**Universidad Tecnológica Metropolitana**

**4F**

**ACTIVIDAD #5 y 6**

**Nombre alumno:**

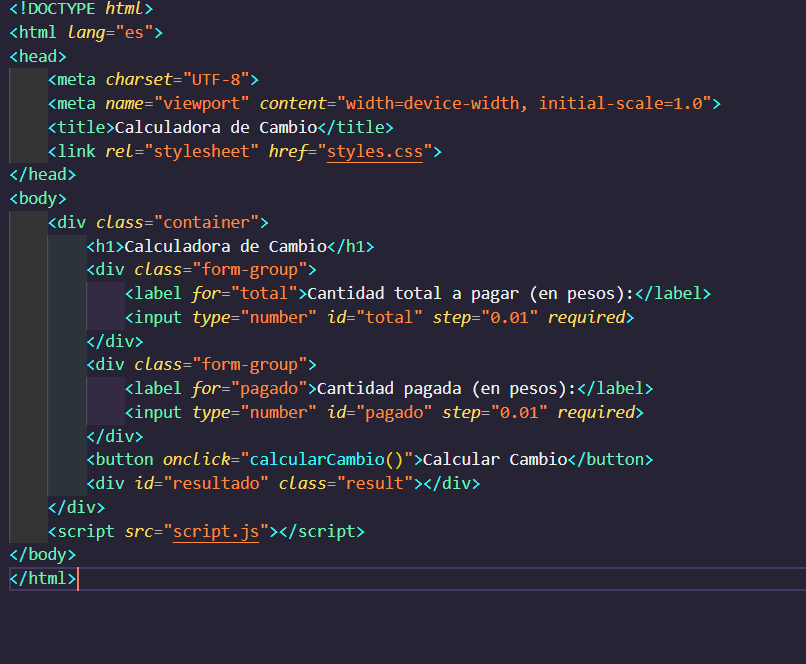
**Fernando David Sanchez Sacnhez**

**Nombre del profesor(a): Ruth Dominguez**

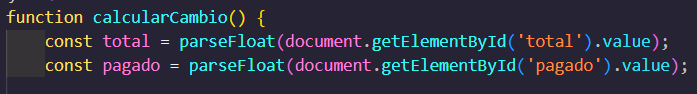
**Fecha de entrega: Mérida, Yucatán a martes 14 de septiembre de 2024**

Actividad 5.

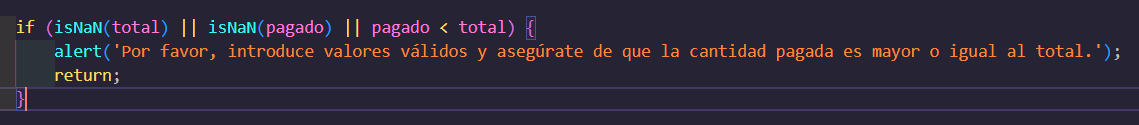
1. El primer paso como en todas las estructuras es hacer el código html con las rutas que estremos utilizando.



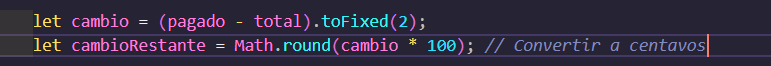
1. Seguidamente empezamos con la creación del script que lo primero que hice fue hacer una función para poder ingresar los datos en la parte de total y pagado para que después se pueda hacer una función con etsas mismas variables que ya declaramos.



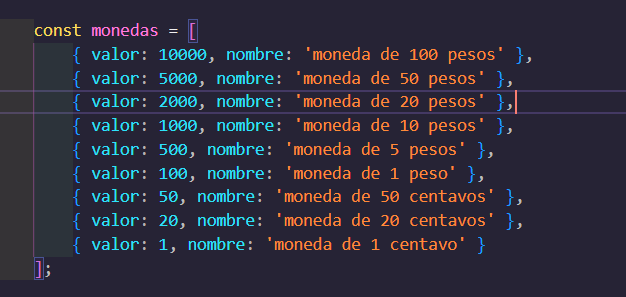
1. En esta parte lo que hice fue aplicar una función para que en el caso de que los valores agregados no sean números, sea menor que la cantidad solicitada o sean valores no válidos, se mande un mensaje de alerta.



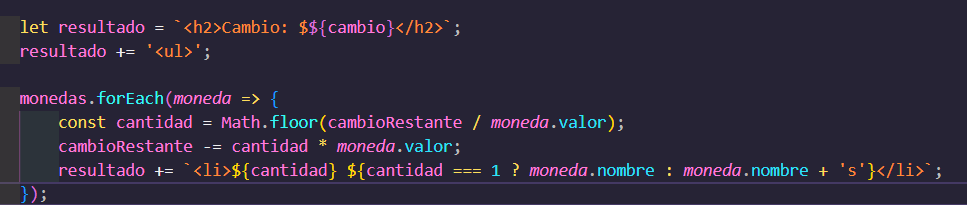
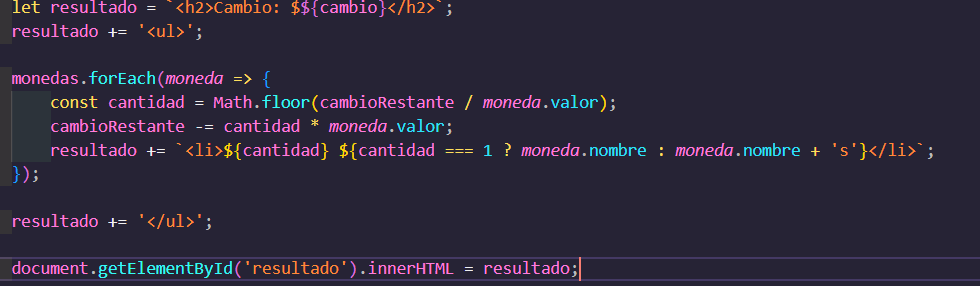
1. En esta parte lo que se hace es calcular el cambio que se debe devolver al usuario, ya que se calcula por medio de la diferencia entre lo pagado y el total y se redondea usando la función toFixed(2) y también de utiliza Math.round ya que se manejaran centavos.



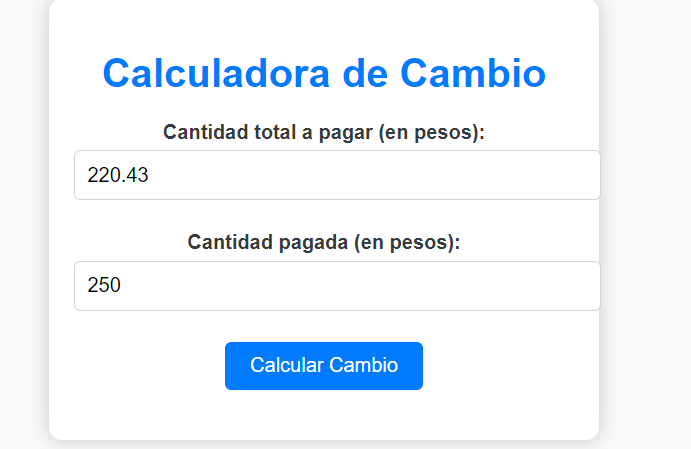
1. En el siguiente paso lo único que hacemos es definir el valor de las monedas que utilizaremos.



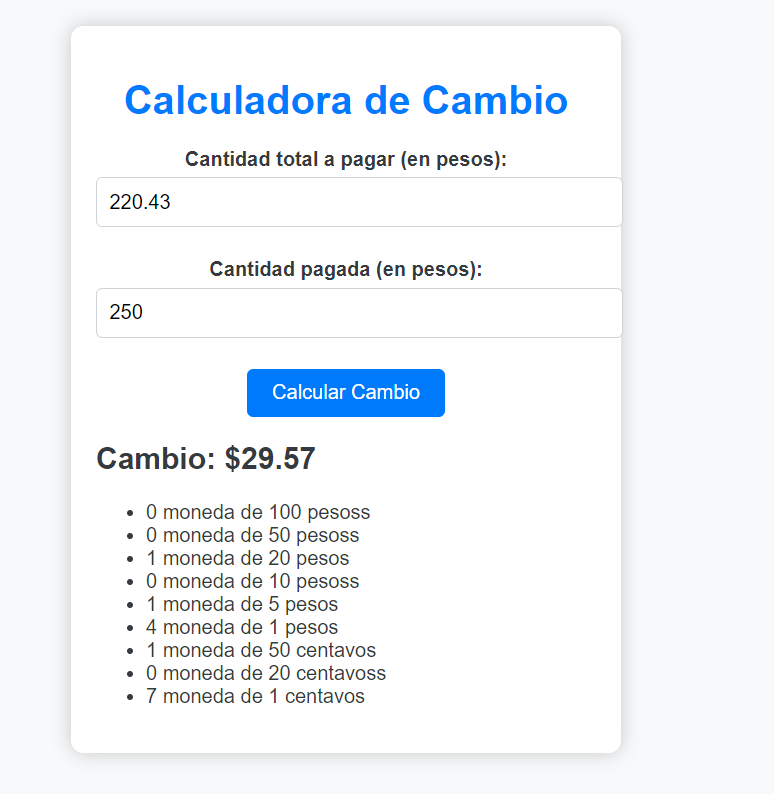
1. En esta parte se calcula la cantidad de monedas que se deben de regresar al usuario y el resultado final de dichas operaciones.



1. Aquí podemos ver un ejemplo en el cual el usuario tiene que pagar la cantidad de 220.43 y pagara con 250

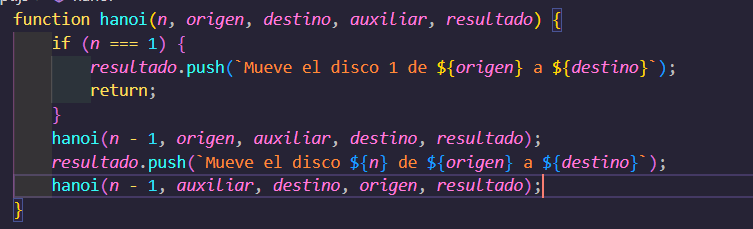


1. Al ejecutarlo nos muestra el siguiente resultado en el cual calculo el cambio de manera adecuada y muestra las monedas que le fueron entregadas al usuario.



Practica 6.

1. Primero creamos una función la cual se asignamos el nombre de function hanoi



Esta función resuelve el problema de las Torres de Hanói para n discos, moviendo los discos desde la torre origen a la torre destino, utilizando una torre auxiliar.

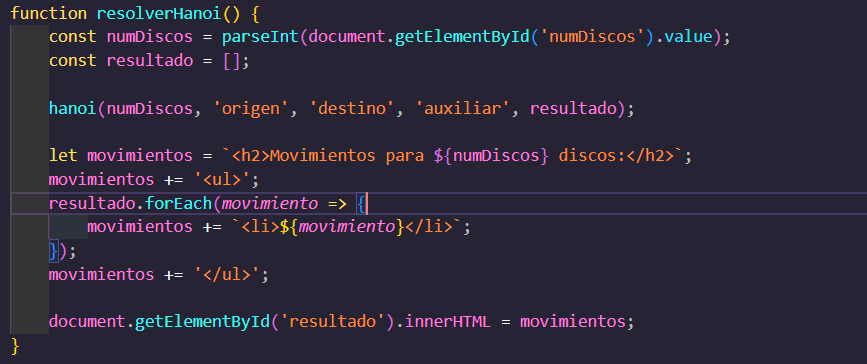
En este caso asigne los valores de la siguiente manera.

* n: El número de discos que se quieren mover.
* origen: La torre de donde se moverán los discos.
* destino: La torre a donde se moverán los discos.
* auxiliar: La torre auxiliar que se usa durante el proceso.
* resultado: Un arreglo que guarda cada uno de los movimientos necesarios para resolver el problema.

Y se aplica la siguiente lógica dependiendo de la cantidad de discos que se ingresen:

* Caso Base: Si n === 1, se mueve el disco directamente de origen a destino.
* Caso Recursivo: Si n > 1, la función realiza lo siguiente:
* Mueve los n-1 discos de origen a auxiliar utilizando destino como torre auxiliar.
* Mueve el disco n desde origen a destino.
* Mueve los n-1 discos desde auxiliar a destino utilizando origen como torre auxiliar.

1. Después desarrollamos la función resolverHanoy



Esta función es la encargada de interactuar con la interfaz de usuario. Toma el número de discos que el usuario ingresa, resuelve el problema de las Torres de Hanói y luego muestra los movimientos necesarios en la página.

* Obtener el Número de Discos:

const numDiscos = parseInt(document.getElementById('numDiscos').value);

Obtiene el valor ingresado por el usuario en el campo de entrada con ID numDiscos y lo convierte a un número entero.

* Preparar el Arreglo resultado:

const resultado = [];

Se inicializa un arreglo vacío que se utilizará para almacenar los movimientos generados por la función hanoi.

* Llamada a la Función hanoi:

hanoi(numDiscos, 'origen', 'destino', 'auxiliar', resultado);

Se llama a la función hanoi con el número de discos y los nombres de las torres. El resultado se almacena en el arreglo resultado.

* Generación del HTML para los Movimientos:

Se construye una lista (<ul>) en HTML que contiene todos los movimientos generados.

Cada movimiento se inserta en un elemento de lista (<li>).

* Mostrar el Resultado en la Página:

document.getElementById('resultado').innerHTML = movimientos;

Se actualiza el contenido del elemento con ID resultado en la página para mostrar la lista de movimientos generados.

1. Y queda de la siguiente manera en nuestra página web, ingresando los discos que desea agregar el usuario.



1. Una vez le damos a resolver, nos muestra todos los movimientos que se deberían de hacer para poder conseguir el objetivo que es llegar a la torre destino.

