**Bài tập bài 12: Javascript nâng cao (Tiết 2)**

**Câu 01: In ra từng phần tử của mảng**

Đề bài:

Viết một hàm đặt tên là **forEachTest(array, callback)** nhận vào hai tham số:

Tham số đầu tiên là một mảng.

Tham số thứ hai là một hàm callback (Đặt tên cho hàm callback là **consoleLog()**).

Gọi hàm callback **consoleLog()** này với từng phần tử của mảng **array** sau đó in ra màn hình console từng phần tử của mảng đó.

Ví dụ:

Mảng [1, 2, 3, 4, 5] trả về:

1

2

3

4

5

Mảng ["Lê Văn A", "Nguyễn Thị B", "Đỗ Thị C"] trả về:

Lê Văn A

Nguyễn Thị B

Đỗ Thị C

Đáp án:

function consoleLog(item) {

console.log(item);

}

function forEachTest(array, callback) {

for (let i = 0; i < array.length; i++) {

callback(array[i]);

}

}

const arrayTest1 = [1, 2, 3, 4, 5];

forEachTest(arrayTest1, consoleLog);

// Trả về:

// 1, 2, 3, 4, 5

const arrayTest2 = ["Lê Văn A", "Nguyễn Thị B", "Đỗ Thị C"];

forEachTest(arrayTest2, consoleLog);

// Trả về:

// Lê Văn A

// Nguyễn Thị B

// Đỗ Thị C

**Câu 02: Cộng từng cặp phần tử liên tiếp của mảng**

Đề bài:

Viết một hàm có tên **forEachPair()** nhận vào hai tham số:

Tham số đầu tiên là một mảng.

Tham số thứ hai là một hàm callback (Đặt tên cho hàm callback là **consoleLog()**).

Gọi hàm callback **consoleLog()** này với từng cặp phần tử liên tiếp của mảng **array** sau đó in ra màn hình console tổng của từng cặp phần tử liên tiếp đó.

Ví dụ:

Mảng [1, 2, 3, 4, 5] trả về:

3 *(Lấy 1 + 2)*

5 *(Lấy 2 + 3)*

7 *(Lấy 3 + 4)*

9 *(Lấy 4 + 5)*

Mảng [5, 20, 30, 60, 90] trả về:

25 *(Lấy 5 + 20)*

50 *(Lấy 20 + 30)*

90 *(Lấy 30 + 60)*

150 *(Lấy 60 + 90)*

Đáp án:

function consoleLog(a, b) {

console.log(a + b);

}

function forEachPair(array, callback) {

for (let i = 0; i < array.length - 1; i++) {

callback(array[i], array[i + 1]);

}

}

const arrayTest1 = [1, 2, 3, 4, 5];

forEachPair(arrayTest1, consoleLog); // 3, 5, 7, 9

const arrayTest2 = [5, 20, 30, 60, 90];

forEachPair(arrayTest2, consoleLog); // 25, 50, 90, 150

**Câu 03: Trả về một số nguyên ngẫu nhiên từ 1 đến 10**

Đề bài:

Tạo một Promise đơn giản để trả về một số nguyên ngẫu nhiên từ 1 đến 10.

Nếu random có trả ra giá trị thì in ra dòng: "Số ngẫu nhiên: 8" *(Với 8 là một số ngẫu nhiên)*.

Nếu random không thành công thì in ra dòng: "Lỗi: Không thể tạo số ngẫu nhiên."

Gợi ý: Dùng Math.random() để tạo ra một số random. (Đọc thêm tại: <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Math/random>)

Ví dụ:

Random ra số 8 —> Số ngẫu nhiên: 8

Random ra undefined —> Lỗi: Không thể tạo số ngẫu nhiên.

Đáp án:

function getRandomNumber() {

return new Promise((resolve, reject) => {

const randomNumber = Math.floor(Math.random() \* 10) + 1; // Giả sử random thành công, trả ra một số bất kỳ.

// const randomNumber = undefined; // Giả sử random lỗi, trả ra undefined.

if (randomNumber) {

resolve(randomNumber);

} else {

reject("Không thể tạo số ngẫu nhiên.");

}

});

}

getRandomNumber()

.then((number) => {

console.log("Số ngẫu nhiên:", number);

})

.catch((error) => {

console.error("Lỗi:", error);

});

**Câu 04: Lấy danh sách các số chẵn**

Đề bài:

Tạo một Promise để lấy danh sách các số chẵn từ một mảng các số nguyên cho trước.

Nếu trong mảng đó có số chẵn thì in ra mảng mới gồm các số chẵn lấy được từ mảng đó.

Nếu trong mảng đó không có số chẵn thì in ra thông báo: "Lỗi: Không tìm thấy số chẵn."

Ví dụ:

Truyền vào mảng [1, 2, 3, 4, 5, 6, 7, 8, 9, 10] trả về [2, 4, 6, 8, 10].

Truyền vào mảng [11, 13, 15, 17, 19] trả về "Lỗi: Không tìm thấy số chẵn.".

Đáp án:

function getEvenNumbers(numbers) {

return new Promise((resolve, reject) => {

const evenNumbers = numbers.filter((number) => number % 2 === 0);

if (evenNumbers.length > 0) {

resolve(evenNumbers);

} else {

reject("Không tìm thấy số chẵn.");

}

});

}

// Test 1

const numbersTest1 = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10];

getEvenNumbers(numbersTest1)

.then((evenNumbers) => {

console.log(evenNumbers);

})

.catch((error) => {

console.error("Lỗi:", error);

});

// Test 2

const numbersTest2 = [11, 13, 15, 17, 19];

getEvenNumbers(numbersTest2)

.then((evenNumbers) => {

console.log(evenNumbers);

})

.catch((error) => {

console.error("Lỗi:", error);

});

**Câu 05: Lấy data thông qua API và in ra màn hình.**

Đề bài:

API danh sách các nước trên thế giới: <https://restcountries.com/v2/all>

Vẽ ra giao diện danh sách các nước trên thế giới: bao gồm thủ đô, tên nước, quốc kỳ.

Giao diện hiển thị:

A group of flags with text

Description automatically generated

Đáp án:

<div id="country" class="country"></div>

\* {

box-sizing: border-box;

}

.country {

display: flex;

flex-wrap: wrap;

}

.country .inner-item {

width: 20%;

margin-bottom: 20px;

}

.country .inner-item .inner-box {

margin: 5px;

border: 1px solid #ddd;

padding: 10px;

height: 100%;

}

.country .inner-item .inner-box .inner-image {

width: 100%;

aspect-ratio: 4/3;

object-fit: contain;

}

.country .inner-item .inner-box .inner-title {

text-align: center;

font-weight: 600;

font-size: 14px;

}

const fetchApi = async (api) => {

const response = await fetch(api);

const result = await response.json();

return result;

};

fetchApi("<https://restcountries.com/v2/all>").then((data) => {

let htmls = data.map((item) => {

return `

<div class="inner-item">

<div class="inner-box">

<img class="inner-image" src="${item.flag}" alt="${item.capital} - ${item.name}" />

<div class="inner-title">

${item.capital} - ${item.name}

</div>

</div>

</div>

`;

});

htmls = htmls.join("");

const divCountry = document.querySelector("#country");

divCountry.innerHTML = htmls;

});

**Câu 06: Lấy địa chỉ IP của người dùng**

Đề bài:

Lấy địa chỉ IP của người dùng khi người dùng truy cập vào website của bạn.

Để lấy địa chỉ IP của người dùng bằng Fetch API trong JavaScript, bạn có thể sử dụng dịch vụ cung cấp thông tin địa chỉ IP, như link api <https://api.ipify.org?format=json>

Đáp án:

<div id="ipAddress"></div>

const API = "https://api.ipify.org?format=json";

fetch(API)

.then((response) => resp <div id="ipAddress"></div>

const API = "https://api.ipify.org?format=json";

fetch(API)

.then((response) => response.json())

.then((data) => {

const ipAddress = data.ip;

const elementIpAddress = document.querySelector("#ipAddress");

elementIpAddress.innerHTML = ipAddress;

})

.catch((error) => {

console.error("Lỗi:", error);

});onse.json())

.then((data) => {

const ipAddress = data.ip;

const elementIpAddress = document.querySelector("#ipAddress");

elementIpAddress.innerHTML = ipAddress;

})

.catch((error) => {

console.error("Lỗi:", error);

});

NOTE : Màu : Code html

Màu : Code CSS

Màu : Code JS