

HTML

HTML, sigla de *HyperText Markup Language* (Lenguaje de Marcado de Hipertexto), constituye el lenguaje estándar utilizado para la creación y estructuración de páginas web. Su función principal es describir la estructura del contenido que se muestra en un navegador. A través de una serie de elementos, HTML permite al navegador interpretar y representar visualmente los distintos componentes de un documento web, como encabezados, párrafos, enlaces, listas, imágenes, entre otros. Cada uno de estos elementos actúa como una etiqueta que indica qué tipo de contenido se está presentando y cómo debe mostrarse.

Para que un documento HTML sea reconocido adecuadamente por los navegadores, debe comenzar con la declaración `<!DOCTYPE html>`, que indica que se trata de un documento compatible con HTML5, la versión actual recomendada por los organismos de estandarización. A partir de esta declaración, se establece una estructura jerárquica compuesta por elementos anidados que definen las distintas secciones del documento:

```
<!DOCTYPE html>

<html>

<head>

  <title>Título de la página</title>

</head>

<body>

  <h1>Mi primer encabezado</h1>

  <p>Mi primer párrafo.</p>

</body>

</html>
```

En el ejemplo anterior, el elemento `<html>` representa la raíz del documento, mientras que `<head>` contiene información meta que no se muestra directamente en el contenido visible de la página, como el título, definido por `<title>`. Por su parte, el elemento `<body>` alberga todo el contenido que será visible para el usuario en el navegador, incluyendo encabezados como `<h1>` y párrafos como `<p>`.

Comprender la estructura de estos elementos es esencial para poder construir documentos HTML funcionales. Cada uno se compone de una etiqueta de apertura, un contenido y una etiqueta de cierre, siguiendo la estructura general: `<etiqueta> contenido </etiqueta>`. Por ejemplo:

```
<h1>Mi primer encabezado</h1>

<p>Mi primer párrafo.</p>
```

Aquí, `<h1>` y `<p>` son etiquetas que delimitan el contenido a representar como un encabezado de nivel 1 y un párrafo, respectivamente. Cabe señalar que algunos elementos, como `
`, son vacíos y no requieren una etiqueta de cierre, ya que simplemente introducen un salto de línea.

Una vez estructurado el contenido, los navegadores web como Google Chrome, Mozilla Firefox, Microsoft Edge o Safari son los encargados de interpretar los documentos HTML. Aunque estos navegadores no muestran directamente las etiquetas HTML al usuario, sí las utilizan para determinar cómo debe renderizarse cada parte del contenido. Así, un texto encerrado entre etiquetas `<h1>` se mostrará con una tipografía grande y destacada, mientras que un bloque definido con `<p>` se representará como un párrafo convencional.

La disposición interna de una página HTML sigue un patrón lógico que puede visualizarse con claridad en el siguiente ejemplo:

```
<html>

<head>

  <title>Título de la página</title>

</head>

<body>

  <h1>Este es un encabezado</h1>

  <p>Este es un párrafo.</p>

  <p>Este es otro párrafo.</p>

</body>

</html>
```

En este esquema, resulta fundamental destacar que el contenido dentro de la sección `<body>` será el que se muestre en la página web, mientras que el texto colocado en `<title>` se visualizará en la barra de título del navegador o en la pestaña correspondiente.

Conocer la evolución del lenguaje HTML permite comprender cómo ha llegado a convertirse en el estándar actual para el desarrollo web. Desde su creación, HTML ha atravesado diversas etapas y actualizaciones significativas que han ampliado sus capacidades:

- **1989:** Tim Berners-Lee concibe la World Wide Web.
- **1991:** El mismo autor propone la primera versión de HTML.
- **1993:** Dave Raggett desarrolla HTML+, una propuesta de mejora.
- **1995:** Se publica la especificación oficial de HTML 2.0.
- **1997:** El W3C lanza HTML 3.2 como recomendación oficial.
- **1999:** Se publica HTML 4.01, ampliamente adoptado.
- **2000:** XHTML 1.0 intenta integrar HTML y XML.
- **2008:** WHATWG publica el primer borrador de HTML5.
- **2012:** HTML5 se consolida como un estándar “vivo”.
- **2014–2017:** Se publican sucesivas actualizaciones: HTML5, HTML 5.1 y HTML5.2, consolidando nuevas funcionalidades como elementos semánticos, formularios enriquecidos y soporte multimedia.

Este recorrido histórico demuestra no solo la importancia del lenguaje HTML en el desarrollo de la web moderna, sino también su constante adaptación a las necesidades tecnológicas emergentes.

Para comenzar a aprender HTML no es necesario contar con un editor profesional: un editor de texto simple es más que suficiente. De hecho, recomendamos utilizar herramientas básicas como Notepad en PC o TextEdit en Mac, ya que permiten concentrarse en la estructura y el funcionamiento del código sin distracciones. Esta aproximación directa favorece el aprendizaje de los fundamentos de HTML.

Como primer ejercicio, podés escribir un fragmento en tu editor de texto

Una vez que tengas listo el código, llega el momento de guardarlo correctamente en tu computadora. En Notepad, hacé clic en “Archivo” y luego en “Guardar como”. Elegí un nombre como index.htm y asegurate de seleccionar la

codificación UTF-8, que es la recomendada para archivos HTML. Cabe destacar que podés usar tanto la extensión .htm como .html, ya que no hay diferencias funcionales entre ellas; la elección depende de tus preferencias personales.

Para visualizar tu página web, simplemente abrí el archivo que acabás de guardar con tu navegador favorito. Podés hacerlo haciendo doble clic sobre el archivo o bien haciendo clic derecho y seleccionando la opción “Abrir con”, eligiendo el navegador que prefieras. El resultado será una página que mostrará un encabezado y un párrafo, exactamente como los definiste en el código.

Estructura Básica de un Archivo HTML

Todo documento HTML debe empezar con una declaración del tipo de documento, conocida como `<!DOCTYPE html>`. Esta línea indica al navegador qué versión de HTML se está utilizando, permitiendo que la página se visualice correctamente. Es fundamental colocarla solo una vez, al principio del documento, y no es sensible a mayúsculas o minúsculas. En el caso de HTML5, la declaración se ve así: `<!DOCTYPE html>`.

El documento HTML como tal comienza con la etiqueta `<html>` y finaliza con `</html>`, englobando todo el contenido que forma parte del archivo. Dentro de él, la parte visible para el usuario se encuentra delimitada entre las etiquetas `<body>` y `</body>`. Por ejemplo:

```
<!DOCTYPE html>

<html>

<body>


<h1>My First Heading</h1>

<p>My first paragraph.</p>


</body>

</html>
```

En este ejemplo, se puede ver claramente cómo se estructura un documento HTML simple, incluyendo un encabezado y un párrafo. Justamente, los encabezados se definen utilizando las etiquetas que van desde `<h1>` hasta `<h6>`, donde `<h1>` representa el nivel más importante y `<h6>` el menos destacado. Así, se puede crear una jerarquía visual del contenido mediante:

```
<h1>Esto Es Encabezado 1</h1>

<h2>Esto Es Encabezado 2</h2>

<h3>Esto Es Encabezado 3</h3>
```

Por otra parte, los párrafos se definen mediante la etiqueta `<p>`, que permite agrupar bloques de texto de manera ordenada. Por ejemplo:

```
<p>This is a paragraph.</p>

<p>This is another paragraph.</p>
```

También es posible incorporar enlaces a otras páginas mediante la etiqueta `<a>`. Para que el vínculo funcione, se debe especificar su destino a través del atributo `href`, como se muestra a continuación:

```
<a href="https://www.w3schools.com">This is a link</a>
```

Este es un buen ejemplo del uso de atributos en HTML, que son fragmentos de información adicional que se colocan dentro de una etiqueta para modificar su comportamiento o aspecto. Aunque este tema se abordará más a fondo más adelante, ya podemos ver cómo funcionan en la práctica.

Otro tipo de contenido muy común en las páginas web son las imágenes. Estas se insertan con la etiqueta ``, que requiere varios atributos esenciales como `src` (la ruta al archivo), `alt` (texto alternativo por si no se carga la imagen), y medidas como `width` y `height`. Por ejemplo:

```

```

Ahora bien, si alguna vez te preguntaste cómo fue creada una página que estás viendo, existe una manera sencilla de averiguarlo. Podés visualizar el código fuente HTML de cualquier sitio web presionando `CTRL + U` en la mayoría de los navegadores o haciendo clic derecho y seleccionando “Ver código fuente de la página”. Esto abrirá una nueva pestaña con el HTML completo de esa página. Para explorar elementos individuales, también podés hacer clic derecho sobre una parte específica del sitio y elegir “Inspeccionar”. Esto abrirá una herramienta de desarrollador donde podrás ver el HTML y el CSS, e incluso modificarlo momentáneamente para ver cómo afectan los cambios en tiempo real.

En HTML, **un elemento** está compuesto por una etiqueta de apertura, contenido y una etiqueta de cierre. Por ejemplo, `<p>Hola</p>` es un elemento donde `<p>` es la etiqueta de apertura, “Hola” es el contenido y `</p>` la etiqueta de cierre. El conjunto completo desde la apertura hasta el cierre se considera un solo elemento HTML.

En estos casos, `<h1>` y `<p>` son etiquetas que encierran contenido: un encabezado y un párrafo, respectivamente. Algunas etiquetas, sin embargo, no contienen ningún contenido ni requieren una etiqueta de cierre. Un caso típico es `
`, que se utiliza para insertar un salto de línea. A estos se los conoce como elementos vacíos, y no deben cerrarse con una etiqueta como `</br>`.

HTML permite además anidar elementos, lo que significa que un elemento puede estar contenido dentro de otro. Esta característica es esencial para estructurar correctamente un documento HTML. Por ejemplo, en el siguiente fragmento:

```
<!DOCTYPE html>

<html>

<body>

<h1>Mi primer encabezado</h1>

<p>Mi primer párrafo.</p>

</body>

</html>
```

Se puede observar cómo el documento está formado por una jerarquía de elementos: `<html>` es el elemento raíz, que contiene el `<body>`, el cual a su vez contiene un `<h1>` y un `<p>`. Cada elemento comienza con una etiqueta de apertura, contiene su contenido o elementos hijos, y finaliza con su respectiva etiqueta de cierre. Esta estructura de anidamiento es la base del diseño de cualquier página web.

Aunque algunos navegadores modernos pueden interpretar correctamente ciertos elementos sin la etiqueta de cierre. La ausencia de etiquetas de cierre puede causar errores inesperados o comportamientos inusuales, especialmente cuando el contenido se vuelve más complejo o cuando se utilizan hojas de estilo y scripts. Para mantener un código limpio y compatible con los estándares, siempre se deben cerrar todos los elementos que lo requieran.

Los elementos vacíos, como se mencionó antes, son una excepción. El más común es `
`, utilizado para insertar un salto de línea, como en el siguiente ejemplo:

`<p>This is a
 paragraph with a line break.</p>`

Otra característica importante es que HTML no distingue entre mayúsculas y minúsculas en sus etiquetas. Por lo tanto, `<P>` y `<p>` son equivalentes. Sin embargo, aunque no sea obligatorio, la recomendación oficial del W3C es utilizar siempre letras minúsculas en las etiquetas HTML. Esta práctica favorece la legibilidad y asegura compatibilidad con versiones más estrictas del lenguaje, como XHTML, donde las etiquetas en mayúsculas causarían errores.

Los **atributos HTML** proporcionan información adicional sobre los elementos. Todos los elementos HTML pueden tener atributos, y estos siempre se colocan dentro de la etiqueta de apertura. Se escriben en pares de nombre y valor, como por ejemplo `name="value"`. La estructura general sería:

Uno de los atributos más utilizados es `href`, que se emplea con la etiqueta `<a>` para definir el destino de un enlace:

```
<a href="https://www.w3schools.com">Visit W3Schools</a>
```

Aquí, el valor de `href` indica la dirección web a la que el enlace llevará al usuario.

Otro atributo fundamental es `src`, utilizado con la etiqueta `` para indicar la ubicación de una imagen. Por ejemplo:

```

```

El atributo `src` puede contener una URL absoluta, que enlaza a una imagen alojada en otro sitio, como en `src="https://www.w3schools.com/images/img_girl.jpg"`, o una URL relativa, que enlaza a una imagen dentro del mismo sitio web, como `src="img_girl.jpg"` o `src="/images/img_girl.jpg"`. En general, se recomienda usar rutas relativas porque permiten mover el sitio entre dominios sin romper los enlaces.

Para controlar el tamaño de las imágenes, se pueden usar los atributos `width` y `height`, que definen las dimensiones en píxeles:

```

```

Otro atributo esencial para las imágenes es `alt`, que define un texto alternativo en caso de que la imagen no pueda cargarse o para mejorar la accesibilidad con lectores de pantalla:

```

```

Si la ruta de la imagen es incorrecta o el archivo no existe, el navegador mostrará el texto alternativo definido en `alt`:

El atributo `style` permite aplicar estilos CSS directamente en un elemento. Por ejemplo, se puede cambiar el color del texto de un párrafo con:

```
<p style="color:red;">This is a red paragraph.</p>
```

Otro atributo importante es `lang`, que debe incluirse dentro de la etiqueta `<html>` para declarar el idioma principal de la página. Esto es útil para los motores de búsqueda y para lectores de pantalla:

```
<!DOCTYPE html>
```

```
<html lang="en">
```

```
<body>
```

```
...
```

```
</body>
```

```
</html>
```

Incluso se puede indicar el país junto con el idioma, como en:

```
<html lang="en-US">
```

El atributo title permite proporcionar información adicional que aparece como un pequeño cartel (tooltip) al pasar el mouse sobre el elemento:

```
<p title="I'm a tooltip">This is a paragraph.</p>
```

Aunque HTML no exige el uso de minúsculas en los nombres de atributos ni el uso de comillas alrededor de los valores, es altamente recomendable seguir estas buenas prácticas para evitar errores y asegurar compatibilidad con estándares más estrictos como XHTML. Por ejemplo:

Correcto (con comillas):

```
<a href="https://www.w3schools.com/html/">Visit our HTML tutorial</a>
```

Incorrecto (sin comillas):

```
<a href=https://www.w3schools.com/html/>Visit our HTML tutorial</a>
```

En casos donde el valor del atributo contiene espacios, las comillas son imprescindibles:

```
<p title="Description of W3Schools">
```

Tanto las comillas dobles como las simples son válidas. Podés elegir una u otra según el contenido:

```
<p title='John "ShotGun" Nelson'>
```

```
<p title="John 'ShotGun' Nelson">
```

El uso correcto de los atributos en HTML no solo mejora la claridad del código, sino que también optimiza la experiencia del usuario y la accesibilidad del sitio.

En HTML, los **encabezados** se utilizan para definir títulos o subtítulos que estructuran el contenido de una página web. Son esenciales tanto para la organización visual como para la jerarquía del contenido, ya que indican la importancia de cada sección y ayudan a los motores de búsqueda a entender la estructura del sitio.

HTML ofrece seis niveles de encabezados, desde <h1> hasta <h6>, siendo <h1> el más importante y generalmente el más grande en tamaño, y <h6> el menos importante y más pequeño:

```
<h1>Esto es encabezado 1</h1>
```

```
<h2> Esto es encabezado 2</h2>
```

```
<h3> Esto es encabezado 3</h3>
```

```
<h4> Esto es encabezado 4</h4>
```

```
<h5> Esto es encabezado 5</h5>
```

```
<h6> Esto es encabezado 6</h6>
```

Visualmente, estos encabezados se diferencian principalmente por su tamaño de fuente predeterminado, aunque también se pueden personalizar con CSS. Por ejemplo:

- <h1>: Título principal de la página.
- <h2>: Subtítulo importante, por debajo del h1.
- <h3>: Sub-sección del h2.
- Y así sucesivamente, disminuyendo jerarquía y tamaño.

Este orden jerárquico es importante tanto para la legibilidad del usuario como para la accesibilidad y el SEO (Search Engine Optimization). Los motores de búsqueda usan esta jerarquía para entender la estructura del contenido y determinar la relevancia de cada sección.

Es una buena práctica usar solo un <h1> por página, reservado para el título principal, y luego utilizar los demás niveles para estructurar subsecciones. Por ejemplo:

```
<h1>Bienvenidos a mi sitio web</h1>

<h2>Sobre mí</h2>

<p>Hola, soy desarrollador web...</p>

<h2>Proyectos</h2>

<h3>Juego de Space Invaders</h3>

<h3>Editor de imágenes con HTML y CSS</h3>
```

Además, los encabezados no solo ayudan a organizar visualmente, sino que también sirven como puntos de referencia para tecnologías de asistencia como lectores de pantalla, lo que mejora la accesibilidad para personas con discapacidades visuales.

Aunque por defecto los encabezados tienen diferentes tamaños de fuente, si se desea, estos pueden ser modificados usando el atributo style o una hoja de estilos CSS. Por ejemplo:

```
<h1 style="font-size: 40px; color: navy;">Título Personalizado</h1>
```

Modificación de Texto en HTML

En HTML, se puede aplicar estilo visual a los elementos utilizando el atributo style. Este atributo permite definir características como color, fuente, tamaño, alineación, y más, directamente en el elemento HTML. El valor del atributo style se escribe como un conjunto de pares propiedad:valor separados por punto y coma, siguiendo la sintaxis de CSS:

```
<tagname style="property:value;">
```

Este sistema es fundamental para personalizar el aspecto visual de una página web sin necesidad (al menos por ahora) de escribir un archivo CSS aparte.

```
<p style="color:red;">Texto Rojo</p>

<p style="color:blue;">Texto Azul</p>

<p style="font-size:30px;">Texto Grande</p>
```

La propiedad background-color define el color de fondo de un elemento HTML. Puede usarse en el <body> para cambiar el fondo de toda la página o en elementos individuales como encabezados, párrafos, divs, etc.

```
<body style="background-color:powderblue;">

  <h1>Esto es un encabezado</h1>

  <p>Esto es un parrafo.</p>

</body>
```

También se puede aplicar a elementos individuales:

```
<h1 style="background-color:powderblue;"> Esto es un encabezado </h1>

<p style="background-color:tomato;">Esto es un parrafo.</p>
```

La propiedad font-family define qué tipo de letra se utilizará. Se pueden especificar varias fuentes, separadas por comas, por si alguna no está disponible en el navegador del usuario.

```
<h1 style="font-family:verdana;">Esto es un encabezado</h1>

<p style="font-family:courier;">Esto es un parrafo.</p>
```

La propiedad font-size ajusta el tamaño del texto. Puede usarse con unidades como píxeles (px), porcentajes (%), em, rem, entre otros:

```
<h1 style="font-size:300%;">Esto es un encabezado</h1>
```

```
<p style="font-size:160%;">Esto es un parrafo.</p>
```

La propiedad text-align define cómo se alinea el texto horizontalmente dentro del contenedor: puede ser left, center, right o justify.

```
<h1 style="text-align:center;">Escabezado Centrado</h1>
```

```
<p style="text-align:center;">Parrafo Centrado.</p>
```

Usar el atributo style es útil para aplicar estilos rápidos y específicos a elementos individuales. Sin embargo, no es recomendable para proyectos grandes, ya que dificulta el mantenimiento del código. En esos casos, se prefiere usar una hoja de estilos CSS externa o interna para separar la estructura HTML del diseño visual.

Ejemplo simple con múltiples propiedades:

```
<p style="color:white; background-color:black; font-size:20px; text-align:right;">
```

```
    Texto personalizado
```

```
</p>
```

En HTML existen múltiples elementos pensados para definir texto con un significado especial o para aplicar estilos que van más allá de lo visual, agregando valor semántico al contenido. Por ejemplo, y generan ambos un texto en negrita, pero con propósitos distintos: mientras que simplemente resalta visualmente un texto sin implicar importancia, sí implica que el contenido tiene un valor relevante, lo cual también puede ser interpretado por lectores de pantalla con un énfasis en la entonación.

De manera similar, <i> y aplican cursiva, pero con distintos fines. El elemento <i> representa una parte del texto con un tono alternativo, como una expresión en otro idioma, el nombre de una obra o una idea interna. En cambio, indica que el texto debe ser leído con énfasis, y eso también influye en tecnologías asistivas como los lectores de pantalla, que lo pronunciarán con una entonación diferente.

HTML también ofrece elementos como <mark>, que permite resaltar texto como si estuviera subrayado con un marcador, útil para llamar la atención sobre una palabra dentro de un párrafo. El elemento <small> reduce el tamaño del texto, ideal para aclaraciones o detalles secundarios. Por otro lado, y <ins> son útiles para representar ediciones: tacha un texto eliminado, y <ins> subraya texto agregado, ambos con significado semántico útil en contextos como documentos legales, artículos modificados o sistemas de revisión de contenido.

También están disponibles elementos que permiten alterar la posición vertical del texto. <sub> crea texto en subíndice, es decir, alineado por debajo de la línea normal, lo cual se utiliza comúnmente en fórmulas químicas como H₂O. En cambio, <sup> genera texto en superíndice, desplazado hacia arriba, lo cual es útil para representar potencias (x²) o referencias como [1].

Todos estos elementos no solo ayudan a personalizar el texto visualmente, sino que también aportan estructura semántica al documento, haciendo que los navegadores, motores de búsqueda y tecnologías de asistencia entiendan mejor el propósito y contexto del contenido. Aunque algunos estilos pueden replicarse con CSS, utilizar estas etiquetas permite mantener un código más accesible y semánticamente correcto.

HTML ofrece un conjunto de etiquetas diseñadas para representar citas, referencias a obras creativas, direcciones y abreviaciones, enriqueciendo la semántica de un documento web. Entre ellas se encuentran <blockquote>, <q>, <abbr>, <address>, <cite> y <bdo>, cada una con una función específica.

El elemento <blockquote> se utiliza para representar citas largas, generalmente de otro autor o fuente. Estas citas se colocan en bloque, es decir, como una sección independiente del texto principal. Los navegadores suelen mostrar el contenido con una sangría automática. Además, se recomienda usar el atributo cite para indicar la fuente.


```
<blockquote cite="https://www.wwf.org">
```

```
    For 60 years, WWF has worked to help people and nature thrive...
```

```
</blockquote>
```

Para citas más breves e insertadas directamente en medio de un párrafo, se utiliza la etiqueta `<q>`. El navegador suele colocar automáticamente comillas alrededor del texto citado.

```
<p>La misión de WWF es <q>construir un futuro donde las personas vivan in armonía con la naturaleza.</q></p>
```

La etiqueta `<abbr>` marca abreviaciones o siglas como "HTML", "CSS" o "Dr." y es útil para mejorar la accesibilidad. Se recomienda usar el atributo `title` para que, al pasar el cursor, se muestre el significado completo.

```
<p>The <abbr title="World Health Organization">WHO</abbr> fue fundado en 1948.</p>
```

Otro elemento importante es `<address>`, que define información de contacto del autor o propietario del documento. Puede incluir correos electrónicos, direcciones físicas, teléfonos, etc. El contenido suele mostrarse en cursiva y separado del resto con saltos de línea.

```
<address>
```

```
    Escrito por Huergo.<br>
```

```
    Visitanos en: ejemplo.com<br>
```

```
    Peru 759, CABA<br>
```

```
    Argentina
```

```
</address>
```

Cuando se desea citar el título de una obra artística o intelectual (como un libro, una canción o una pintura), se usa `<cite>`. El texto se muestra habitualmente en cursiva, ayudando a diferenciarlo del resto.

```
<p><cite>The Scream</cite> por Edvard Munch, pintada en 1893.</p>
```

Finalmente, la etiqueta `<bdo>` permite forzar la dirección del texto (de izquierda a derecha o de derecha a izquierda), lo que es útil para idiomas como árabe o hebreo, o para mostrar visualmente algún efecto.

```
<bdo dir="rtl">Texto de derecha a izquierda</bdo>
```

Como Programar Correctamente en HTML

En HTML, los comentarios son elementos que permiten insertar notas dentro del código sin que se muestren en el navegador. Son útiles para dejar recordatorios, explicar partes del código, ocultar contenido temporalmente o incluso ayudar durante el proceso de depuración. Los comentarios no tienen ningún efecto visual en la página web, ya que están destinados exclusivamente al desarrollador.

Para escribir un comentario en HTML se usa la siguiente sintaxis:

```
<!-- Esto es un comentario -->
```

El comentario comienza con `<!--` y termina con `-->`. Es importante notar que el símbolo de exclamación solo aparece al inicio. El contenido escrito entre estas dos etiquetas no será interpretado ni mostrado por el navegador.

Una de las funciones más comunes de los comentarios es dejar anotaciones dentro del código, como recordatorios o aclaraciones.

```
<!-- Aquí empieza la sección principal -->
```

```
<p>Esto es un parrafo.</p>
```

```
<!-- Agregar más información sobre el producto acá -->
```

También se pueden usar comentarios para ocultar temporalmente una parte del contenido. Esto es útil cuando se quiere probar diferentes variantes sin borrar el código original.

```
<p>Este un párrafo visible.</p>
```

```
<!-- <p>Este párrafo esta comentado y no aparecerá en el navegador.</p> -->
```

HTML permite comentar múltiples líneas, no solo una. Todo lo que esté entre <!-- y --> será ignorado, sin importar cuántas líneas ocupe.

```
<!--
```

```
<p>Este contenido esta comentado:</p>
```

```

```

```
-->
```

Además, los comentarios son muy útiles para detectar errores. Al comentar secciones de código de forma progresiva, se puede aislar qué parte está provocando un fallo en el diseño o en la estructura de la página.

Incluso se pueden usar para ocultar solo una pequeña parte del contenido, como dentro de un párrafo:

```
<p>This <!-- hidden word --> is a paragraph.</p>
```