**Lab 10: Flying First Class**

For a better view of the code, please check the GitHub repo in the link below:

HTML GitHub link:

<https://github.com/AhmedAbdelRazak/RCC/blob/master/Chapter10/Lab10.html>

JavaScript GitHub link:

<https://github.com/AhmedAbdelRazak/RCC/blob/master/Chapter10/Lab.js>

Console output: <https://github.com/AhmedAbdelRazak/RCC/blob/master/Chapter10/Lab10_Console%20Screenshot.jpg>

Alert window output: <https://github.com/AhmedAbdelRazak/RCC/blob/master/Chapter10/Lab10_Alert%20Window%20Output.jpg>

Here is the HTML for the code:

<!DOCTYPE html>

<html lang="en">

    <head>

        <meta charset="utf-8" />

        <title>Flying First Class</title>

        <script src="Lab.js"></script>

    </head>

    <body></body>

</html>

Here is the JavaScript:

//need an array of 4 or more passengers with name, paid status: true or false, and ticket type (firstclass, premium, and coach)

var passengers = [

    { name: "Jane Doloop", paid: true, ticket: "coach" },

    { name: "Dr. Evel", paid: true, ticket: "firstclass" },

    { name: "Sue Property", paid: false, ticket: "firstclass" },

    { name: "John Funcall", paid: true, ticket: "premium" },

];

//need a function to process the passengers. Iterate over the length of the array and call the function on each passenger. If the passenger fails the test, they can't fly. (either they haven't paid or they are on the no fly list)

function processPassengers(passengers, test) {

    for (var i = 0; i < passengers.length; i++) {

        if (test(passengers[i])) {

            return false;

        }

    }

    return true;

}

//Put Dr. Evel on the No Fly List because he is Evil! Spelling and case must match the name that you put in the array. If you want to put me on the no fly list instead of Dr. Evel, go for it....

function checkNoFlyList(passenger) {

    //IMPORTANT\*\*\*\* I will never put Dr. Ian in this list ;)

    return passenger.name === "Dr. Evel";

}

//Check to see who hasn't paid... it will get that from the array.

function checkNotPaid(passenger) {

    return !passenger.paid;

}

//This will print out who has and has not paid using string concatenation into a var called message. If the passenger has paid then the message will read message + " has paid"

function printPassenger(passenger) {

    var message = passenger.name;

    if (passenger.paid) {

        message = message + " has paid";

    } else {

        message = message + " has not paid";

    }

    console.log(message);

    return false;

}

//

// plane can only fly if every passenger is on the fly list

//

var allCanFly = processPassengers(passengers, checkNoFlyList);

if (!allCanFly) {

    console.log(

        "The plane can't take off: we have a passenger on the no fly list.",

    );

}

//

// plane can only fly if every passenger has paid

//

var allPaid = processPassengers(passengers, checkNotPaid);

if (!allPaid) {

    console.log("The plane can't take off: not everyone has paid.");

}

//

// we don't care about the result here; we're just using

// processPassengers to display the passenger list

//

//Passing passengers and printPassenger() as two arguments to "processPassengers" function

processPassengers(passengers, printPassenger);

//first class gets a cocktail or wine, premium gets wine, cola, or water, and coach gets cola or water.

function createDrinkOrder(passenger) {

    var orderFunction;

    if (passenger.ticket === "firstclass") {

        orderFunction = function () {

            alert("Would you like a cocktail or wine?");

        };

    } else if (passenger.ticket === "premium") {

        orderFunction = function () {

            alert("Would you like wine, cola or water?");

        };

    } else {

        orderFunction = function () {

            alert("Your choice is cola or water.");

        };

    }

    return orderFunction;

}

//first class gets chicken or pasta, premium gets a snack box or cheese plate, and coach gets peanuts or pretzels.

function createDinnerOrder(passenger) {

    var orderFunction;

    if (passenger.ticket === "firstclass") {

        orderFunction = function () {

            alert("Would you like chicken or pasta?");

        };

    } else if (passenger.ticket === "premium") {

        orderFunction = function () {

            alert("Would you like a snack box or cheese plate?");

        };

    } else {

        orderFunction = function () {

            alert("Would you like peanuts or pretzels?");

        };

    }

    return orderFunction;

}

//alert window to pick up the trash

function pickupTrash() {

    alert("Can I have your trash, please?");

}

//serve customer will serve drinks, get dinner order, and pick up trash.

//8th to be executed and the functions w/n function serveCustomer should be the 9th to be executed

function serveCustomer(passenger) {

    var getDrinkOrderFunction = createDrinkOrder(passenger);

    var getDinnerOrderFunction = createDinnerOrder(passenger);

    getDrinkOrderFunction();

    getDinnerOrderFunction();

    getDrinkOrderFunction();

    // getDrinkOrderFunction();

    // getDrinkOrderFunction();

    // pick up trash

    pickupTrash();

}

//this will iterate over the customers, serving them one by one.

function servePassengers(passengers) {

    for (var i = 0; i < passengers.length; i++) {

        serveCustomer(passengers[i]);

    }

}

//of course we need to call servePassengers ( ) with passengers as a parameter below. I didn't include it, you have to write it.

servePassengers(passengers);

Here is the Output in the console:

Graphical user interface, text, application, email

Description automatically generated

Here is the output in the alert window:

Graphical user interface

Description automatically generated