FORMULACIÓN DE LAS ACTIVIDADES DE APRENDIZAJE

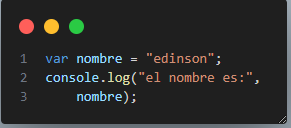
1. Actividades de reflexión inicial.

Desarrollo de la actividad tipo investigación en la cual el aprendiz debe consultar en internet los siguientes interrogantes y dar un ejemplo de cada uno:

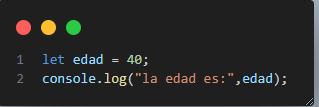
• Tipos de datos:

o Que son las variables var y las variables let (de un ejemplo)

* Var: tiene un ámbito de función y puede ser redeclarada o reasignada

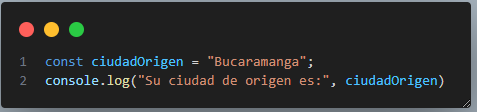


* Let: tiene un ámbito de bloque y puede ser reasignada, pero no redeclarada dentro del mismo ámbito



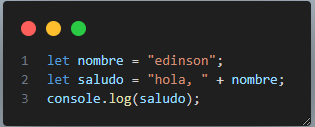
o Que son las constantes (conts) (de un ejemplo)

* Const: tiene un ámbito de bloque, no puede ser redeclarada ni reasignada.



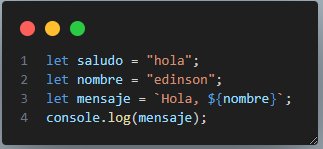
o Que son las cadenas de texto (Strings) (de un ejemplo)

* Es una secuencia de caracteres almacenadas entre comillas simples o dobles.



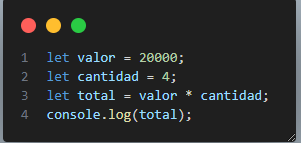
o Que son las plantillas de cadenas texto (Template Strings) (de un ejemplo)

los backticks (`…`) permiten interpolaicon de variables y hacer cadenas multilínea de forma sencilla.



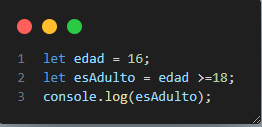
o Que son los números (Number) (de un ejemplo)

es un tipo de dato que se utiliza para representar valores numéricos, ya sean enteros o decimales



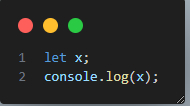
o Que son los Booleans (de un ejemplo)

es un tipo de dato que solo puede tener dos valores: true(verdadero) o false (falso). Los valores booleanos se utilizan para representar estados lógicos, como resultado de comparaciones o condiciones.

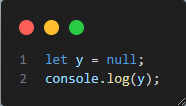


o Que son los undefined, null y Nan (de un ejemplo)

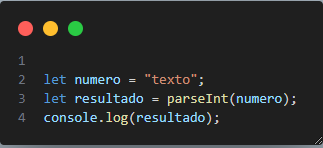
el valor undefined se asigna automáticamente a una variable que ha sido declarada pero no se le ha asignado ningún valor.



Null es valor que representa intencionalmente la ausencia de cualquier valor u objeto, es usado para indicar que una variable no tiene ningún valor asignado de manera dliberada.



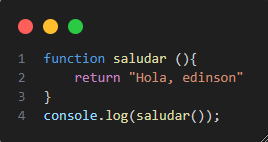
Nan es un valor que representa algo que no e numero valido. Aparece como resultado de operaciones matemáticas que no tienen sentido



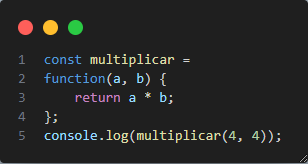
o Que son las funciones y que tipos de funciones existente en JavaScript (de ejemplos)

es un bloque de código diseñado para realizar una tarea especifica, las funciones periten reutilizar código y mejorar la modularidad del programa, una funicon se puede invocar o llamar en cualquier momento, lo que ejecuta el bloque de código definido dentro de ella.

**Function declarations**

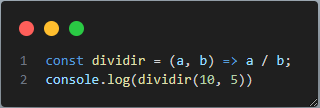


**Function Expressions**

****

**Arrow functions**

Las funciones flecha son una sintaxis mas concisa para escribir funciones introducida



Funciones anonimas

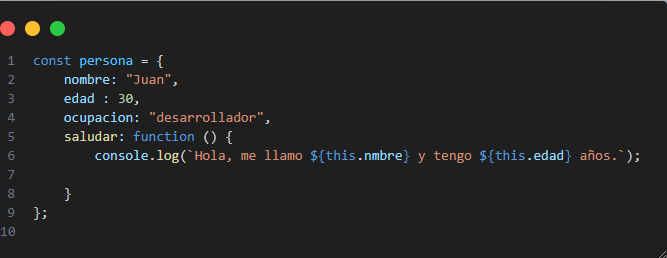
o Que son los arreglos (Arrays) (de ejemplos)

son estructuras de datos que permiten almacenar multiples elementos en una sola variable, donde cada elemeto tiene una posición o inidce, los arreglos son útiles para agrupar datos relacionados y accedera ellos mediante índices.



o Que son los objetos en JavaScript (de un ejemplo)

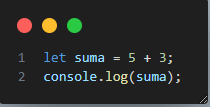
es una estructura que permite almacenar datos y funcionalidades relacionadas en forma de propiedades (valores) y métodos (funciones). Los objetos son fundamentales para representar entidades complejas, como personas, productos o configuraciones.



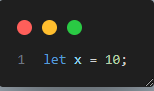
• Estructuras de control:

o Tipos de operadores (de un ejemplo de cada uno)

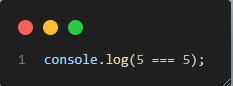
**Operador Aritmético**: Realiza operaciones matemáticas.



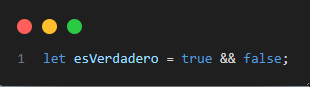
**Operador de Asignación**: Asigna valores a las variables.

****

**Operador de Comparación**: Compara dos valores.

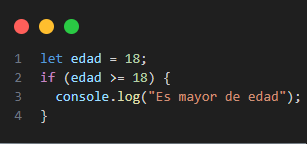
****

**Operador Lógico**: Realiza operaciones lógicas.

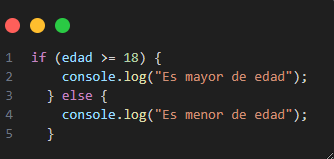


o Tipos de condicionales (if, if else) (de un ejemplo)

**if**: Ejecuta el bloque si la condición es verdadera.

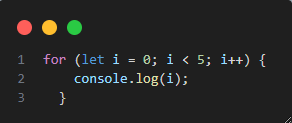


**if-else**: Ejecuta un bloque si la condición es verdadera, de lo contrario, ejecuta otro bloque.

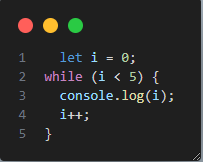


o Ciclos (Loops) (de un ejemplo de cada uno)

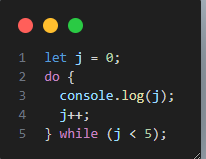
**for**: Ejecuta el bloque un número específico de veces.



**while**: Ejecuta el bloque mientras la condición sea verdadera.

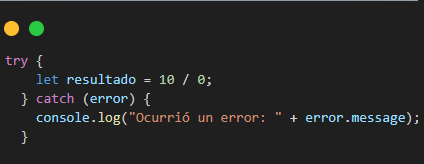


**do-while**: Ejecuta el bloque al menos una vez, luego verifica la condición.



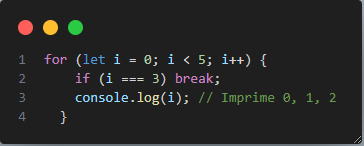
o Manejo de errores (de un ejemplo)

**try-catch**: Intenta ejecutar un bloque de código y captura cualquier error.

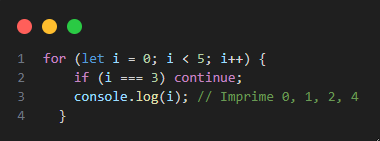


o Break & Continue (de un ejemplo de cada uno)

**break**: Termina un bucle.

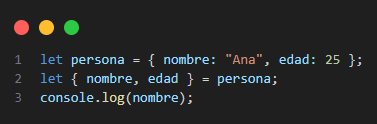


**continue**: Salta a la siguiente iteración del bucle.



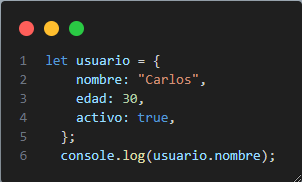
o Que es la destructuración (de un ejemplo)

La destructuración permite extraer valores de arreglos u objetos.



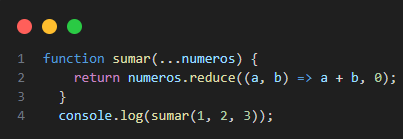
o Que son los objetos literales (de un ejemplo)

Son objetos creados con una sintaxis literal.

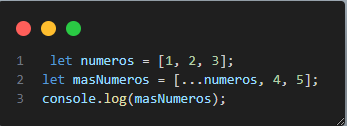


o Que son los parámetros REST y Operador Spread (de un ejemplo de cada uno)

**Parámetros REST**: Agrupan múltiples argumentos en un arreglo.

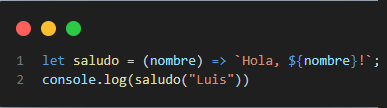


**Operador Spread**: Expande elementos de un arreglo o un objeto.



o Que son las arrow functions (de un ejemplo)

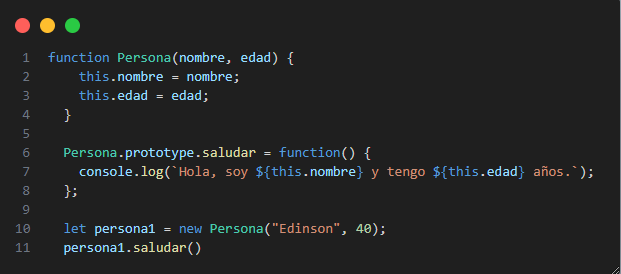
Son funciones de sintaxis concisa.



• Programación orienta a Objetos

o Que son los prototipos (de un ejemplo)

En JavaScript, todos los objetos tienen un prototipo, que es un objeto del cual heredan propiedades y métodos. Un prototipo es como una plantilla para crear objetos similares, permitiendo compartir funcionalidades sin duplicar código.



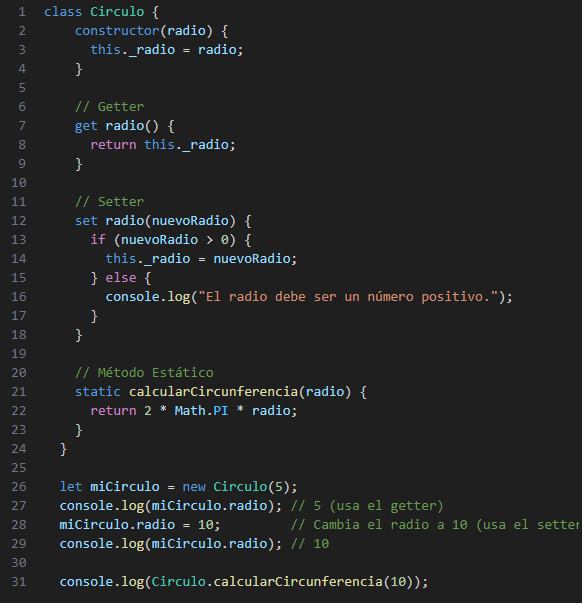
o Que es la herencia prototípica (de un ejemplo)

La herencia prototípica es una característica en JavaScript que permite que un objeto herede propiedades y métodos de otro. Para heredar, simplemente se asigna el prototipo de un objeto a otro, creando una cadena de prototipos.



o Que son los métodos estáticos, getters y setters (de un ejemplo de cada uno)

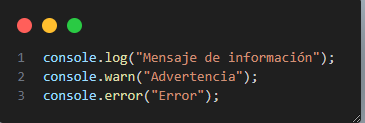
* **Métodos Estáticos**: Son métodos que pertenecen a la clase en lugar de a una instancia. Son útiles para crear utilidades relacionadas con la clase.
* **Getters y Setters**: Permiten obtener y establecer valores de propiedades de manera controlada.



• Objetos y funciones del lenguaje

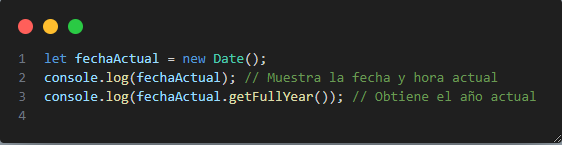
o Que es el objeto console (de un ejemplo)

El objeto console proporciona métodos para mostrar información de depuración en la consola del navegador o en la terminal. Es útil para imprimir valores, advertencias y errores durante el desarrollo.



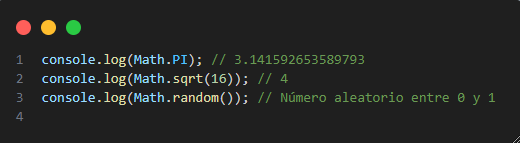
o Que es el objeto date (de un ejemplo)

El objeto Date se utiliza para trabajar con fechas y horas en JavaScript. Permite obtener y manipular información de tiempo.



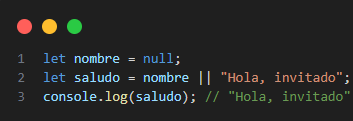
o Que es el objeto Math (de un ejemplo)

El objeto Math ofrece propiedades y métodos para realizar operaciones matemáticas, como redondeos, exponentes y trigonometría.



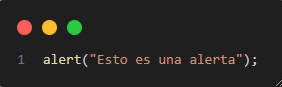
o Que es el operador de cortocircuito en javascript (de un ejemplo)

El operador de cortocircuito (|| y &&) evalúa expresiones y, según la condición, corta la evaluación y devuelve un valor.

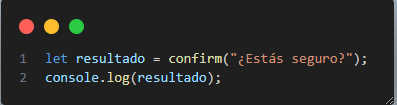


o Que es el alert, confirm y pormpot (de un ejemplo de cada una)

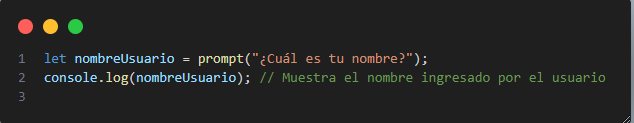
**alert**: Muestra un mensaje de alerta.



**confirm**: Muestra un mensaje de confirmación y devuelve true si el usuario acepta y false si cancela.



**prompt**: Muestra un cuadro de texto donde el usuario puede ingresar información.

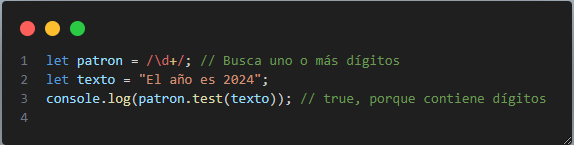


o Que son las expresiones regulares (de un ejemplo)

Las expresiones regulares son patrones utilizados para buscar coincidencias en cadenas de texto, útiles para validación y manipulación de texto.

o Que son las funciones anónimas autoejecutables (de un ejemplo)

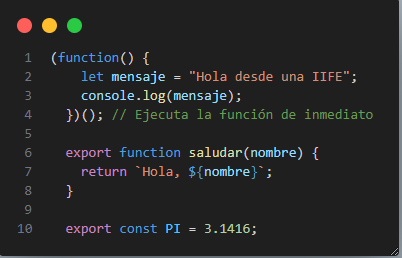
Una IIFE (Immediately Invoked Function Expression) es una función que se define y ejecuta de inmediato. Se utiliza para crear un ámbito de variables sin contaminar el global.

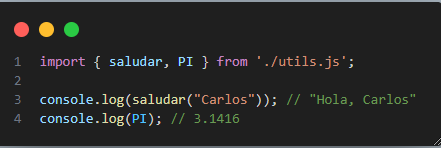


o Que son los módulos (Import y export) (de un ejemplo)

Los módulos permiten dividir el código en archivos separados para mantener la estructura y la reutilización. export se usa para hacer que funciones o variables estén disponibles fuera del archivo, y import se usa para traer esas funcionalidades a otros archivos.

Supongamos que tienes dos archivos: utils.js y main.js.





Pantallazo de eliminación de la rama desarrollo

