

## LAB 2

### MỤC TIÊU

Kết thúc bài thực hành sinh viên có khả năng:

- ✓ Ôn tập lại các phương thức xử lý mảng trong javascript.
- ✓ Ôn tập lại các phương thức xử lý object trong javascript
- ✓ Hiểu cách xử lý bất đồng bộ trong javascript

### NỘI DUNG

#### BÀI 1: XỬ LÝ DỮ LIỆU VỚI CÁC PHƯƠNG THỨC CỦA MẢNG

Nhà trường FPT POLYTECHNIC đang cần bạn thống kê tìm ra sinh viên có điểm số cao nhất, để chuẩn bị trao giải cho buổi lễ tôn vinh ong vàng, được tổ chức định kỳ sau khi kết thúc một kỳ học. Những sinh viên được tuyên dương là sinh viên nằm trong TOP 100 sinh viên có điểm số cao nhất hoặc TOP 10 sinh viên có điểm rèn luyện cao nhất. Điểm số và điểm rèn luyện là 2 giải thưởng khác nhau, thế nên cần bạn sắp xếp và lọc ra 2 lại danh sách sinh viên có điểm số từ cao xuống thấp để ban tổ chức dễ thông báo đến các bạn sinh viên lọt top để trao thưởng. Sinh viên có điểm số cao nhất sẽ là Ong vàng. Mẫu dữ liệu đầu vào các các danh sách từ các lớp khác nhau, cần bạn gộp lại để thống kê, lưu ý rằng sinh viên nào có trạng thái “failed” không được bỏ vào 2 sách trao thưởng này.

Yêu cầu đầu ra:

- Danh sách sinh viên có điểm số từ cao xuống thấp.
- Danh sách sinh viên có điểm rèn luyện từ cao xuống thấp.

- Xuất ra các thông tin của Ong vàng.

Dữ liệu các bạn được cung cấp như sau:

```
const class1 = [  
  {  
    mssv: 'PS0000',  
    name: 'Nguyen Van A',  
    avgPoint: 8.9,  
    avgTraningPoint: 7,  
    status: 'pass',  
  },  
  {  
    mssv: 'PS0001',  
    name: 'Nguyen Van B',  
    avgPoint: 4.9,  
    avgTraningPoint: 10,  
    status: 'pass',  
  }  
];
```

```
const class2 = [  
  {  
    mssv: 'PS0002',  
    name: 'Nguyen Van C',  
    avgPoint: 4.9,  
    avgTraningPoint: 10,  
    status: 'failed',  
  },  
  {  
    mssv: 'PS0003',  
    name: 'Nguyen Van D',
```

```

    avgPoint: 10,
    avgTraningPoint: 10,
    status: 'pass',
  },
  {
    mssv: 'PS0004',
    name: 'Nguyen Van E',
    avgPoint: 10,
    avgTraningPoint: 2,
    status: 'pass',
  },
]

```

Giải thích:

- + **avgPoint**: là điểm số trung bình của sinh viên.
- + **avgTraningPoint**: là điểm số rèn luyện trung bình.
- + **status**: là trạng thái học tập của sinh viên (failed = rớt môn, pass = học đúng tiến độ)

Hướng dẫn:

- Đầu tiên bạn cần gộp danh sách sinh viên của nhiều lớp lại với nhau. Chúng ta sử dụng phương thức **concat** của mảng:

```
const allStudent = class1.concat(class2)
```

- Tiếp theo bạn cần loại ra những sinh viên có **status** bằng failed ra khỏi danh sách trao thưởng này. Chúng ta sử dụng phương thức **filter**:

```
const allStudent = class1.concat(class2)
                          .filter(student => student?.status !== 'failed')
```

- Sử dụng phương thức **sort** sắp xếp danh sách theo điểm.
- Để lấy được thông tin Ong vàng, bạn có thể lấy sinh viên có index = 0, tại danh sách sinh viên có điểm số cao nhất.

## BÀI 2: Xử lý dữ liệu với các phương thức của object

Một công ty bán mĩ phẩm có rất nhiều sản phẩm, việc tìm kiếm một sản phẩm mất rất nhiều thời gian. Ở hệ thống cũ, để tìm kiếm một sản phẩm, nhân viên phải nhập vào mã sản phẩm, chương trình sẽ phải lặp từng sản phẩm trong mảng, để tìm sản phẩm có đúng mã đang tìm kiếm. Việc này không được tối ưu, và sếp mong muốn các bạn tối ưu hoá hệ thống tìm kiếm. Các mĩ phẩm của công ty là một mảng danh sách sản phẩm, yêu cầu phải chuyển đổi dữ liệu từ mảng sang object, để tối ưu hoá công cụ tìm kiếm sản phẩm bằng key trong object. Bởi vì đây là dữ liệu cũ, một dữ liệu đã bị lỗi, hoặc không có dữ liệu, nên giám đốc yêu cầu bạn phải xoá bỏ nó ra.

Xem đoạn code bên dưới để hiểu hơn về đề bài. **newData** là kết quả của đề bài yêu cầu, bạn phải sử dụng các phương thức trong object.

```
// Dữ liệu đầu vào được cung cấp
const oldData = [
  {code: 'ab', name: 'Son môi'},
  {code: 'ac', name: 'Sữa rửa mặt'},
  {code: null, name: null},
  {code: null, name: ''},
];

// Dữ liệu đã được xử lý bằng các phương thức của mảng
const newData = {
  ab: {code: 'ab', name: 'Son môi'},
  ac: {code: 'ac', name: 'Sữa rửa mặt'},
};

// Lấy thông tin mỹ phẩm có mã "ab" newData['ab']
// Kết quả: {code: 'ab', name: 'Son môi'}
```

Hướng dẫn:

- Đầu tiên bạn cần viết một hàm để chuyển đổi từ array sang object. Các bạn sẽ sử dụng phương thức **Object.fromEntries**, code mẫu như bên dưới:

```
const parseArrayToObject = ({array = [], keyId = ''}) =>
  Object.fromEntries(
    array?.map(item => [keyId ? item?.[keyId] : item, item]) || [],
  );
```

- Sau khi đã có được object, các bạn sẽ cần loại bỏ các dữ liệu không đạt yêu cầu (không có **code** hoặc **name** hoặc dữ liệu **null**, **undefined**). Để thực hiện điều này các bạn cần lặp ra các phần tử trong object, sử dụng phương thức **Object.keys**

```
const filterObject = obj => {  
  Object.keys(obj).forEach(key => { ...  
  });  
  return obj;  
};
```

- Sau đó bạn cần thêm các điều kiện bên trong forEach để kiểm tra sản phẩm có dữ liệu hợp lệ không.

### BÀI 3: Xử lý bất đồng bộ trong javascript

Hãy viết một chương trình bất đồng bộ như nhau. Bạn được cung cấp 3 promise, công việc của bạn là phải chờ 3 promise này thực hiện xong.

Yêu cầu thứ nhất, nếu một promise thất bại, thì phải lập tức dừng không chạy các promise khác nữa, và log ra màn hình lỗi của promise. Nếu tất cả promise hoàn thành, log ra màn hình thông báo thành công.

Yêu cầu thứ hai, chương trình sẽ luôn chạy bất kể promise nào thất bại hay thành công. Nếu hoàn thành xong tất cả promise log ra màn hình thông báo thành công cho người dùng.

3 promise được cung cấp sẽ như sau:

```
const firstPromise = new Promise((resolve, reject) => {
  setTimeout(() => {
    resolve('foo');
  }, 2000);
});

const secondPromise = new Promise((resolve, reject) => {
  setTimeout(() => {
    reject('Error: some bug');
  }, 2000);
});

const getList = async () =>
  await fetch('https://64d8a86c5f9bf5b879ce6dd9.mockapi.io/api/v1/moviesNow');
```

Hướng dẫn:

- Bạn có thể thấy hàm **getList** sử dụng **async await** được javascript xem là một **Promise**.
- Để hoàn thành yêu cầu thứ nhất, các bạn sử dụng **Promise.all** kèm theo sử dụng **then, catch** để log ra màn hình khi promise kết thúc công việc.
- Để hoàn thành yêu cầu thứ 2, các bạn sử dụng **Promise.allSettled** kèm theo sử dụng **then, catch** và **finally** để log ra màn hình khi promise kết thúc công việc.

#### BÀI 4: GV CHO THÊM

#### **\*\*\* YÊU CẦU NỘP BÀI:**

Sv nén file bao gồm các yêu cầu đã thực hiện trên, nộp lms đúng thời gian quy định của giảng viên. Không nộp bài coi như không có điểm.

--- Hết