# Inicio

En este paso a paso, entenderemos los básicos sobre desarrollo web aprendiendo como utilizar HTML, CSS y Javascript para construir un sitio web. Además, como ejercicio extra entenderemos como podemos realizar el despliegue de este sitio web para sea accesible públicamente en internet con Azure.

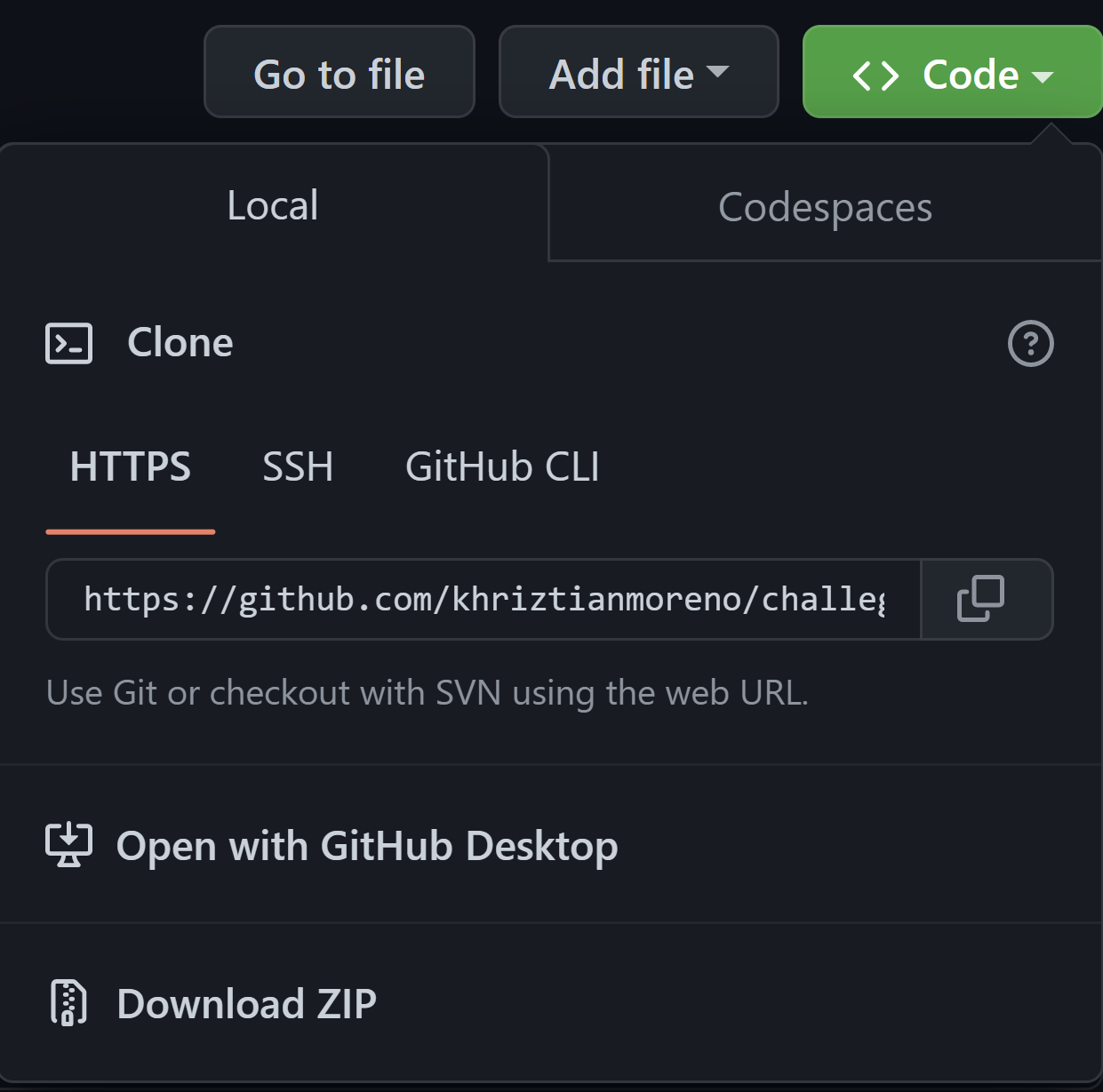
## Pre requisitos

### Instalar Microsoft Visual Studio Code

1. Descarga e instala Microsoft Visual Studio Code desde aquí:

### Tener una copia local del código

* Abre el siguiente link:
* Una vez dentro, da click en la pestaña que dice <>Code y da click en Download ZIP



* Toda la documentación necesaria para realizar el paso a paso y seguir la clase estará dentro de este ZIP

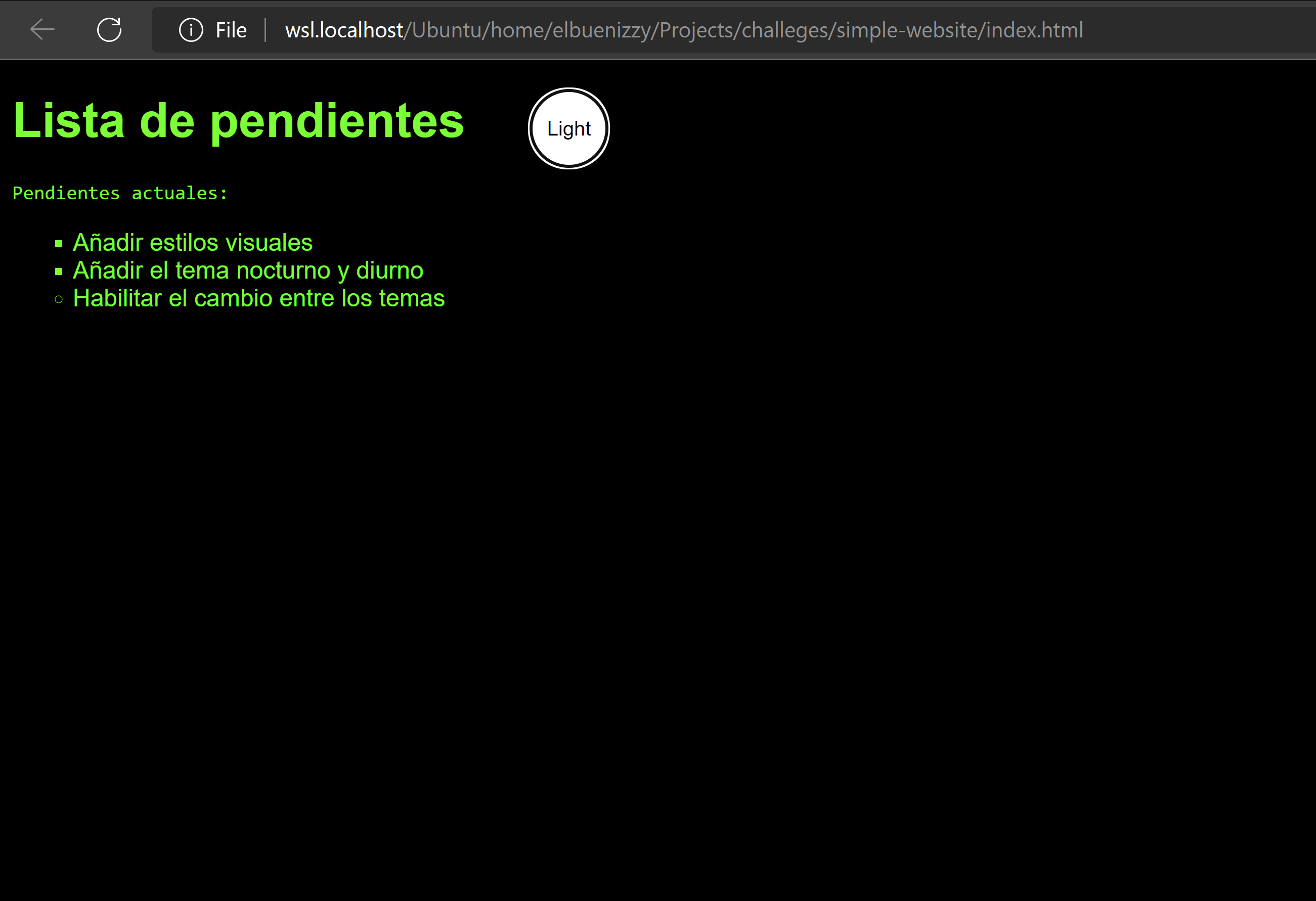
# Introducción

Dentro del Desarrollo web, somos capaces de crear sitios que estarán disponibles dentro de todos los dispositivos, que tengan todos los tipos de multimedia posibles, que realicen computación compleja y que hasta creen aplicativos que se vean y sientan como aplicaciones nativas, o sea, aplicaciones creadas específicamente para dispositivo móvil, aunque haya sido creada para web.

Si te interesa iniciarte en el desarrollo web, probablemente has encontrado multiples opciones para comenzar. En esta ocasión aprenderás los componentes básicos de un sitio web y algunas herramientas que puedes utilizar para la gestión del mismo. Utilizarás HTML, CSS y Javascript para construir un sitio web. De igual forma utilizarás herramientas de desarrollo en el navegador para comprender que está sucediendo.

### Que es lo que lograremos

1. Imagina que eres un desarrollador web y te han pedido tener un sitio web accesible para todos, para esto necesitas habilitar los temas nocturno y diurno



## Componentes de una página web

El contenido, el estilo y la lógica interactiva se separan en archivos HTML, CSS y JavaScript, respectivamente. Dos de los muchos motivos para separar los intereses son la simplicidad y la reutilización. Por ejemplo, al aplicar estilo a elementos HTML con CSS, puede simplificar el código HTML. En lugar de codificar el aspecto dentro de cada elemento, puede aplicar estilos CSS a todos los elementos de una página, independientemente de la complejidad de la página. Además, puede vincular varias páginas HTML a un único archivo CSS, lo que puede ayudarle a simplificar una apariencia coherente en todo el sitio web

1. Note:
2. Default Admin: sysadmin
3. Default Password: Password$123

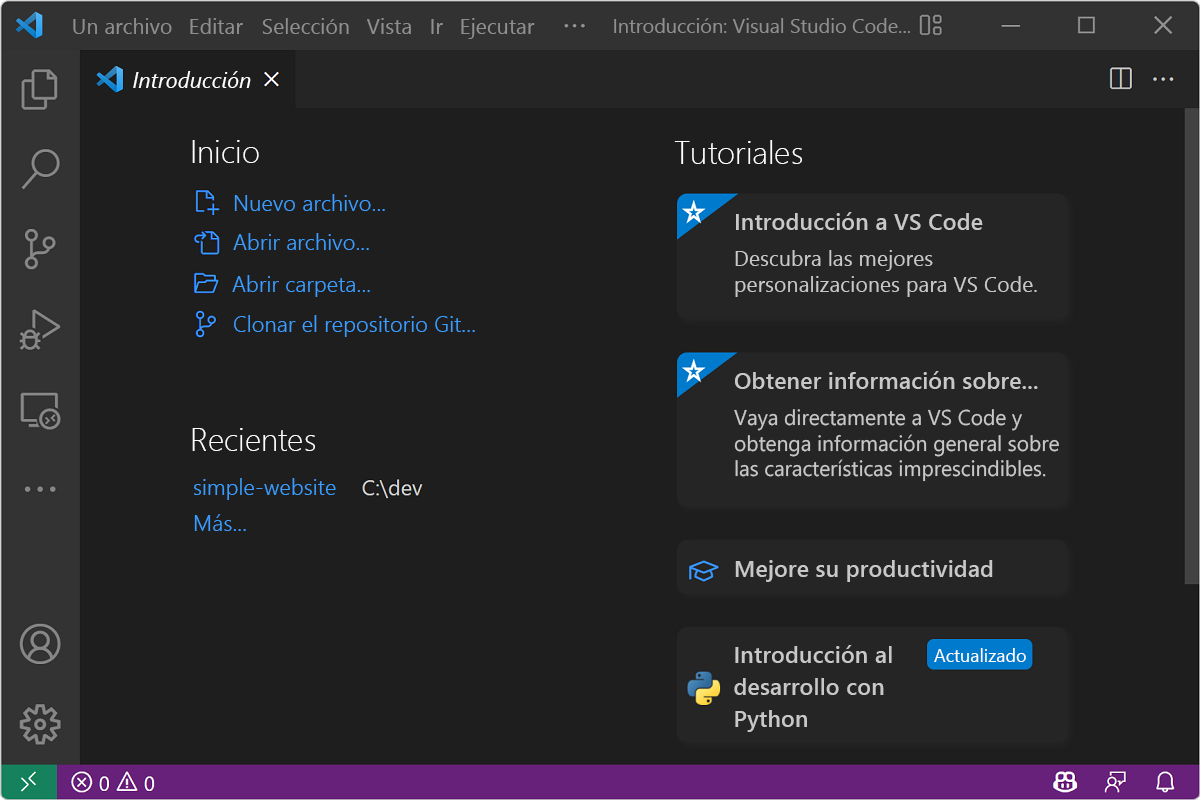
# Configuración de la estructura de la aplicación web

Al crear un sitio web, es habitual que la estructura del proyecto cambie con el tiempo a medida que se vuelve más complicado. La clave consiste en mantener la apariencia de organización, y existen estrategias comunes que ayudan a que así sea. Los proyectos de gran tamaño suelen requerir un mayor grado de cuidado y atención para que muchos usuarios puedan mantenerlo todo organizado.

En esta unidad, creará una estructura de proyecto pequeña con Visual Studio Code. El proyecto tendrá tres archivos: un archivo HTML, un archivo CSS y un archivo JavaScript. También agregará una extensión de Visual Studio Code para simplificar la ejecución del sitio web en el explorador.

## Creación de una carpeta nueva para el sitio web

* Abre el explorador de Windows
* Dentro de Documentos crea una carpeta nueva llamada ClaseDeSoftware
* Abre Visual Studio Code



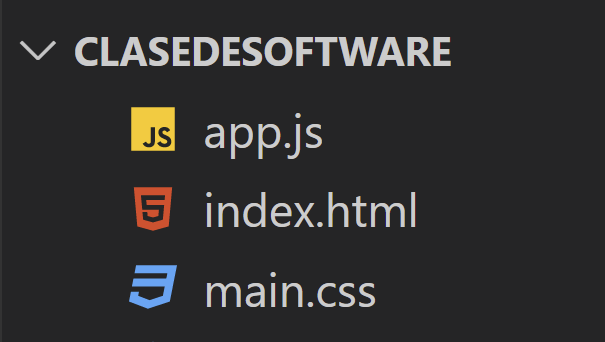
* Seleccionamos Archivo -> Abrir folder y seleccionamos la carpeta que creamos en documentos

## Creación de archivos

* Cree un archivo; para ello, seleccione **Archivo > Nuevo archivo**
* Escriba  **index.html** como nombre de archivo y, a continuación, seleccione **Guardar**.

Repita los pasos anteriores para crear dos archivos más, main.css y app.js. Cuando haya terminado, en el explorador de Visual Studio Code, verá que la carpeta del proyecto simple-website contiene el siguiente archivo, que incluye el sitio web:

* index.html
* main.css
* app.js



Puede crear un sitio web incluyendo todo el código HTML, los estilos CSS y el código JavaScript en un único archivo. Sin embargo, en este ejercicio está usando un archivo HTML para el contenido, un archivo CSS para los estilos y un archivo JavaScript para la interactividad.

La configuración de tres archivos ayuda a mantener organizado el proyecto del sitio web. La separación de contenido, estilos y lógica es un ejemplo de *mejora progresiva*. Si JavaScript no está habilitado o no es compatible con los clientes, HTML y CSS seguirán funcionando. Sin embargo, si CSS no es compatible con los clientes, al menos su contenido HTML aparecerá.

# Adición de HTML básico a la aplicación web

Por el momento, el sitio web tiene un archivo HTML vacío. Se agregará algo de código. El objetivo es usar el lenguaje de marcado de hipertexto (HTML) para describir la página web que deben mostrar los exploradores de los clientes. ¿No sería buena idea tener una plantilla inicial? De forma cómoda, los editores pueden rellenar automáticamente parte del texto reutilizable o la estructura HTML típica.

En esta sección, agregará contenido HTML básico, abrirá la página HTML en un explorador y echará el primer vistazo a las herramientas de desarrollo.

## Incorporación de código HTML

Visual Studio Code proporciona compatibilidad básica para la programación de HTML desde el principio. Incluye resaltado de sintaxis, finalizaciones inteligentes con IntelliSense y formato personalizable.

* Abra el archivo index.html seleccionando el archivo index.html en el **Explorador**.
* En la página **index.html**, escriba html:5 y después seleccione **ENTRAR**. El código de plantilla HTML5 se agrega al archivo.

1. <!DOCTYPE html>
2. <html lang="en">
3. <head>
4. <meta charset="UTF-8">
5. <meta http-equiv="X-UA-Compatible" content="IE=edge">
6. <meta name="viewport" content="width=device-width, initial-scale=1.0">
7. <title>Document</title>
8. </head>
9. <body>
10. </body>
11. </html>

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.

## Edición del element Head

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

Los *metadatos* definen datos sobre el documento HTML, como el juego de caracteres, los scripts y el explorador en el que se abre la página web.

El *título* de una página web aparece en la parte superior de la ventana del explorador y es significativo por muchos motivos. Por ejemplo, el título lo usan y lo muestran los motores de búsqueda. Vamos a agregar un título

En el editor, modifique el elemento <title> para que se parezca al ejemplo siguiente.

1. ...
2. <head>
3. <meta charset="utf-8">
4. <meta http-equiv="X-UA-Compatible" content="IE=edge">
5. <meta name="viewport" content="width=device-width, initial-scale=1.0">
6. <title> Sitio web simple
7. </title>
8. ...

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. Tener una página CSS independiente se denomina *CSS externa*.

El código suele ser más fácil de leer cuando es conciso y compartimentado. Puede usar una o varias hojas de estilos externas para dar servicio a varias páginas web. En lugar de actualizar cada página HTML con código CSS replicado, puede realizar cambios en un único archivo CSS y hacer que esas actualizaciones se apliquen a todas las páginas web dependientes. Vamos a vincular a una hoja de estilos externa.

En el editor de **VS Code**, agregue una línea en blanco después del elemento <title>, escriba link y, a continuación, seleccione ENTRAR. **VS Code** debe agregar la siguiente línea al archivo index.html.

1. <link rel="stylesheet" href="">

Actualice href= a href="main.css" y guarde el archivo

1. ...
2. <head>
3. <meta charset="utf-8">
4. <meta http-equiv="X-UA-Compatible" content="IE=edge">
5. <meta name="viewport" content="width=device-width, initial-scale=1.0">
6. <title> Sitio web simple</title>
7. <link rel="stylesheet" href="main.css">
8. </head>
9. ...

## Edición del cuerpo

Para empezar, ahora se rellenará el elemento <body>.

El elemento <body> contiene el contenido del sitio web visible para sus clientes en sus exploradores.

* Agregue un elemento <h1> de *encabezado*, seguido de un elemento <p> de *párrafo* y, después, cree una *lista sin ordenar*<ul> que contenga varios elementos <li> de *elementos de lista*.
* Edite el código, o cópielo y péguelo, para que tenga un aspecto similar al siguiente ejemplo.

1. <!DOCTYPE html>
2. <html lang="en">
3. <head>
4. <meta charset="utf-8">
5. <meta http-equiv="X-UA-Compatible" content="IE=edge">
6. <meta name="viewport" content="width=device-width, initial-scale=1.0">
7. <title>Sitio web simple</title>
8. <link rel="stylesheet" href="main.css">
9. </head>
10. <body>
11. <h1>Lista de pendientes</h1>
12. <p id="msg">Pendientes actuales:</p>
13. <ul>
14. <li class="list">Añadir estilos visuales</li>
15. <li class="list">Añadir el tema nocturno y diurno</li>
16. <li>Habilitar el cambio entre los temas </li>
17. </ul>
18. </body>
19. </html>

Se puede usar un atributo ID (que se usa en el elemento <p>) para aplicar estilo a un elemento, mientras que el atributo de clase (que se usa en el elemento <li>) sirve para aplicar un estilo a todos los elementos de la misma clase.

Antes del paso siguiente, asegúrese de que el archivo se guarda presionando Control+S o Comando+S.

# Abrir en el explorador

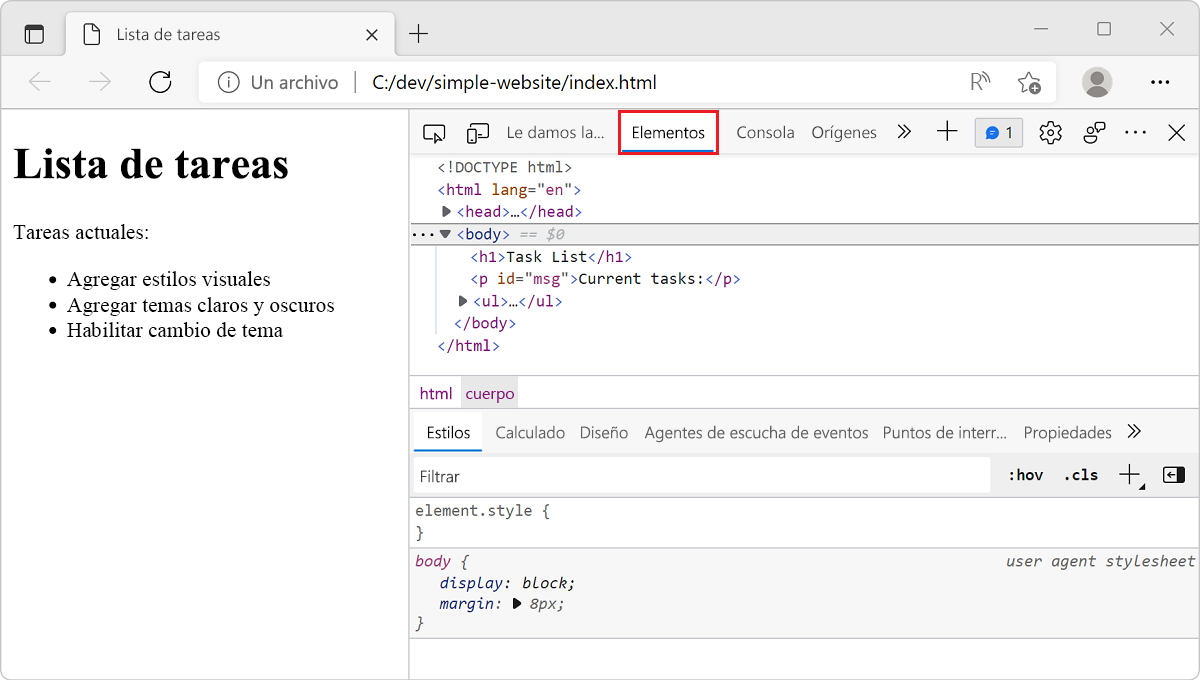
Puede obtener una vista previa local de la página web si abre el archivo HTML en un explorador. En lugar de una dirección de sitio web que comienza con https://, el explorador apunta a la ruta de acceso del archivo local, que debería tener un aspecto similiar a C:/dev/simple-website/index.html.

Para poder obtener una vista previa de lo que hemos avanzado, simplemente tenemos que dirigirnos hacia la carpeta y abrir el archivo creado

# Visualización de la página con las herramientas de desarrollo

Puede inspeccionar una página web mediante las herramientas para desarrolladores del explorador. Vamos a probarlo.

1. Abra Herramientas de desarrollo; para ello, haga clic con el botón derecho del ratón en la página web y seleccione **Inspeccionar**, o bien pruebe estos métodos abreviados de teclado:
   * Presione el método abreviado de teclado para **Herramientas de desarrollo**, que es F12.
   * Presione Ctrl+Mayús+I en Windows y Linux u Opción+Comando+I en Mac.
2. Haga click en la pestaña **Elements**



1. Mueva el mouse sobre los elementos HTML que se muestran en la pestaña **Elementos** y expanda el contenido de los distintos elementos.

La pestaña **Elementos** de las herramientas de desarrollo muestran el modelo de objetos de documento (DOM) tal como se representa en el explorador. Al depurar, a menudo es importante ver cómo el explorador interpreta el código fuente.

La inspección de la página en un explorador proporciona todo tipo de información útil y puede ayudar a solucionar problemas. Con el inspector también puede ver detalles de CSS, como verá en la sección siguiente.

# Aplicación de estilos al código HTML con CSS

Las Hojas de estilos en cascada (CSS) le permiten especificar la apariencia de la página. La idea básica es definir cuál debe ser el estilo para los elementos que se usan en las páginas HTML. Mientras que los elementos HTML definen el contenido, los estilos CSS definen el aspecto de este contenido.

Por ejemplo, puede aplicar esquinas redondeadas o asignar un fondo degradado a un elemento. También puede usar CSS para especificar el aspecto de los hipervínculos y su respuesta al interactuar con ellos. También puede realizar diseños de página sofisticados y efectos de animación.

Puede aplicar estilos a elementos concretos, a todos los elementos de un tipo en particular o usar clases para aplicar estilos a muchos elementos diferentes.

En este ejercicio, aplicará estilos CSS a los elementos de la página HTML y agregará código CSS para definir los temas claros y oscuros. También comprobará los resultados en las herramientas de desarrollo del explorador.

## CSS externas

En la parte anterior sobre HTML, ha creado un vínculo a un archivo CSS externo desde HTML.

Una ventaja de las CSS externas es que se pueden vincular varias páginas HTML al mismo archivo CSS. Si realiza un cambio en la CSS, se actualizará el estilo de cada página. El uso de un archivo HTML para el contenido de la página, de un archivo CSS para aplicar estilos y de un archivo JavaScript para la interacción se denomina *separación de preocupaciones*.

Tal como se ha descrito anteriormente, también puede escribir CSS directamente en HTML, lo que se denomina *CSS interno*. Incluso para un sitio web básico, hay tantas reglas CSS que la página HTML se puede desordenar muy rápidamente. Con más de una página, la misma CSS se repetiría con frecuencia y sería difícil de administrar.

## Reglas de CSS

Las reglas CSS son la forma en que se aplican los estilos a los elementos HTML. Las reglas CSS tienen un **selector**, que se usa para expresar a qué elemento o elementos se deben aplicar los estilos.

En **Visual Studio Code**, abra el archivo main.css y escriba lo siguiente.

1. body {
2. font-family: monospace;
3. }
4. ul {
5. font-family: helvetica;
6. }

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 (}).

Por ejemplo, el selector ul selecciona el elemento HTML <ul> de la página para aplicarle estilos. La declaración es font-family: helvetica y determina cuál debe ser el estilo. El nombre de la propiedad es font-family y el valor es helvetica.

Como verá a continuación, puede definir nombres personalizados propios para los elementos.

## Selectores

Los selectores *id.* y *clase* permiten aplicar estilos a los nombres de atributo personalizados del código HTML. Un identificador se usa para aplicar estilo a un elemento, mientras que las clases se pueden usar para aplicar estilo a varios elementos.

1. Copie el código siguiente en el archivo CSS, después de la llave de cierre del selector ul que ha agregado antes.
2. li {
3. list-style: circle;
4. }
5. .list {
6. list-style: square;
7. }
8. #msg {
9. font-family: monospace;
10. }

El código anterior contiene tres reglas CSS, y las dos últimas usan atributos personalizados para seleccionar elementos: .list y #msg.

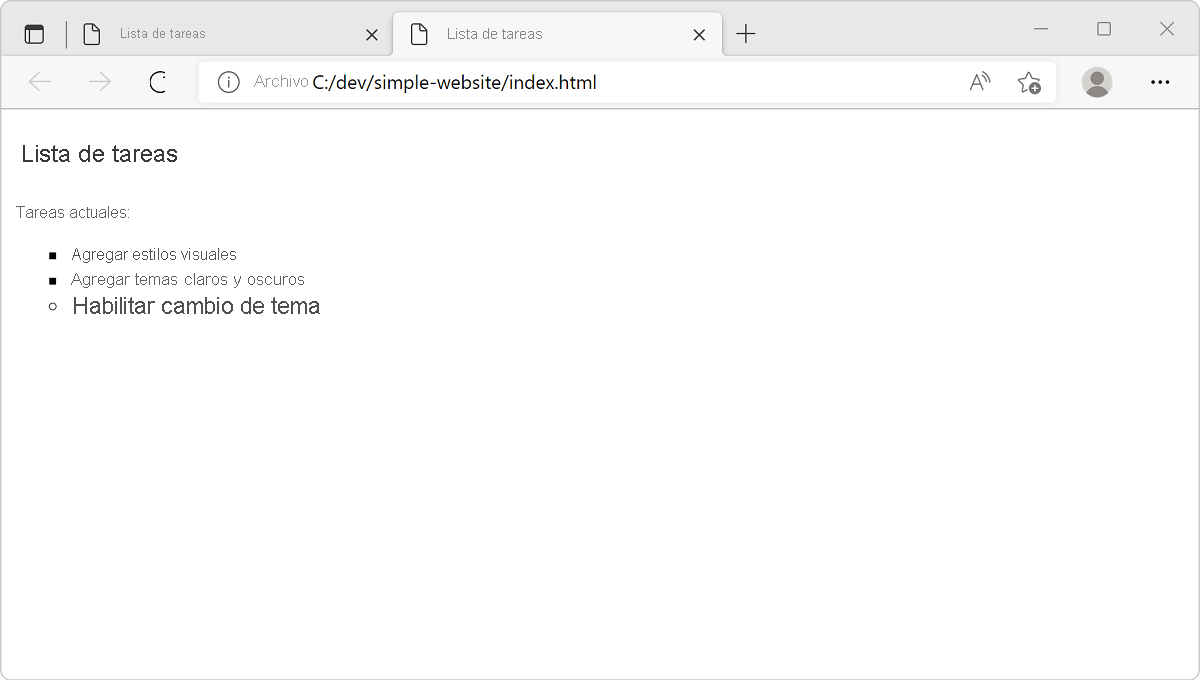
* + .list es un *selector de clases*. Cada elemento HTML que contenga un atributo class establecido en list obtendrá los estilos definidos dentro de este selector.
  + #msg es un *selector de Id*. El elemento HTML que tenga su atributo id establecido en msg obtendrá los estilos definidos dentro de este selector.

Los nombres que use para los selectores pueden ser arbitrarios, siempre que coincidan con lo que haya definido en el código HTML.

1. Guarde el trabajo mediante Control+S en Windows o Comando+S en macOS.

## Visualización en un explorador

Para ver los cambios realizados bastará con dar actualizar a la página o abrir el archivo de nuevo



¿Los estilos de fuente son los que esperaba ver? Es interesante cómo 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.

Tenga en cuenta que, al usar **Abrir en el explorador predeterminado** en **VS Code**, cada vez se abre una nueva pestaña en el explorador. Para evitar abrir una pestaña nueva, puede volver a cargar la pestaña que ya contiene su sitio web.

Para volver a cargar la pestaña, presione F5, que es el método abreviado de teclado para actualizar, o bien presione Ctrl+R en Windows o Linux, y Comando+R en Mac.

## Incorporación de un tema claro

1. En el archivo CSS, agregue el código siguiente al final del archivo.
2. .light-theme {
3. color: #000000;
4. background: #00FF00;
5. }

En este ejemplo, #000000 especifica el negro para el color de fuente y #00FF00 especifica el verde para el color de fondo.

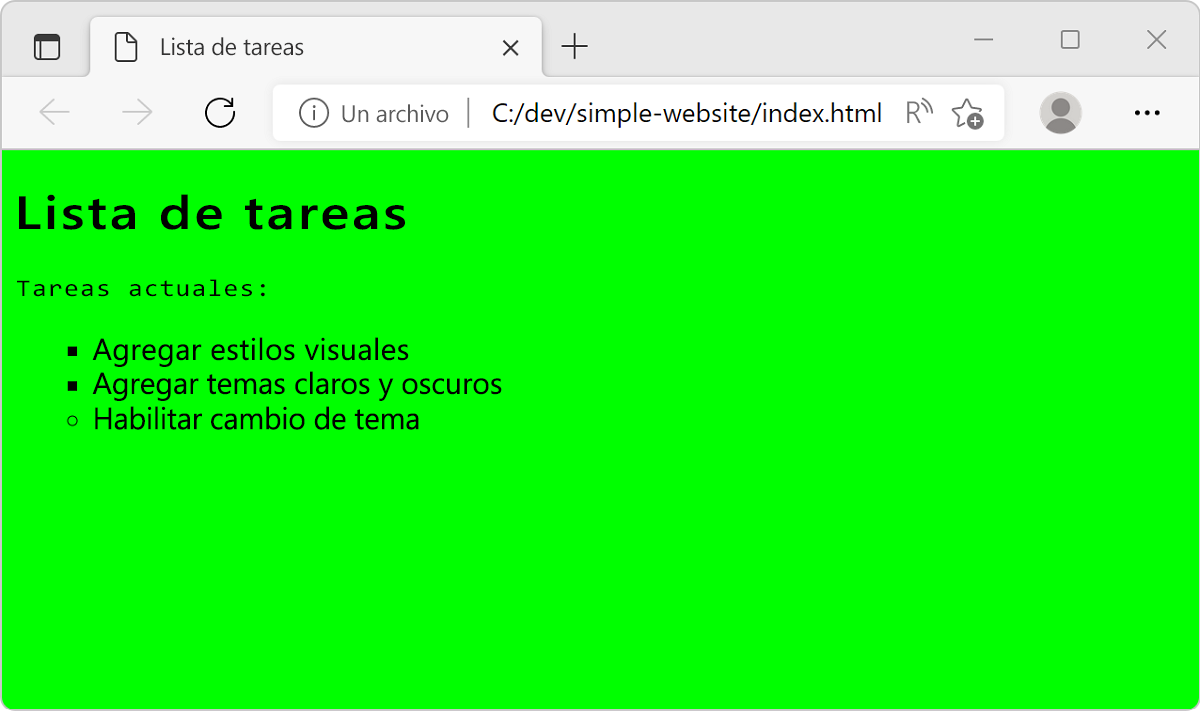
En el archivo HTML, actualice el elemento <body> con un nombre de clase, light-theme, para que el selector de clases del tema claro aplique correctamente los estilos.

1. <body class="light-theme">

## Visualización en un explorador

## .

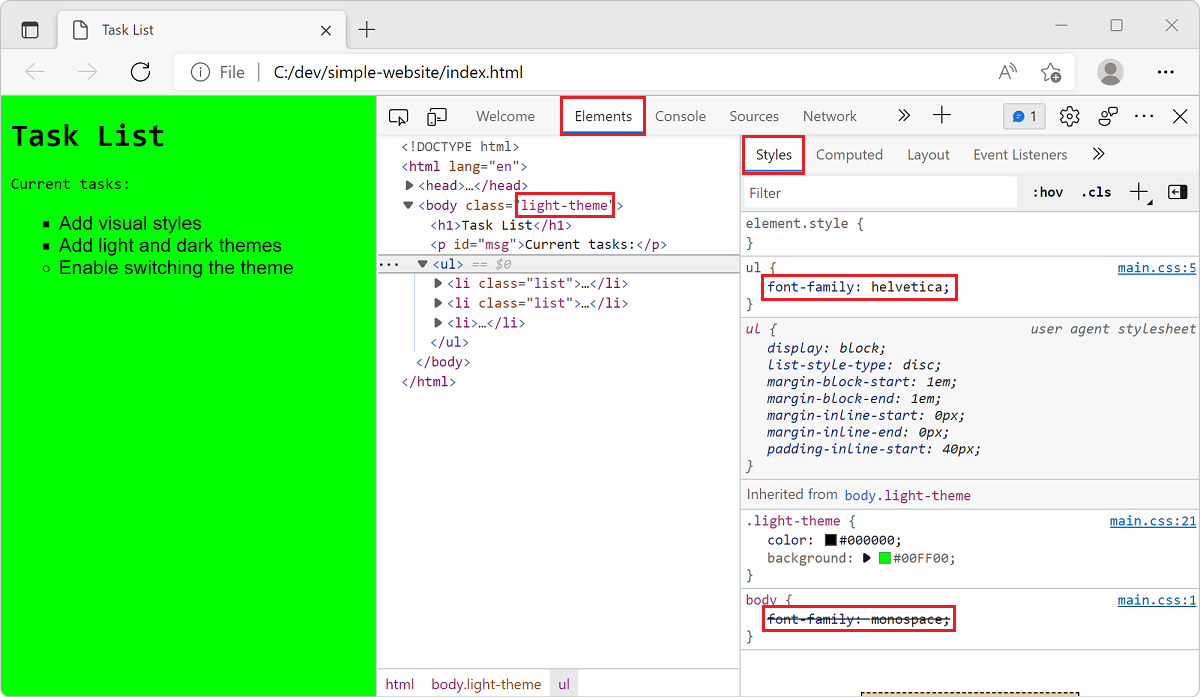
Para ver los cambios realizados bastará con dar actualizar a la página o abrir el archivo de nuevo



## Visualización de CSS aplicado

En la vista del explorador, abra Herramientas de desarrollo.

1. Haga clic con el botón derecho en la página y seleccione Inspeccionar, o bien seleccione el método abreviado de teclado F12 o Ctrl+Mayús+I.
2. Seleccione la pestaña Elementos y, dentro de la pestaña Elementos, seleccione la pestaña Estilos (ya debería estar seleccionada de manera predeterminada).
3. Mantenga el puntero sobre los distintos elementos HTML y, a medida que seleccione algunos, observe cómo las herramientas de desarrollo muestran qué estilos se han aplicado a dichos elementos en la pestaña Estilos.
4. Seleccione el elemento <body>. Observe el elemento light-theme aplicado.
5. Seleccione el elemento <ul> de la lista sin ordenar. Observe el estilo personalizado font-family: helvetica;, que reemplaza el estilo del elemento <body>.



## Incorporación de un tema oscuro

Para agregar compatibilidad con un tema oscuro para su CSS, siga estos pasos.

1. Agregue algunas constantes a la raíz de la página en la parte superior del archivo CSS.
2. :root {
3. --green: #00FF00;
4. --white: #FFFFFF;
5. --black: #000000;
6. }

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. A continuación, podrá usar estas variables en otras reglas CSS.

Al final del archivo CSS, reemplace la regla light-theme por el código siguiente para actualizarlo y agregar el selector dark-theme.

1. .light-theme {
2. --bg: var(--green);
3. --fontColor: var(--black);
4. }
5. .dark-theme {
6. --bg: var(--black);
7. --fontColor: var(--green);
8. }

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.

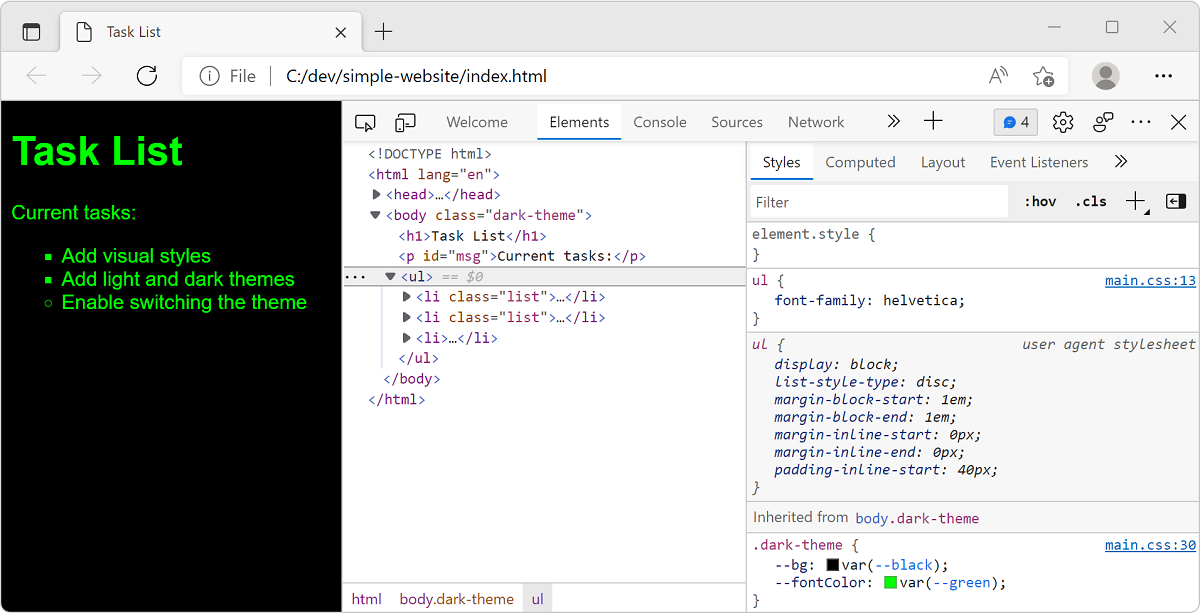
A continuación, en el archivo CSS, reemplace el selector body actual por el código siguiente.

1. body {
2. background: var(--bg);
3. color: var(--fontColor);
4. font-family: helvetica;
5. }

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>.

En el archivo CSS, quite las reglas con los selectores #msg y ul para que también hereden la misma fuente de <body>.

Para ver el tema oscuro, abra el archivo index.html, edite manualmente el tema predeterminado del atributo de clase <body> en tema oscuro (dark-theme) y, a continuación, vuelva a cargar la página en el explorador.



Edite el atributo de clase <body> para volver a cambiar el valor predeterminado al tema claro.

---------------------------------------------FIN DE LA CLASE--------------------------------------------------

# Reto 1: Adición de interactividad con javascript

JavaScript (o *ECMAScript*) es un lenguaje de programación que le ayuda a agregar interactividad a las páginas web.

Por ejemplo, puede usar JavaScript para definir el comportamiento que se va a producir cuando un usuario selecciona un botón, como por ejemplo, abrir una ventana emergente. Con JavaScript, puede agregar o quitar contenido de una página web sin volver a cargarla.

En esta unidad, configurará un archivo JavaScript de ejemplo para la página web. Creará un botón para cambiar entre los temas claro y oscuro. A continuación, adjuntará el botón al código JavaScript que realiza el cambio de tema real. Por último, comprobará el proyecto terminado en las herramientas de desarrollo del explorador.

## Vínculo a javascript

Al igual que CSS, puede agregar JavaScript directamente al archivo HTML, pero un procedimiento recomendado es guardar el código JavaScript en un archivo independiente. Agregar el código JavaScript a un archivo independiente facilita su reutilización en varias páginas web. Por ejemplo, podría crear una alerta emergente agregando <script>alert('Hello World')</script> en cualquier parte del cuerpo de las páginas web; sin embargo, es mejor agregar el código JavaScript a un archivo independiente que se pueda vincular a todos los archivos que necesiten su funcionalidad personalizada.

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.

1. En **VS Code**, abra el archivo index.html.
2. En una nueva línea antes del elemento script:src de cierre, escriba </body> y después seleccione ENTRAR. Las etiquetas de apertura y cierre de un script se agregan al código.
3. Modifique el elemento <script> para cargar el archivo app.js como se muestra en el ejemplo siguiente, y asegúrese de que se encuentra después del elemento </ul> de cierre de la lista.
4. ...
5. <ul>
6. <li class="list">Add visual styles</li>
7. <li class="list">Add light and dark themes</li>
8. <li>Enable switching the theme</li>
9. </ul>
10. <script src="app.js"></script>
11. ...

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.

## Adición de un botón

Necesita un modo de permitir que el usuario cambie entre los temas claro y oscuro en la página web. En este ejercicio, implementará esa funcionalidad con un elemento <button> HTML.

1. En el archivo HTML, agregue un elemento <button>. Coloque el botón al final de la lista dentro de un elemento <div>.
2. ...
3. <ul>
4. <li class="list">Add visual styles</li>
5. <li class="list">Add light and dark themes</li>
6. <li>Enable switching the theme</li>
7. </ul>
8. <div>
9. <button class="btn">Dark</button>
10. </div>
11. <script src="app.js"></script>
12. ...

Observe que el elemento <button> de este ejemplo tiene un atributo class que usará para aplicar estilos CSS.

1. En el archivo CSS, agregue una regla nueva con un n selector de clases .btn para el botón HTML. Para que los colores de los botones sean distintos de los colores generales de los temas claro u oscuro, establezca las propiedades color y background-color en esta regla. Invalidarán las predeterminadas establecidas en la regla body del archivo CSS.
2. .btn {
3. color: var(--btnFontColor);
4. background-color: var(--btnBg);
5. }
6. A continuación, modifique la regla .btn para agregar algunos estilos para el tamaño, la forma, la apariencia y la ubicación del botón. El CSS siguiente crea un botón redondo a la derecha del encabezado de la página.
7. .btn {
8. position: absolute;
9. top: 20px;
10. left: 250px;
11. height: 50px;
12. width: 50px;
13. border-radius: 50%;
14. border: none;
15. color: var(--btnFontColor);
16. background-color: var(--btnBg);
17. }
18. A continuación, actualice el CSS para el tema claro y oscuro. Defina algunas variables nuevas, --btnBg y --btnFontColor, para especificar el color de fondo y el color de la fuente del botón.
19. .light-theme {
20. --bg: var(--green);
21. --fontColor: var(--black);
22. --btnBg: var(--black);
23. --btnFontColor: var(--white);
24. }
25. .dark-theme {
26. --bg: var(--black);
27. --fontColor: var(--green);
28. --btnBg: var(--white);
29. --btnFontColor: var(--black);
30. }

## Agregar un controlador de eventos

Para hacer que el botón realice alguna acción cuando lo seleccione, necesita un controlador de eventos en el archivo JavaScript. Un controlador de eventos es una forma de ejecutar una función JavaScript cuando se produce un evento en la página. Para el botón, vamos a agregar un controlador de eventos para el evento click; la función del controlador de eventos se ejecuta cuando se produce el evento click.

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

1. En el archivo Javascript, use document.querySelector para obtener la referencia de botón.
2. 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.

1. 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');

});

En el código anterior, ha usado el método toggle para modificar el atributo de clase del elemento <body>. Este método agrega o quita automáticamente las clases light-theme y dark-theme. Este código aplica los estilos oscuros en lugar de los estilos claros al hacer clic y, a continuación, los estilos claros en lugar de los oscuros al volver a hacer clic.

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.

El código JavaScript completo debería verse así.

'use strict';

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

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

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

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

const className = document.body.className;

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

this.textContent = "Dark";

} else {

this.textContent = "Light";

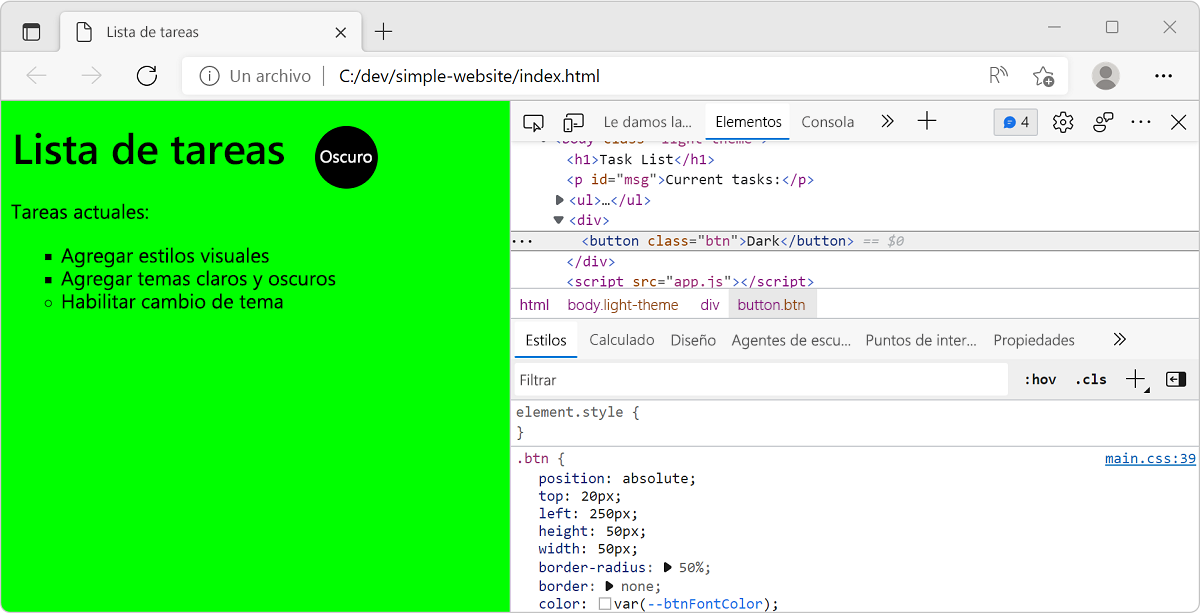
}

});

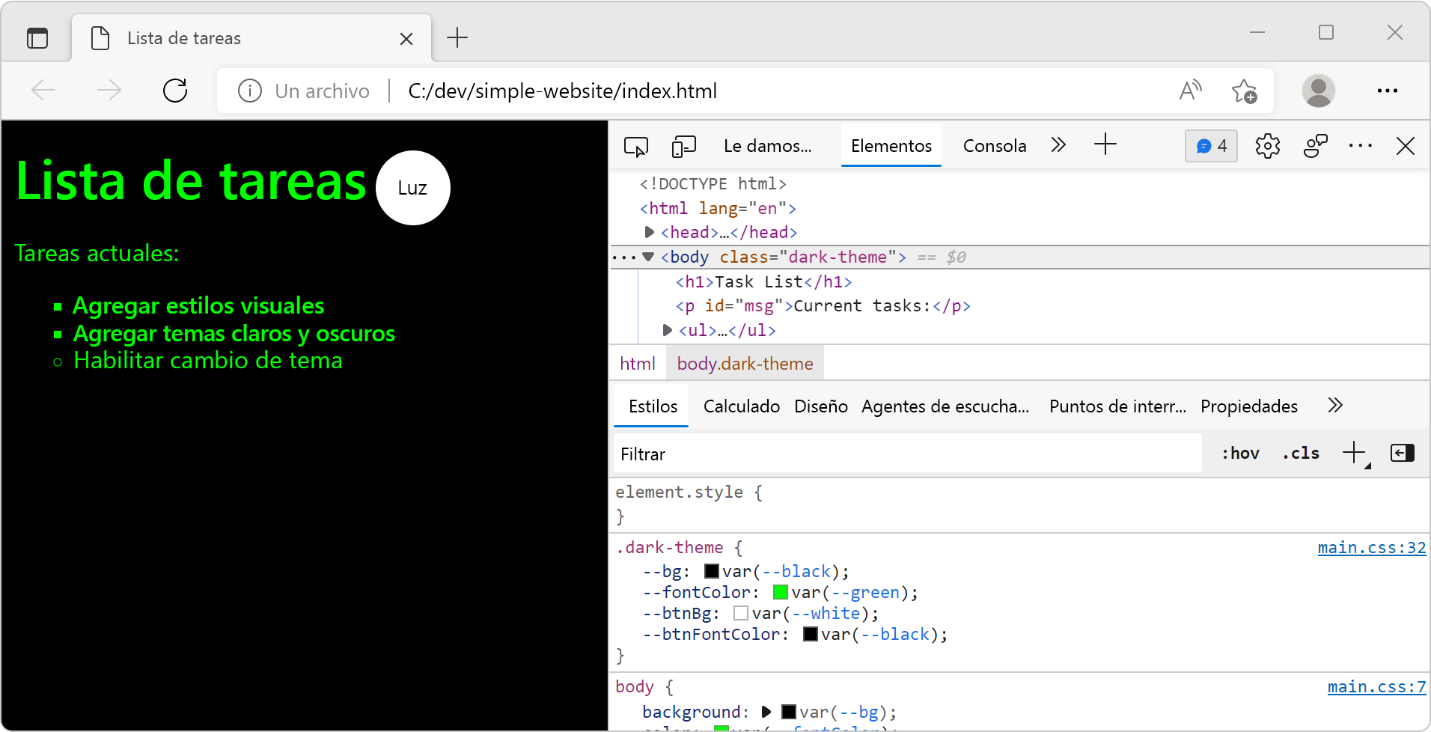
Por convención de JavaScript, se usan *mayúsculas y minúsculas concatenadas* para los nombres de variables con más de una palabra; por ejemplo, la variable className.

## Abrir en el explorador

Para obtener una vista previa, seleccione index.html y seleccione **Abrir en el explorador predeterminado**, o bien vuelva a cargar la misma pestaña del explorador presionando F5.



Seleccione el botón **Oscuro** nuevo para cambiar al tema oscuro.



# Reto 2: Despliegue en Azure