Modulo 5

**Unidad 13: Creá una página web simple con Visual Studio Code**

**Presentación**

En esta unidad aprenderemos a realizar un sitio web simple con Visual Studio Code. Para eso veremos algunas instrucciones y pautas muy sencillas de HTML, el lenguaje que da estructura a las páginas, CSS, que es el lenguaje que le dice al navegador qué aspecto tendrá (colores, tipografías, tamaños, posiciones, etc.) y Java Script, un lenguaje de programación que, en este caso, servirá para dar algo de interacción a nuestro sitio.

Lo que veremos aquí es una base para introducirte en un tema muy amplio y con muchos recorridos posibles. Al finalizar estas dos unidades no serás un experto en diseño web, pero tendrás los conocimientos para entender sus fundamentos y profundizarlos.

Esta unidad representa un cambio respecto a lo que venimos viendo. Hasta aquí trabajamos con programas que nos daban “todo servido”. Ya sea desde la computadora o desde la “nube” nos permitían realizar distintas operaciones, como agregar imágenes, escribir un texto o realizar cálculos tan solo con un par de clics. Con estos programas no necesitábamos saber “código”, simplemente teníamos que identificar las funciones y los íconos.

Las cosas aquí son un poco diferentes, Visual Studio Code, el programa que veremos necesita que les demos las instrucciones necesarias para realizar las tareas utilizando distintos lenguajes, cada uno con sus reglas específicas. Tendremos que hablarle no en uno sino en tres “idiomas” para llegar a nuestro producto final: un sitio web simple con interacción.

En esta primera parte veremos algunas cuestiones básicas de esos lenguajes: HTML para dar estructura al sitio y CSS para darle estilo. También veremos de qué se trata JavaScript y para qué puede servirnos.

|  |
| --- |
| **Escribir código requiere mucha atención. Si nos equivocamos en un signo, una letra, un espacio, el programa no podrá interpretarlo adecuadamente y no obtendremos el resultado deseado.** |

**Cómo se crea una página web**

Una página web consiste en uno o varios archivos escritos en código. Estos archivos, que se crean desde una computadora, luego se envían a un servidor, que es una computadora más potente que está conectada a Internet las 24 horas del día los 365 días del año. Hay empresas que se dedican a alojar estos archivos ya sea en forma gratuita (por lo general con publicidades) o con un costo mensual o anual. Es lo que se conoce como **Hosting**.

Pero, además, para que podamos ver la página desde cualquier navegador tenemos que crear y registrar un dominio, que es el nombre de la dirección que tendrá nuestra página en internet. Por ejemplo, “miempresa.com.ar”

Los pasos para crear una página web son los siguientes:

* Se escribe el código de la página
* Se sube al servidor de la empresa de hosting
* Se crea y registra un dominio
* Se asocia el dominio a la página alojada en el hosting

|  |
| --- |
| Para leer más sobre cómo alojar páginas web podés leer el [**siguiente artículo**](https://es.wikipedia.org/wiki/Alojamiento_web).  Para saber [**cómo registrar un dominio**](https://nic.ar/es/ayuda/instructivos/registro-de-dominio) podés consultar la web nic.ar. |

También existen servicios para crear webs que no requieren aprender código; nos ofrecen una serie de plantillas ya diseñadas que luego es posible modificar. Sin embargo, el conocimiento del código permite una mayor personalización de los sitios.

**Tres lenguajes para un solo sitio**

Para diseñar nuestro sitio web simple con interactividad vamos a necesitar crear tres archivos usando tres lenguajes distintos. Estos son HTML, CSS y JavaScript. Veamos de qué se tratan:

* **HTML**: Son las siglas de Hyper Text Markup Language o lenguaje de marcado de hipertexto. Es el lenguaje que usamos para **dar estructura a las páginas web**. Con él le decimos al navegador **qué elementos tiene nuestra página y cómo está organizada**.
* **CSS**: CSS o Cascade Style Sheet (Hojas de estilos en cascada) es un lenguaje que nos permite decirle al navegador web cómo debe verse la página que escribimos en HTML. Si HTML es el esqueleto de la página, CSS es la piel y la vestimenta. Con CSS podemos establecer qué tipo de letra tiene una web, su color, tamaño o la posición de los objetos; también podemos decir qué color de fondo queremos y darle una apariencia particular a distintas partes de la página.
* **Java** **Script**: Java Script es un lenguaje de programación con el que pueden desarrollarse distintas aplicaciones. Uno de sus usos en las páginas web es darles interactividad, es decir que cuando las personas, por ejemplo, hagan clic en un botón (cuando ocurre un “evento”) algo (un “comportamiento”), ocurra (por ejemplo, que nos salga un mensaje de error).

**Nociones básicas de HTML**

Como mencionamos, HTML nos permite crear la estructura de la página y sus contenidos, ya sean textos, imágenes o multimedia. En su versión más simple un sitio web simple puede tener un código HTML como el siguiente:

|  |
| --- |
| <!DOCTYPE html> <html> <head> <title>Mi sitio web</title> </head> <body> <p>Este es mi primer sitio web </p> <img src=imágenes/globos.jpg> </body> </html> |

Este código indica al navegador:

* qué tipo de documento es (en este caso, HTML)
* que el título de la página es Mi sitio web
* que la página contiene un párrafo que dice “Este es mi primer sitio web”
* y una imagen que está guardada en una carpeta llamada imágenes dentro de mi computadora.

Si escribimos este simple código en un archivo tendremos nuestra primera web. Podemos hacer esto en nuestro bloc de notas, pero existen programas, editores de código que nos facilitan la tarea y nos dan muchas más posibilidades.

A continuación, veremos cómo se ve este sitio web simple usando Visual Studio Code.

**Instalando Visual Studio**

VSCode nos permite (y facilita) escribir el código y obtener un resultado, en nuestro caso, un sitio web, aunque con él pueden desarrollarse aplicaciones y programas en diferentes lenguajes.

Para esta capacitación te vamos a pedir que lo descargues de la siguiente página:

[**https://code.visualstudio.com/**](https://code.visualstudio.com/)

Una vez allí, hace clic en **download** para descargarlo en tu computadora.

Cuando la descarga termine hace doble clic en el archivo y tildá la opción “Acepto el acuerdo” y luego hace clic en Siguiente.

Las siguientes pantallas te indican dónde se instalará el programa. Hacé clic en Siguiente sin marcar otras opciones hasta que te aparezca la opción de instalar.

Cuando termine la instalación tendrás la posibilidad de iniciar el programa automáticamente si dejás marcada la opción “Ejecutar Visual Studio Code”.

Para volver a usar el programa en otra oportunidad tendrás que buscarlo en la lista de programas de tu Windows haciendo clic en el ícono de Windows y bajando hasta la letra V o bien escribiendo Visual Studio en la caja de búsqueda.

**Primeras configuraciones con VS Code**

La pantalla de bienvenida te permite elegir qué aspecto tendrá el programa. Podés elegir entre fondo claro (light), fondo negro (dark) y alto contraste (high contrast).

Antes de avanzar vamos a asegurarnos de que el programa esté en castellano. Para eso deberemos instalar un “paquete” de código adicional desde el mismo programa.

1. Presioná simultáneamente las teclas **Ctrl**, **Mayúsculas y P** de tu teclado.
2. En la barra de búsqueda que aparece escribí el siguiente texto: **Configure Display Language**.
3. Hacé clic sobre **Install Aditional Languages** (“Instalar idiomas adicionales”)
4. Seleccioná el paquete de idioma **Spanish** (castellano) en la lista de la izquierda.
5. Hacé clic en **Install**.
6. Hacé clic en **Restart** **Now** para reiniciar el programa.

**Instalar la extensión “Abrir en el navegador”**

Para que podamos ir viendo el avance del sitio web simple necesitaremos instalar una extensión a VS Code que nos permita abrir el código en el navegador. **Esto no significa que nuestro sitio esté en internet sino que podemos verlo desde el navegador de nuestra computadora sin necesidad de conexión**. Recordemos que para que un sitio web esté “on-line” necesitaremos contar con un hosting y un dominio.

Para instalar esta extensión, seguí estos pasos:

1. Seleccioná el icono **Extensiones** en la **barra de actividad** vertical (el panel izquierdo).
2. En el cuadro de búsqueda **escribí "abrir en"** y, a continuación, seleccioná la extensión **abrir en el explorador**.
3. Seleccioná **Instalar**, y Visual Studio Code instalará la extensión.
4. Volvé al **Explorador** de VS Code; para ello, hacé clic en el icono superior de la**barra de actividad** o presioná **Control+Maýus-E** en Windows.

Este proceso de instalación y configuración tarda un poco, pero solo debe realizarse una vez. Ahora ya estás listo para crear un sitio web.

**Nuestro primer sitio web**

Ahora que tenemos nuestro editor de código configurado veamos cómo se ve el HTML que escribimos más arriba. Para eso, vamos a:

1. abrir VS Code
2. crear una carpeta
3. crear un documento HTML nuevo y guardarlo.

**Creación de una carpeta nueva para la página web**

1. Abrí VS Code. Verás la página de **Bienvenida**. A la izquierda hacé clic en Abrir carpeta (también podés hacerlo desde **Archivo** > **Abrir carpeta**).
2. Se abrirá un cuadro de diálogo en tu computadora. Buscá la ubicación donde quieras crear los archivos del sitio web (por ejemplo, Mis Documentos, Disco C).
3. Una vez que alcanzaste la ubicación deseada verás la opción crear una **Nueva carpeta en la parte superior del cuadro**. Hacé clic allí.
4. Dale nombre a la carpeta creada. En este caso vamos a ponerle: **Carpeta de prueba**. Hacé clic en **Seleccionar carpeta**.

Ya tenemos la carpeta para nuestra primera prueba… pero aún está vacía. Tenemos que crear el archivo HTML.

**Creación de un archivo HTML**

1. En VSCode hacé clic en **Archivo** > **Nuevo archivo** en el menú superior o presioná las teclas **Control+N**.
2. Guardá el archivo presionando las teclas Control+S o haciendo clic en Archivo y luego en Guardar.
3. Escribí **prueba.html como nombre de archivo** y, a continuación, seleccioná **Guardar**.

Ahora tenemos nuestro documento HTML. Sobre la parte derecha de la pantalla, en la pestaña prueba.html, escribí o copia el código que vimos más arriba. Acá te lo recordamos:

|  |
| --- |
| <!DOCTYPE html> <html> <head> <title>Mi sitio web</title> </head> <body> <p>Este es mi primer sitio web </p> <img src= imágenes/globos.jpg> </body> </html>  **Guardá el documento.** |

Para ver el sitio web hace clic con el botón derecho del mouse sobre el archivo prueba.html en la columna de la izquierda y hace clic en **Open in Default Browser** (Abrir en navegador predeterminado).

Deberás ver algo como esto:

Seguramente hay algo diferente en tu página y es que **no te está mostrando ninguna imagen**. Eso es porque el navegador no la encuentra en tu computadora. Y tiene sentido, ya que no bajaste ninguna imagen. Para que te muestre una imagen tendrás que descargar una imagen de internet y guardarla en la carpeta imágenes dentro de la carpeta prueba que creamos.

**Descargar una imagen**

1. Buscá en [**Google imágenes**](https://www.google.com.ar/imghp?hl=es&ogbl) una imagen que te guste.
2. Hacé clic sobre ella con el botón derecho del mouse y seleccioná **Guardar** **imagen**.
3. Guardá la imagen con el nombre imagen1 en una carpeta llamada imágenes dentro de la carpeta prueba.
4. En VSCode reemplazá el nombre globos.jpg en el código HTML por el de imagen1.jpg
5. Guardá el archivo con CTRL+S
6. Hacé clic con el botón derecho del mouse y seleccioná Abrir en navegador predeterminado.

Ahora debería mostrarte la imagen en tu página.

**Estructura básica de una página web**

Como dijimos, HTML es el lenguaje que permite crear las bases de una página web, sus ladrillos.

La estructura más básica de una página web es la que vimos en el ejemplo más arriba.

* Definición del tipo de documento (HTML)
* El comienzo del código HTML
* Head o cabeza: es la parte que contiene instrucciones para los navegadores y buscadores de internet
* Cuerpo, que es la parte visible de la página, que puede contener, texto, imágenes, videos, etc.

**Etiquetas de HTML**

HTML incluye una serie de instrucciones que deben escribirse siguiendo una estructura particular, una sintaxis. En el caso de HTML las instrucciones se llaman “etiquetas” que se escriben dentro de signos de mayor y menor <>. Por ejemplo, la siguiente etiqueta indica el comienzo de un párrafo:

<p>

A continuación de la etiqueta de apertura escribimos el texto que irá en el párrafo, y cuando terminamos, **escribimos una etiqueta de cierre** para indicar que el elemento párrafo ha terminado.

<p> Esto es un párrafo </p>

La mayoría de los elementos de HTML tienen etiqueta de apertura y de cierre (</>), aunque existen algunas que solo tienen etiqueta de apertura, como en el caso del salto de línea <br>.

**¡Si nos olvidamos de las etiquetas de cierre el código no funcionará!**

**Etiquetas meta**

Las etiquetas meta contienen información para navegadores y buscadores. **Van dentro del head del HTML**, y por lo tanto no pueden verse en el cuerpo de la página. Por ejemplo, indican la codificación del idioma, la descripción del sitio web (que aparece al buscarla desde Google u otro buscador) o el explorador en el que se abre la página web.

Algunas etiquetas meta:

* Charset: esta etiqueta indica al navegador la codificación del idioma castellano para su correcta interpretación. <meta charset= "iso-8859-1">
* La etiqueta viewport es importante para que la página web se adapte a los distintos dispositivos (pc, celular, Tablet). <meta name="viewport" content="width=device-width, initial-scale=1.0">
* Description: es la descripción que muestra Google de nuestro sitio cuando aparece en una búsqueda. No tiene etiqueta de cierre. <meta name="description" content="texto de la descripción del sitio">

**Etiqueta Title**

Se utiliza para indicar el título de la página. Este título puede verse en la parte superior izquierda de los navegadores y es lo primero que devuelven los buscadores como Google cuando el sitio aparece en los resultados de la búsqueda. También va siempre dentro de las etiquetas del Head.

|  |
| --- |
| <head> <title>Título de la página </title> </head> |

**Etiqueta body**

Es el cuerpo del documento HTML. El cuerpo encierra todos los contenidos que se muestran (texto, imágenes, tablas). Dentro del cuerpo pondremos las instrucciones o etiquetas que queremos que se vean en nuestra página web.

Algunos ejemplos:

* **Párrafos <p> </p>**: Permite escribir texto en párrafos.
* **Títulos <H1> </H1> <H2> </H2>…<H6> </H6>**: Esta etiqueta indica que lo que está dentro de ella es el título más importante dentro del contenido de la página. Además de H1 hay otros 5 niveles de títulos que van de mayor a menor importancia y sirven para organizar el contenido.

Ejemplo:

* + <H1> Esto es un título </H1>  
    <H2> Esto es un subtítulo </H2>
* **Vínculo o enlace <a> </a>**: Permite crear un enlace a una página web.

<p> Hacé clic en <a href= “<https://www.google.com.ar/>”>aquí</a> para visitar... </p>

En este ejemplo, la palabra aquí será un enlace a la página de google

* **Énfasis <strong> </strong>**: El texto que va dentro de estas etiquetas se verá en negrita, para darle más énfasis.

Ejemplo:  
<p> <strong>esto irá en negritas </strong> </p>  
En este caso primero tenemos que cerrar la etiqueta strong y luego el párrafo.

* **Salto de línea <br>**: Con esta etiqueta HTML le podemos decir al navegador que deje un espacio entre una línea y otra.
* **Imágenes <img> </img>**: Esta etiqueta va siempre acompañada del atributo src o “source”, que indica la ubicación de la imagen. En este caso la carpeta imágenes dentro de la carpeta en la que está el archivo html.  
  <img src= “imágenes/casa.jpg”>
* **Listas <OL> <li></li> <OL> | <UL> <li></li> <UL>**: Las etiquetas ol y li nos sirven para crear listas, de manera similar a cómo utilizamos las viñetas en Word o PowerPoint. La etiqueta OL (ordered list o lista ordenada) se usa para listas con números y la etiqueta UL (unordered list, lista no ordenada) para listas con viñetas. Dentro de las listas, cada elemento se identifica con la etiqueta li.  
  Ejemplo HTML:  
  <ul>  
  <li>Primer elemento</li>  
  <li>Segundo elemento</li>  
  <li>Tercer elemento</li>  
  <ul>  
  Esto se leerá así:
  + Primer elemento
  + Segundo elemento
  + Tercer elemento

Notá que en este ejemplo tenemos que cerrar cada elemento y luego cerrar la etiqueta lista.

**Atributos HTML**

Dentro de las etiquetas, además de los elementos tenemos, en algunos casos, los atributos. Los atributos sirven para modificar o agregar alguna característica al contenido. Por ejemplo, si queremos cambiar la tipografía, la posición de una imagen, la alineación de un texto vamos a agregar un atributo al elemento. Los atributos HTML **van siempre dentro de la etiqueta de apertura**.

Ahora bien, los atributos nos dicen qué característica queremos agregar o cambiar, pero no nos dicen de qué manera cambiará. Para eso necesitamos agregar un valor al atributo.

Por ejemplo, si tengo un texto, pero quiero que aparezca alineado al centro, al elemento tengo que agregarle un atributo de alineación, y a continuación un valor (en este caso, el tipo de alineación). Se vería así:

|  |
| --- |
| <p align=“center”> Esto es un párrafo alineado al centro </p>  Esto es un párrafo alineado al centro |

De todas maneras, actualmente no se recomienda aplicar atributos de estilo con HTML. Para eso, como veremos en seguida, es conveniente usar CSS.

|  |
| --- |
| [**Más información sobre los atributos y valores**](https://desarrolladoresweb.org/html/atributos-html/) en HTML. |

|  |
| --- |
| **Probando etiquetas**  En nuestro archivo prueba.html vamos a borrar todo lo que hemos escrito hasta aquí. Nos quedará completamente vacío.  Ya vimos que todo archivo HTML tiene una estructura típica y que tiene algunas líneas que siempre se repiten. VS Code lo tiene en cuenta y escribirá algunas de ellas por nosotros.  Dentro de la solapa prueba.html hace clic en la línea 1 y escribí ! Cuando veas la línea con la llave inglesa, presioná Enter. También podés escribir html:5 en lugar de!.  Veremos ya desplegada la estructura básica del HTML.  En el head vamos a reemplazar la etiqueta  Después de la etiqueta de apertura del body vamos a agregar un párrafo, dos saltos de línea y un enlace. Finalmente, agregaremos una lista. Veamos cómo  <body> <h1 align= “center”> Esto es un título principal </h1> <p> Esto es un ejemplo de párrafo. Siempre se escribe entre una etiqueta de apertura y una de cierre </p> <br> <br> <p> Esto es un ejemplo de <a href=“[http://www.google.com.ar](http://www.google.com.ar/)”> vínculo a una página web</a></p> <p> Estos son los días de la semana </p> <ol> <li>lunes</li> <li>martes</li> <li>miércoles</li> <li>…</li> </ol> </body>  Al copiar o escribir el código no tenés que incluir las etiquetas de body (ya las creamos con anterioridad). No olvides guardarlo antes de verlo en el navegador. |

**CSS**

CSS es un complemento de HTML. Nos sirve para aplicar el estilo que queremos a los elementos que ya creamos con ese lenguaje. Por ejemplo:

* Con HTML creamos un párrafo en nuestra web
* Con CSS le diremos de qué color será el texto, qué tipografía usaremos, en qué tamaño, si irá subrayado y mucho más.

Para dar estilo al html utilizamos “reglas” que, básicamente dicen cómo se verán los distintos elementos. Por ejemplo, podemos tener un título H1 de un color y tamaño, un H2 de otro color, un párrafo con una tipografía, otro con otra, etc.

Para aplicar CSS tenemos que conocer **las propiedades** que modifican los diferentes elementos. Por ejemplo, la propiedad que modifica el tipo de fuente se llama font-family; la propiedad para alinear el texto se llama text-align y la propiedad del color de texto se llama color.

Algunas propiedades de CSS

|  |  |
| --- | --- |
| **Propiedad** | **Función** |
| Font-family: | Cambia el tipo de fuente. |
| Font-size: | Define el tamaño de la fuente. |
| Color: | Define el color de la fuente. |
| Width: | Define el **ancho** de un elemento |
| Height: | Define el **alto** de un elemento |
| Margin: | Es la distancia entre un elemento y otro (desde el borde de un elemento hacia afuera) |
| Background: | Define los fondos de un objeto. El fondo puede ser una **imagen** o un **color**. |

Para conocer más [**propiedades CSS podés consultar el siguiente enlace**](http://w3.unpocodetodo.info/css3/chuletacss3.php).

**Reglas de CSS**

CSS aplica el estilo al código HTML mediante reglas. Estas reglas se componen del elemento al que se va aplicar el estilo (selector) y la declaración, que es lo que se va a modificar de ese elemento, es decir, la propiedad, y de qué manera se modificará, es decir, con qué valor.

Para que el estilo se aplique debemos escribir la regla utilizando una sintaxis diferente a la que usamos en HTML. En este caso

El selector es el elemento o conjunto de elementos al o a los que se aplicará el cambio de estilo. En el ejemplo anterior, a los títulos h1.

Luego abrimos una llave y dentro de ella colocamos la propiedad, dos puntos, el valor y punto y coma ;. La propiedad es lo que vamos a cambiar del elemento. En este caso, el tipo de letra del h1.

Si quisiéramos cambiar el color del texto, debemos conocer el nombre de la propiedad (color) y los valores posibles, es decir, los nombres de los colores o sus códigos.

p {color: blue;  
}

Para cambiar el tamaño de la tipografía tendríamos que usar la propiedad Font-size y elegir el valor, que en muchos casos se expresa en un número de píxeles (px).

Por ejemplo:

p {font-size:20px;  
}

Si queremos cambiar el estilo, agregando negritas, la propiedad es font-weight y el valor es “bold” (si queremos que sea en cursiva, el valor será “italic”).

p {font-weight: bold;  
}  
p {font-weight: italic;  
}

Por otra parte, si queremos agregar varias propiedades a un elemento (cada una con su valor), podemos hacerlo de manera consecutiva sin olvidar el ; luego de cada valor.

p {font-family:Arial;  
font-size: 20px;  
font-weight:bold;  
}

En este caso tendremos una fuente Arial de 20 px y en negrita.

**No olvides el ; final. Sin él el navegador no interpretará la instrucción que le damos.**

**Otros selectores**

CSS también nos da la posibilidad de aplicar un mismo estilo a varios elementos o aplicar un estilo diferente a elementos de un mismo tipo. Para esto usamos diferentes tipos de selectores:

**Selector grupal:**

Los mismos estilos se aplican a varios elementos. Por ejemplo:

h1, p {Font-family: Arial  
}

Aquí se cambiará la fuente en todos los párrafos y en los títulos h1.

**Selector de clase:**

El selector de clase comienza con un punto (.) y selecciona todo los elementos que nosotros agrupemos por alguna razón.

Supongamos que quiero que algunos párrafos, pero no todos, tengan un color de fuente rojo y tamaño 14px. Entonces, lo que puedo hacer es crear una clase para todos los párrafos que quiera en rojo.

Para eso creo una clase y le pongo un nombre, en este caso, “rojo” y escribo la propiedad y el valor que quiero que tenga.

.rojo {color: red;  
Font-size:14px;  
}

Luego, en mi documento de HTML le asignaré esa clase a los párrafos que quiero cambiar agregando un atributo a la etiqueta de apertura.

<p class="rojo"> este párrafo irá en rojo y con tamaño 14 </p>

Lo que estamos diciendo aquí es que todo párrafo que tenga el atributo de clase “rojo” tendrá las características que definimos en nuestro archivo CSS.

**Selector Id:**

El selector de identidad se aplica a un único elemento que queremos modificar. Ya no serán todos los párrafos ni un grupo de párrafo sino un determinado párrafo (u otro elemento, como una tabla, un titular, etc.). Un Id se puede utilizar una sola vez en cada documento, y a cada elemento solo se le puede aplicar un único id. Para indicar un id en un archivo CSS usamos el signo #.

Para que un título h2 tenga color verde, podemos entonces crear un Id en nuestro archivo CSS de la siguiente manera:

#verde {  
color: green;  
}

Y luego, en nuestro archivo HTML incorporar ese id a un h2.

<h2 id="verde">este título aparecerá en verde</h2>

**Selector universal**

Se utiliza para aplicar estilo a todos los elementos del html, salvo a aquellos para los que se especifique lo contrario.

El selector universal se especifica con un \*

\* {

Existen otros selectores y formas de combinarlos, pero tenemos los suficientes para avanzar en nuestro proyecto.

|  |
| --- |
| **A practicar - Agregando estilos a nuestro archivo de prueba.**  Para agregar estilos vamos a tener que trabajar simultáneamente en nuestro archivo html y en un archivo Css. Para empezar, tendremos abierto nuestro documento prueba.html   1. En el editor de **VS Code**, agregá una línea en blanco presionando enter después del elemento <title> de nuestro documento prueba.html 2. Escribí **link** y, a continuación, presioná ENTER. **VS Code** agregará la siguiente línea. <link rel="stylesheet" href="">   Esta línea de código le dice al navegador que prueba.html está vinculado con una hoja de estilos o archivo CSS. Ahora necesitamos decirle el nombre de nuestro archivo CSS y decirle dónde encontrarlo.  Para eso, donde dice href=“ ” vamos a agregar el nombre de la hoja de estilos (pruebaestilos.css). Quedaría así:  <link rel="stylesheet" href=" pruebaestilos.css"> (no olvides las comillas “”)  Guardá el documento HTML presionando CTRL+S o haciendo clic en Archivo/Guardar.  Este es nuestro nuevo head:  <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>Document</title> <link rel="stylesheet" href=" pruebaestilos.css"> </head>  Ahora vamos a crear un archivo CSS. Desde el explorador de VSCode vamos a Archivo/Nuevo Archivo (o presionamos CTRL+N).  Guardamos este documento vacío con el siguiente nombre dentro de la misma carpeta en la que está el html: pruebaestilos.css  En la línea 1 escribimos nuestras reglas css:  li {color: green; } p{ Font-family: impact; Font-weight: italic; Font-size:16 px; }  Guardá el archivo  Hacé clic derecho sobre el archivo prueba.html (no sobre el CSS) y seleccioná abrir en el navegador. |

**Variables en CSS**

Una última cuestión antes de poner manos a la obra…las variables en CSS.

Las variables en CSS son “espacios” para guardar información sobre estilos, que podemos aplicar en distintos momentos y a distintos elementos. Por ejemplo, podemos guardar información sobre colores, tamaños o tipografías, entre otras, y luego indicar en el archivo HTML cuándo usar esos colores y esas tipografías.

Para agregar una variable CSS tenemos que hacer dos cosas:

1. Crear la variable en la que guardamos la información de estilo.
2. Llamar a la variable es decir, decirle al elemento que “vaya a buscar” las propiedades y los valores a esa variable.

Para crear la variable, dentro de un selector usamos dos guiones y especificamos el nombre de la variable y el valor. Por ejemplo, para indicar que se aplique a todo el documento usaremos un nuevo selector, el selector root.

root {  
--color:blue;  
}

Luego, dentro de un estilo podemos decirle que tome el color de texto de esa variable.

.parrafo {  
var(--color-texto);  
}

Acá le estamos diciendo que a la clase llamada párrafo le aplique el color que guardamos en la variable “color”.

**Unidad 14: Manos a la obra**

**Manos a la obra**

Ya recorrimos de manera muy rápida algunas cuestiones básicas de dos de nuestras tres herramientas principales: Html, CSS. En esta unidad vamos a mostrarte de manera muy sintética qué es Javascript y luego vamos a combinar nuestros tres lenguajes para crear un sitio web simple con interactividad.

**Un par de cuestiones sobre JavaScript**

Como mencionamos más arriba, JavaScript nos permite darle interactividad a nuestra web. Por ejemplo, con JavaScript podemos:

* Cambiar los valores y atributos de un HTML.
* Cambiar los estilos CSS.
* Ocultar algunos elementos y mostrar otros.

Una de las formas de hacerlo es agregar un**botón a nuestro HTML** y decirle que cuando ocurra un evento (por ejemplo, que se haga clic sobre el botón), realice uno de estos cambios.

Veamos un ejemplo adaptado de la página w3School:

En nuestro HTML de prueba reemplazá el código que venimos trabajando por el siguiente código:

<!DOCTYPE html>  
<html>  
<body>  
<h2>Probando JavaScript</h2>  
<p id="prueba">JavaScript puede cambiar el estilo de un elemento de HTML.</p>  
<button type="button" onclick="document.getElementById('prueba').style.fontSize='35px'">Haz clic!</button>  
</body>  
</html>

En este ejemplo al cliquear el botón (el “evento”), el programa modifica el texto con el estilo que guardamos en el id “prueba”.

En este caso agregamos el código JavaScript en el mismo HTML, pero más adelante veremos cómo hacerlo desde un archivo externo.

**Condicionales en JS**

En JavaScript podemos generar una instrucción que dé un cierto resultado dependiendo de si se cumple o no una condición. Esto resulta muy útil en un sitio web interactivo ya que dependiendo de la acción del usuario podemos ejecutar distintas respuestas.

Para eso podemos usar, entre otros:

* El comando *if* para que se ejecute un bloque de código cuando se verifica una condición,
* *Else* para que se ejecute el mismo bloque de código si la condición no se cumple
* *Switch*, para especificar varios bloques alternativos de código para que se ejecuten.

Veremos el uso de estas variables y condicionales a la hora de darle algo de interactividad a nuestra página.

**Manos a la obra**

Ya estamos en condiciones de combinarlas en una página web básica usando VSCode. Para eso vamos a olvidarnos de nuestros archivos de prueba y vamos a crear una carpeta nueva con crear tres archivos:

* El documento html, que contiene los elementos de la página web.
* El documento CSS que aplica estilos al html
* El documento de Javascript, para aplicar interactividad a un elemento.

**Crear una carpeta nueva para la página web**

Ya vimos cómo crear una carpeta en VSCode, pero vamos a repasarlo.

1. Cerrá los documentos que tengas abiertos en VSCode
2. A la izquierda hace clic en Abrir carpeta (también podés hacerlo desde **Archivo > Abrir carpeta**)
3. Buscá la ubicación donde quieras crear los archivos del sitio web y hacé clic en **Nueva carpeta en la parte superior del cuadro**.
4. Dale nombre a la carpeta creada. En este caso será: sitio-web-simple y hacé clic en **Seleccionar** **carpeta**.

Ya tenemos la carpeta del sitio web. Ahora tenemos que crear los archivos que mencionamos más arriba: html, css y jss.

**Creación de los archivos**

1. Hacé clic en **Archivo** > **Nuevo archivo** en el menú superior o presioná las teclas Control+N (si estás usando Windows).
2. Guardá el archivo vacío presionando las teclas Control+S.
3. Escribí **index.html** como nombre de archivo y, a continuación, seleccioná **Guardar**.  
   Index.html será nuestro archivo principal (así suele llamarse la página principal de un sitio).
4. Repetí los pasos anteriores para crear dos archivos más, con los nombres main.css y app.js. Cuando hayas terminado, en el explorador de Visual Studio Code, verás que la carpeta del proyecto**sitio-web-simple** contiene los siguientes archivos:
   * index.html
   * main.css
   * app.js

|  |
| --- |
| **Es posible incluir todo el código en un único archivo HTML, sin embargo, tener tres archivos separados para el contenido (HTML), estilos (CSS) y lógica (JSS) ayuda a mantener organizado el proyecto y modificarlo más fácilmente en caso de necesidad.**  **Tener una página CSS independiente se denomina CSS externa. El código suele ser más fácil de leer cuando es conciso y compartimentado. En lugar de actualizar cada página HTML con código CSS, es posible realizar cambios en un único archivo CSS y hacer que esas actualizaciones se apliquen a todas las páginas web vinculadas con este archivo.**  **Por otra parte, esta separación es un ejemplo de mejora progresiva. Si JavaScript no está habilitado o no es compatible con los navegadores, por ejemplo, HTML y CSS seguirán funcionando. Sin embargo, si CSS no es compatible al menos aparecerá su contenido HTML.** |

**Agregar HTML básico a la aplicación web**

Por el momento, el sitio web tiene tres archivos vacíos. Vamos a agregar algo de código a nuestro HTML para crear la página web que deben mostrar los navegadores.

En la parte central arriba deberías ver tus tres archivos vacíos. Hacé clic en la solapa index.html, es decir en el archivo sobre el que trabajaremos.

|  |
| --- |
| **Si no ves las solapas con los tres archivos significa que los cerraste. Tendrás que abrirlos nuevamente desde Archivo / Abrir y buscarlos en la carpeta en la que los guardaste. Empecemos, entonces, por index.html.** |

Dentro de la solapa index.html hace clic en la línea 1 y escribí !. Cuando veas la línea con la llave inglesa, presioná Enter.

Ya tenemos nuestra estructura básica de HTML. Ahora vamos a modificar el head y, luego, el body.

|  |
| --- |
| **Si el código de plantilla HTML5 no se agrega al archivo index.html, probá cerrar y volver a abrir el archivo.** |

**Editando el Head**

Vamos a hacer algunas modificaciones en el Head.

En la línea <html lang="en"> vamos a reemplazar “en” por “es”. Con esto le estaremos diciendo al navegador que el idioma de la página es español y no inglés.

También vamos ponerle un título a nuestra página, que se verá en la parte superior del navegador. Para eso vamos a reemplazar “Document” por “sitio web simple”. Ese será el nombre de nuestra página. Quedará así

<title>Sitio web simple</title>

Este es, entonces, nuestro código por el momento:

<!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>Sito web simple</title>

**Vincular el archivo HTML con los estilos CSS**

Como ya mencionamos, una buena práctica es mantener el código que se refiere a los estilos separados del código HTML del contenido. Para que esos estilos se puedan aplicar necesitamos vincular.

1. En el editor de **VS Code**, agregá una línea en blanco después del elemento <title> presionando enter.
2. Escribí **link** y, a continuación, presioná ENTER. **VS Code** agregará la siguiente línea.  
   <link rel="stylesheet" href="">

Para eso, donde dice href=“ ” vamos a agregar el nombre de la hoja de estilos (main.css). Quedaría así:

<link rel="stylesheet" href="main.css"> (no olvides las comillas “”)

Guardá el documento HTML presionando CTRL+S o haciendo clic en Archivo/Guardar.

Este es nuestro nuevo head:

<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>Sitio web simple</title>  
<link rel="stylesheet" href="main.css">  
</head>

**Edición del cuerpo (body)**

En la unidad anterior vimos que las etiquetas de los elementos visibles de la página web deben escribirse después de la etiqueta de apertura de <body>. Empecemos a darle contenido a la página. ¡No olvides cerrar las etiquetas!

Para empezar, agreguemos el título principal o h1.

1. Agregá al código que venimos trabajando un elemento <h1> con el siguiente texto: “Lista de tareas”. Presioná enter para que el siguiente elemento se escriba en una nueva línea. De esa manera será más fácil de leer.
2. A continuación, agregá un elemento <p> de párrafo con el siguiente texto: “Tareas actuales:”
3. Agregá un atributo id con el nombre “msg” al párrafo (esto le da un nombre al párrafo para que después podamos aplicarle un estilo único).  
   Debería verse así: <p id="msg">Tareas actuales:</p>
4. Ahora vamos a agregar una lista no ordenada <ul> que contenga tres elementos <li> . Con los siguientes textos:
   * “Agregar estilos visuales”
   * “Agregar temas claro y oscuro”
   * “Permitir el cambio de tema”.
5. Agregá un atributo class con el nombre “lista” a los dos primeros elementos de la lista (esto nos permitirá luego aplicarles un mismo estilo).
6. Revisá el código para que coincida con el siguiente ejemplo o copialo desde aquí.  
   <!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>Sitio web simple </title>  
   <link rel="stylesheet" href="main.css">  
   </head>  
   <body>  
   <h1>Lista de tareas</h1>  
   <p id="msg">Tareas actuales:</p>  
   <ul>  
   <li class="lista">Agregar estilos visuales </li>  
   <li class="lista">Agregar tema claro y oscuro </li>  
   <li>Permitir cambio de temas </li>  
   </ul>  
   </body>  
   </html>

7- Guardá el archivo presionando Control+S.  
8- Abrir el proyecto en el explorador

|  |
| --- |
| **IMPORTANTE**  **Asegurate de hacer clic con el botón derecho del mouse en el icono o el texto del nombre de archivo. Si aparece el cuadro de diálogo Visual Studio Code, seleccione Yes, I trust the authors (Sí, confío en los autores).** |

Nuestro sitio web simple es eso… simple. Por ahora lo vinculamos con la hoja de estilos css, pero no hemos escrito nada en ella, lo que significa que lo que vemos es solo la apariencia predeterminada del HTML. A continuación, veremos cómo darle más vida aplicando estilos.

**Aplicación de estilos al código HTML con CSS**

En esta sección

1. aplicaremos estilos CSS a los elementos de la página HTML,
2. agregaremos algo de código CSS para definir los temas claro y oscuro (esto es la apariencia de la pantalla según sea un fondo negro con letras claras o a la inversa).
3. comprobaremos los resultados en el explorador.

En **Visual Studio Code**, abrí el archivo main.css. Si ya lo tenés abierto, hacé clic sobre la pestaña con el nombre main.css. Escribí o copia en ese documento vacío el siguiente código.

body {  
font-family: monospace;  
}  
  
ul {  
font-family: helvetica;  
}

Con esto le estamos diciendo al navegador que la fuente de nuestro sitio web (es decir de su parte visible, que es el body) será monospace, y que la fuente de la lista será helvética.

**No olvides cerrar las llaves y poner punto y coma luego del valor de la propiedad.**

Vamos a especificar ahora cómo se verán algunos de los elementos del HTML a través de los selectores Id y de clase que definimos anteriormente. Recordá que le dimos al párrafo el id=msg y a los primeros elementos de la lista la clase llamada “lista”.

En nuestro archivo main.css

1. Copiá o escribí el siguiente el código.  
li {  
list-style: circle;  
}  
  
.lista {  
list-style: square;  
}  
  
#msg {  
font-family: monospace;  
}

**El prefijo de punto para .list indica que se trata de un selector de clases. El # indica que se trata de un selector de Id.**

En el fragmento de código anterior estamos diciendo que la lista llevará viñetas circulares, salvo aquellos ítems que pertenezcan a la clase llamada “lista”, que llevarán un estilo de viñeta cuadrado.

También estamos diciendo que nuestro párrafo identificado como msg llevará otra tipografía, en este caso, monospace.

Los nombres que se usan para los selectores pueden ser arbitrarios, pero siempre tienen que coincidir con lo que hayas definido en el código HTML.

2. Guardá el trabajo mediante Control+S.  
3. Abrí el HTML en el explorador (Recordá guardar primero tu archivo CSS).

**Aunque solo estuvieras editando el archivo main.css, para obtener una vista previa de los cambios, tenés que seleccionar el archivo index.html ya que es allí donde se reflejarán los cambios.**

**Incorporación de un tema claro**

Un tema es la apariencia general de un sitio. Probablemente hayas escuchado de temas claros y oscuros. Los temas oscuros son aplicaciones o webs con fondo oscuro y letra clara, que muchos usuarios prefieren a la hora de leer en pantalla. Otros prefieren los temas claros que son los convencionales de fondo blanco o de otro color claro y letras oscuras.

Ahora vamos empezar a agregar los dos tipos de temas para que al final, con JavaScript, podamos optar por uno de ellos. Tenemos que trabajar por un lado, en el archivo CSS, por otro, en el HTML.

Para empezar vamos a definir un color claro mediante una clase CSS.

1. En el **archivoCSS**, agregá el código siguiente al final del archivo.  
   .tema-claro {  
   color: #000000;  
   background: #00FF00;  
   }  
   En este ejemplo, #000000 especifica el negro para el color de fuente y #00FF00 especifica el verde para el color de fondo.
2. En el **archivo HTML**, vamos a agregar la clase tema-claro en la etiqueta de apertura del <body> para que el selector de clases del tema claro se aplique a todos los elementos visibles de nuestro sitio web.  
   <body class="tema-claro">
3. Guardá ambos documentos
4. Visualizá el resultado en el explorador

La página web tendrá el fondo verde que especificamos mediante CSS. Si no funciona, controlá que hayas guardado ambos documentos antes de visualizar el html en el explorador.

**Incorporación de un tema oscuro**

Ahora vamos incorporar la compatibilidad con un tema oscuro.

1. Para eso vamos a agregar algunas variables a la raíz de la página, en la parte superior del archivo CSS.

:root {  
--green: #00FF00;  
--white: #ffffff;  
--black: #000000;  
}

En este ejemplo definimos tres variables de color que están asociadas a la raíz de la página.

2. Al final del archivo CSS, reemplazá el selector tema-claro por el código siguiente, y agregá el selector tema-oscuro.

.tema-claro {  
--bg: var(--green);  
--fontColor: var(--black);  
}  
.tema-oscuro {  
--bg: var(--black);  
--fontColor: var(--green);  
}

En el código anterior definimos algunas variables nuevas, bg y fontColor, para especificar el color de fondo y el color de la fuente.Usamos la palabra clave var para que tome los valores de propiedad de las variables llamadas –black y --green.

3. A continuación, en el archivo CSS, después del selector :root, reemplazá el selector body actual por el código siguiente.

\* {  
color: var(--fontColor);  
font-family: helvetica;  
}  
body {  
background: var(--bg);  
}

En este ejemplo, se usa el selector \* para establecer la propiedad color predeterminada para todos los elementos de la página. Esto no afectará a las propiedades color y background, que toman los valores de las variables definidas en los selectores de los temas claro y oscuro.

4. En el archivo CSS, quitá el selector #msg para que podamos aplicar la misma fuente a todos los elementos.

5. Para ver el tema oscuro, abrí el archivo index.html y editá manualmente el tema predeterminado del atributo de clase <body> en tema oscuro (tema-oscuro).

6- Abrí el HTML en el explorador.

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

**Adición de interactividad con JavaScript**

Ahora vamos a configurar un archivo JavaScript que nos permita pasar del tema claro al oscuro haciendo clic en un botón. Vamos a:

* Vincular un documento JavaScript con nuestro HTML
* Crear un botón en HTML para cambiar entre los temas claro y oscuro
* Conectaremos el botón al código JavaScript para que se realice el cambio.

**Vínculo a JavaScript**

Al igual que con CSS, podemos agregar JavaScript directamente al archivo HTML, pero es recomendable guardarlo en un archivo independiente, lo que facilita la reutilización del código en varias páginas web.

Para vincular ambos archivos vamos a utilizar la etiqueta de script <script> en HTML.

1. En **VS Code**, abrí el archivo index.html.
2. En una nueva línea **antes del elemento de cierre </body>**, escribí **script:src** y después seleccioná ENTRAR. Las etiquetas de apertura y cierre de un script se agregarán al código.
3. Ahora tenemos que indicar cuál es el archivo que se va a vincular. Para eso modificá el elemento <script> para agregar el archivo app.js como fuente (src) como se muestra a continuación (asegúrate de que se encuentra después del elemento </ul> de cierre de la lista.

**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.**

**Agregamos un botón**

En nuestro sitio web simple definimos, a través de variables un tema claro y un tema oscuro. Necesitamos ahora la posibilidad de cambiar de uno a otro. Para eso, vamos a crear un botón en nuestro HTML con la etiqueta <button>.

1. En el archivo HTML, agregá un elemento <button> con una clase (“btn”) al final de la lista. Vamos a hacerlo dentro de un elemento <div>. La etiqueta HTML div sirve para “contener” elementos de manera de poder aplicarles un estilo a todo lo que esté dentro de ese contenedor.

<ul>  
<li class="lista">Agregar estilos visuales </li>  
<li class="lista">Agregar temas claro y oscuro</li>  
<li>Permitir el cambio de temas</li>  
</ul>  
<div>  
<button class="btn">Oscuro</button>  
</div>  
<script src="app.js"></script>

En el archivo CSS, vamos a agregar un selector de clase para el botón HTML. Para que los colores de los botones sean distintos de los colores generales de los temas claro u oscuro vamos a establecer las propiedades color y background-color en el selector de botón.

Este selector de clases es específico del botón e invalida el selector universal (\*) que usamos para aplicar colores de fuente en el archivo CSS.

Vamos a agregar algunas reglas 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.

.btn {  
position: absolute;  
top: 20px;  
left: 250px;  
height: 50px;  
width: 50px;  
border-radius: 50%;  
border: none;  
color: var(--btnFontColor);  
background-color: var(--btnBg);  
}

Ahora vamos a actualizar el CSS para el tema claro y oscuro. Vamos a definir algunas variables nuevas, btnBg y btnFontColor, para especificar el color de fondo y el color de la fuente del botón dentro de cada tema.

.tema-claro {  
--bg: var(--green);  
--fontColor: var(--black);  
--btnBg: var(--black);  
--btnFontColor: var(--white);  
}  
  
.tema-oscuro {  
--bg: var(--black);  
--fontColor: var(--green);  
--btnBg: var(--white);  
--btnFontColor: var(--black);  
}

**Agregar un controlador de eventos**

Para hacer que el botón realice una acción al hacer clic en él necesitamos un controlador de eventos en el archivo JavaScript.

Para poder agregar el controlador de eventos necesitamos una referencia al botón.

1. En el archivo JavaScript, vamos a usar document.querySelector para obtener la referencia de botón mediante el siguiente código:

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

2. A continuación, agregaremos la escucha de eventos y el controlador de eventos para el evento clic.

switcher.addEventListener('click', function() {  
document.body.classList.toggle('dark-theme')  
});

En el código anterior, usamos el método toggle para cambiar el elemento a la clase tema-oscuro. Esto aplica automáticamente los estilos del tema oscuro en lugar del tema claro.

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

El código JavaScript completo debería verse así.

Guardá todos los archivos y abrí index.html desde VS-Code. Cuando se abra la página, hace clic en el botón Oscuro para cambiar al tema oscuro. Cuando se cargue el tema oscuro, presioná en Claro para volver al tema claro.

**Si no te funciona, es muy probable que hayas cometido un error al escribir alguna instrucción, como una letra incorrecta o faltante. Revisá tu código con cuidado y línea por línea sin descuidar signos de puntuación, llaves, corchetes, etc. ¡Escribir código requiere mucha atención y paciencia para revisar los errores!**

|  |
| --- |
| **Próximos pasos**  Hasta aquí llegamos con las nociones básicas de los tres pilares para crear un sitio web interactivo. Esperamos que este recorrido rápido te haya sido de interés y puedas profundizarlo con otros recursos si así lo deseás.  Hay mucho material y tutoriales gratuitos en la web para seguir explorando este camino. En este [**artículo podrás encontrar algunos de estos recursos**](https://www.bbc.com/mundo/noticias/2015/06/150615_tecnologia_como_aprender_programacion_gratis_ig).  ¡Mucha Suerte! |