UTN FACULTAD REGIONAL CÓRDOBA
INGENIERÍA EN SISTEMAS DE INFORMACIÓN

Desarrollo de Software
Notas de Clase

Clase 17 - Desarrollo Web

¿Qué es el desarrollo Web?

Desarrollo Web significa construir y mantener sitios web. Al comienzo los sitios se componían de un conjunto de **páginas web** muy sencillas. Sin embargo, en la actualidad, y con el avance de la tecnología, en una web se puede hacer cualquier cosa. Es por eso que el desarrollo web se ha popularizado tanto que se ha convertido en un área bastante demandada.

Los términos sitio web y aplicación web suenan parecido y con frecuencia se utilizan sin distinción, hablando de ellos como si fueran lo mismo. Sin embargo, se trata de plataformas digitales con propósitos específicos que responden a necesidades muy diferentes. Cada una ofrece un conjunto distinto de funcionalidades a los usuarios, las que dependen directamente de los objetivos de un proyecto.

En pocas palabras, un **sitio web** es un conjunto de páginas estáticas que entregan información. Por su parte, las **aplicaciones web** son plataformas principalmente interactivas que se centran en que los usuarios realicen acciones y accedan a contenido dinámico recuperado generamente de diferentes fuentes de información. Una aplicación web puede ser parte de un sitio en un proyecto, pero no al revés.

Las webs que solo tienen frontend se les conoce como webs estáticas. Se les denomina así precisamente porque siempre muestran la misma información para todos los usuarios. Ejemplo de eso son las páginas informativas o personales, puede ser perfectamente estáticas porque todos los usuarios que entren en la web la van a ver igual, es la misma información.

Pero si se piensa en una web como una red social, no puede ser estática porque necesita guardar usuarios en una base de datos y cada uno de estos usuarios ven cosas distintas en su timeline.

Características de las aplicaciones Web

- **Utilizan distintos lenguajes de programación:** A nivel técnico las aplicaciones web dinámicas son mucho más complejas que las estáticas y suelen utilizar los siguientes lenguajes de programación: PHP, JavaScript y Asp, pues son los que permiten estructurar de mejor forma el contenido.
- **Se gestionan en un CMS:** A nivel técnico las aplicaciones web dinámicas son mucho más complejas que las estáticas y suelen utilizar los siguientes lenguajes de programación: PHP, JavaScript y Asp, pues son los que permiten estructurar de mejor forma el contenido.
- **No es necesario entrar a un servidor:** El diseño, la base de datos, el contenido y las actualizaciones que se deseen hacer se pueden ejecutar sin necesidad de utilizar un servidor. Incluso se puede modificar el diseño fácilmente desde el panel de administración. Todos los cambios realizados se actualizan en tiempo real.

Ventajas

- Las aplicaciones web dinámicas pueden ser visitadas desde cualquier dispositivo tecnológico con conexión a internet y se pueden ejecutar en todos los sistemas operativos.
- Son mucho más fácil de actualizar que una aplicación estática o una página web.
- No es necesario descargarlaso instalarlas en el móvil o el ordenador.
- Se cargan mucho más rápido que una página web.
- Se desarrollan en muy poco tiempo.
- Permiten implementar distintas funcionalidades como bases de datos y foros, lo cual es muy usado para facilitar procesos empresariales en la intranet.
- Puede ser gestionada por un editor de contenido, no es necesario que la administre un webmaster.
- No exige el uso de servidor.
- Facilita a los usuarios la localización del contenido que desean visitar y la manipulación del mismo es mucho más amigable.
- Es una excelente herramienta para recopilar los datos de los usuarios que han navegado en la app web.

HTML (*)

Para construir un sitio Web es necesario diseñar un conjunto páginas WEB. Dichas páginas no son otra cosa que documentos de texto plano estructurados mediante un Lenguaje de Marcas de Hipertexto o **HTML**.

Hipertexto hace referencia a los enlaces que conectan páginas web entre sí, ya sea dentro de un único sitio web o entre sitios web. Los enlaces son un aspecto fundamental de la Web.

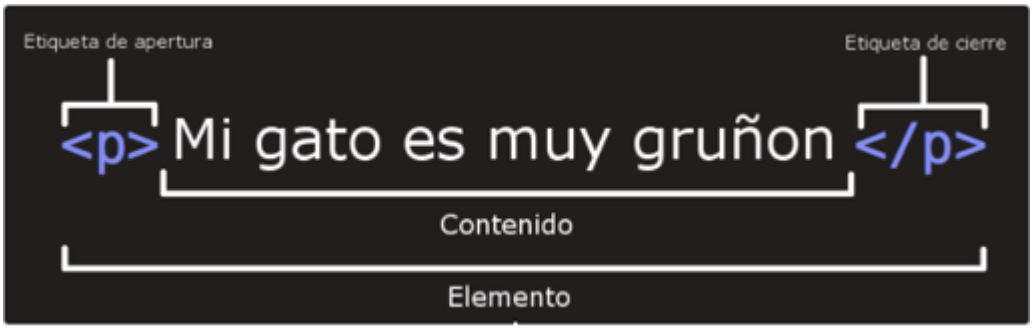
HTML es un estándar a cargo del World Wide Web Consortium (W3C), organización dedicada a la estandarización de casi todas las tecnologías ligadas a la web, sobre todo en lo referente a su escritura e interpretación. En su última versión **HTML5** varias de sus anteriores etiquetas (HTML4) se han simplificado, por lo que es mucho más fácil y rápido escribir código.

HTML5 es mucho más dinámico e incluye elementos multimedia. Soporta de forma nativa el vídeo y el audio, e incluso es posible crear juegos o animaciones con él.

Si se piensa en la construcción de una casa, los cimientos, las vigas (columnas) y las paredes conforman la estructura del proyecto. De igual manera HTML define la estructura y el significado del contenido web.

Para editar una página HTML y posteriormente visualizarla, todo lo que se necesita es un editor de texto y un navegador Web. Para ver una página HTML no es necesario una conexión a Internet, cualquier explorador Web permite hacerlo de manera local.

A diferencia de los lenguajes de programación convencionales, HTML utiliza una serie de etiquetas (o marcas) intercaladas llamadas **elementos HTML**. Dichos elementos serán posteriormente interpretadas por los exploradores encargados de visualizar la página o el documento Web con el fin de establecer el formato.



Las partes principales del elemento son:

- La etiqueta de apertura: consiste en el nombre del elemento (en este caso, p), encerrado por paréntesis angulares (< >) de apertura y cierre. Establece dónde comienza o empieza a tener efecto el elemento —en este caso, dónde es el comienzo del párrafo—.
- La etiqueta de cierre: es igual que la etiqueta de apertura, excepto que incluye una barra de cierre (/) antes del nombre de la etiqueta. Establece dónde termina el elemento —en este caso dónde termina el párrafo—.
- El contenido: este es el contenido del elemento, que en este caso es sólo texto.

- El elemento: la etiqueta de apertura, más la etiqueta de cierre, más el contenido equivale al elemento.

Todo elemento HTML puede tener **atributos** en forma de clave=valor, tal como se muestra en la siguiente imagen:

Los atributos contienen información adicional acerca del elemento, la cual no quieres que aparezca en el contenido real del elemento. Aquí **class** es el nombre del atributo y **editor-note** el valor del atributo. En este caso, el atributo class permite darle al elemento un nombre identificativo, que se puede utilizar luego para apuntarle al elemento información de estilo y demás cosas.

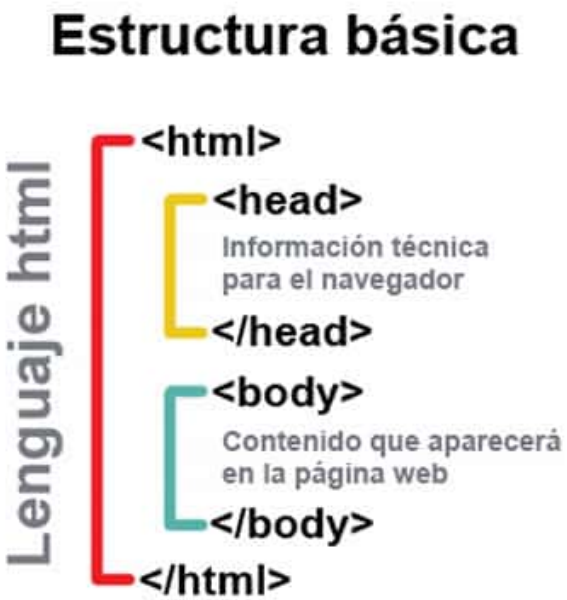
Un atributo debe tener siempre:

- Un espacio entre este y el nombre del elemento (o del atributo previo, si el elemento ya posee uno o más atributos).
- El nombre del atributo, seguido por un signo de igual (=).
- Comillas de apertura y de cierre, encerrando el valor del atributo.

Los atributos siempre se incluyen en la etiqueta de apertura de un elemento, nunca en la de cierre.

Estructura básica de un documento HTML (*)

Tal como se muestra en la siguiente imagen, todo documento HTML se compone de un conjunto de elementos individuales que son combinados para formar una página HTML entera.



- `<html></html>`. Este elemento encierra todo el contenido de la página entera y, a veces, se le conoce como el elemento raíz (root element).
- `<head></head>`. Este elemento actúa como un contenedor de todo aquello que quieres incluir en la página HTML que no es contenido visible por los visitantes de la página. Incluye palabras clave (keywords), una descripción de la página que quieres que aparezca en resultados de búsquedas, código

CSS para dar estilo al contenido, declaraciones del juego de caracteres, etc. Dentro del `<head>` comunmente se incluyen:

- `<meta charset="utf-8">`. Este elemento establece el juego de caracteres que tu documento usará en utf-8, que incluye casi todos los caracteres de todos los idiomas humanos. Básicamente, puede manejar cualquier contenido de texto que puedas incluir.
- `<title></title` .Establece el título de tu página, que es el título que aparece en la pestaña o en la barra de título del navegador cuando la página es cargada, y se usa para describir la página cuando es añadida a los marcadores o como favorita.
- `<body></body>`. Encierra todo el contenido que deseas mostrar a los usuarios web que visiten tu página, ya sea texto, imágenes, videos, juegos, pistas de audio reproducibles, y demás.

> Al comienzo del documento, antes del elemento raíz, suele indicarse el elemento `<!DOCTYPE html>`. Este elemento no es una etiqueta pero sirve para identificar la versión de HTML en la que está escrita el documento, así de sencillo. `<!DOCTYPE html>` indica que el documento es HTML 5.

Etiquetas basicas HTML (*)

Las etiquetas indican a los exploradores Web cómo tienen que mostrar el texto y los gráficos. La siguiente tabla muestra las etiquetas básicas como encabezados, enlaces, imágenes, listas y formularios.

Etiqueta	Significado	Detalle
`<h1>` `<h2>` `<h3>` `<h4>` `<h5>` `<h6>`	Cabeceras	Los documentos HTML, al igual que los documentos Word pueden tener cabeceras para indicar Títulos y subtítulos. El tamaño del texto es mayor cuanto mayor es el nivel (siendo 1 el nivel más alto).
`<hr>`	Línea horizontal	Permite dibujar un separador con forma de línea horizontal
`<p>`	Párrafos	Se utiliza para escribir un párrafo de texto. Todo párrafo va predecida automáticamente por una línea en blanco. Junto con el texto es posible incluir etiquetas en línea para resaltar partes específicas del texto: `<span><strong>` `<small><del>`.
` `	Retorno de carro	Los retornos de carro en el texto son ignorados; por lo tanto, para introducir uno se utiliza esta etiqueta
`<a>`	Enlaces	Permite definir un enlace. Esta etiqueta delimita el texto que se quiere utilizar para enlazar con otra página o parte de la misma. Por ejemplo: `<a href='#identificador'>` Sección dentro del documento `</a` para referenciar una sección dentro del documento, o: `<a href='/path de la pagina'>` Otra página del sitio `</a>` para referenciar a otra página dentro del sitio, o bien: `<a href='URL externa+protocolo' target='blank' rel="noopener noreferrer" >Otra página en Internet</a>` para enlazar a otro documento fuera del sitio.

Etiqueta	Significado	Detalle
`<img>`	Imágenes	Mediante el atributo **src** permite especificar el nombre del fichero que contiene la imagen. "<img src='URL del recurso' alt='Texto alternatico'>". El atributo **alt** permite indicar un texto alternativo en caso de que la imagen no pueda ser cargada exitosamente.
`<ol>`	Listas ordenadas	`<ol><li>item</li></ol>`. El elemento interno `<li>` (list item) permite indicar cada elemento de la lista enumerada. Salida: 1. item
`<ul>`	Listas desordenas	`<ul><li>item</li></ul>`. Idem anterior con viñetas
`<form>`	Formularios	Son uno de los principales puntos de interacción entre un usuario y un sitio web. El atributo más importante es **action**='recurso' que permite definir la acción por defecto del formulario cuando se envían datos a un servidor. Este atributo va acompañado de **method**='GET/POST' para indicar el tipo de método utilizado para generar la petición HTTP.
`<imput>`	Campos de entrada	Existe una amplia variedad de controles de entrada de datos. Para crearlo se utiliza la etiqueta **input** con los atributos **type** y **name**. Para ver todas las posibilidades acceder a: MDN Reference - Input

> Para ver el detalle completo de etiquetas de HTML acceder al siguiente pdf: Etiquetas HTML

Formularios HTML (*)

Un formulario HTML es una sección de una página web que permite al usuario ingresar y enviar datos a un servidor web. Los formularios HTML están compuestos por una serie de elementos, como campos de entrada de texto, botones de opción, casillas de verificación, menús desplegables, etc.

Cuando un usuario completa y envía un formulario HTML, los datos ingresados en el formulario se envían al servidor web que está detrás del sitio web. El servidor web luego procesa los datos y realiza alguna acción basada en la información que recibió. Por ejemplo, si el formulario es un formulario de contacto, el servidor podría enviar un correo electrónico al propietario del sitio web con los detalles de la consulta del usuario.

Los formularios HTML son una parte importante del diseño web y se utilizan comúnmente en una amplia variedad de aplicaciones, como encuestas, registros de usuarios, compras en línea, entre otros. Los desarrolladores web utilizan una combinación de lenguaje HTML, CSS y JavaScript para crear formularios interactivos y atractivos para los usuarios.

A continuación, se enumeran algunas de las propiedades principales de la etiqueta HTML `<form>`:

- **action**: es la URL a la que se enviarán los datos del formulario cuando se envíe. Es importante que se especifique una URL válida para que los datos del formulario se puedan procesar correctamente.
- **method**: especifica el método HTTP que se utilizará para enviar los datos del formulario. Los dos métodos más comunes son GET y POST. GET se utiliza para obtener información de la URL, mientras que POST se utiliza para enviar información a un servidor.

- **enctype**: especifica cómo se codificarán los datos del formulario antes de enviarlos. El valor predeterminado es application/x-www-form-urlencoded, pero también se pueden usar otros valores, como multipart/form-data para enviar archivos.
- **target**: especifica el destino donde se abrirá la respuesta del servidor después de enviar el formulario. Si se omite, el valor predeterminado es _self, lo que significa que la respuesta se mostrará en la misma ventana o pestaña que el formulario.
- **name**: especifica un nombre para el formulario que se puede utilizar para referirse al formulario en JavaScript y CSS.
- **autocomplete**: especifica si el navegador debe completar automáticamente los campos del formulario. Los valores posibles son on y off.
- **novalidate**: especifica que el formulario no debe ser validado cuando se envíe. Esto se usa comúnmente en pruebas y en formularios que no requieren validación.
- **class y id**: especifican clases y un identificador únicos para el formulario, que se pueden utilizar para aplicar estilos y scripts personalizados al formulario.

Ejemplo de aplicación

Se propone como actividad paso a paso diseñar un página con un formulario de contacto que permita ingresar los datos: nombre completo, correo electrónico, teléfono, un mensaje y una opción desde una lista de opciones. Los datos del formulario serán enviados a: https://labsys.frc.utn.edu.ar/dds-express/eco quien recibe los datos del formulario y devuelve una tabla HTML con los campos/valores enviados.

> Nota: Todos los campos serán obligatorios y los tipos de entrada deben ser los más adecuados según el valor del campo del formulario.

Ver código del ejemplo aquí (Se sugiere instalar la extensión **Live Server** de Visual Studio Code)

Bibliografía o Referencia

- Aprende sobre desarrollo web - Conceptos básicos de HTML
- Ceballos Fco. J. (2006). *Interfaces gráficas y aplicaciones para Internet* (2° Edición). Editorial RA-MA.

¿Qué es CSS? (*)

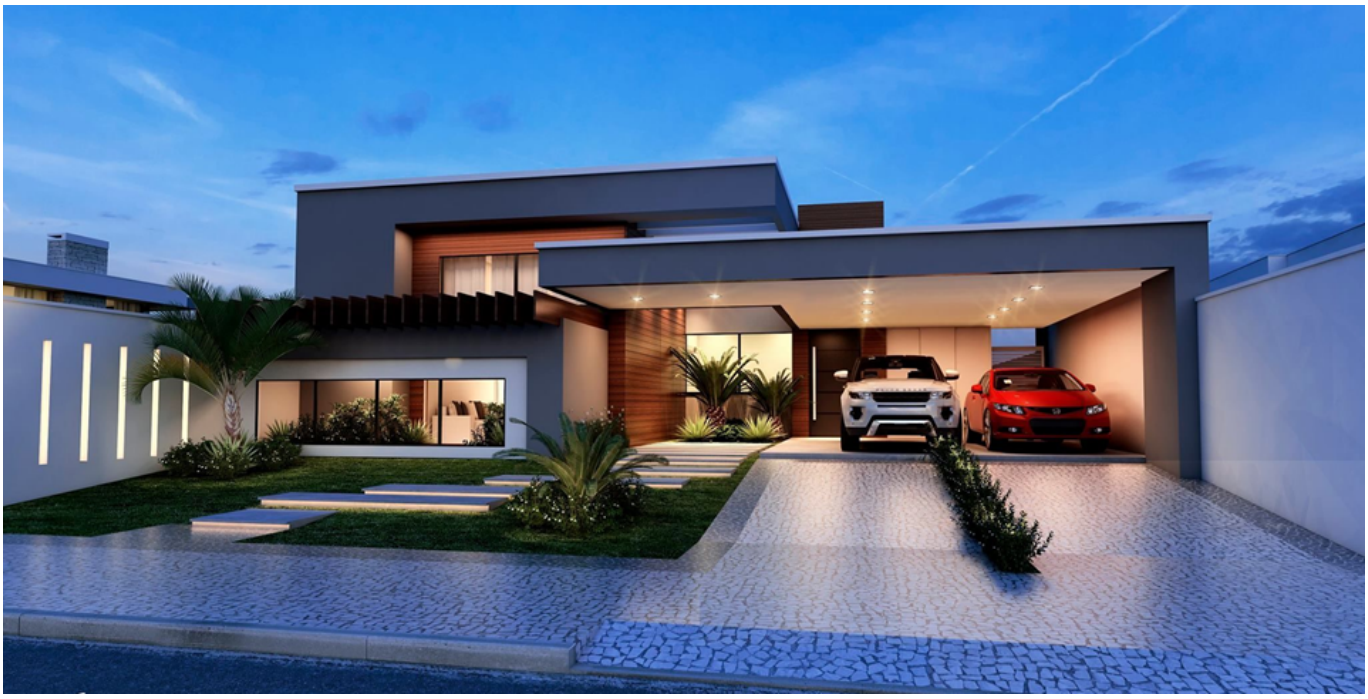
CSS es lo que se denomina lenguaje de hojas de estilo en cascada y se usa para **estilizar elementos** escritos en un lenguaje de marcado como HTML.

CSS separa el contenido de la representación visual del sitio. La diferencia entre un sitio web que implementa CSS y uno que no, es enorme y definitivamente se nota. Antes de CSS, todo el estilo debía incluirse en el marcado HTML. Esto significa que había que describir por separado todos los fondos, los colores de fuente, las alineaciones, etc.

CSS permite estilizar todo en un archivo diferente, creando el estilo allí y después integrando el archivo CSS sobre el marcado HTML. Esto hace que el marcado HTML sea mucho más limpio y fácil de mantener.En

resumen, con CSS no tienes que describir repetidamente cómo se ven los elementos individuales. Esto ahorra tiempo, hace el código más corto y menos propenso a errores.

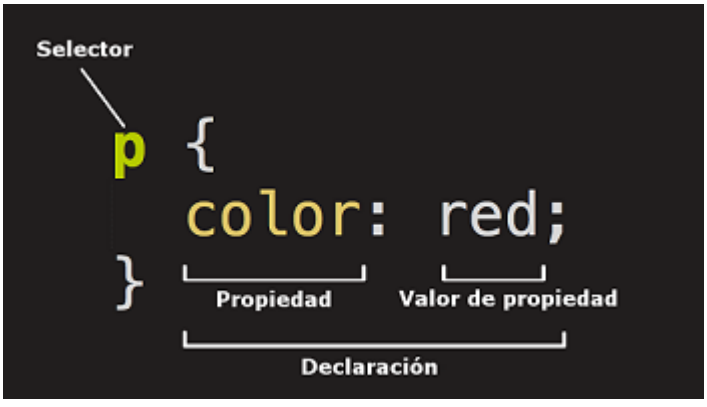
CSS permite tener múltiples estilos en una página HTML, y esto hace que las posibilidades de personalización sean casi infinitas.



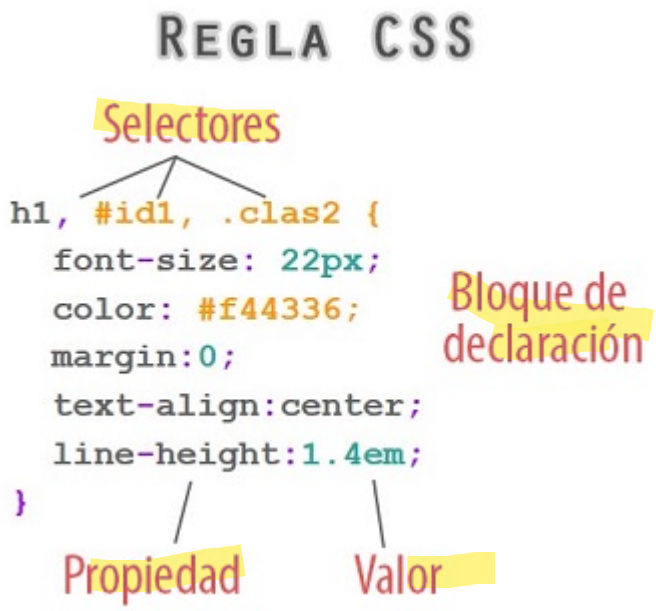
Continuando con la análogía utilizada con HTML, la imagen anterior muestra el resultado final del proyecto de casa, ya con las terminaciones correspondientes (estilos). Para lograr estos detalles finales sobre el HTML, CSS utiliza de manera selectiva a sus elementos mediante **reglas**.

Sintaxis de una regla CSS (*)

> Las reglas de estilo siempre se aplican sobre **selectores**, que son los responsables de definir sobre qué elemento se aplicarán. La siguientes imágenes muestran cómo se escribe una regla CSS.



> o también:



Tipos de Selectores

Existen distintos tipos de selectores. Se pueden mencionar los siguientes:

- (*) De *elemento*: elementos de tipo especificados

> "p{color:white;}"

- (*) De *identificación*:elemento en la página con el ID especificado

> "#lista-ordenada{font-size:small;}"

- (*) De *clase*: elementos en la página con la clase especificada

> ".clase-titulo{text-align:justify;}"

- De *atributo*: elementos en una página con el atributo especificado.

> "a[ref]{text-decoration:none;}"

- De *pseudo-clase*: elementos especificados, pero solo cuando esté en el estado especificado

> "a:visited{color:red;}"

> Otro tipo de selector comúnmente utilizado es el **selector descendiente**. Los selectores descendientes tienen más prioridad sobre otros según el elemento desde el se parte (según su ascendente). En el ejemplo mostrado se aplican los estilos a cualquier enlace **a** que esté contenido dentro de un **div**, por lo que no se indica si es un div class="hijo", class="padre" o id="abuelo". Como existen tres div por encima del enlace, se podría crear un selector "div div div a" que aplicaría a todos los enlaces que tengan tres div ascendentes, y estos estilos tendrían más prioridad que los del selector "div a" del ejemplo anterior. Regla de ejemplo:

> **div a{background-color:#009cde;color:#ffffff;}**

> **div div div a{background-color:red;color:yellow;}**

> Cabe mencionar que también es posible combinar elementos hermanos mediante el operador '~' o hermanos adyancentes mediante '+'; o hijos directos mediante A '>' B.

¿Dónde escribir reglas CSS? (*)

Podemos escribir reglas CSS en:

1. Dentro de un archivo externo con extensión .css:

- Crear un nuevo archivo en una carpeta styles (por convención) y copiar las líneas de código.
- Luego abrir el archivo html pegar la siguiente línea dentro del :

```
<link  href="styles/style.css"  rel="stylesheet"  type="text/css">
```

- Guardar el archivo .html y recargar la página para ver los estilos aplicados

2. Dentro de un bloque "$<$style$>$" agregar atributo style:

- Dentro del bloque "$<$head$>$" agrega la etiqueta "$<$style$>$$<$/style$>$". Por ejemplo:

```
<head>
    <style>
        p: red;
    </style>
</head>
```

Con lo que todo texto encerrado en una etiqueta "$<$p$>$" se mostrará de color rojo.

3. En línea:

- Directamente sobre un elemento HTML agregar el atributo style. Por ejemplo:

```
<p  style="color: blue; text-align: center;">
```

Esto causaría un efecto similar al ejemplo anterior, pero solo para el texto del párrafo correspondiente, dejándolo de color azul.

Variables

Los sitios web complejos tienen una gran cantidad de CSS, a menudo con muchos valores repetidos. Por ejemplo, el mismo color puede usarse en cientos de lugares diferentes, lo que requiere una búsqueda global y reemplazo si ese color necesita cambiar. Las **propiedades personalizadas** o simplemente **variables CSS** permiten que un valor se almacene en un lugar y luego se haga referencia en muchos otros lugares. Un

beneficio adicional son los identificadores semánticos. Por ejemplo, "--main-text-color" es más fácil de entender que #00ff00, especialmente si este mismo color también se usa en otros contextos.

Las propiedades personalizadas están sujetas a la cascada y heredan su valor de su padre.

¿Cómo crear variables CSS?

- Para crear una variable: "--nombre-variable: unValor"
- Para usar una variable declarada en un ámbito: "var(--nombre-variable)"
- También podemos definir un valor de respaldo, por ejemplo:

```
html{
--color-defecto: #000
}
.mi-clase{
border-right: 1px  solid  var(--defecto, black);
}
```

Bibliografía o Referencia

- MDN References - CSS