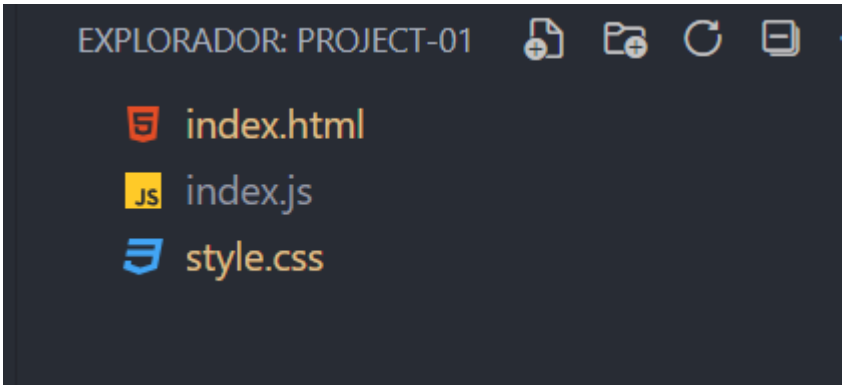
Manejo del DOM - Mensajes

Continuamos con el manejo del DOM en este ejercicio realizaremos el envio de un mensaje a través de un input y un botón para mostrarlo y asi manipular nuestro DOM.

Estructura de nuestro proyecto:



@hdtoledo

Aquí el código HTML:

index.html > html

```html
<!DOCTYPE html>
<html lang="es">

<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Un mensaje para ti</title>
    <link rel="stylesheet" href="style.css">
</head>

<body>
    <div class="container">
        <h1>Un mensaje para enviar</h1>
        <input type="text" value="" placeholder="Escribe tu mensaje ...">
        <div class="btn">Enviar</div>
        <p class="error">Por favor ingresa un mensaje!</p>
        <h3>El ultimo mensaje fue ...</h3>
        <p class="message"></p>
    </div>
    <script src="index.js"></script>
</body>

</html>
```

Resumen del código html inicial:

1. **<!DOCTYPE html>**: Esto define la versión de HTML que se está utilizando, en este caso, HTML5.
2. **<html lang="es">**: Aquí comienza el elemento HTML, con la especificación del lenguaje "es" (español).
3. **<head>**: Dentro de la cabeza del documento, se incluyen metadatos y enlaces a recursos externos.
    - **<meta charset="UTF-8">**: Establece la codificación de caracteres del documento como UTF-8.
    - **<meta http-equiv="X-UA-Compatible" content="IE=edge">**: Especifica la compatibilidad con Internet Explorer.
    - **<meta name="viewport" content="width=device-width, initial-scale=1.0">**: Define las características de la ventana gráfica para dispositivos móviles.
    - **<title>Un mensaje para ti</title>**: Establece el título de la página.
    - **<link rel="stylesheet" href="style.css">**: Enlaza una hoja de estilos externa llamada "style.css".
4. **<body>**: Aquí comienza el cuerpo del documento, que contiene el contenido visible de la página.
    - ** **: Crea un contenedor principal para organizar los elementos dentro de la página.

- **<h1>Un mensaje para enviar</h1>**: Título principal de la página.
- **<input type="text" value="" placeholder="Escribe tu mensaje ...">**: Campo de entrada de texto para escribir el mensaje.
- ** Enviar **: Un botón con la clase "btn" para enviar el mensaje.
- ** Por favor ingresa un mensaje! **: Párrafo con un mensaje de error que se mostrará si no se ingresa ningún mensaje antes de enviar.
- **<h3>El último mensaje fue ...</h3>**: Título para indicar que se mostrará el último mensaje enviado.
- ** **: Párrafo con la clase "message" que se utilizará para mostrar el último mensaje enviado.

- **<script src="index.js"></script>**: Vincula un archivo JavaScript externo llamado "index.js", que probablemente contiene la lógica para manejar el envío de mensajes.

Nuestro código CSS:

style.css > ...

```css
@import url("https://fonts.googleapis.com/css2?family=Poppins:wght@100;200;300;400;500;600;700;800;900&display=swap");

* {
  margin: 0;
  padding: 0;
  box-sizing: border-box;
}

body {
  background-color: rgb(51, 196, 253);
  display: flex;
  justify-content: center;
  align-items: center;
  min-height: 100vh;
  font-family: "Poppins", sans-serif;
}

.container {
  background-color: white;
  width: 700px;
  padding: 20px;
  border-radius: 10px;
  box-shadow: 3px 3px 20px rgba(0, 0, 0, 0.8);
  display: flex;
  flex-direction: column;
  justify-content: center;
  align-items: center;
  gap: 10px;
}

h1 {
  font-size: 28px;
  margin: 10px;
}
```

style.css > ...

```css
}

input {
  width: 90%;
  height: 40px;
  border: 2px solid rgba(0, 0, 0, 0.6);
  padding: 5px 10px;
  font-size: 18px;
  margin-bottom: 10px;
}

.btn {
  background-color: rgb(219, 51, 253);
  padding: 10px 30px;
  border-radius: 10px;
  border: 2px solid rgba(0, 0, 0, 0.4);
  font-size: 18px;
  margin-top: 5px;
  font-weight: 500;
  box-shadow: 2px 2px 4px rgba(0, 0, 0, 0.6);
  cursor: pointer;
  /* display: inline-block; */
}

.error {
  background-color: rgb(255, 139, 139);
  padding: 5px 0;
  width: 70%;
  text-align: center;
  display: none;
}

h3 {
  margin-top: 5px;
  font-size: 22px;
}

.message {
  font-size: 18px;
  color: rgb(255, 29, 29);
  font-weight: bold;
  margin-top: -5px;
}
```

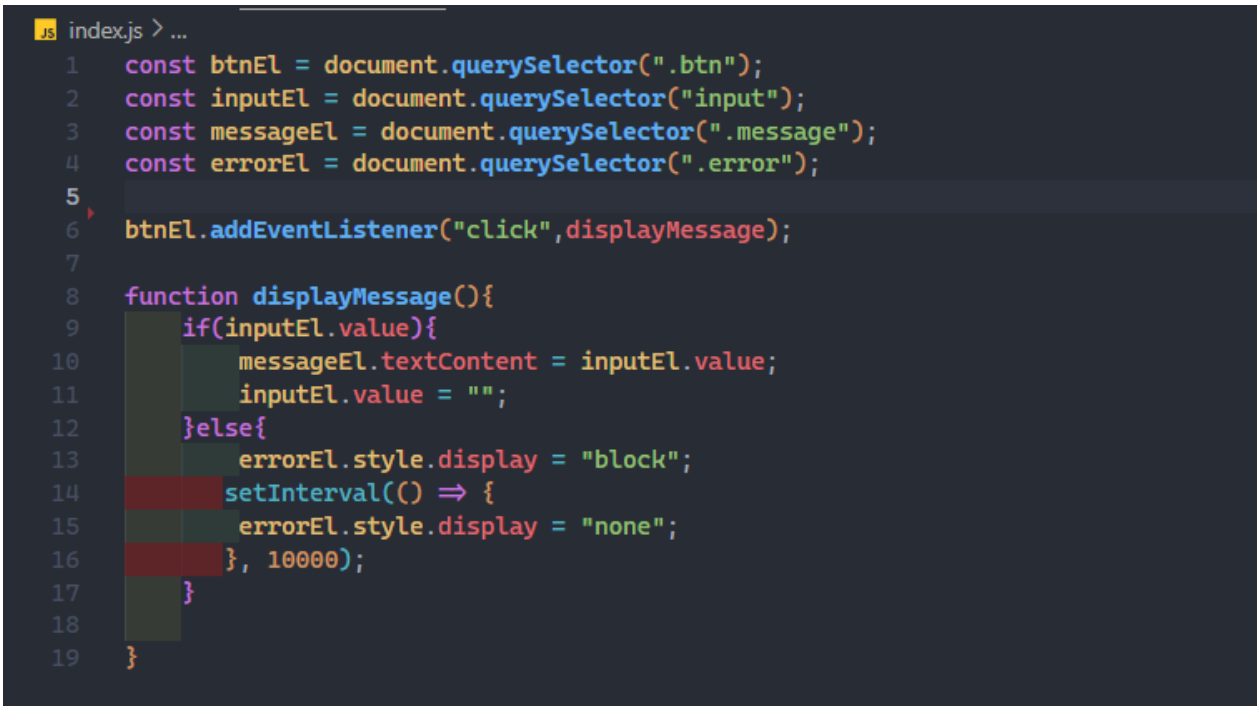
Resumen del CSS:

1. **@import url("https://fonts.googleapis.com/css2?family=Poppins:wght@100;200;300;400;500;600;700;800;900&display=swap");**: Importa la fuente "Poppins" desde Google Fonts con diferentes pesos (100 a 900).
2. *** { margin: 0; padding: 0; box-sizing: border-box; }**: Establece un estilo de reinicio, eliminando márgenes y rellenos predeterminados y asegurándose de que el modelo de caja sea de tipo "border-box".
3. **body { ... }**: Establece estilos para el cuerpo de la página.
    - **background-color: rgb(51, 196, 253);**: Establece el color de fondo del cuerpo en un tono de azul.

- **display: flex; justify-content: center; align-items: center;**: Utiliza el modelo de diseño flexible (flexbox) para centrar contenido tanto horizontal como verticalmente en la página.
- **min-height: 100vh;**: Establece la altura mínima del cuerpo al 100% de la altura de la ventana visible (viewport height).
- **font-family: "Poppins", sans-serif;**: Define la fuente para todo el cuerpo de la página.

4. **.container { ... }**: Define estilos para un contenedor principal en la página.
    - **background-color: white;**: Establece el color de fondo del contenedor en blanco.
    - **width: 700px;**: Fija el ancho del contenedor en 700 píxeles.
    - **padding: 20px;**: Agrega relleno alrededor del contenido del contenedor.
    - **border-radius: 10px;**: Agrega esquinas redondeadas al contenedor.
    - **box-shadow: 3px 3px 20px rgba(0, 0, 0, 0.8);**: Agrega una sombra al contenedor para dar un efecto de elevación.
    - **display: flex; flex-direction: column; justify-content: center; align-items: center;**: Utiliza flexbox para organizar los elementos hijos del contenedor.
    - **gap: 10px;**: Agrega espacio entre los elementos hijos del contenedor.
5. **h1, input, .btn, .error, h3, .message { ... }**: Establece estilos para diferentes elementos dentro del contenedor.
    - **h1 { ... }**: Define estilos para el elemento **<h1>**.
    - **input { ... }**: Define estilos para los campos de entrada de texto.
    - **.btn { ... }**: Define estilos para el botón.
    - **.error { ... }**: Define estilos para el mensaje de error.
    - **h3 { ... }**: Define estilos para el elemento **<h3>**.
    - **.message { ... }**: Define estilos para el párrafo con la clase "message".

Nuestro código JS:



```
index.js > ...
1   const btnEl = document.querySelector(".btn");
2   const inputEl = document.querySelector("input");
3   const messageEl = document.querySelector(".message");
4   const errorEl = document.querySelector(".error");
5
6   btnEl.addEventListener("click",displayMessage);
7
8   function displayMessage(){
9       if(inputEl.value){
10          messageEl.textContent = inputEl.value;
11          inputEl.value = "";
12      }else{
13          errorEl.style.display = "block";
14          setInterval(() => {
15          errorEl.style.display = "none";
16          }, 10000);
17      }
18
19  }
```

Resumen del código JS:

1. **const btnEl = document.querySelector(".btn");**: Selecciona el primer elemento con la clase "btn" en el documento y asigna el elemento a la variable **btnEl**.
2. **const inputEl = document.querySelector("input");**: Selecciona el primer elemento **<input>** en el documento y asigna el elemento a la variable **inputEl**.
3. **const messageEl = document.querySelector(".message");**: Selecciona el primer elemento con la clase "message" en el documento y asigna el elemento a la variable **messageEl**.
4. **const errorEl = document.querySelector(".error");**: Selecciona el primer elemento con la clase "error" en el documento y asigna el elemento a la variable **errorEl**.
5. **btnEl.addEventListener("click", displayMessage);**: Agrega un evento de clic al elemento con la clase "btn", para que cuando se haga clic en el botón, se ejecute la función **displayMessage**.
6. **function displayMessage() { ... }**: Define la función **displayMessage** que se ejecuta cuando se hace clic en el botón.
    - **if (inputEl.value) { ... }**: Verifica si el valor del campo de entrada (**inputEl**) no está vacío.
        - **messageEl.textContent = inputEl.value;**: Si el campo de entrada no está vacío, establece el contenido del elemento con la clase "message" con el valor del campo de entrada.
        - **inputEl.value = "";**: Limpia el campo de entrada después de mostrar el mensaje.
    - **else { ... }**: Si el campo de entrada está vacío:

- **errorEl.style.display = "block";**: Muestra el elemento con la clase "error" estableciendo su estilo de visualización en "block" (visible).
- **setInterval(() => { errorEl.style.display = "none"; }, 10000);**: Utiliza **setInterval** para programar que después de 10 segundos (10000 milisegundos), el elemento con la clase "error" tenga su estilo de visualización cambiado a "none" (oculto). Esto simula un mensaje de error temporal que desaparece después de un tiempo. La función **setInterval** ejecuta la función proporcionada a intervalos regulares especificados. En este caso, se ejecuta después de 10 segundos y oculta el mensaje de error.