

LAB 2: DATA STRUCTURES, FUNCTIONS

MỤC TIÊU

Kết thúc bài thực hành này bạn có khả năng

- ✓ Sử dụng Data Structures (Array, Object, Strings,...)
- ✓ Modern Operators (Spread Operators, Rest Pattern and Parameters, Short Circuiting, The Nullish Coalescing Operators,...)
- ✓ Sử dụng hàm nâng cao: default parameters, value vs reference,...

NỘI DUNG

LAB2.1: (SECTION 9: DATA STRUCTURES. MODERN OPERATORS AND STRINGS > CODING CHALLENGE #3)

Hãy tiếp tục với app đặt cược của chúng ta! Lúc này, chúng ta có 1 mảng kết hợp theo dạng [key,value] tương ứng với những đoạn log ghi lại những sự kiện xảy ra trong trận đấu.

Trong đó:

-> key = phút diễn ra

-> value = tên hành động diễn ra (ghi bàn, thay người, nhận thẻ đỏ,...)

Bây giờ chúng ta có 1 mảng liên kế gồm các sự kiện xảy ra trong trận đấu cùng thời gian tương ứng

1. Hãy tạo ra 1 mảng các sự kiện khác nhau xảy ra trong suốt trận đấu (lưu ý là không trùng nhau)
2. Sau khi trò chơi kết thúc, phát hiện ra rằng thẻ vàng phút 64 là không đúng. Hãy xóa nó khỏi danh sách sự kiện trong trận đấu.
3. In ra console theo định dạng như sau: "1 sự kiện xảy ra, trung bình mỗi 9 phút". Lưu ý rằng mỗi trận đấu có 90 phút.

4. Lập toàn bộ mảng sự kiện và in ra màn hình, đánh dấu đối với mỗi sự kiện trong trận đấu diễn ra trong hiệp 1 hay hiệp 2,

Ví dụ: [FIRST HALF] 17: ⚽ GOAL

LAB2.2: (SECTION 9: DATA STRUCTURES. MODERN OPERATORS AND STRINGS > CODING CHALLENGE #4)

Viết 1 chương trình nhận vào 1 danh sách các biến được viết theo kiểu underscore (ví dụ: ten_bien) và chuyển nó sang dạng camelCase (ví dụ: tenBien)
Đầu vào sẽ đến từ một vùng văn bản được chèn vào DOM (xem mã bên dưới) và chuyển đổi sẽ xảy ra khi nhấn nút.

- Dưới đây là dữ liệu đầu vào cần test:

underscore_case
first_name
Some_Variable
calculate_AGE
delayed_departure

- Đầu ra mong muốn (Hãy log các kết quả này ra màn hình để đối chiếu với kết quả phía dưới)

underscoreCase ✓
firstName ✓✓
someVariable ✓✓✓
calculateAge ✓✓✓✓
delayedDeparture ✓✓✓✓✓

Lưu ý 1: Hãy lưu ý rằng ký tự nào xuất hiện 1 ký tự xuống dòng "\n" trong đoạn text 😊

Lưu ý 2: Giải pháp đơn giản cho phần này là tách ra 2 từ từ ký tự đầu vào ví dụ: ("a_b" -> "a" và "b")

Lưu ý 3: Đừng để ý về dấu tick kia ✓. Vấn đề sẽ được giải quyết khi bạn chuyển đổi 😊

Lưu ý 4: Thử thách này có thể sẽ khó với bạn, vì vậy hãy bắt đầu xem cách giải trong trường hợp cảm thấy không có hướng giải quyết.

Sau cùng, hãy test các kết quả đầu vào của bạn xem có đúng không ?

LAB2.3: (SECTION 9: DATA STRUCTURES. MODERN OPERATORS AND STRINGS > STRING METHOD PRACTICE)

Thực hiện code ví dụ sau

```
// String Methods Practice

const flights =

'_Delayed_Departure;fao93766109;txl2133758440;11:25+_Arrival;bru0943384722;fao93766109;11:45+_Delayed_Arrival;hel7439299980;fao93766109;12:05+_Departure;fao93766109;lis2323639855;12:30';

// 🚫 Delayed Departure from FAO to TXL (11h25)
//           Arrival from BRU to FAO (11h45)
// 🚫 Delayed Arrival from HEL to FAO (12h05)
//           Departure from FAO to LIS (12h30)

const getCode = str => str.slice(0, 3).toUpperCase();

for (const flight of flights.split('+')) {
  const [type, from, to, time] = flight.split(';');
  const output = `${type.startsWith('_Delayed') ? '🚫' : ''}${type.replaceAll(
    '_',
    ' '
  )} ${getCode(from)} ${getCode(to)} (${time.replace(':', 'h')})`.padStart(36);
  console.log(output);
}
```

LAB2.4: (SECTION 10: A CLOSER LOOK AT FUNCTIONS > CODING CHALLENGE #2)

Lấy hàm IIFE dưới đây, và ở cuối function hãy gắn 1 event thay đổi màu của thẻ 'h1' sang màu xanh dương (blue) sau mỗi lần click vào element body. Chúng ta không select đến thẻ h1 đó nữa!

Và sau đó hãy tự giải thích lại cho chính mình hoặc bạn học của mình Tại sao nó lại hoạt động!

Hãy bình tĩnh nghĩ xem chính xác thì khi nào callback được thực thi, và biến được sử dụng trong ví dụ này có ý nghĩa gì?.

```
(function () {
```

```
const header = document.querySelector('h1');  
header.style.color = 'red';  
document.querySelector('body').addEventListener('click', function () {  
    header.style.color = 'blue';  
});  
})();
```

--- Hết ---