện bài tập sau:

Mô tả:

Bạn là một lập trình viên xây dựng hệ thống quản lý khách hàng cho một công ty. Hãy tạo một class để lưu trữ thông tin khách hàng và các phương thức để thao tác với dữ liệu này.

Yêu cầu:

- Tạo một class `Customer` chứa các thuộc tính: `id, name, email, phone`.
- Tạo một phương thức `displayInfo` để hiển thị thông tin khách hàng.
- Tạo một phương thức `updateEmail` để cập nhật email của khách hàng.
Phương thức nhận vào một tham số duy nhất là `newEmail`

2. Tạo file `02-js-class-order.ts` để thực hiện bài tập sau:

Mô tả:

Bạn đang làm việc cho một cửa hàng trực tuyến và cần tạo một hệ thống quản lý đơn hàng. Hãy tạo một class để lưu trữ thông tin đơn hàng và các phương thức để thao tác với dữ liệu này.

Yêu cầu:

- Tạo một class `Order` chứa các thuộc tính: `orderId, customerName, items` (mảng các sản phẩm), `totalAmount`.
 - Sản phẩm bao gồm các thuộc tính: `name, price, amount, discount`
- Tạo một phương thức `addItem` để thêm sản phẩm vào đơn hàng.
- Tạo một phương thức `calculateTotal` để tính tổng số tiền của đơn hàng.

3. Tạo file `02-js-class-library.ts` để thực hiện bài tập sau:

Mô tả: Bạn đang xây dựng một ứng dụng quản lý thư viện. Hãy tạo một class để lưu trữ thông tin sách và các phương thức để thao tác với dữ liệu này.

Yêu cầu:

- Tạo một class `Library` chứa các thuộc tính: `name, location, books` (mảng các sách).
- Tạo một phương thức `addBook` để thêm sách vào thư viện.
- Tạo một phương thức `findBook` để tìm sách theo tiêu đề.

4. Tạo file `02-js-class-football.ts` để thực hiện bài tập sau:

Mô tả: Bạn đang phát triển một ứng dụng quản lý đội bóng. Hãy tạo một class để lưu trữ thông tin cầu thủ và các phương thức để thao tác với dữ liệu này.

Yêu cầu:

- Tạo một class `team` chứa các thuộc tính: `name`, `players` (mảng các cầu thủ).
- Tạo một phương thức `addPlayer` để thêm cầu thủ vào đội.
- Tạo một phương thức `listPlayers` để liệt kê tất cả các cầu thủ trong đội.