Ha habido diferentes versiones de HTML. El tipo de documento<!DOCTYPE html> indica que este documento HTML contiene código HTML5.

Si bien no vamos a profundizar demasiado en el significado de todos los elementos HTML, sí señalaremos algunos elementos importantes. La etiqueta meta indica información de metadatos que normalmente no será visible para el usuario que visualice la página, a menos que vea el código fuente en su explorador. Los elementos o etiquetas meta proporcionan información descriptiva sobre la página web. Por ejemplo, ayuda a los motores de búsqueda a procesar la información de las páginas web que se devuelve en los resultados de la búsqueda.

El juego de caracteres (charset) para UTF-8 puede parecer insignificante, pero es fundamental para establecer cómo interpretan los caracteres los equipos. Si faltan los metadatos del juego de caracteres, la seguridad podría ponerse en peligro. Hay bastante historia e información técnica detrás del atributo charset, pero lo importante de este ejercicio es que el código reemplazable de **VS Code** proporciona algunos valores de manera predeterminada.

El elemento <head> del código HTML contiene información sobre el sitio web no visible dentro de la pestaña del explorador.

Para aplicar estilos a los elementos HTML de la página web, podría escribir el código CSS directamente en el encabezado de la página web. La escritura de CSS en la página HTML se denomina CSS interno. Pero un procedimiento recomendado consiste en separar la estructura HTML y los estilos CSS.

Tocando link y tab: <link href="#" rel="stylesheet" /> para vincular el html y el css.

<!DOCTYPE html>

<html lang="en">

<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>Simple Website</title>

<link href="Main.css" rel="stylesheet" />

</head>

<body>

<h1>Task List </h1>

<p id="msg">Current tasks: </p>

<ul>

<li class=" list">Add visual styles </li>

<li class="list"> Add light and dark themes </li>

<li class=" list"> Enable switching and theme</li>

</ul>

</body>

</html>

ID permite darle estilo a solo un elemento; class le da estilo a todos los miembros de esa class.

body {

font-family: monospace;

}

ul {

font-family: helvetica;

}

El fragmento de código anterior contiene dos reglas. Cada regla tiene lo siguiente:

* Un *Selector*. body y ul son los selectores de las dos reglas y se usan para seleccionar a qué elementos se aplican los estilos.
* Una llave de apertura ({).
* Una lista de *declaraciones* de estilo que determinan el aspecto que deben tener los elementos seleccionados.
* Una llave de cierre (}).

Id y class, también son selectores:

.list es un selector de clases.

#msg es un selector de Id.

Heredabilidad de los estilos

 los estilos aplicados a <body> se heredan en el elemento <h1>. No hemos definido nada para <h1>, pero aun así ha obtenido la fuente que se definió en <body>. Este mecanismo de herencia de elementos principales a sus descendientes es uno de los aspectos clave de CSS. Sin embargo, los elementos <li> tienen una fuente diferente, que reemplaza la que se definió en <body>, porque también son descendientes del elemento <ul> para el que definió un estilo.

:root {

--green: #00FF00;

--white: #FFFFFF;

--black: #000000;

}

El selector :root representa el elemento <html> en la página HTML. Para este tipo de tarea, un procedimiento recomendado es definir un conjunto de variables de CSS globales en una regla CSS con el selector :root. En este ejemplo, ha definido tres variables de color.

.light-theme {

--bg: var(--green);

--fontColor: var(--black);

}

.dark-theme {

--bg: var(--black);

--fontColor: var(--green);

}

En el código anterior, ha definido dos variables nuevas, bg y fontColor, que especifican un color de fuente y de fondo. Estas variables usan la palabra clave var para establecer sus valores de propiedad en las variables especificadas anteriormente en el selector :root.

body {

background: var(--bg);

color: var(--fontColor);

font-family: helvetica;

}

En este ejemplo, se usa el selector body para establecer las propiedades background y color y, dado que los elementos visibles en la página web están dentro del elemento <body>, heredarán los colores establecidos en <body>.

JavaScript

La etiqueta de script HTML <script> nos permitirá vincular a un archivo JavaScript externo, que es la forma en que configurará la aplicación web en este ejercicio.

<script src="app.js"></script>

El elemento <script> se puede colocar en <head> o en otra parte de <body>. Sin embargo, colocar el elemento <script> al final de la sección <body> permite que primero se muestre todo el contenido de la página y, después, se cargue el script.

En el archivo HTML, agregue un elemento <noscript> después de la etiqueta </script> de cierre, que se puede usar para mostrar un mensaje si JavaScript está desactivado.

<noscript>You need to enable JavaScript to view the full site.</noscript>

La adición del elemento <noscript> es un ejemplo de *tolerancia a errores* o *degradación correcta*. Al utilizar el elemento <noscript>, su código puede detectar y planificar cuando una característica no es compatible o no está disponible.

**Establecimiento del modo strict**

JavaScript se diseñó para ser fácil de aprender y permite que el desarrollador cometa ciertos errores. Por ejemplo, JavaScript no produce un error al usar una variable mal escrita y, en su lugar, crea una nueva variable global. Aunque tener menos errores es tentador al empezar a aprender JavaScript, puede llevar a escribir código más difícil de optimizar para los exploradores y más difícil de depurar.

Cambie al modo strict para obtener errores más útiles al cometer equivocaciones.

* En **VS Code**, abra el archivo app.js y escriba lo siguiente.

'use strict';

Agregar botón

En el archivo HTML, agregue un elemento <button>. Coloque el botón al final de la lista dentro de un elemento <div>.

Para poder agregar el controlador de eventos, necesita una referencia al elemento de botón.

const switcher = document.querySelector('.btn');

La función document.querySelector usa selectores de CSS, igual que los que ha usado en el archivo CSS. switcher ahora es una referencia al botón de la página.

A continuación, agregue el controlador de eventos para el evento click. En el código siguiente, agregue un cliente de escucha para el evento click y defina una función de controlador de eventos que el explorador ejecutará cuando se produzca el evento click.

switcher.addEventListener('click', function() {

document.body.classList.toggle('light-theme');

document.body.classList.toggle('dark-theme');

});

Método toggle, para modificar el atributo de clase del elemento <body>. Este método agrega o quita automáticamente las clases light-theme y dark-theme.

Pero también es necesario actualizar la etiqueta del botón para mostrar el tema correcto, por lo que debe agregar una instrucción if para determinar el tema actual y actualizar la etiqueta del botón.

const className = document.body.className;

if(className == "light-theme") {

this.textContent = "Dark";

} else {

this.textContent = "Light";

}

});