TRABAJO ENTREGABLE

En base a las definiciones técnicas tomadas con el equipo deberás declarar la variable autos. Esta debe contener los siguientes vehículos:

1. El primer auto es un Ford Fiesta Azul, del 2019, con 200 kilómetros, cuyo precio es 150000, disponible en 12 cuotas, con la patente APL123 que no está vendido.
2. El segundo auto es un Toyota Corolla Blanco, del 2019, 0 kilómetros, cuyo precio es 100000, disponible en 14 cuotas, con la patente JJK116 que no está vendido.

Cada auto debe tener los siguientes atributos: marca, modelo, precio, km, color, cuotas, anio, patente, vendido.

RESPUESTA:

let autos = [

{

marca: "Ford",

modelo: "Fiesta",

precio: 150000,

km: 200,

color: "Azul",

cuotas: 12,

anio: 2019,

patente: "APL123",

vendido: false

},

{

marca: "Toyota",

modelo: "Corolla",

precio: 100000,

km: 0,

color: "Blanco",

cuotas: 14,

anio: 2019,

patente: "JJK116",

vendido: false

}

]

module.exports = autos;

# Etapa 2

En esta primera etapa, necesitamos requerir tu módulo autos y guardarlo en la variable autosImportados que se encuentra en la misma carpeta del archivo donde estás trabajando. Recordá que es el mismo archivo que creaste en la etapa anterior.

Además, necesitarás crear un objeto literal llamado concesionaria que contendrá todas las funcionalidades que el cliente solicita.

Por último, nuestro objeto literal debe tener un atributo llamado autos que contenga la lista de automóviles importada anteriormente.

RESPUESTA

let autosImportados = require("./autos.js");

const concesionaria = {

autos: autosImportados

};

Ahora que la concesionaria tiene los autos, es posible crear la funcionalidad buscarAuto que reciba por parámetro la patente y devuelva el auto al cual le corresponde. En caso de no encontrar el mismo, deberá retornar null.

Para que todo funcione tenés que agregar el código que escribiste en el ejercicio anterior.

Te sugerimos utilizar this.autos en lugar de autosImportados.

Recordá que cuando trabajamos con Arrays tenemos varios métodos para iterar sobre ellos, y uno de ellos nos sirve para **buscar**.

PISTAS:

Cuidado al usar arrow functions, te recomendamos usar function().

Recordá que podes usar Array.prototype.find para buscar un objeto dentro de un array

# Etapa 4

Ahora, María les pide que agreguen la funcionalidad de venderAuto que recibe la patente y, en caso de encontrar al automóvil, le asigna el estado de vendido.

¿Cómo lo resuelven?

**Recordatorio**: Para comenzar tenés que agregar el código que escribiste en el ejercicio anterior. Para resolver esta nueva funcionalidad, tendrás que utilizar la función buscarAuto, **la cual ya nos devuelve el objeto literal a modificar.**

PISTAS:

Recordá que tenés que usar el this para acceder a la funcionalidad del mismo objeto literal.

Para esta etapa, recomendamos utilizar funciones normales en vez de arrow functions

# Etapa 5

La primera es poder contar, como concesionaria, con la habilidad de poder tener la lista de autos para la venta. A lo cual, María, cree que es una tarea sencilla que Juan y vos pueden encarar solos, usando la función autosParaLaVenta, aunque por las dudas ella les recuerda que no deberían de aparecer los autos que ya fueron vendidos.

Para comenzar, tenés que agregar el código que escribiste en el ejercicio anterior. Tené en cuenta que estamos optimizando nuestro código, por lo cual, deberíamos utilizar el método filter.

PISTAS:

Recordá utilizar el this para llamar a una función definida dentro del mismo objeto literal.

Recordá que el filter devuelve una lista.

# Etapa 6

María, contenta con el trabajo que realizaron, les pide otra funcionalidad extra. Resulta que a la concesionaria le suelen preguntar muy seguido cuáles de los autos **para la venta**son 0 km. Tené en cuenta que María considera que un auto 0 km es aquel que tenga un kilometraje menor a 100. Vas a tener que desarrollar la funcionalidad autosNuevos.

¿Cómo podés resolver esto reutilizando la función autosParaLaVenta?

Para comenzar, tenés que agregar el código que escribiste en el ejercicio anterior.

PISTASS:

Recordá utilizar el this para ejecutar funciones dentro del mismo objeto literal.

# Etapa 7

El cliente le pidió saber cuánto dinero generaron las ventas.

María te pide que completes la función listaDeVentas que devuelve una lista que contiene el precio de venta de cada auto vendido. A esto, Juan, que está al lado tuyo, se le escapa la frase "mmm.....estoy seguro que alguna función de arrays nos va a servir, pero no me acuerdo".

Para comenzar tenés que agregar el código que escribiste en el ejercicio anterior.

¿Te acuerdas cuál es el método de Array con el que puedes generar un nuevo Array?

PISTAS:

Recordá utilizar el this para llamar a una funcionalidad dentro del mismo objeto literal.

# Etapa 8

Terminada esta función, María te pide que resuelvas la funcionalidad de totalDeVentas, que justamente nos devuelva la sumatoria del valor de todas las ventas realizadas. Acá el único requerimiento técnico explícito es que utilices la función reduce, ¡a codear!

Recordá que si la lista esta vacía, el reduce falla en la ejecución. ¿Por qué puede pasar esto?

# Etapa 9

Muy contento el equipo por cómo viene el desarrollo, por la tarde, María te comenta que se agrega una funcionalidad muy importante: la de verificar si una persona puede comprar o no un auto. Esta permite al sistema definir si una persona al consultar por un auto, puede comprarlo. Las personas solo sacan los autos en cuotas y tomando dos factores como condición de compra. Una es el **costo total**: si el total de un auto excede lo que la persona considera caro, no va a comprar el auto. Otra condición es su **capacidad de pago en cuotas**: si la capacidad de pago en cuotas supera al costo de la cuota, va a poder pagarlo. Si **ambas condiciones se cumplen**, se realiza la compra.

Es por esto que María te pide que desarrolles la función puedeComprarque reciba por parámetro**un auto y una persona**y devuelva true si la misma puede comprar el auto.

Una persona va a ser representada mediante un objeto literal de la siguiente forma:

{  
nombre: “Juan”,  
capacidadDePagoEnCuotas: 20000,  
capacidadDePagoTotal: 100000  
}

Para comenzar tenés que agregar el código que escribiste en el ejercicio anterior.

PISTAS

Ambas condiciones deben cumplirse a la vez: una persona debe de poder pagar cada cuota y el total del auto.

# Etapa 10

Ahora, te comprometiste a realizarla. Así que manos a la obra. Hay que escribir la función autosQuePuedeComprar, que recibe una persona y devuelve la lista de autos que puede comprar.

La función debe de realizar los siguientes pasos:

1) Obtener los autos para la venta

2) Luego debemos retornar los que pueda comprar, ¿hay alguna manera de poder filtrar la lista de autos para la venta del punto 1 con el paso 2?

3) Por cada uno de los autos debe de probar si la persona puede comprarlo, ¿ya hay alguna funcionalidad que me permita hacer esto?.

Para comenzar tenés que agregar el código que escribiste en el ejercicio anterior.

PISTAS

No utilizar la función buscarautos