Cho đoạn code js :

// The scope of `random` is too loose

const random = Math.floor(Math.random() \* 3);

const getRandEvent = () => {

  if (random === 0) {

    return 'Marathon';

  } else if (random === 1) {

    return 'Triathlon';

  } else if (random === 2) {

    return 'Pentathlon';

  }

};

// The scope of `days` is too tight

const getTrainingDays = event => {

  if (event === 'Marathon') {

    let days = 50;

  } else if (event === 'Triathlon') {

    let days = 100;

  } else if (event === 'Pentathlon') {

    let days = 200;

  }

  return days;

};

// The scope of `name` is too tight

const logEvent = event => {

  const name = 'Nala';

  console.log(`${name}'s event is: ${event}`);

};

const logTime = days => {

  const name = 'Nala';

  console.log(`${name}'s time to train is: ${days} days`);

};

const event = getRandEvent();

const days = getTrainingDays(event);

// Define a `name` variable. Use it as an argument after updating logEvent and logTime

logEvent(event);

logTime(days);

Là một vận động viên dày dạn kinh nghiệm, một trong những hoạt động yêu thích của bạn là chạy marathon. Bạn sử dụng một dịch vụ có tên là Ngày đào tạo sẽ gửi cho bạn tin nhắn về sự kiện bạn đã đăng ký và những ngày bạn còn lại để tập luyện. Chương trình hiện đang sử dụng sai phạm vi cho các biến.

Todo List:

1) Hãy bắt đầu bằng cách chạy tệp trainingDays.js. Trong giao diện điều khiển, chúng ta có thể thấy rằng chương trình đã bị hỏng!

**Lý tưởng** hàm getRandEvent () chọn ngẫu nhiên một sự kiện. Hàm getTrainingDays () trả về số ngày để đào tạo dựa trên sự kiện đã chọn. Các hàm logEvent() và logTime() in tên vận động viên, sự kiện và số ngày vào bảng điều khiển.

Nhưng các biến có phạm vi kém(poorly scoped variables) đang gây ra lỗi.

2) Để tránh ReferenceError, hãy khai báo days trong hàm getTrainingDays, trước câu lệnh if.

3) Chạy lại chương trình: không có lỗi, nhưng số ngày là undefined! Các biến ngày mới đang được xác định trong phạm vi của mỗi câu lệnh if / else if.

Xóa ba let trong câu lệnh if / else if

4) Chạy lại chương trình: đã sửa! Bây giờ các câu lệnh if / else if đang thay đổi days ban đầu thay vì xác định một ngày mới.

5) Các hàm log – logEvent () và logTime () - sử dụng cùng một biến tên. Dường như có một vấn đề với việc xác định phạm vi; có code trùng lặp! Code trùng lặp có thể chỉ ra rằng một biến có phạm vi hẹp.

Hãy tránh sự lặp lại này bằng cách thêm name là param đầu tiên cho mỗi hàm.

6) Di chuyển biến name sang phạm vi toàn cục.

7) Truyền name làm tham số đầu tiên cho logEvent () và logTime ().

8) Chạy chương trình

9) Hãy thử các chức năng cho đối thủ cạnh tranh khác. Sao chép và dán mã này cuối file code js.

const event2 = getRandEvent();  
const days2 = getTrainingDays(event2);  
const name2 = 'Warren';  
  
logEvent(name2, event2);  
logTime(name2, days2);

10) Chạy chương trình. Các sự kiện được chỉ định ngẫu nhiên, nhưng Nala và Warren đang chạy các events giống nhau!

11) Chúng ta thấy rằng biến random được xác định trong phạm vi toàn cục. Mỗi lần getRandEvent () được gọi, nó sẽ sử dụng cùng một giá trị.

Ở đầu tệp, di chuyển biến ngẫu nhiên từ phạm vi toàn cục sang phạm vi khối trong hàm getRandEvent.

12) Chương trình dễ bảo trì hơn và ít sai sót hơn nhờ vào việc tối ưu lại code. Chạy chương trình một vài lần để đảm bảo kết quả là ngẫu nhiên.