

ESCUELA POLITÉCNICA NACIONAL

# *Facultad de Ingeniería Software*

**Nombre:**  Sánchez Doménica **Fecha:** 01/12/2023

**COMPILADORES**

**TAREA PARTE 1**

**Tarea 1**

Juguemos a la floristería. Declara seis variables, recordando nombrarlas según su propósito:

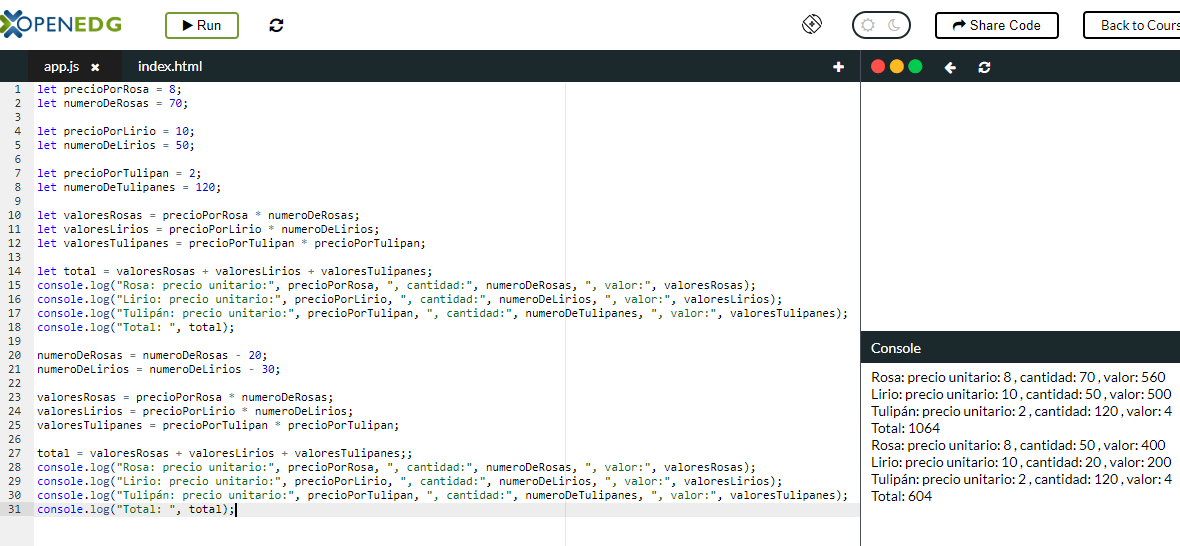
* el precio de una sola rosa (8) y el número de rosas que tienes (70)
* el precio de un solo lirio (10) y el número de lirios que tienes (50)
* el precio de un solo tulipán (2) y la cantidad de tulipanes que tienes (120)

Ahora declara tres variables, una para cada una de las rosas, lirios y tulipanes que tienes, en las que colocas su precio total. Inserta los valores correspondientes en las variables utilizando las variables declaradas en el paso anterior. Finalmente, declara una variable en la que almacenes el precio de todas tus flores (nuevamente, usa las variables anteriores para la inicialización). Muestra toda la información del inventario en la consola de la siguiente forma:



**Tarea 2**

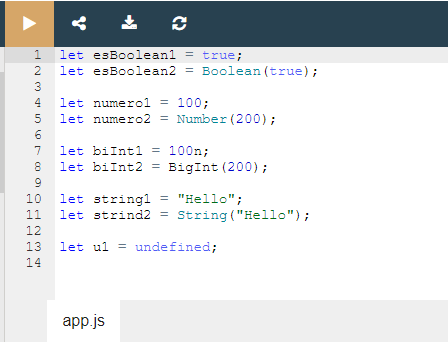
Modifica el código del ejemplo anterior. Supón que los precios de las flores serán constantes (no cambiarán). Declara e inicializa las variables restantes de la misma manera que en el ejemplo anterior. Muestra toda la información recopilada en la consola. Ahora disminuye el número de rosas en 20 y el de lirios en 30. Vuelve a mostrar toda la información recopilada en la consola.



**TAREAS DATOS PRIMITIVOS PARTE 2**

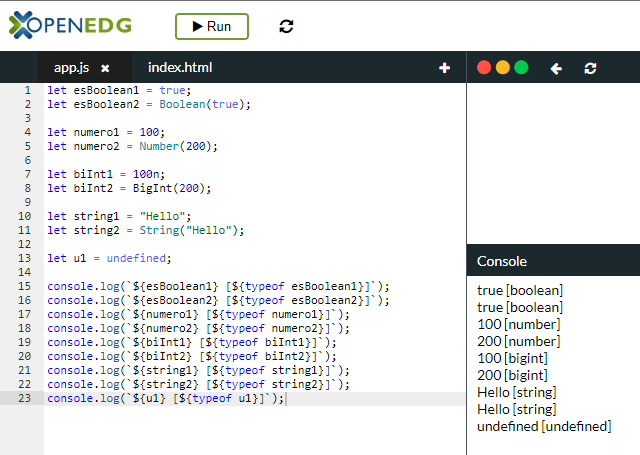
**Tarea 1**

Escribe un código que cree variables y las inicialice con valores Boolean, Number, BigInt, String y tipos undefined usando (cuando sea posible) literales y funciones constructoras.



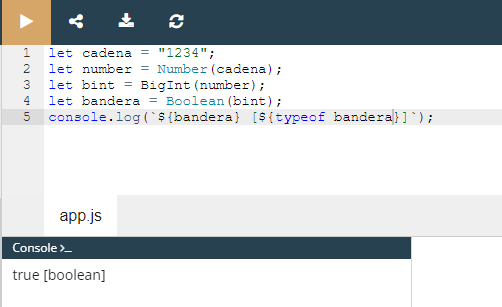
**Tarea 2**

Imprime todos los valores y todos los tipos de esos valores usando console.log. Intenta usar la interpolación de cadenas para mostrar el valor y el tipo al mismo tiempo con una sola llamada a console.log.



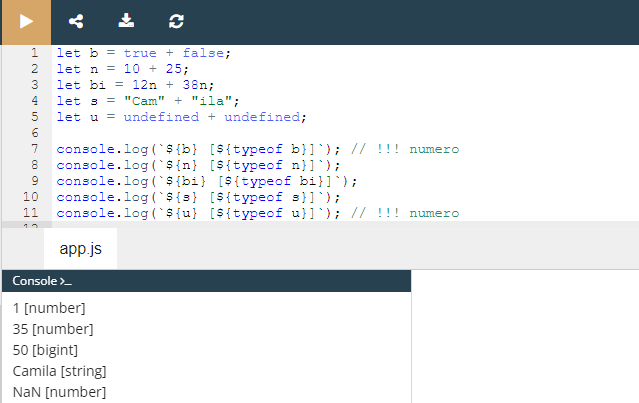
**Tarea 3**

Realizar una cadena de conversiones: crear un Boolean a partir de un BigInt creado a partir de un Number que se creó a partir de un String . Comienza con el valor "1234". ¿Es posible?



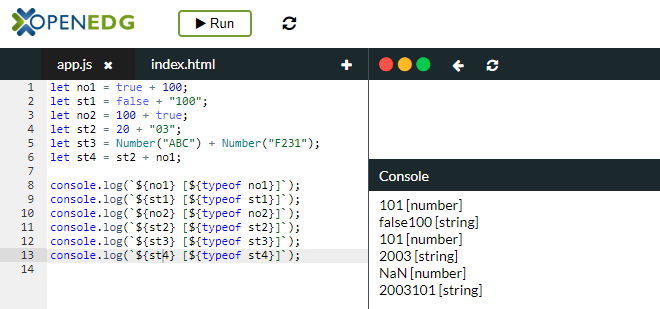
**Tarea 4**

Intenta agregar dos valores del mismo tipo y verifica el tipo de resultado. Pruébalo para todos los tipos primitivos.



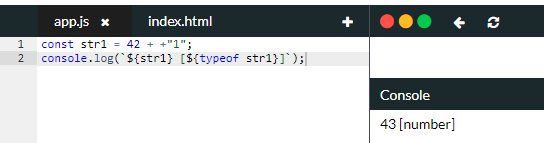
**Tarea 5**

Prueba sumar dos valores de diferentes tipos y verifica los resultados.



**Tarea 6**

Intenta modificar la línea const str1 = 42 + "1"; para obtener el resultado 43 (sin eliminar las comillas alrededor del 1 ).



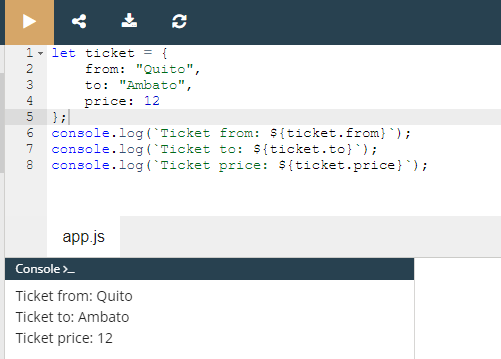
**DATOS COMPLEJOS PARTE 3**

**Tarea 1**

Crea un objeto que describa un boleto de tren y guárdalo en la variable ticket. El objeto debe tener tres campos:

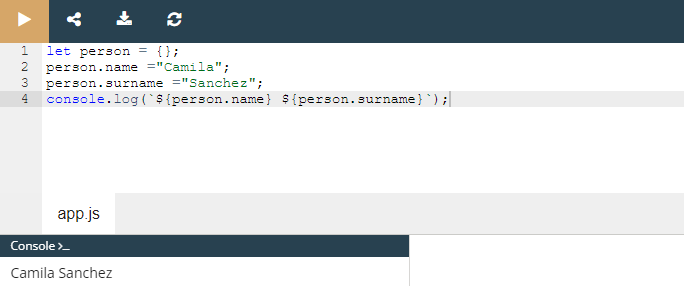
* estación inicial (el nombre clave es from, y como valor, proporciona el nombre de la estación más cercana en tu área)
* estación final (el nombre clave es to, y como valor, dar cualquier otra estación dentro de 100 km)
* el precio del boleto (el nombre clave es price, y como valor, proporciona la cantidad que te gustaría pagar por este boleto)

El objeto debe crearse usando llaves {}, en los que todos los campos se enumerarán inmediatamente. Luego muestra los valores de todos los campos del ticket en la consola.

. 

**Task 2**

Declara un objeto vacío y guárdalo en la variable person. Usando la notación de punto, agrega los campos name y surname al objeto ingresando tus datos como valores. Intenta mostrar los campos individuales en la consola.

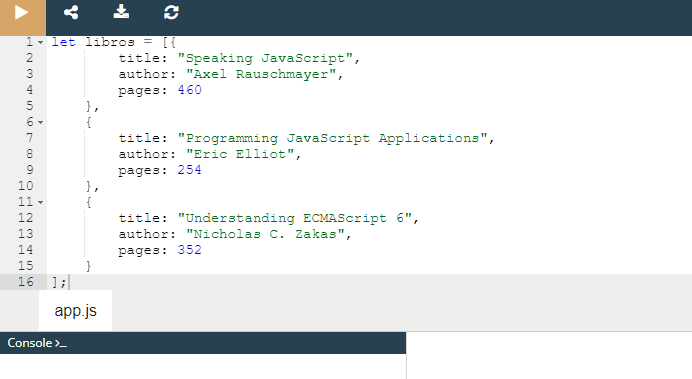


**Tarea 3**

Estamos creando una pequeña biblioteca de libros sobre programación en JavaScript. Tenemos tres libros y queremos preparar una lista de ellos. Almacenaremos tres datos de cada libro: el título, el autor y el número de páginas:

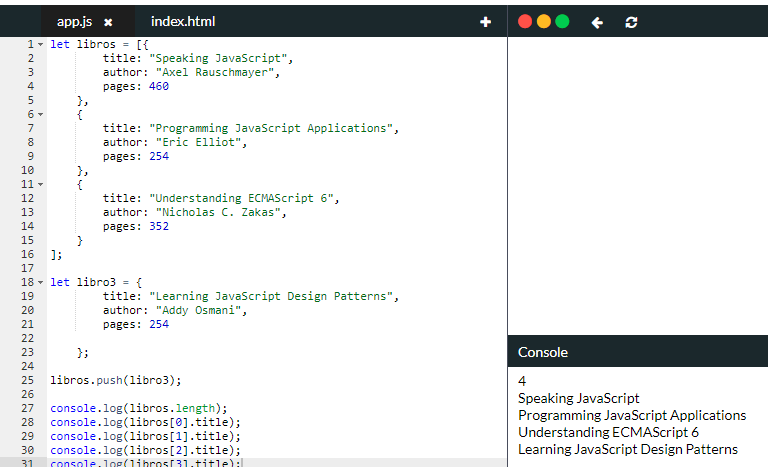
*Speaking JavaScript*, Axel Rauschmayer, 460;

* + *Programming JavaScript Applications*, Eric Elliott, 254;
  + *Understanding ECMAScript 6*, Nicholas C. Zakas, 352.
* Crea un arreglo de tres objetos que representen los libros. Cada objeto debe tener las siguientes propiedades: title, author, pages



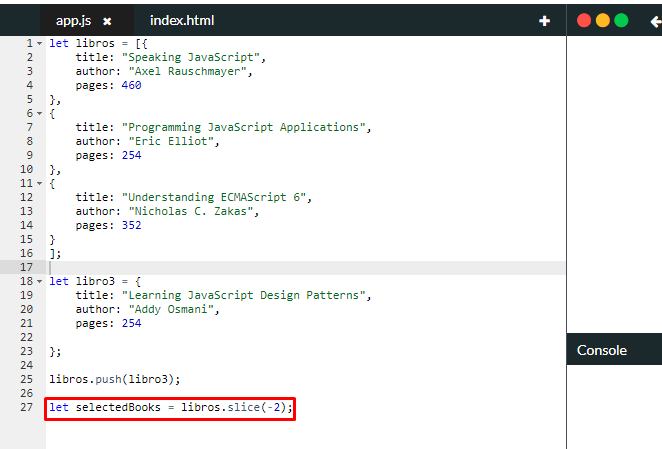
**Tarea 4**

Agregar un nuevo libro a la colección: *Learning JavaScript Design Patterns*, por Addy Osmani, 254 páginas. Usa el método apropiado para adjunta el libro al final del arreglo. Muestra la longitud del arreglo y, a su vez, todos los nombres de los libros en la colección.



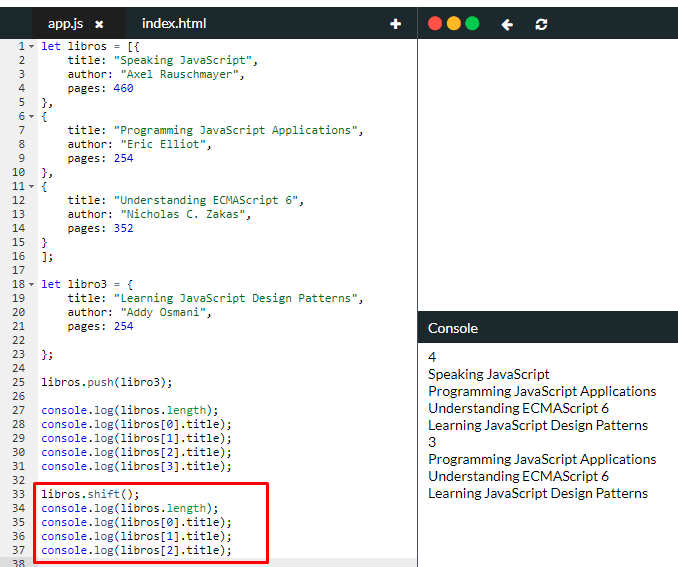
**Task 5**

Utiliza el comando slice para copiar los dos últimos libros al nuevo arreglo.



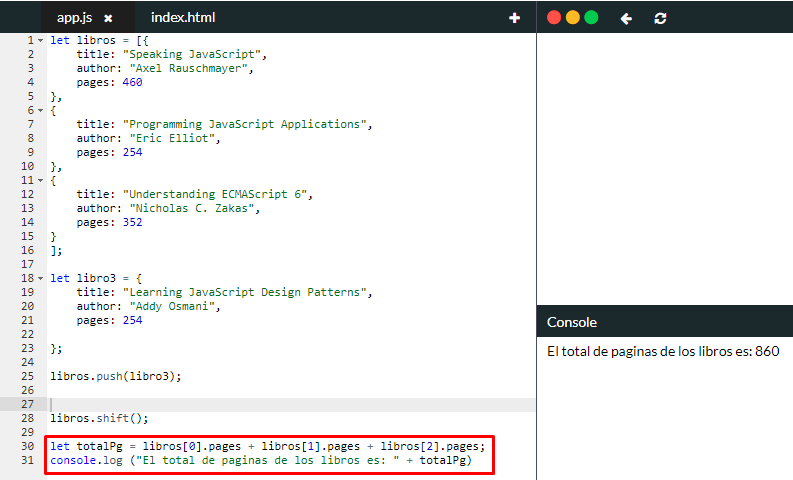
**Tarea 6**

El primer libro de la colección se pierde en circunstancias inexplicables. Ya has aceptado la pérdida, así que ahora elimínalo del arreglo. ¿Cuál método usarás para este propósito? Muestra la longitud del arreglo y todos los nombres de los libros de la colección a su vez.



**Tarea 7**

Muestra la suma de las páginas de todos los libros de la colección.



**COMENTARIOS CAP 4**

Existe un código que actualmente no funciona. Intenta arreglarlo usando solo comentarios. Intenta, si es posible, usar los atajos de teclado en tu editor para este propósito.  
"use strict";

const prefix = "username\_";

let userName = "Jack";

const userName = "Adam";

let prefixedUserName;

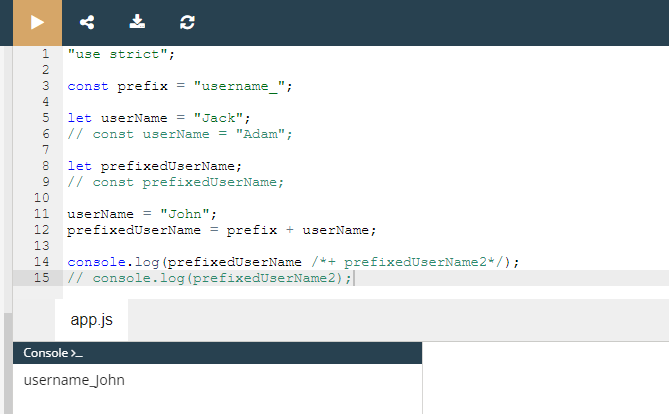
const prefixedUserName;

userName = "John";

prefixedUserName = prefix + userName;

console.log(prefixedUserName + prefixedUserName2);

console.log(prefixedUserName2);



**ARBOL NAVIDEÑO**

