

Avance 4

Consideraciones Proyecto semana 4

Duración: 2 horas

Comencemos ahora a presentar de manera dinámica los datos generados. Para hacerlo, realiza lo siguiente:

Avance del proyecto semanal.

Comencemos ahora a presentar de manera dinámica los datos generados. Para hacerlo, realiza lo siguiente:

Cabecero:

- En el archivo **index.html**, agrega
 - El id **presupuesto** al div con clase **presupuesto_valor**.
 - El id **ingresos** al div con clase **presupuesto_ingreso--valor**.
 - El id **egresos** al div con la clase **presupuesto_egreso--valor**.
 - El id **porcentaje** al div con la clase **presupuesto_egres--porcentaje**.
- En el archivo **app.js**, modifica la función **cargarCabecero** para que, en lugar de imprimir en consola los valores generados, puedas recuperar por id los siguientes elementos:
 - **Presupuesto:** en la propiedad **innerHTML**, modifica su valor, asignando el contenido de la variable **presupuesto** pasada por la función **formatoMoneda**.
 - **Porcentaje:** en la propiedad **innerHTML**, modifica su valor, asignando el contenido de la variable **porcentajeEgreso** pasada por la función **formatoPorcentaje**.
 - **Ingresos:** en la propiedad **innerHTML**, modifica su valor, asignando el valor resultante de la invocación de la función **totalIngresos** pasada por la función **formatoMoneda**.
 - **Egresos:** en la propiedad **innerHTML**, modifica su valor, asignando el valor resultante de la invocación de la función **totalEgresos** contenido de la variable **presupuesto** pasada por la función **formatoMoneda**.
- En este momento, al recargar el documento, debería sustituir el contenido del elemento que tiene el id **presupuesto, porcentaje, ingresos y egresos** del contenedor cabecero, sin embargo, no lo hace. Para que lo haga, crea la función **cargarApp** en la que vas a mandar llamar a la función **cargarCabecero()**.
- En el documento html, indícale en el body que, cuando cargue (con el evento **onload**), debe mandar llamar a la función **cargarApp()**.
- Prueba tu aplicación, ahora se deben mostrar en pantalla los valores calculados en lugar de los valores asignados estáticamente.

Avance 4

Ingresos:

- Para cargar los ingresos dinámicamente, es necesario recorrer el arreglo **ingresos**. Para ello, crea la siguiente función:

- Nómbrala **cargarIngresos**.
- Debe ser una función tipo flecha.
- Declara la variable vacía **ingresosHTML**.
- Recorre el arreglo **ingresos** con un ciclo for of.
- En el cuerpo del ciclo, concatena el resultado de una función llamada **crearIngresoHTML** y pásale como parámetro cada uno de los elementos del arreglo **ingreso**.
- Una vez que termine el ciclo, recupera el elemento **lista-ingresos** a través de su id o asígnale el contenido de la variable **ingresosHTML**.

- Como se desea que cada elemento generado por **crearIngresoHTML** cree dinámicamente el contenido del **div lista-ingresos**, debes crear todo el contenido html dentro de la función.

Para lograrlo, haz lo siguiente:

- Declara la función flecha **crearIngresoHTML**.
- Pásale como parámetro **ingreso**.
- Declara la variable **ingresoHTML** y asígnale, por medio de template string, el contenido del div lista-ingresos con los siguientes cambios:
 - En lugar de escribir una cadena en el **div elemento-descripcion**, toma el contenido de **ingreso.descripcion**.
 - En el contenido del **div elemento-valor**, asígnale el valor del elemento ingreso pasado por la función **formatoMoneda**.
 - En el ícono **close-circle-outline**, asígnale el evento onclick e iguálalo a la función **eliminarIngreso** y pásale como parámetro el id del elemento ingreso.

- Retorna la cadena **ingresoHTML**.
- Manda llamar esta función **cargarIngresos()** desde la función **cargarApp()**.
- Modifica tu arreglo **ingresos** (cambia valores y agrega un nuevo elemento al arreglo) y valida que se estén cargando correctamente los elementos en tu aplicación.

Avance 4

Egresos:

- Para cargar los egresos dinámicamente, es necesario recorrer el arreglo egresos. Para ello, crea la siguiente función:
 - Nómbrala **cargarEgresos**.
 - Debe ser una función tipo flecha.
 - Declara la variable vacía **egresosHTML**.
 - Recorre el arreglo ingresos con un ciclo for of.
 - En el cuerpo del ciclo, concatena el resultado de una función llamada **crearEgresoHTML** y pásale como parámetro cada uno de los elementos del arreglo egreso.
 - Una vez que termine el ciclo, recupera el elemento **lista-egresos** a través de su id o asígnale el contenido de la variable **egresosHTML**.
- Como se desea que cada elemento generado por **crearEgresoHTML** cree dinámicamente el contenido del **div lista-ingresos**, debes crear todo el contenido html dentro de la función. Para lograrlo, haz lo siguiente:
 - Declara la función flecha **crearEgresoHTML**.
 - Pásale como parámetro **egreso**.
 - Declara la variable **egresoHTML** y asígnale, por medio de template string, el contenido del div lista-egresos con los siguientes cambios:
 - En lugar de escribir una cadena en el **div elemento-descripcion**, toma el contenido de `egreso.descripcion`.
 - En el contenido del **div elemento-valor**, asígnale el valor del elemento egreso pasado por la función **formatoMoneda**.
 - En el ícono **close-circle-outline**, asígnale el evento onclick e iguálalo a la función **eliminarEgreso** y pásale como parámetro el id del elemento **egreso**.
 - Retorna la cadena **egresoHTML**.
 - Manda llamar a la función **cargarEgresos()** desde la función **cargarApp()**.
 - Modifica tu arreglo egresos (cambia valores y agrega un nuevo elemento al arreglo) y valida que se estén cargando correctamente los elementos en tu **aplicación**.

Avance 4

- Para programar la función `eliminarEgreso` llamada desde el evento `onclick`, realiza lo siguiente:
 - Define la función flecha.
 - Recibe el parámetro `id`.
 - En el cuerpo de la función, crea una variable llamada `indiceEliminar` y asígnale el método `findIndex` del arreglo `egresos`. En este método, obtén el índice del elemento que se desea eliminar, a través de una función flecha que pasa como parámetro. Lo que hace esa función es que cuando el `id` pasado como argumento coincide con el `id` del elemento egreso evaluado, regresa el `id` del elemento.
 - Utilizando el método `splice` del arreglo `egresos`, indícale que elimine el elemento contenido en la variable `indiceEliminar` y, como segundo parámetro, indícale que solo se desea eliminar 1 elemento.
 - Recarga el cabecero llamando a la función `cargarCabecero()` para que se realicen los cálculos con los elementos que queden en el arreglo `egresos`.
 - Recarga el arreglo de los egresos con la función `cargarEgresos()` para que se dibujen solo los elementos restantes del arreglo `egresos`.

Una vez que ya estamos mostrando los datos dinámicamente en el cabecero, en la lista de ingresos y en la lista de los egresos, es momento de darle funcionalidad al formulario.

- En el archivo `html`:
 - Agrega el evento `onclick` al botón con la clase `agregar-btn`.
 - Manda llamar a la función **`agregarDato()`**.
 - Agrega el evento **`onsubmit`**, asignándole los valores “return false” al formulario con `id` **`forma`**.
- En el archivo `app.js`:
 - Declara la función flecha **`agregarDato`**.
 - En la variable `forma`, recupera el formulario con `id` **`forma`**.
 - Recupera el tipo de **`select`** que seleccionó el usuario, asignando a la variable `tipo` el contenido del **`select`** con `id` tipo del formulario **`forma`**.
 - Recupera la **`descripción`** del formulario `forma` y asígnalo a la variable `descripción`.
 - Recupera el valor del formulario `forma` y asígnalo a la variable `valor`.
 - Valida que el valor de `descripción` y de `valor` no estén vacíos. Si la condición se cumple:
 - Valida si el valor de `tipo` es igual a `ingreso`:
 - Agrega el valor de la **`descripción`** y el **`valor`** al arreglo **`ingresos`** a través del método `push` e invocando a un nuevo objeto del tipo **`Ingreso`**.
 - Invoca a la función **`cargarCabecero`** y **`CargarIngreso`**.

Ahora tu aplicación esta lista para que puedas probar toda su funcionalidad.

Avance 4

Consideraciones

Maestro:

- En esta parte del proyecto se utilizan eventos y manejo del DOM, por tanto, asegúrate de que el aprendedor comprenda ambos conceptos y la diferencia entre ellos. Invítalo a que lea la bibliografía adicional.

Alumno:

- Asegúrate de poder manipular el DOM y hacer el backtesting en la consola antes de presentarlo en el HTML.

a) Importante: En la sesión sincrónica del tema 8 se deberá establecer que el aprendedor debe tener listo su proyecto para entregarlo en la semana 5. El proyecto se entregará por los medios indicados por el instructor.

La obra presentada es propiedad de ENSEÑANZA E INVESTIGACIÓN SUPERIOR A.C. (UNIVERSIDAD TECMILENIO), protegida por la Ley Federal de Derecho de Autor; la alteración o deformación de una obra, así como su reproducción, exhibición o ejecución pública sin el consentimiento de su autor y titular de los derechos correspondientes es constitutivo de un delito tipificado en la Ley Federal de Derechos de Autor, así como en las Leyes Internacionales de Derecho de Autor.

El uso de imágenes, fragmentos de videos, fragmentos de eventos culturales, programas y demás material que sea objeto de protección de los derechos de autor, es exclusivamente para fines educativos e informativos, y cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por UNIVERSIDAD TECMILENIO.

Queda prohibido copiar, reproducir, distribuir, publicar, transmitir, difundir, o en cualquier modo explotar cualquier parte de esta obra sin la autorización previa por escrito de UNIVERSIDAD TECMILENIO. Sin embargo, usted podrá bajar material a su computadora personal para uso exclusivamente personal o educativo y no comercial limitado a una copia por página. No se podrá remover o alterar de la copia ninguna leyenda de Derechos de Autor o la que manifieste la autoría del material.