



APRENDE
HTML
DESDE CERO
HASTA AVANZADO

A C A D E M I A X

Contenido

1	Introducción	11
1.1	Bienvenida	11
1.1.1	Libro vivo	11
1.1.2	Alcance	12
1.2	Prerequisitos	12
1.3	¿Cómo sacarle el mayor provecho a este libro?	14
2	Primeros pasos	14
2.1	Visión general	14
2.1.1	¿Qué es y porqué debes aprenderlo?	14
2.1.2	¿En dónde se utiliza?	15
2.1.3	¿Qué trabajos puedes conseguir?	16
2.1.4	¿Cuánto puedes ganar?	17
2.1.5	¿Cuales son las preguntas más comunes?	17
2.2	Historia, evolución, y versiones	18
2.3	Características y ventajas	18
2.4	Diferencias con otros lenguajes de programación	19
2.5	Configuración	20
2.5.1	IDE	20
2.5.2	Entorno	20
2.6	Hola Mundo	21
3	Gramática	22
3.1	Partes	22
3.1.1	Etiquetas	22
3.1.2	Contenido	23
3.1.3	Atributos	24
3.1.4	Elementos	26
3.1.5	Comentarios	27

Contenido

3.2	Sintaxis	27
3.2.1	DOCTYPE	28
3.2.2	Estructura de árbol	29
3.3	Comentarios (una sola línea y multilínea)	30
3.4	Bloques e indentación	31
3.5	Conjunto de caracteres	32
3.6	Sensibilidad de mayúsculas y minúsculas	32
4	Elementos	33
4.1	Documento	33
4.1.1	<html>	34
4.2	Metadatos	35
4.2.1	<head>	36
4.2.2	<title>	38
4.2.3	<base>	39
4.2.4	<link>	40
4.2.5	media	42
4.2.6	type	43
4.2.7	<meta>	44
4.2.8	<style>	45
4.3	Secciones	46
4.3.1	<body>	49
4.3.2	<article>	51
4.3.3	<section>	52
4.3.4	<nav>	55
4.3.5	<aside>	57
4.3.6	<h1-6>	59
4.3.7	<hgroup>	62
4.3.8	<header>	63
4.3.9	<footer>	65
4.3.10	<address>	67

4.4	Agrupaciones	69
4.4.1	<p>	69
4.4.2	<hr>	70
4.4.3	<pre>	71
4.4.4	<blockquote>	73
4.4.5		74
4.4.6		75
4.4.7	<menu>	76
4.4.8		77
4.4.9	<dl>	79
4.4.10	<dt>	80
4.4.11	<dd>	81
4.4.12	<figure>	83
4.4.13	<figcaption>	84
4.4.14	<main>	86
4.4.15	<div>	90
4.5	Texto	91
4.5.1	<a>	92
4.5.2		94
4.5.3		95
4.5.4	<small>	96
4.5.5	<s>	98
4.5.6	<cite>	99
4.5.7	<q>	100
4.5.8	<dfn>	102
4.5.9	<abbr>	103
4.5.10	<ruby>	105
4.5.11	<rt>	106
4.5.12	<rp>	107
4.5.13	<data>	109

Contenido

4.5.14	<time>	110
4.5.15	<code>	112
4.5.16	<var>	114
4.5.17	<samp>	115
4.5.18	<kbd>	116
4.5.19	<sub>	117
4.5.20	<sup>	118
4.5.21	<i>	119
4.5.22		120
4.5.23	<u>	121
4.5.24	<mark>	122
4.5.25	<bdi>	123
4.5.26	<bdo>	124
4.5.27		126
4.5.28	 	127
4.5.29	<wbr>	128
4.6	Ediciones	129
4.6.1	<ins>	129
4.6.2		130
4.7	Incrustaciones	131
4.7.1	<picture>	133
4.7.2	<source>	135
4.7.3		137
4.7.4	<iframe>	139
4.7.5	<embed>	141
4.7.6	<object>	142
4.7.7	<video>	145
4.7.8	<audio>	147
4.7.9	<track>	148
4.7.10	<map>	151

Contenido

4.7.11	<area>	151
4.7.12	Tablas	154
4.7.13	<table>	156
4.7.14	<caption>	158
4.7.15	<colgroup>	160
4.7.16	<col>	162
4.7.17	<tbody>	164
4.7.18	<thead>	166
4.7.19	<tfoot>	168
4.7.20	<tr>	170
4.7.21	<td>	172
4.7.22	<th>	173
4.8	Formularios	175
4.8.1	<form>	176
4.8.2	<label>	178
4.8.3	<input>	179
4.8.4	<button>	182
4.8.5	<select>	183
4.8.6	<datalist>	184
4.8.7	<optgroup>	186
4.8.8	<option>	187
4.8.9	<textarea>	188
4.8.10	<output>	190
4.8.11	<progress>	191
4.8.12	<meter>	192
4.8.13	<fieldset>	193
4.8.14	<legend>	195
4.9	Interactividad	197
4.9.1	<details>	198
4.9.2	<summary>	199

4.9.3	<dialog>	200
4.10	Programación	201
4.10.1	<noscript>	203
4.10.2	<template>	204
4.10.3	<slot>	205
4.10.4	<canvas>	207
4.11	Desuso	208
5	Atributos	209
5.1	Globales	209
5.1.1	accesskey	210
5.1.2	autocapitalize	211
5.1.3	autofocus	212
5.1.4	contenteditable	213
5.1.5	class	214
5.1.6	dir	216
5.1.7	draggable	217
5.1.8	enterkeyhint	219
5.1.9	hidden	220
5.1.10	id	221
5.1.11	inert	222
5.1.12	inputmode	223
5.1.13	is	224
5.1.14	itemid	225
5.1.15	itemprop	226
5.1.16	itemref	227
5.1.17	itemscope	228
5.1.18	itemtype	230
5.1.19	lang	231
5.1.20	nonce	232
5.1.21	Atributo slot	233

5.1.22	spellcheck	235
5.1.23	Atributo style	236
5.1.24	tabindex	237
5.1.25	Atributo title	238
5.1.26	translate	239
5.2	Eventos	240
5.2.1	onauxclick	241
5.2.2	onbeforeinput	242
5.2.3	onblur	243
5.2.4	oncancel	244
5.2.5	oncanplay	244
5.2.6	oncanplaythrough	245
5.2.7	onchange	246
5.2.8	onclick	247
5.2.9	onclose	248
5.2.10	oncontextlost	249
5.2.11	oncontextmenu	250
5.2.12	oncontextrestored	251
5.2.13	oncopy	252
5.2.14	oncuechange	252
5.2.15	oncut	253
5.2.16	ondblclick	254
5.2.17	ondrag	255
5.2.18	ondragend	256
5.2.19	ondragenter	257
5.2.20	ondragleave	258
5.2.21	ondragover	259
5.2.22	ondragstart	260
5.2.23	ondrop	261
5.2.24	ondurationchange	262

5.2.25 onemptied	263
5.2.26 onended	264
5.2.27 onerror	264
5.2.28 onfocus	265
5.2.29 onformdata	266
5.2.30 oninput	267
5.2.31 oninvalid	267
5.2.32 onkeydown	268
5.2.33 onkeypress	269
5.2.34 onkeyup	270
5.2.35 onload	270
5.2.36 onloadeddata	271
5.2.37 onloadedmetadata	272
5.2.38 onloadstart	273
5.2.39 onmousedown	274
5.2.40 onmouseenter	276
5.2.41 onmouseleave	277
5.2.42 onmousemove	278
5.2.43 onmouseout	279
5.2.44 onmouseover	280
5.2.45 onmouseup	281
5.2.46 onpaste	282
5.2.47 onpause	283
5.2.48 onplay	284
5.2.49 onplaying	285
5.2.50 onprogress	286
5.2.51 onratechange	287
5.2.52 onreset	288
5.2.53 onresize	289
5.2.54 onscroll	290

Contenido

5.2.55 onscrollend	292
5.2.56 onsecuritypolicyviolation	293
5.2.57 onseeked	294
5.2.58 onseeking	294
5.2.59 onselect	295
5.2.60 onslotchange	296
5.2.61 onstalled	298
5.2.62 onsubmit	299
6 Siguietes pasos	300
6.1 Herramientas	300
6.2 Recursos	300
6.3 Despedida	301
6.4 Preguntas de entrevista	302

1 Introducción

1.1 Bienvenida

Bienvenid@ a este libro de Academia X en el aprenderás HTML desde cero hasta avanzado.

Hola, mi nombre es Xavier Reyes Ochoa y soy el autor de este libro. He trabajado como consultor en proyectos para Nintendo, Google, entre otros proyectos top-tier, trabajé como líder de un equipo de desarrollo por 3 años, y soy Ingeniero Ex-Amazon. En mis redes me conocen como Programador X y comparto videos semanalmente en YouTube desde diversas locaciones del mundo con el objetivo de guiar y motivar a mis estudiantes mientras trabajo haciendo lo que más me gusta: la programación.

En este libro vas a aprender estos temas:

- Gramática
- Elementos
- Atributos

La motivación de este libro es darte todo el conocimiento técnico que necesitas para elevar la calidad de tus proyectos y al mismo tiempo puedas perseguir metas más grandes, ya sea utilizar esta tecnología para tus pasatiempos creativos, aumentar tus oportunidades laborales, o si tienes el espíritu emprendedor, incluso crear tu propio negocio en línea. Confío en que este libro creado con mi experiencia te dará los recursos que necesitas para que tengas éxito en este campo.

Empecemos!

1.1.1 Libro vivo

Este libro fue planeado, editado, revisado, y publicado manualmente por Xavier Reyes Ochoa con el soporte de Hilde Clark-Snustad. El contenido fue en su mayor

1 INTRODUCCIÓN

parte generado por inteligencia artificial usando ChatGPT (Jan 30 Version) de OpenAI. Puedes ver más detalles en <https://openai.com/>

Este trabajo es lo que llamo un **LIBRO VIVO**, esto quiere decir que será actualizado en el tiempo a medida que existan cambios en la tecnología. La versión actual es 1.0.0 editada el 23 de febrero de 2023. Si tienes correcciones importantes, puedes escribirnos a nuestra sección de contacto en <https://www.academia-x.com>

1.1.2 Alcance

El objetivo de este libro es llenar el vacío que existe sobre esta tecnología en Español siguiendo el siguiente enfoque:

- Se revizan los temas desde un nivel básico hasta avanzado de forma práctica con ejemplos y retos.
- Se evita sobresimplificar estos temas.
- Se evita incluir material de relleno.
- Se evita entrar en detalle en temas que no son prácticos. Si deseas profundizar en algún tema, te dejo varios recursos populares y avanzados en la lección de recursos como el estándar de esta tecnología (que puede ser difícil de leer si recién estás empezando).

1.2 Prerequisitos

Para que tengas éxito vas necesitar entender estos conceptos básicos:

1. Los desarrolladores utilizan un **computador** de escritorio con monitores grandes para realizar un trabajo eficiente. Para practicar tu código, es un requisito tener un computador conectado al internet.
2. Una **página web** está compuesta de código escrito en archivos de HTML, CSS, y JavaScript. Es un requisito que tengas experiencia previa visitando páginas web.

1 INTRODUCCIÓN

3. Los archivos de HTML se abren con una aplicación que es tu **navegador** (los navegadores más populares son Chrome, Safari, Microsoft Edge, y Firefox). Es un requisito que tengas previa experiencia abriendo un navegador y utilizando sus funcionalidades básicas antes.
4. El navegador es una aplicación que corre en tu **sistema operativo** (los más populares son Windows, MacOS, y Linux). Los dispositivos móviles como iPads y teléfonos inteligentes también tienen sistemas operativos (los más populares son Android y iOS). Es un requisito que tengas previa experiencia utilizando funcionalidades básicas de tu sistema operativo.
5. Los sistemas operativos trabajan con **archivos y carpetas**. Es un requisito que tengas previa experiencia creando, cambiando el nombre y la extensión, reorganizando, y borrando archivos y carpetas.
6. Los archivos tienen **extensiones** que indican el tipo de contenido que contienen y la aplicación que los puede abrir. Por ejemplo: la extensión .txt es para archivos de texto simple, la extensión .doc es para archivos que se abren con la aplicación Word, los archivos con extensión .bmp son archivos de imágenes que se abren con la aplicación Paint. Es un requisito que tengas previa experiencia abriendo archivos de varias extensiones en sus aplicaciones.
7. Hay miles de **aplicaciones** que se pueden instalar en tu sistema operativo. A estas aplicaciones se les llama software. Puedes empezar instalando el navegador Chrome si no lo tienes. Es un requisito que tengas previa experiencia descargando e instalando software en tu sistema operativo.
8. El **internet** está compuesto de **servidores** (que simplemente son computadoras sin monitores en muchas partes del mundo) que transmiten información (a través de cables bajo tierra, bajo el mar, y de forma inalámbrica) hasta llegar a tu hogar cuando pides esta información. Eso quiere decir, que cuando visitas una página web lo que haces es simplemente pedir archivos que un desarrollador creó, subió a un servidor, y configuró una dirección. Este es un buen resumen, pero es un requisito que tengas un entendimiento básico de cómo funciona la web.

1.3 ¿Cómo sacarle el mayor provecho a este libro?

Para sacarle el mayor provecho a este libro:

1. No solo leas el libro. La práctica es esencial en este campo. Practica todos los días y no pases de lección hasta que un concepto esté 100% claro.
2. No tienes que memorizarlo todo, solo tienes que saber donde están los temas para buscarlos rápidamente cuando tengas dudas.
3. Cuando tengas preguntas usa [Google](#), [StackOverFlow](#), y [ChatGPT](#)
4. Acepta que en esta carrera, mucho de tu tiempo lo vas utilizar investigando e innovando, no solo escribiendo código.
5. No tienes que aprender inglés ahora pero considera aprenderlo en un futuro porque los recursos más actualizados están en inglés y también te dará mejores oportunidades laborales.
6. Si pierdas la motivación, recuerda tus objetivos. Ninguna carrera es fácil pero ya tienes los recursos para llegar muy lejos. Te deseo lo mejor en este campo!

2 Primeros pasos

2.1 Visión general

2.1.1 ¿Qué es y por qué debes aprenderlo?

HTML (HyperText Markup Language) es un lenguaje de marcado que se utiliza para crear páginas web. Está compuesto por etiquetas que se utilizan para definir la estructura y el contenido de una página web.

Aprender HTML te permite crear y personalizar proyectos web de manera más efectiva y te ofrece una mayor comprensión de cómo funcionan los sitios web y cómo se pueden mejorar. Además, conocer HTML puede ser útil para trabajar en áreas relacionadas con la optimización de motores de búsqueda (SEO).

2 PRIMEROS PASOS

En resumen, aprender HTML es esencial para cualquiera que quiera desarrollar o trabajar con sitios web, y puede abrir muchas oportunidades en el mundo laboral y en el desarrollo personal.

2.1.2 ¿En dónde se utiliza?

HTML es una tecnología fundamental que se utiliza en muchas otras tecnologías y plataformas para crear y mostrar contenido web. Algunos ejemplos de tecnologías que utilizan HTML incluyen:

- **Navegadores web:** Los navegadores web, como Google Chrome, Firefox y Safari, utilizan HTML para interpretar y mostrar el contenido de las páginas web que visitas.
- **Sitios web:** La mayoría de los sitios web utilizan HTML para estructurar y dar formato a su contenido. Esto incluye sitios de noticias, redes sociales, blogs, tiendas en línea y muchos más.
- **Aplicaciones web:** Las aplicaciones web son programas que se ejecutan en un navegador y se utilizan para realizar tareas específicas, como el correo electrónico o la edición de documentos. Muchas aplicaciones web utilizan HTML para estructurar y dar formato a su interfaz y contenido.
- **Correos electrónicos:** Aunque los correos electrónicos suelen ser enviados en formato de texto simple, muchos correos electrónicos modernos también utilizan HTML para dar formato al contenido y hacer que los mensajes se vean más atractivos.
- **Aplicaciones móviles híbridas:** Las aplicaciones móviles híbridas son aplicaciones que se ejecutan en un navegador y se pueden instalar como aplicaciones nativas en un dispositivo móvil. Muchas aplicaciones móviles híbridas utilizan HTML para crear su interfaz y contenido.

Estos son solo algunos ejemplos de las muchas tecnologías que utilizan HTML. Como se puede ver, es una tecnología muy versátil y ampliamente utilizada en muchas

áreas diferentes.

2.1.3 ¿Qué trabajos puedes conseguir?

Aprender HTML puede abrir muchas oportunidades laborales en el mundo del desarrollo web y la tecnología. Algunos trabajos que puedes conseguir al aprender HTML incluyen:

- **Desarrollador web:** Los desarrolladores web utilizan lenguajes de programación como HTML, CSS y JavaScript para crear y mantener sitios web y aplicaciones web.
- **Diseñador web:** Los diseñadores web utilizan HTML y CSS para crear la apariencia y la estructura de un sitio web. También trabajan en la usabilidad y el diseño de la interfaz de usuario.
- **Diseñador gráfico:** Los diseñadores gráficos utilizan HTML y CSS para crear contenido para sitios web, correos electrónicos y otras plataformas en línea.
- **Especialista en SEO:** Los especialistas en SEO utilizan HTML para optimizar los sitios web para los motores de búsqueda. Esto incluye la optimización del contenido y la estructura del sitio, así como la investigación y el uso de palabras clave relevantes.
- **Técnico de soporte:** Los técnicos de soporte a menudo trabajan con HTML y otras tecnologías web para resolver problemas técnicos y brindar asistencia a los usuarios.

Estos son solo algunos ejemplos de los muchos trabajos que puedes conseguir al aprender HTML. Conocer HTML puede ser una habilidad valiosa en muchas áreas diferentes, y puede ayudarte a encontrar trabajo en una amplia variedad de industrias.

Es difícil decir con certeza cuánto puedes ganar al utilizar HTML en tu trabajo, ya que hay muchos factores que pueden afectar tu salario, como tu nivel de experiencia, la

2 PRIMEROS PASOS

industria en la que trabajas, la región en la que vives y la demanda de trabajadores con habilidades en HTML.

En general, se espera que los trabajadores con conocimientos de HTML tengan salarios más altos que los trabajadores que no tienen esta habilidad. Sin embargo, el salario exacto que puedes ganar dependerá de tu rol y tu nivel de experiencia.

2.1.4 ¿Cuánto puedes ganar?

Por ejemplo, según el sitio web de empleo Glassdoor, en Estados Unidos, los salarios promedio para los trabajos relacionados con HTML incluyen:

- Desarrollador web: \$74,000 anuales
- Diseñador web: \$54,000 anuales
- Diseñador gráfico: \$47,000 anuales
- Especialista en SEO: \$50,000 anuales

Estos salarios son solo una guía y pueden variar significativamente dependiendo de la región y la industria en la que trabajes. Además, es importante tener en cuenta que estos salarios son promedios y pueden ser más altos o más bajos en función de tu nivel de experiencia y otras habilidades y conocimientos que tengas.

2.1.5 ¿Cuales son las preguntas más comunes?

- ¿Cómo se crea un enlace en HTML?
- ¿Cómo se inserta una imagen en HTML?
- ¿Cómo se crea una tabla en HTML?
- ¿Cómo se crea un formulario en HTML?
- ¿Cómo se crea una lista en HTML?
- ¿Cómo se crea una cabecera en HTML?
- ¿Cómo se crea una sección en HTML?

- ¿Cómo se crea una tabla de contenido en HTML?
- ¿Cómo se crea una página web en HTML?
- ¿Cómo se crea una plantilla en HTML?

Después de finalizar este libro podrás responder a estas preguntas sin problemas.

2.2 Historia, evolución, y versiones

HTML (HyperText Markup Language) es un lenguaje de marcado de hipertexto que se utiliza para crear páginas web. Fue creado por Tim Berners-Lee en 1990, y es el lenguaje de marcado más utilizado para crear páginas web.

La primera versión de HTML fue HTML 1.0, que fue publicada en 1993. Esta versión incluía etiquetas básicas para la creación de documentos web, como `<html>`, `<head>`, `<title>`, `<body>`, `<h1>`-`<h6>`, `<p>`, `<a>`, `<img>`, etc.

Desde entonces, HTML ha evolucionado a través de varias versiones, cada una con nuevas características y mejoras. La última versión de HTML es HTML 5, que fue publicada en 2014. Esta versión incluye nuevas etiquetas, como `<video>`, `<audio>`, `<canvas>`, `<svg>`, `<header>`, `<footer>`, etc., así como mejoras en la semántica y la accesibilidad.

Además de HTML 5, hay varias versiones anteriores de HTML, como HTML 4.01, XHTML 1.0, XHTML 1.1, etc. Estas versiones incluyen características similares a las de HTML 5, pero no son tan ampliamente utilizadas.

2.3 Características y ventajas

HTML (HyperText Markup Language) es un lenguaje de marcado de hipertexto que se utiliza para crear páginas web. Está compuesto por etiquetas que se utilizan para definir la estructura y el contenido de una página web.

Características:

2 PRIMEROS PASOS

- Es un lenguaje de marcado de hipertexto, lo que significa que se utiliza para crear páginas web.
- Está compuesto por etiquetas que se utilizan para definir la estructura y el contenido de una página web.

Ventajas:

- Es fácil de aprender y usar.
- Es compatible con la mayoría de los navegadores web.
- Es un lenguaje estándar, lo que significa que es compatible con la mayoría de los sistemas operativos.
- Es un lenguaje de alto nivel, lo que significa que es fácil de entender y usar.
- Es un lenguaje de código abierto y gratuito.
- Es un lenguaje de alto rendimiento, lo que significa que es rápido y eficiente.

2.4 Diferencias con otros lenguajes de programación

HTML es un lenguaje de marcado, mientras que otros lenguajes son lenguajes de programación. HTML se utiliza para definir la estructura y el contenido de una página web, mientras que los lenguajes de programación se utilizan para escribir código que realiza una tarea específica. HTML no es un lenguaje de programación, por lo que no se pueden escribir programas con él. Por otro lado, los lenguajes de programación se utilizan para escribir programas que realizan tareas específicas. HTML no tiene ninguna capacidad de lógica, mientras que los lenguajes de programación tienen capacidades de lógica y control. HTML no tiene ninguna capacidad de almacenamiento de datos, mientras que los lenguajes de programación tienen capacidades de almacenamiento de datos.

2.5 Configuración

2.5.1 IDE

Los archivos de HTML son archivos de texto. Puedes editarlos con **editores de texto** como Notepad en Windows o Notes en MacOS pero es recomendado utilizar un **IDE** (Integrated Development Environment) que es una aplicación de edición de código más avanzado que le da colores a tu código para que sea más fácil de leer y tengas funciones de autocompletado, entre otras. Algunos IDEs populares son [Brackets](#), [Atom](#), [Sublime Text](#), [Vim](#), y [Visual Studio Code](#).

El editor recomendado para practicar el código que vamos a ver es Visual Studio Code (o VSCode) que puedes bajar desde <https://code.visualstudio.com/>

2.5.2 Entorno

1. **Instalar un navegador web:** Para ver archivos HTML, necesitarás un navegador web como Google Chrome, Mozilla Firefox, Microsoft Edge, Safari, etc. Puedes descargar uno de estos navegadores desde sus respectivos sitios web:
 - Google Chrome: <https://www.google.com/chrome/>
 - Mozilla Firefox: <https://www.mozilla.org/en-US/firefox/new/>
 - Microsoft Edge: <https://www.microsoft.com/en-us/edge>
 - Safari: <https://support.apple.com/downloads/safari>
2. **Verificar la instalación:** Una vez que hayas instalado un navegador web, puedes verificar la instalación abriendo el navegador y visitando un sitio web. Si el sitio se carga correctamente, entonces la instalación se ha realizado correctamente.
3. **Correr archivos de HTML:** Para correr archivos de HTML, primero necesitas abrir el archivo en tu navegador web. Esto se puede hacer de dos maneras:

2 PRIMEROS PASOS

- Abre el archivo desde el navegador: en la barra de direcciones del navegador, escribe la ruta completa del archivo (por ejemplo, C:\Users\username\Documents\miarchivo.html).
- Abre el archivo desde tu computadora: en tu computadora, busca el archivo HTML que desees abrir. Luego, da doble click en el archivo o haz clic derecho en el archivo y selecciona “Abrir con” y luego selecciona el navegador web que desees usar.

2.6 Hola Mundo

“Hola mundo” significa “Hello world” en inglés y es una expresión utilizada para probar un nuevo lenguaje tecnológico de forma simple.

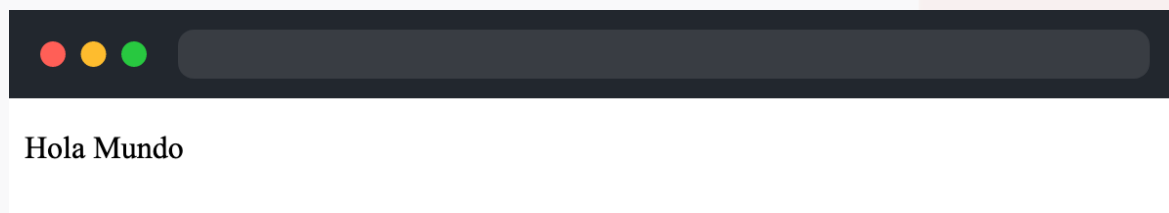
Su objetivo es imprimir la cadena “Hola mundo” para indicar que el programa se ha ejecutado correctamente, ayudando a comprender los conceptos básicos de la sintaxis y estructura de la tecnología en aprendizaje.

Para escribir “Hola mundo” en HTML, primero debes crear un archivo de texto con la extensión .html. Luego, dentro del archivo y usando un editor de texto como VSCode, escribe el siguiente código. Guarda el archivo y ábrelo en un navegador web para ver el resultado “Hola mundo” en la pantalla.



El código se renderiza de esta manera al abrirlo en un navegador.

3 GRAMÁTICA



Felicitaciones! has escrito tu primera línea de HTML. Es el texto “Hola Mundo” rodeado por la etiqueta `<p>` que simboliza párrafo. La etiqueta de apertura `<p>` junto con el contenido “Hola Mundo” y la etiqueta de cierre `</p>` se llaman en conjunto, un elemento.

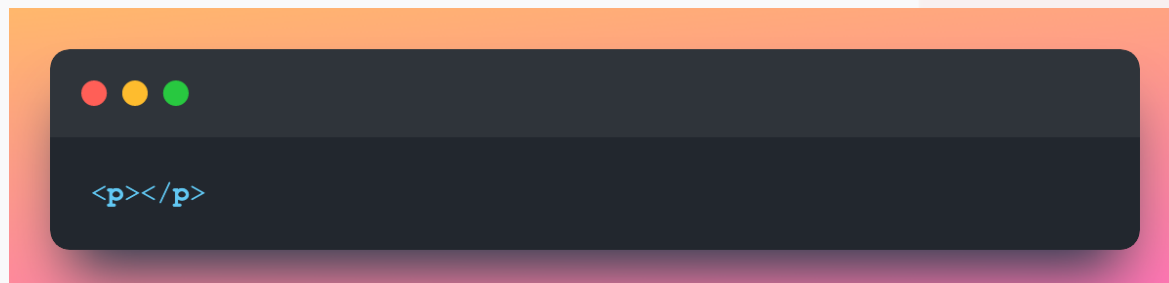
3 Gramática

3.1 Partes

Algunas de las partes fundamentales de HTML incluyen:

3.1.1 Etiquetas

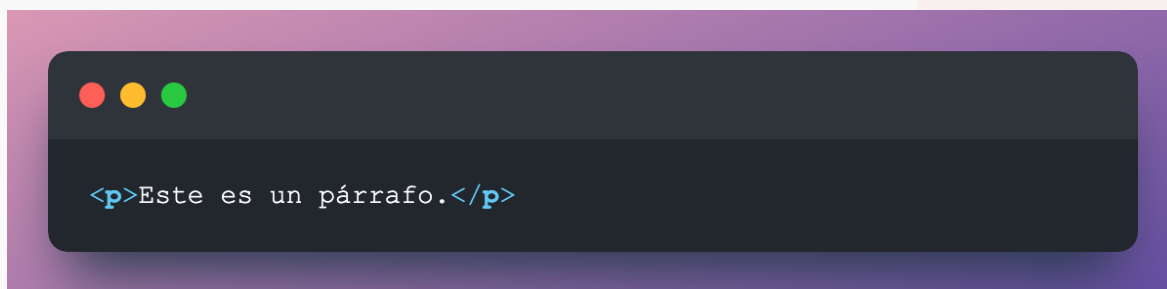
Una etiqueta es una palabra o conjunto de caracteres rodeados por corchetes angulares, como `<html>`, que indica el tipo de contenido que se encuentra dentro de ella. Por ejemplo, la etiqueta `<p>` indica que el contenido dentro de ella es un párrafo de texto.



3 GRAMÁTICA

3.1.2 Contenido

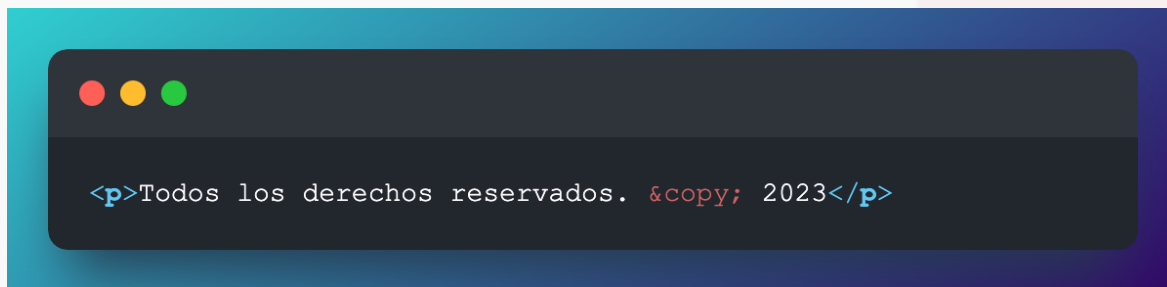
El contenido es el texto o los elementos dentro de una etiqueta. Por ejemplo, el contenido de una etiqueta `<p>` podría ser “Este es un párrafo.”



El código se renderiza de esta manera al abrirlo en un navegador.

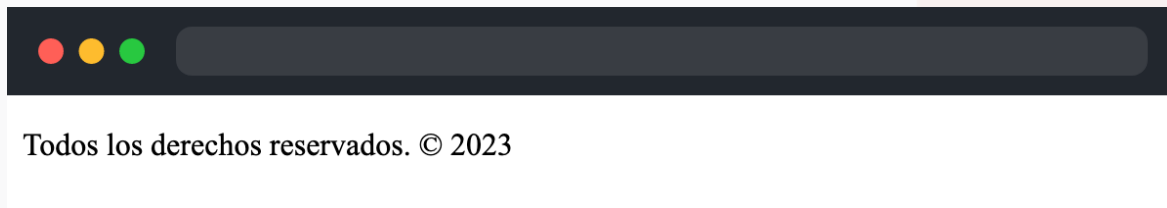


Entidades de caracteres Las entidades de caracteres son un tipo de contenido que se utilizan para mostrar caracteres especiales o caracteres que no se pueden ingresar directamente en el código fuente. Por ejemplo, la entidad de caracteres para renderizar el símbolo de copyright © es `©`; Las entidades de caracteres empiezan con el símbolo de `&` y terminan con el símbolo de `;`



3 GRAMÁTICA

El código se renderiza de esta manera al abrirlo en un navegador.



Aquí puedes ver otros ejemplos de entidades de caracteres y existen muchas más que puedes visitar en el estándar de HTML en <https://html.spec.whatwg.org/multipage/named-characters.html>

HTML	Símbolo
&	&
©	©
ñ	ñ
®	®
@	@

3.1.3 Atributos

Un atributo es un valor adicional que se puede agregar a una etiqueta para proporcionar información adicional sobre el elemento. Por ejemplo, la etiqueta **** puede tener un atributo “src” que indica la URL de la imagen a mostrar. URL son las siglas en inglés de Uniform Resource Locator, que en español significa Localizador Uniforme de Recursos.

3 GRAMÁTICA

```

```

El código se renderiza de esta manera al abrirlo en un navegador.



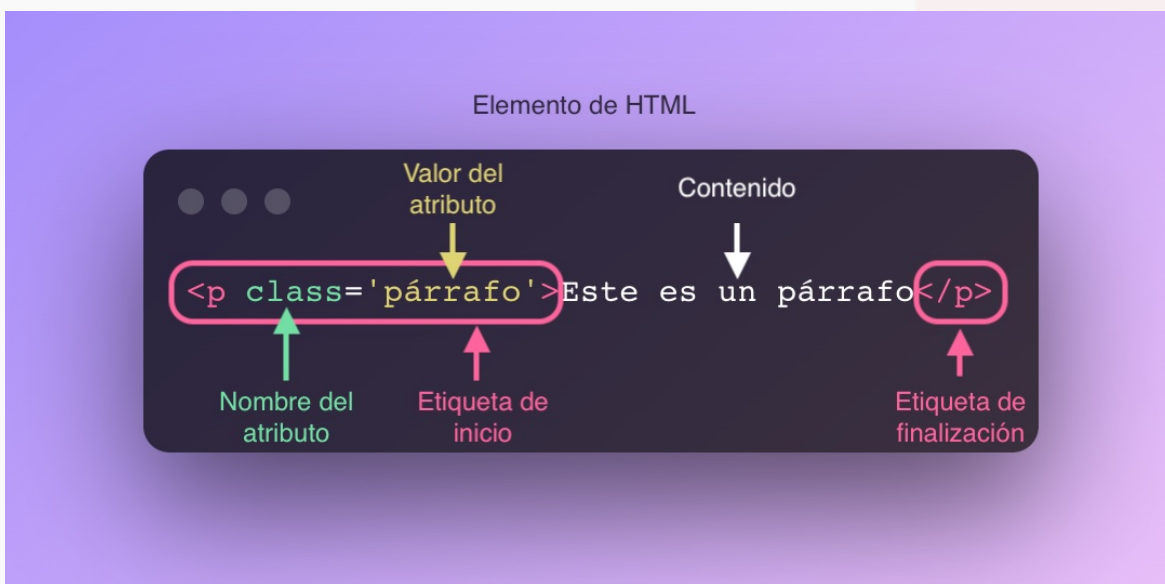
Los valores de los atributos los puedes escribir entre comillas simples (") o comillas dobles ("). Algunos atributos que tienen el valor de **true** no requieren valores, por ejemplo **<video autoplay></video>** en lugar de **<video autoplay="true"></video>**

3 GRAMÁTICA

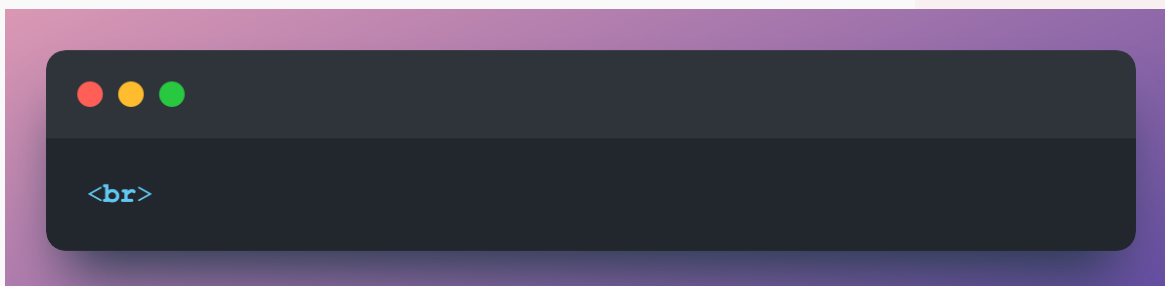
3.1.4 Elementos

Un elemento es un conjunto de una o varias etiquetas que indican una parte específica del contenido. Un elemento puede tener contenido dentro de él, como un párrafo, o puede estar vacío, como una imagen.

Esta imagen muestra las partes de HTML al crear un elemento.



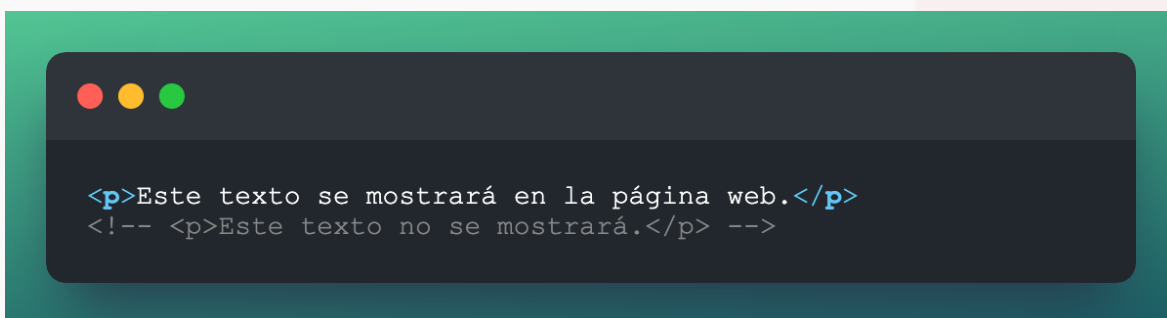
También existen **elementos** que no requieren **etiquetas** de cierre si no tienen contenido de texto. Por ejemplo para imágenes puedes usar `<img />` en lugar de `<img></img>` y también las puedes escribir sin barra oblicua. Por ejemplo este elemento de quiebre de línea: `<br>`



3 GRAMÁTICA

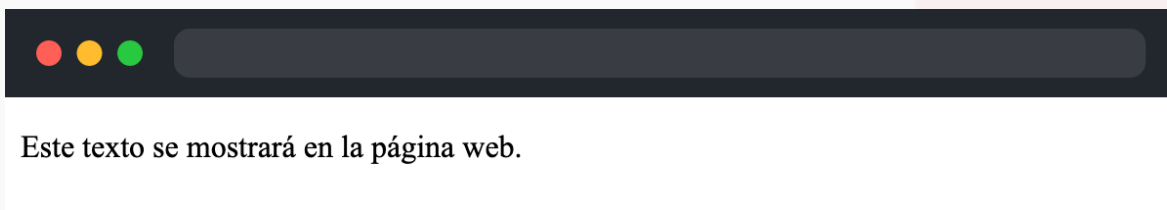
3.1.5 Comentarios

Los comentarios se utilizan para agregar notas o descripciones en el código fuente que no se mostrarán en la página web final. Los comentarios comienzan con la etiqueta `<!--` y terminan con `-->`. Todo lo que esté entre estas etiquetas será considerado un comentario y no se mostrará en la página web. Los comentarios son útiles para proporcionar información sobre el código o para deshabilitar temporalmente una sección del código sin eliminarlo permanentemente.

A screenshot of a code editor window with a dark background and a teal border. The window has three colored window control buttons (red, yellow, green) in the top-left corner. The code inside the editor is:

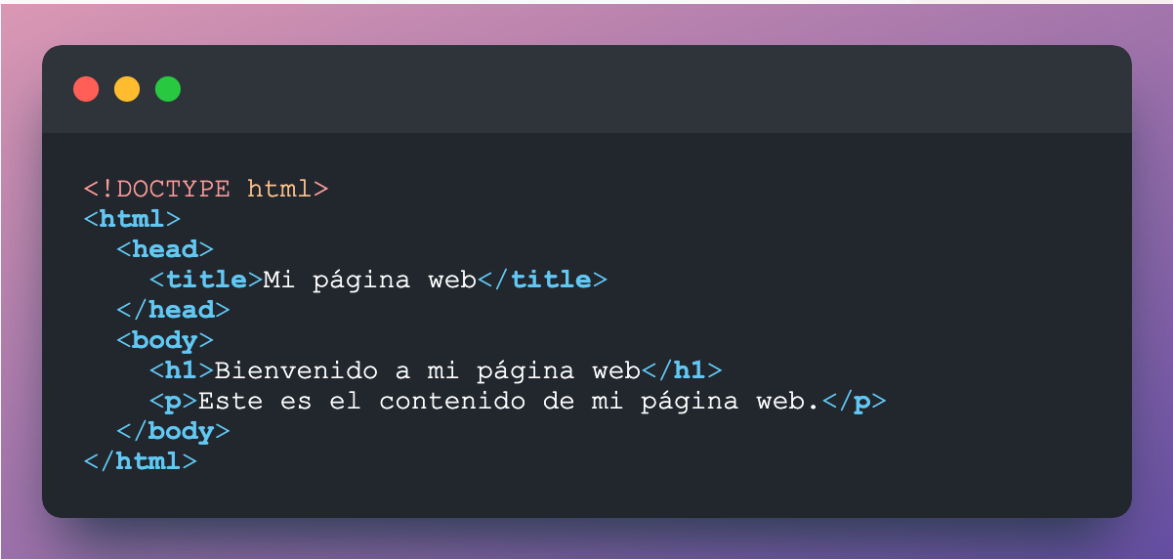
```
<p>Este texto se mostrará en la página web.</p>  
<!-- <p>Este texto no se mostrará.</p> -->
```

El código se renderiza de esta manera al abrirlo en un navegador.



3.2 Sintaxis

Para escribir un archivo de HTML, puedes usar tu IDE y guardar el archivo con la extensión “.html” . A continuación, escribe el código HTML en el archivo. Por ejemplo:

A code editor window with a dark background and three colored window control buttons (red, yellow, green) in the top-left corner. The code is written in a light blue font and is an HTML document structure.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Mi página web</title>
  </head>
  <body>
    <h1>Bienvenido a mi página web</h1>
    <p>Este es el contenido de mi página web.</p>
  </body>
</html>
```

El código se renderiza de esta manera al abrirlo en un navegador.

A web browser window with a dark header bar containing three colored window control buttons (red, yellow, green) and a search bar. The main content area is white and displays the rendered HTML.

Bienvenido a mi página web

Este es el contenido de mi página web.

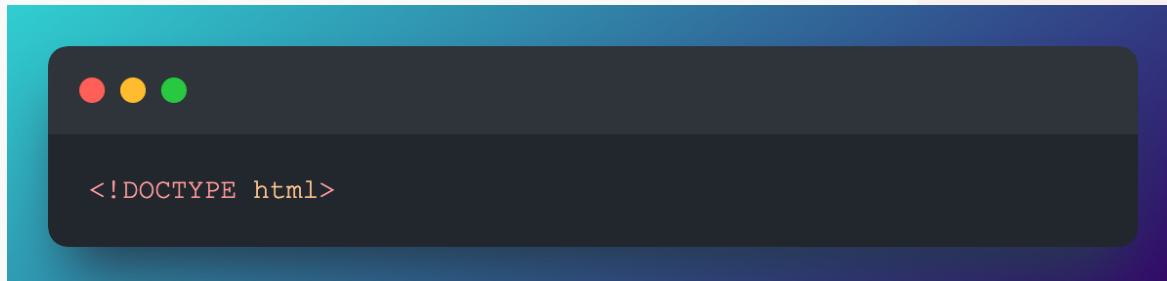
Una vez guardado el archivo, puedes abrirlo en un navegador web (como Chrome, Firefox, Safari, etc.) para ver tu página web.

3.2.1 DOCTYPE

DOCTYPE es una instrucción de tipo de documento que se utiliza en la primera línea de un documento HTML para indicar al navegador web qué versión de HTML se está utilizando. El DOCTYPE es importante porque los navegadores web utilizan esta información para determinar cómo deben mostrar el contenido de una página.

3 GRAMÁTICA

La sintaxis para especificar el tipo de documento en HTML es la siguiente:

A code editor window with a dark background and a blue-to-purple gradient border. It displays the HTML5 DOCTYPE declaration in a light blue monospace font.

```
<!DOCTYPE html>
```

Este es el DOCTYPE para HTML5, la versión más reciente de HTML.

Hay diferentes versiones de HTML, y cada una tiene una sintaxis diferente para el DOCTYPE. Por ejemplo, el DOCTYPE para HTML 4.01 Strict es el siguiente:

A code editor window with a dark background and a purple-to-pink gradient border. It displays the HTML 4.01 Strict DOCTYPE declaration in a light blue monospace font.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
    "http://www.w3.org/TR/html4/strict.dtd">
```

Es importante asegurar que el DOCTYPE sea correcto para que el navegador web pueda mostrar el contenido de la página de manera correcta.

3.2.2 Estructura de árbol

La estructura de árbol de HTML se refiere a la jerarquía de etiquetas que se utilizan para organizar el contenido de una página web. Cada etiqueta HTML representa un nodo en el árbol, y las etiquetas contenidas dentro de otras etiquetas se consideran nodos secundarios o hijos del nodo principal.

La raíz de la estructura de árbol de HTML es la etiqueta **<html>**, que contiene dos nodos principales: **<head>** y **<body>**.

3 GRAMÁTICA

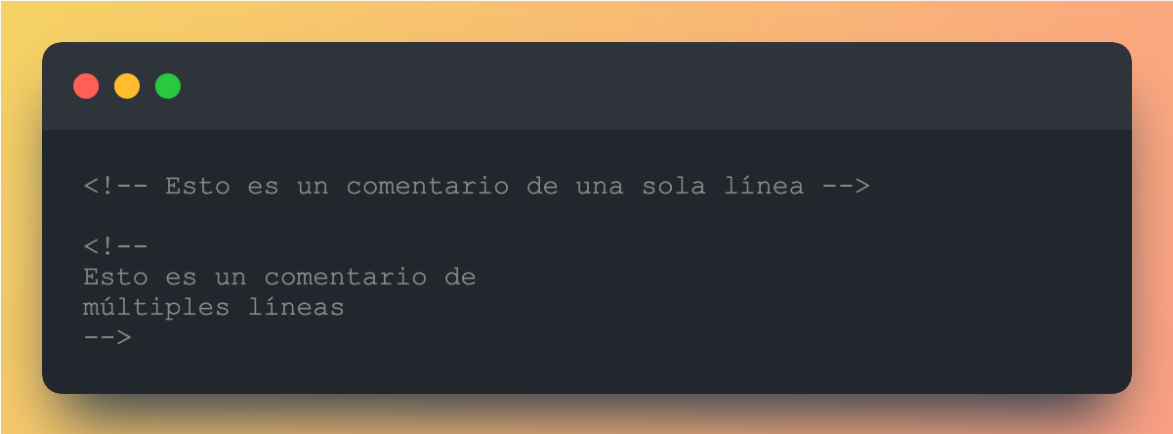
La etiqueta **<head>** contiene información sobre la página web, como el título de la página, metadatos y enlaces a hojas de estilo CSS.

La etiqueta **<body>** contiene el contenido visible de la página web, como encabezados, párrafos, imágenes, enlaces y otros elementos. Dentro del body se puede organizar utilizando diferentes etiquetas como: **<header>**, **<nav>**, **<main>**, **<article>**, **<section>**, **<aside>**, **<footer>** entre otras, dependiendo del contenido y estructura de la página.

3.3 Comentarios (una sola línea y multilínea)

Los comentarios se pueden escribir en una línea o múltiples líneas.

Ejemplo:



```
<!-- Esto es un comentario de una sola línea -->

<!--
Esto es un comentario de
múltiples líneas
-->
```

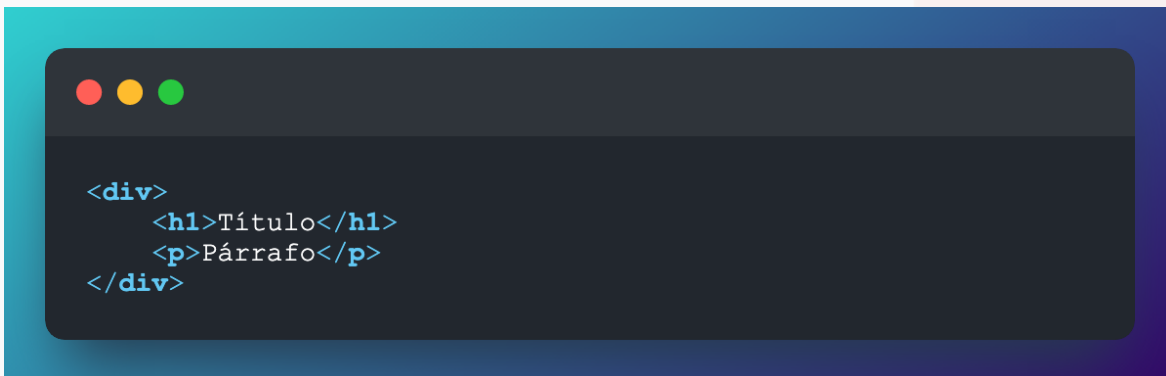
Reto:

Intenta agregar un comentario a tu código HTML para explicar qué hace una etiqueta específica.

3.4 Bloques e indentación

Los bloques y la indentación son conceptos importantes en HTML. Los bloques son elementos HTML que se usan para contener otros elementos HTML. Estos elementos se pueden anidar para crear estructuras jerárquicas. La indentación se usa para mostrar la jerarquía de los elementos. Esto ayuda a los desarrolladores a leer y comprender el código HTML.

Ejemplo:

A code editor window with a dark background and a teal-to-blue gradient border. It contains the following HTML code:

```
<div>
  <h1>Título</h1>
  <p>Párrafo</p>
</div>
```

El código se renderiza de esta manera al abrirlo en un navegador.



Reto:

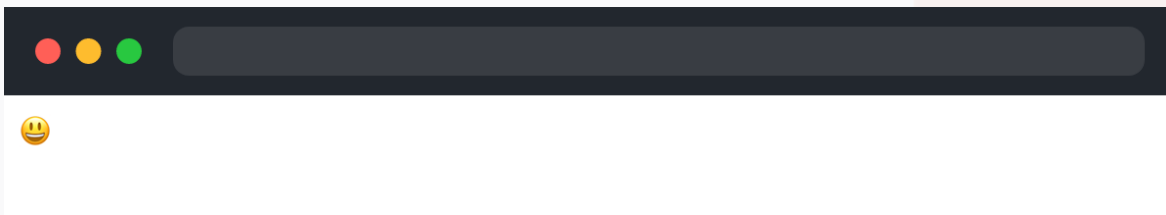
Intenta crear una estructura HTML usando bloques y indentación como el ejemplo.

3.5 Conjunto de caracteres

El conjunto de caracteres usado por HTML es el conjunto de caracteres Unicode. Esto significa que HTML puede mostrar caracteres de cualquier idioma, incluyendo caracteres especiales como símbolos y emojis. Un ejemplo práctico y básico de uso de caracteres Unicode en HTML es el uso de la etiqueta `<span>` para mostrar un emoji:



El código se renderiza de esta manera al abrirlo en un navegador.



Reto:

Usa la etiqueta `<span>` para mostrar una carita sonriente en tu página web.

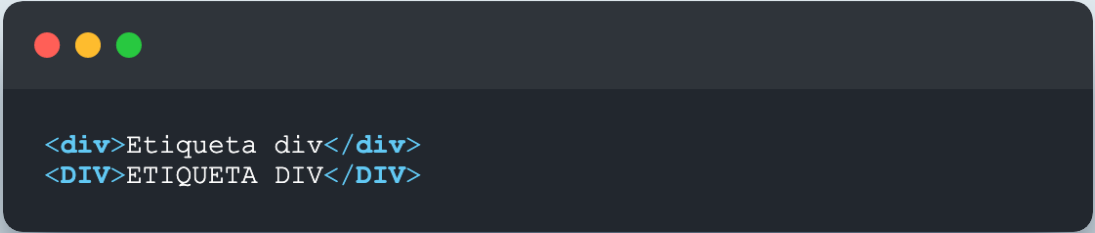
3.6 Sensibilidad de mayúsculas y minúsculas

HTML no es sensible a mayúsculas y minúsculas en etiquetas. Puedes escribir `<span>` o `<SPAN>` y tendrá el mismo efecto pero es una buena práctica escribirlas en minúsculas. El texto o contenido si es sensible. Se recomienda escribir el código como los ejemplos de este libro y el estándar de HTML ya que si se escriben en

4 ELEMENTOS

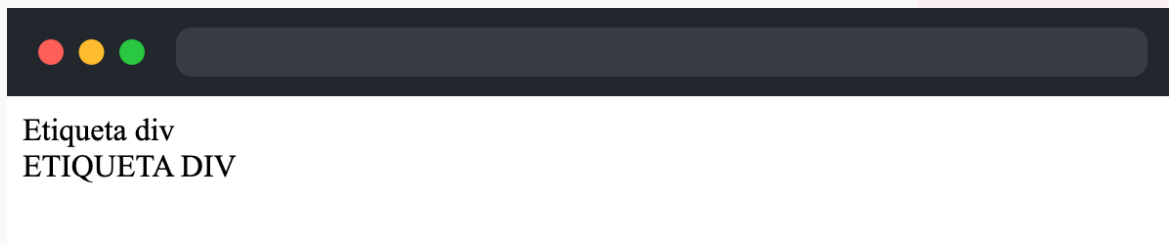
mayúsculas o minúsculas incorrectas, el navegador puede no interpretar el código correctamente.

Ejemplo:



```
<div>Etiqueta div</div>
<DIV>ETIQUETA DIV</DIV>
```

El código se renderiza de esta manera al abrirlo en un navegador.



Reto:

Intenta escribir el mismo código HTML usando mayúsculas en lugar de minúsculas.
¿Qué sucede?

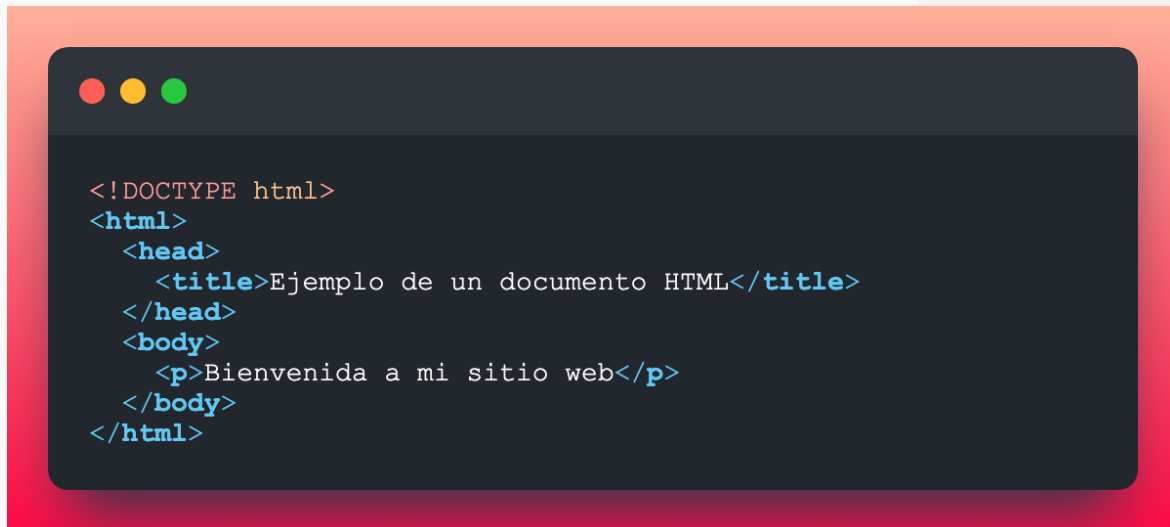
4 Elementos

4.1 Documento

La etiqueta raíz del documento en HTML (The document element) es **<html>** e indica el inicio y fin del documento, proporcionando información acerca del documento.

Ejemplo:

4 ELEMENTOS



El código se renderiza de esta manera al abrirlo en un navegador.



En este ejemplo se puede observar como se utilizan las etiquetas **<html>**, **<head>** y **<body>** para estructurar el documento, el cual comienza con la declaración de tipo de documento “!DOCTYPE html” y la etiqueta **<html>** en la cual se encuentran las etiquetas **<head>** que contiene el título de la página, y la etiqueta **<body>** donde se encuentra el contenido principal de la página.

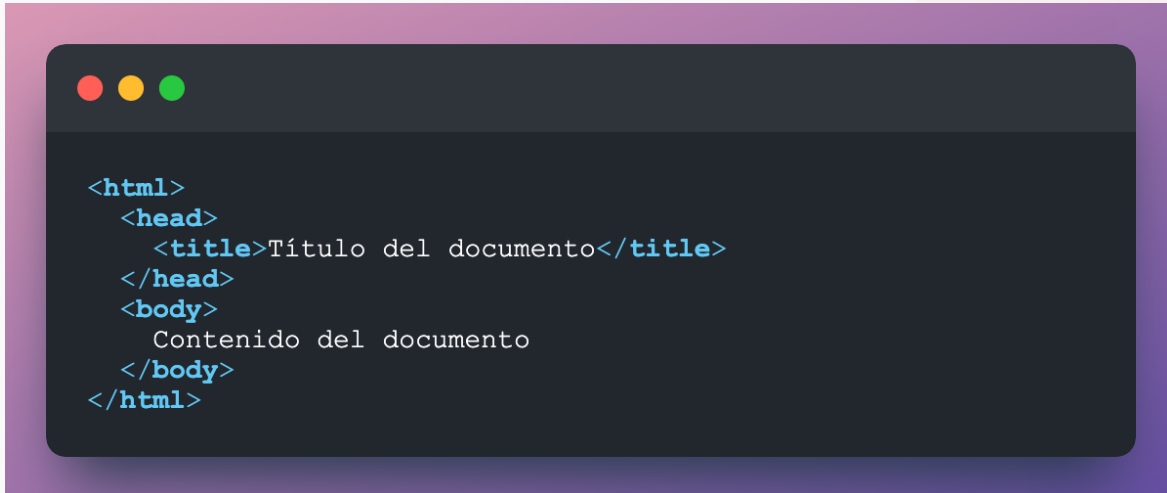
Esta es la etiqueta utilizada como raíz del documento a detalle:

4.1.1 <html>

La etiqueta **<html>** es la etiqueta raíz de un documento HTML. Todos los demás elementos del documento deben estar contenidos dentro de esta etiqueta.

4 ELEMENTOS

Por ejemplo, aquí hay una estructura básica de un documento HTML:



El código se renderiza de esta manera al abrirlo en un navegador.



En este ejemplo, la etiqueta **<html>** engloba todo el contenido del documento, incluyendo la etiqueta **<head>** que contiene información sobre el documento y la etiqueta **<body>** que contiene el contenido visible del documento.

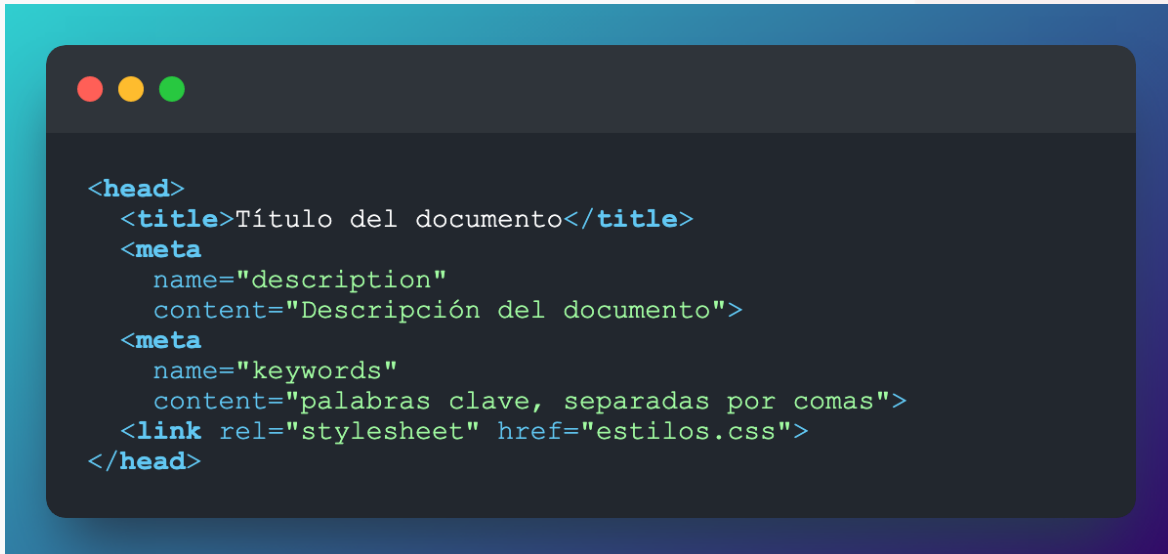
4.2 Metadatos

Las etiquetas de metadatos del documento en HTML (Document metadata) incluyen información en un documento HTML para proporcionar contexto y describir el contenido del documento. Los metadatos no se muestran directamente al usuario, sino que se utilizan por los buscadores y otros servicios para comprender y clasificar el contenido del documento.

4 ELEMENTOS

Los metadatos se incluyen en el elemento **<head>** del documento HTML, y se pueden especificar mediante diferentes etiquetas, como **<title>**, **<meta>**, **<link>**, entre otras.

Aquí hay un ejemplo de un elemento **<head>** con algunos metadatos:



En este ejemplo, la etiqueta **<title>** especifica el título del documento que se mostrará en la pestaña del navegador y en los resultados de búsqueda. La etiqueta **<meta>** con el atributo **<name>** igual a “description” proporciona una descripción del contenido del documento, y la etiqueta **<meta>** con el atributo **<name>** igual a “keywords” proporciona una lista de palabras clave asociadas al documento. La etiqueta **<link>** se utiliza para vincular un archivo de estilos CSS al documento, lo que permite darle formato y estilo al contenido del mismo.

Estas son las etiquetas utilizadas para metadatos a detalle:

4.2.1 <head>

La etiqueta **<head>** en HTML es utilizada para contener información meta acerca del documento que no es visible directamente para el usuario. Los metadatos incluidos

4 ELEMENTOS

en la sección **<head>** incluyen información como el título de la página, la descripción, las palabras clave, las hojas de estilo, entre otros. La información en el head no se mostrará en la página, sino que se utilizará para proporcionar contexto al documento y ayudar a los motores de búsqueda y navegadores a comprender y clasificar el contenido del mismo.

Aquí hay un ejemplo básico de una estructura **<head>**:

A code editor window with a dark background and light-colored text. The code is HTML for the head section of a document. It includes a title, a charset meta tag, a description meta tag, a keywords meta tag, and a link to a stylesheet. The code is as follows:

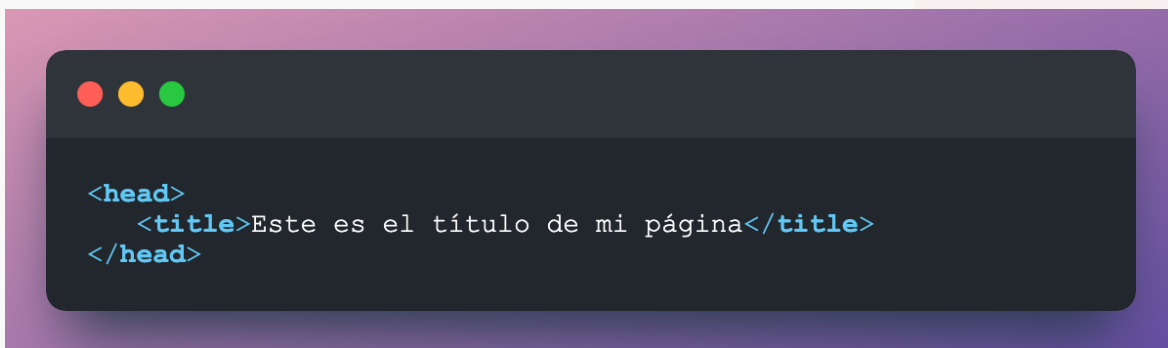
```
<head>
  <title>Título del documento</title>
  <meta charset="UTF-8">
  <meta
    name="description"
    content="Breve descripción del contenido">
  <meta
    name="keywords"
    content="palabras clave, separadas por comas">
  <link rel="stylesheet" href="estilos.css">
</head>
```

En este ejemplo, la etiqueta **<title>** especifica el título de la página que se mostrará en la pestaña del navegador y en los resultados de búsqueda. La etiqueta **<meta>** con el atributo “charset” especifica el conjunto de caracteres utilizado en el documento, generalmente utf-8. El resto de **<meta>** proporciona una descripción del contenido del documento y una lista de palabras clave asociadas al documento, y la etiqueta **<link>** se utiliza para vincular un archivo de estilos CSS al documento, lo que permite darle formato y estilo al contenido del mismo.

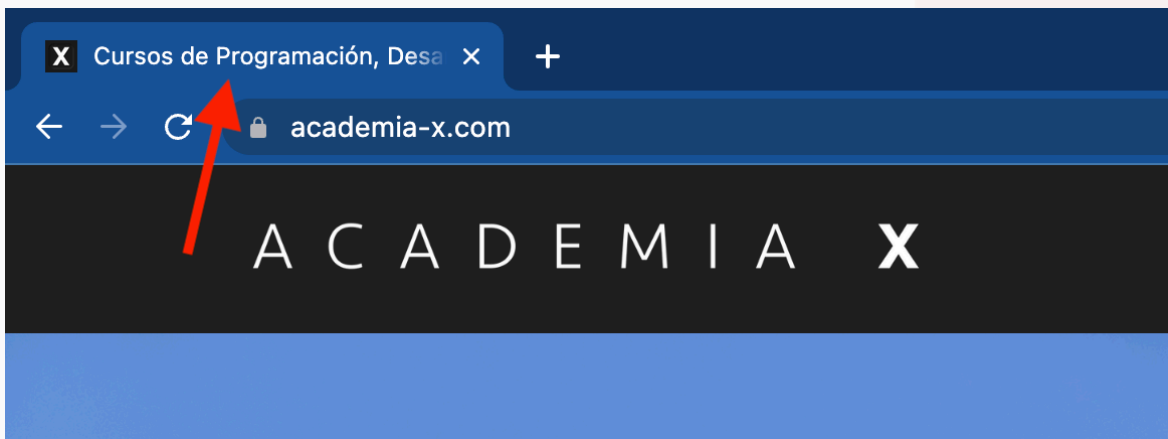
4.2.2 <title>

La etiqueta **<title>** en HTML es utilizada para especificar el título de una página web. El título se muestra en la pestaña del navegador y es utilizado por los motores de búsqueda para identificar y clasificar la página. Es importante que los títulos sean descriptivos y representativos del contenido de la página para mejorar el SEO.

La estructura de una etiqueta title es como sigue:



Este es un ejemplo en el que la etiqueta **<title>** es igual a “Cursos de Programación, Desarrollo Web, e Ingeniería de Software.”



La etiqueta **<title>** siempre se encuentra dentro de la etiqueta **<head>**, y se recomienda que sean cortos y descriptivos, preferiblemente menos de 60 caracteres. Es importante mencionar que el título es uno de los factores más importantes para

4 ELEMENTOS

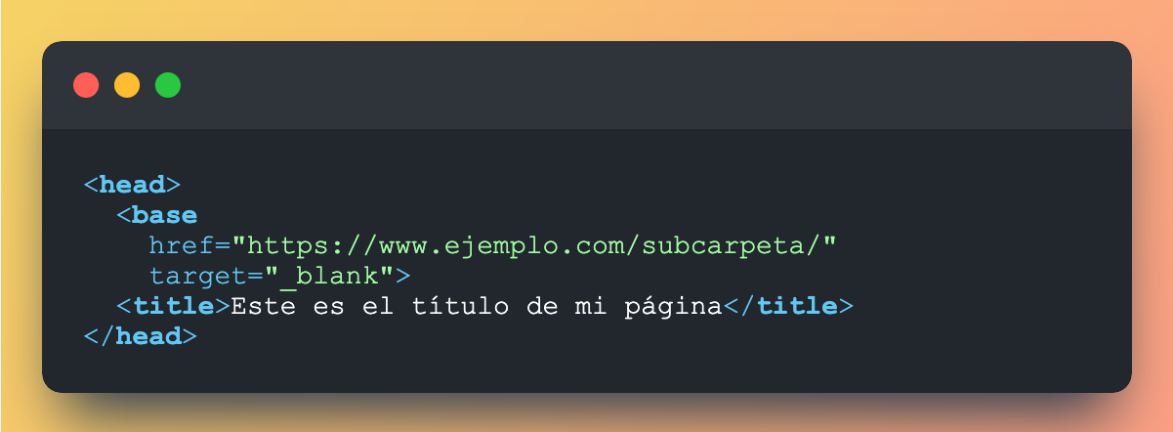
el SEO, por lo que debe ser escrito pensando en las palabras clave relevantes y buscadas por los usuarios para esa página específica.

4.2.3 <base>

La etiqueta **<base>** en HTML es utilizada para establecer una URL base para todas las URLs relativas en un documento HTML. Esto permite especificar una ruta base para todos los enlaces, imágenes y recursos incluidos en una página, de manera que se pueden utilizar URLs relativas en lugar de URLs absolutas.

La etiqueta **<base>** se utiliza dentro de la sección head del documento HTML, y solo se permite una etiqueta **<base>** por documento.

Aquí hay un ejemplo de una etiqueta **<base>** en un documento:

A code editor window with a dark background and orange border. It contains the following HTML code:

```
<head>
  <base
    href="https://www.ejemplo.com/subcarpeta/"
    target="_blank">
  <title>Este es el título de mi página</title>
</head>
```

En este ejemplo, el atributo "href" especifica la URL base a utilizar, en este caso <https://www.ejemplo.com/subcarpeta/>. El atributo target especifica donde se abra los enlaces, en este caso _blank significa que se abra en una nueva pestaña.

Con esta etiqueta **<base>**, todos los enlaces relativos en el documento, como por ejemplo: ****, se convertirán en:

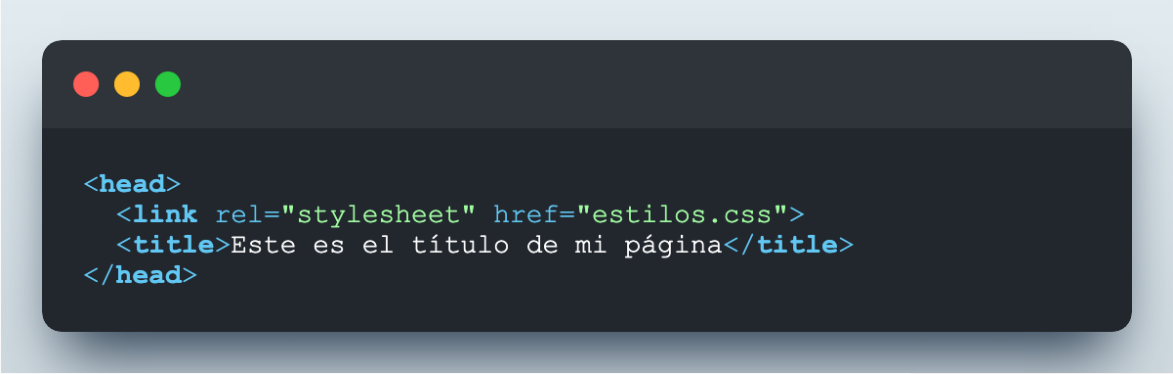
****, y los recursos como imágenes ****, se convertirán en:

Es importante mencionar que esta etiqueta es útil cuando se está creando una estructura de carpetas o se va a mover una página a otra ruta, pero en muchos casos su uso no es esencial.

4.2.4 <link>

La etiqueta **<link>** en HTML se utiliza para vincular un archivo externo a un documento HTML, como una hoja de estilos CSS o un archivo de iconos. La etiqueta se utiliza dentro de la sección **<head>** del documento y no tiene contenido, sino que se utilizan sus atributos para especificar la relación y la ruta del archivo vinculado.

Un ejemplo de una etiqueta **<link>** utilizando para vincular una hoja de estilos es:

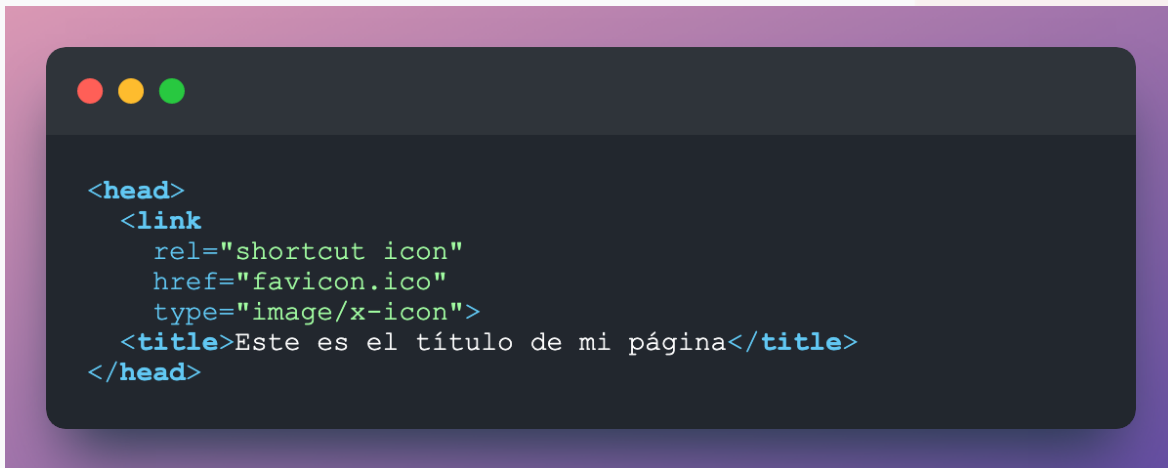


```
<head>
  <link rel="stylesheet" href="estilos.css">
  <title>Este es el título de mi página</title>
</head>
```

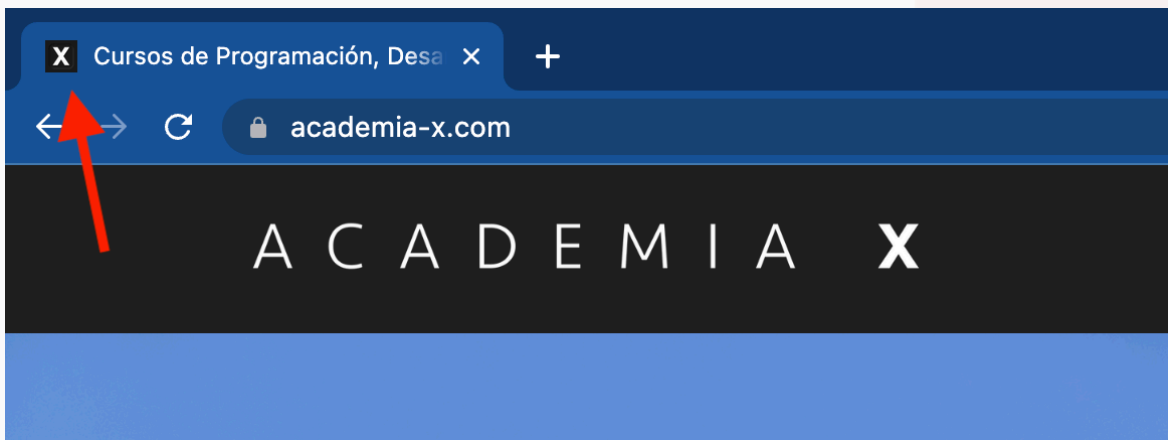
En este ejemplo el atributo `rel="stylesheet"` indica que se está vinculando una hoja de estilo al documento, y el atributo `href="estilos.css"` especifica la ruta del archivo de estilos CSS.

También puede utilizar **<link>** para vincular un archivo de iconos y se especifica mediante `rel="shortcut icon"` o `rel="icon"` en la etiqueta, y el atributo `href` especifica la ruta al archivo.

4 ELEMENTOS



Este es un ejemplo de la renderización del ícono "favicon.ico" en un navegador.



Además, la etiqueta **<link>** también se puede utilizar para vincular un archivo RSS para una página, se especifica mediante `rel="alternate"` y atributo `type="application/rss+xml"` en la etiqueta y el atributo `href` especifica la ruta al archivo RSS

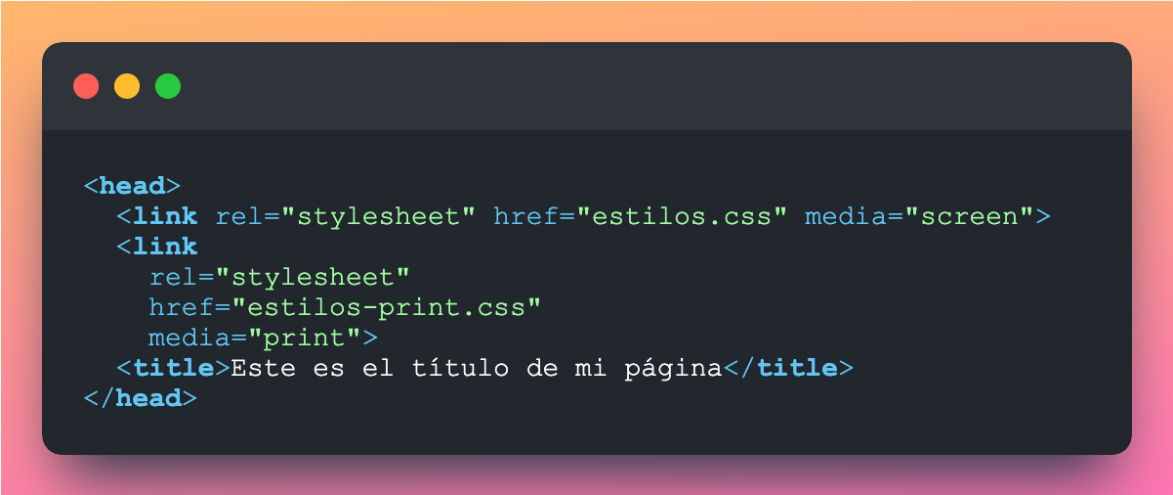
Es importante mencionar que la etiqueta **<link>** es una etiqueta vacía, es decir no tiene contenido, y su uso es específicamente para vincular archivos externos, no se usa para contenido dentro del documento HTML.

4.2.5 media

El atributo “media” en la etiqueta **<link>** en HTML se utiliza para especificar a qué dispositivos o tipos de pantalla se aplicará una hoja de estilos vinculada. Este atributo se utiliza para optimizar la experiencia del usuario en dispositivos móviles y pantallas con diferentes resoluciones, al permitir que los diseños se adapten automáticamente.

El valor del atributo “media” se puede especificar mediante una serie de reglas de medios, como "screen" para pantallas de ordenador, "print" para impresión, "all" para todos los dispositivos, entre otros.

Un ejemplo de cómo se utiliza el atributo “media” en una etiqueta **<link>** es:



```
<head>
  <link rel="stylesheet" href="estilos.css" media="screen">
  <link
    rel="stylesheet"
    href="estilos-print.css"
    media="print">
  <title>Este es el título de mi página</title>
</head>
```

En este ejemplo se está especificando que la hoja de estilo estilos.css se aplicará solo a dispositivos con pantalla (media="screen") y estilos-print.css solo se aplicará al momento de imprimir (media="print"). Esto permite tener un control más preciso de cómo se verá la página en diferentes dispositivos y situaciones.

Por defecto el valor para “media” es "all" si no se especifica el valor, entonces se aplicará a todos los dispositivos.

Es importante mencionar que este atributo solo es aplicable para etiquetas **<link>**

4 ELEMENTOS

con `rel="stylesheet"`, ya que estas son las que se utilizan para vincular hojas de estilo.

4.2.6 type

El atributo `“type”` en la etiqueta **<link>** en HTML se utiliza para especificar el tipo de archivo vinculado. Este atributo es opcional pero recomendado, ya que ayuda al navegador a determinar cómo interpretar el archivo y a optimizar la carga de la página.

El valor del atributo `“type”` se especifica utilizando una notación MIME (Multipurpose Internet Mail Extensions), que es un sistema estándar utilizado para identificar diferentes tipos de archivos. Por ejemplo, el tipo MIME para una hoja de estilos CSS es `"text/css"` y el tipo MIME para una imagen PNG es `"image/png"`.

Un ejemplo de cómo se utiliza el atributo `“type”` en una etiqueta **<link>** es:



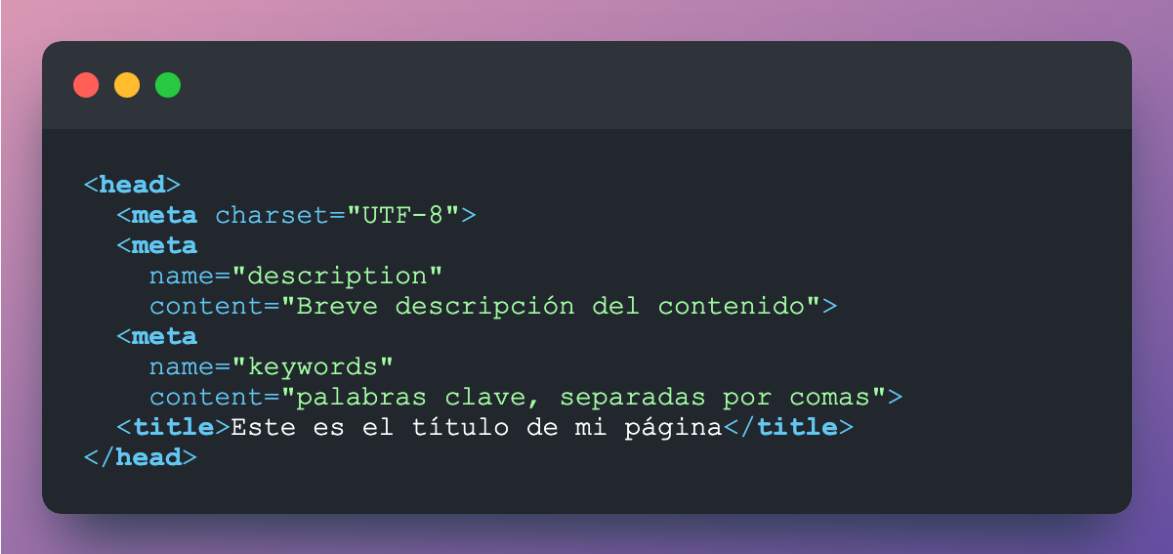
En este ejemplo se está especificando que el tipo de archivo vinculado en la primera etiqueta es una hoja de estilos en formato CSS con `"text/css"` y en la segunda etiqueta es un archivo de icono con `"image/x-icon"`.

Es importante mencionar que algunos navegadores pueden inferir el tipo.

4.2.7 <meta>

La etiqueta **<meta>** en HTML se utiliza para incluir información adicional acerca del documento HTML que no es visible directamente para el usuario. Los metadatos incluidos en la sección **<meta>** incluyen información como la descripción de la página, las palabras clave, la codificación de caracteres, la configuración de robots, entre otros. Los metadatos proporcionan contexto al documento y ayudan a los motores de búsqueda y navegadores a comprender y clasificar el contenido del mismo.

Aquí hay un ejemplo básico de algunas etiquetas **<meta>** utilizadas para proporcionar información sobre el documento:

A screenshot of a code editor window with a dark background and a purple border. The window has three colored window control buttons (red, yellow, green) in the top-left corner. The code inside is HTML code for the head section of a document, showing three meta tags and a title tag. The code is color-coded: tags are blue, attributes are green, and content is green.

```
<head>
  <meta charset="UTF-8">
  <meta
    name="description"
    content="Breve descripción del contenido">
  <meta
    name="keywords"
    content="palabras clave, separadas por comas">
  <title>Este es el título de mi página</title>
</head>
```

En este ejemplo, la primera etiqueta **<meta>** con el atributo “charset” especifica el conjunto de caracteres utilizado en el documento, generalmente utf-8. El segundo **<meta>** con atributo name= “description” y el contenido es una descripción breve del contenido del documento, esta descripción es utilizada por los motores de búsqueda como un resumen del contenido del documento. El tercer **<meta>** con atributo name= “keywords” especifica una lista de palabras clave asociadas al documento, separadas por comas, estas son utilizadas por los motores de búsqueda para identificar el contenido del documento.

4 ELEMENTOS

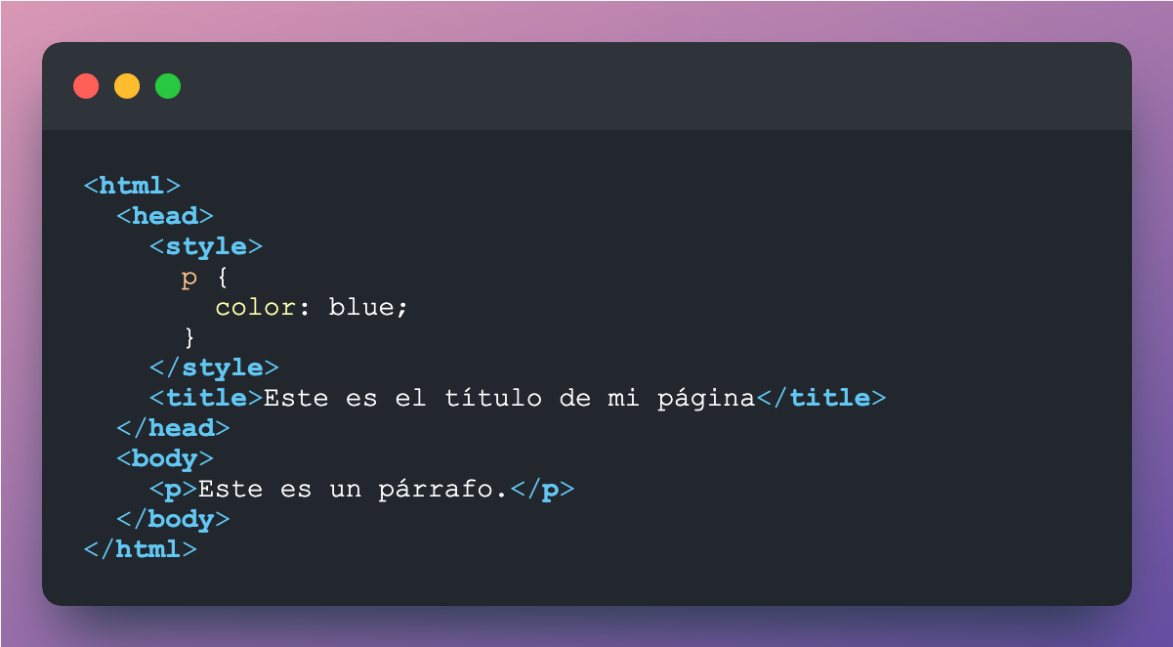
Es importante mencionar que no todas las etiquetas **<meta>** son interpretadas por los motores de búsqueda, sino solo algunas específicas, por lo que es recomendable investigar cuales son las mas relevantes para el contenido de tu documento.

4.2.8 <style>

La etiqueta **<style>** en HTML se utiliza para incluir hojas de estilo CSS (Cascading Style Sheets) en un documento HTML. Esta etiqueta se utiliza para aplicar estilos visuales a los elementos de una página web, como el color de fondo, el tamaño de letra, el tipo de letra, el espaciado, entre otros.

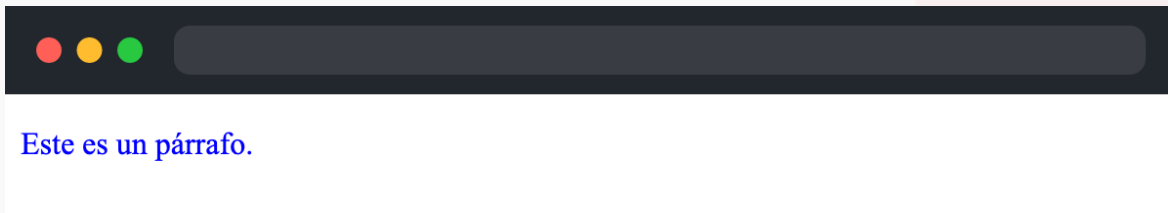
La etiqueta **<style>** se coloca dentro de la sección **<head>** del documento HTML, y su contenido consta de reglas CSS escritas en un lenguaje CSS.

Aquí hay un ejemplo de cómo se utiliza la etiqueta **<style>** para aplicar un estilo a los elementos de una página:

A code editor window with a dark background and light-colored text. The code is HTML, showing the structure of a document with a style tag. The code is as follows:

```
<html>
  <head>
    <style>
      p {
        color: blue;
      }
    </style>
    <title>Este es el título de mi página</title>
  </head>
  <body>
    <p>Este es un párrafo.</p>
  </body>
</html>
```

El código se renderiza de esta manera al abrirlo en un navegador.



En este ejemplo, se establece que todos los elementos `<p>` tendrán color azul. No te preocupes de entender CSS ya que ese es un tema fuera del alcance de HTML.

4.3 Secciones

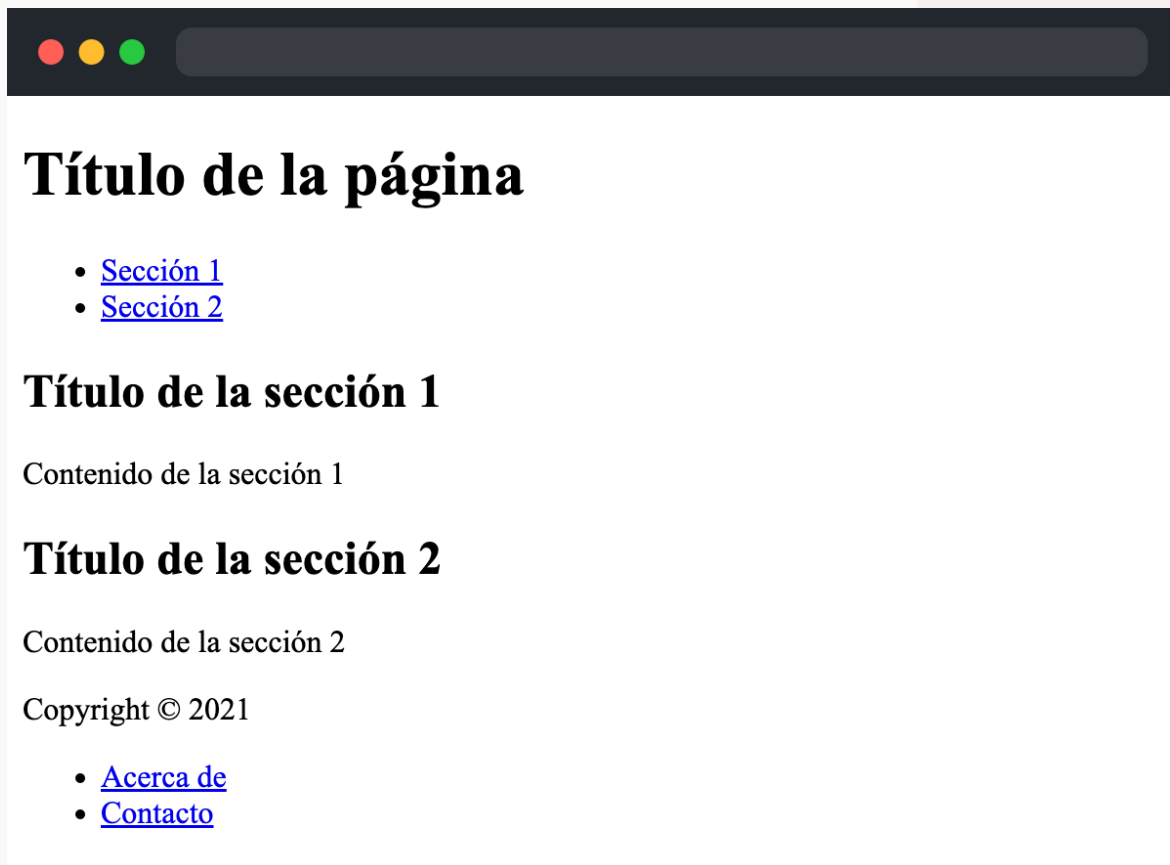
Las etiquetas de secciones en HTML (Sections) permiten organizar el contenido de una página web con un enfoque semántico. Estas etiquetas indican al navegador y a los buscadores el tipo de contenido que se encuentra dentro de la sección, permitiendo una mejor comprensión y accesibilidad del contenido.

Algunas etiquetas comunes utilizadas para definir secciones incluyen **`<header>`**, **`<nav>`**, **`<main>`**, **`<article>`**, **`<section>`**, **`<aside>`**, y **`<footer>`**.

Aquí hay un ejemplo de cómo se utilizan varias etiquetas para organizar el contenido de una página web:

```
<body>
  <header>
    <h1>Título de la página</h1>
    <nav>
      <ul>
        <li><a href="#seccion1">Sección 1</a></li>
        <li><a href="#seccion2">Sección 2</a></li>
      </ul>
    </nav>
  </header>
  <main>
    <section id="seccion1">
      <h2>Título de la sección 1</h2>
      <p>Contenido de la sección 1</p>
    </section>
    <section id="seccion2">
      <h2>Título de la sección 2</h2>
      <p>Contenido de la sección 2</p>
    </section>
  </main>
  <footer>
    <p>Copyright © 2021</p>
    <ul>
      <li><a href="#">Acerca de</a></li>
      <li><a href="#">Contacto</a></li>
    </ul>
  </footer>
</body>
```

El código se renderiza de esta manera al abrirlo en un navegador.



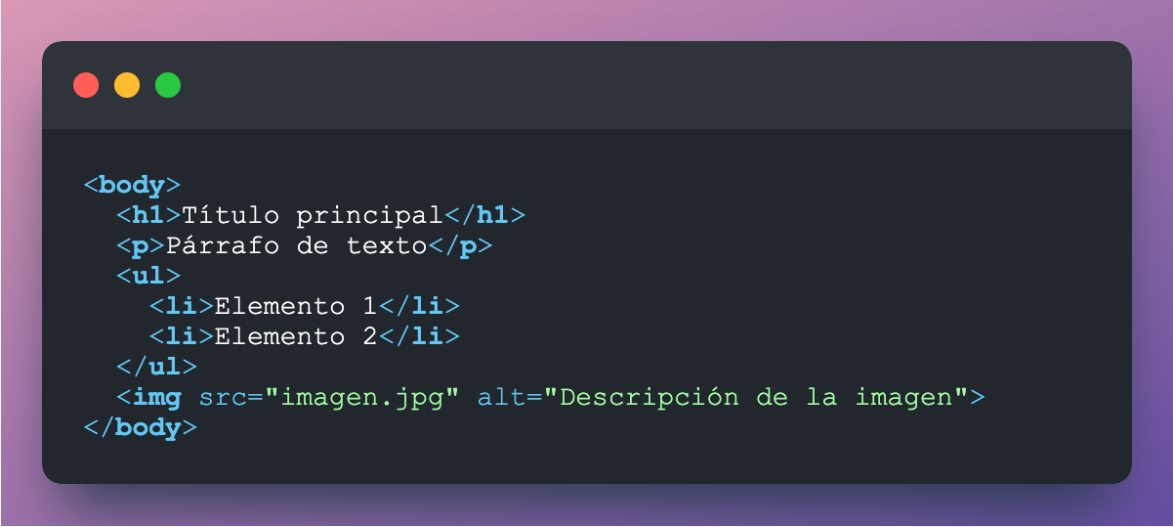
En este ejemplo se puede ver como se utilizan varias etiquetas para organizar el contenido de una página web, como **<header>** para el encabezado, **<main>** para el contenido principal, **<section>** para agrupar contenido relacionado, **<footer>** para el pie de página, entre otras. Utilizar etiquetas semánticas ayuda a los buscadores y navegadores a entender el contenido de la página, y mejora la accesibilidad y experiencia del usuario.

Estas son las etiquetas utilizadas para secciones a detalle:

4.3.1 <body>

El elemento **<body>** del documento HTML es el contenedor principal para el contenido visible de una página web. Todos los elementos que se deseen mostrar en la página deben estar contenidos dentro de este elemento.

Aquí hay un ejemplo básico de un elemento **<body>** con contenido:

A code editor window with a dark background and a purple gradient border. It contains the following HTML code:

```
<body>
  <h1>Título principal</h1>
  <p>Párrafo de texto</p>
  <ul>
    <li>Elemento 1</li>
    <li>Elemento 2</li>
  </ul>
  
</body>
```

El código se renderiza de esta manera al abrirlo en un navegador.



Título principal

Párrafo de texto

- Elemento 1
- Elemento 2



En este ejemplo, el elemento **<body>** contiene un título principal (**<h1>**), un párrafo de texto (**<p>**), una lista sin ordenar (****) con dos elementos (****), y una imagen (****). Todos estos elementos se mostrarán en la página web una vez que se cargue el documento HTML.

4.3.2 <article>

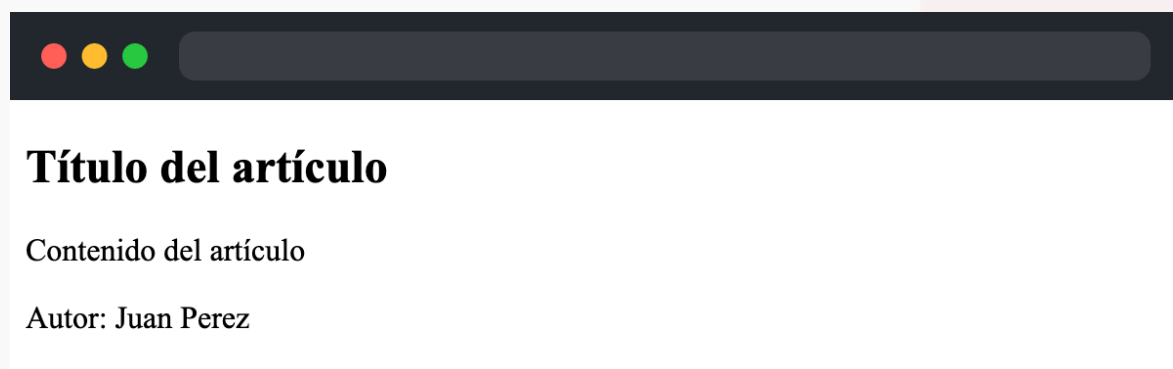
La etiqueta **<article>** en HTML es una etiqueta semántica que se utiliza para identificar contenido independiente y auto-contenido dentro de un documento HTML. El contenido dentro de esta etiqueta debería ser capaz de entenderse de manera independiente del resto del contenido de la página. Ejemplos de contenido que se podría incluir en una etiqueta **<article>** incluyen noticias, artículos de blog, comentarios, entre otros.

Aquí hay un ejemplo de cómo se utiliza la etiqueta **<article>** en un documento:

A code editor window with a dark background and a red-to-orange gradient border. It displays the following HTML code:

```
<body>
  <article>
    <h2>Título del artículo</h2>
    <p>Contenido del artículo</p>
    <footer>
      <p>Autor: Juan Perez</p>
    </footer>
  </article>
</body>
```

El código se renderiza de esta manera al abrirlo en un navegador.



4 ELEMENTOS

En este ejemplo se muestra como se estructura un articulo dentro de la etiqueta `article`, donde se incluye un titulo h2, un contenido y un pie de pagina con el autor del mismo.

Es importante mencionar que una etiqueta `<article>` no es necesaria para todos los contenidos, solo si el contenido dentro de esa sección es independiente y no necesita ser contextualizado con otras partes del documento. Además, puede haber varias etiquetas `<article>` dentro de una página y pueden estar anidadas dentro de otras etiquetas como `<section>`, `<header>`, `<footer>`, entre otras.

4.3.3 `<section>`

La etiqueta `<section>` en HTML es una etiqueta semántica que se utiliza para agrupar contenido relacionado dentro de un documento HTML. El contenido dentro de esta etiqueta debería estar relacionado temáticamente o tener algún propósito común.

La diferencia entre `<article>` y `<section>` es que `<article>` suele ser utilizado para contenido independiente y que podría ser compartido o leído por sí mismo mientras que `<section>` es utilizado para agrupar contenido relacionado dentro de una página y no necesariamente independiente.

Aquí hay un ejemplo de cómo se utiliza la etiqueta `<section>` en un documento:

```
<body>
  <header>
    <h1>Este es el título de mi página</h1>
  </header>
  <nav>
    <ul>
      <li><a href="#seccion1">Sección 1</a></li>
      <li><a href="#seccion2">Sección 2</a></li>
      <li><a href="#seccion3">Sección 3</a></li>
    </ul>
  </nav>
  <main>
    <section id="seccion1">
      <h2>Título de la sección 1</h2>
      <p>Contenido de la sección 1</p>
    </section>
    <section id="seccion2">
      <h2>Título de la sección 2</h2>
      <p>Contenido de la sección 2</p>
    </section>
    <section id="seccion3">
      <h2>Título de la sección 3</h2>
      <p>Contenido de la sección 3</p>
    </section>
  </main>
  <footer>
    <p>Copyright © 2021</p>
  </footer>
</body>
```

El código se renderiza de esta manera al abrirlo en un navegador.



En este ejemplo se muestra como se estructura una página con diferentes secciones dentro de la etiqueta **<section>** y cómo estas son agrupadas dentro del **<main>** de la página. Además, se pueden observar cómo cada una tiene un título y contenido distinto, y como estan vinculadas desde el menú de navegación mediante anclajes.

Es importante mencionar que una etiqueta **<section>** no es necesaria para todos los contenidos, solo si el contenido dentro de esa sección es relacionado temáticamente.

4.3.4 <nav>

La etiqueta **<nav>** en HTML es una etiqueta semántica que se utiliza para identificar una sección de una página web que contiene enlaces de navegación. Estos enlaces pueden ser internos o externos a la página y pueden incluir links a otras páginas, a secciones dentro de la misma página, etc. La idea de utilizar **<nav>** es marcar específicamente las secciones de una página web que contienen enlaces de navegación para que los buscadores y navegadores puedan entender mejor el contenido de la página.

Aquí hay un ejemplo de cómo se utiliza la etiqueta **<nav>** en un documento:

```
<body>
  <header>
    <h1>Este es el título de mi página</h1>
  </header>
  <nav>
    <ul>
      <li><a href="#seccion1">Sección 1</a></li>
      <li><a href="#seccion2">Sección 2</a></li>
      <li><a href="#seccion3">Sección 3</a></li>
    </ul>
  </nav>
  <main>
    <section id="seccion1">
      <h2>Título de la sección 1</h2>
      <p>Contenido de la sección 1</p>
    </section>
    <section id="seccion2">
      <h2>Título de la sección 2</h2>
      <p>Contenido de la sección 2</p>
    </section>
    <section id="seccion3">
      <h2>Título de la sección 3</h2>
      <p>Contenido de la sección 3</p>
    </section>
  </main>
  <footer>
    <p>Copyright © 2021</p>
  </footer>
</body>
```

El código se renderiza de esta manera al abrirlo en un navegador.



En este ejemplo, se puede observar que la etiqueta `<nav>` se utiliza para contener una lista de enlaces de navegación que permiten al usuario acceder a diferentes secciones del documento y también podría redireccionar a otras páginas.

4.3.5 `<aside>`

La etiqueta `<aside>` en HTML es una etiqueta semántica que se utiliza para identificar contenido secundario relacionado con el contenido principal de una página web. El contenido dentro de esta etiqueta puede incluir información como notas al pie,

4 ELEMENTOS

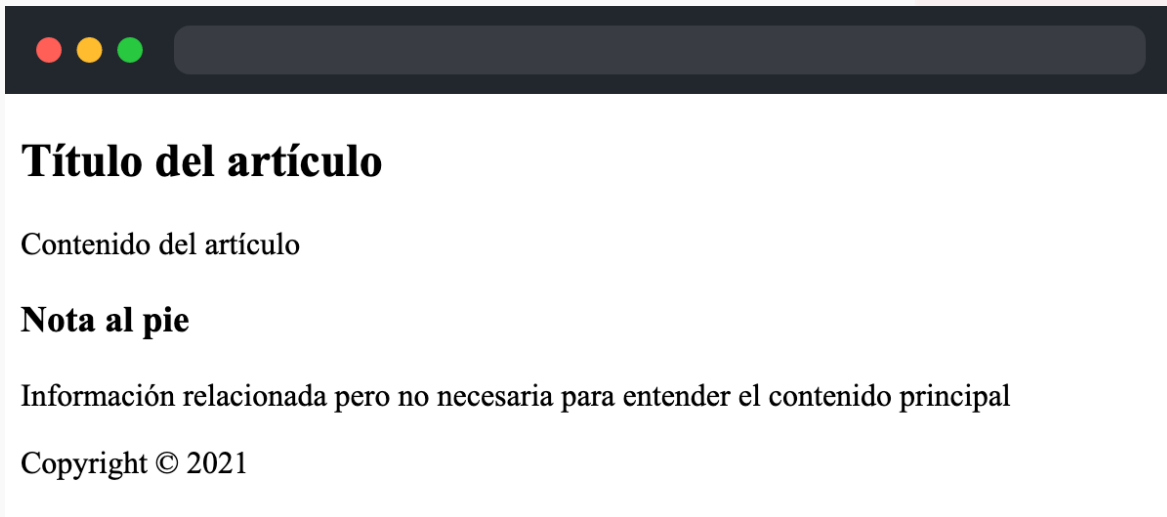
citas, definiciones, comentarios, entre otros. La idea de utilizar **<aside>** es marcar específicamente los contenidos que son relevantes pero no necesarios para entender el contexto principal de la página.

Aquí hay un ejemplo de cómo se utiliza la etiqueta **<aside>** en un documento:

A code editor window with a dark background and a green title bar. The code is written in a light blue font and shows the structure of an HTML document. It includes a main section with a title and content, followed by an aside section with a note, and a footer with a copyright notice.

```
<body>
  <main>
    <section>
      <h2>Título del artículo</h2>
      <p>Contenido del artículo</p>
    </section>
    <aside>
      <h3>Nota al pie</h3>
      <p>Información relacionada pero no necesaria para
        entender el contenido principal</p>
    </aside>
  </main>
  <footer>
    <p>Copyright © 2021</p>
  </footer>
</body>
```

El código se renderiza de esta manera al abrirlo en un navegador.



En este ejemplo se puede observar que se incluye una sección principal (**<section>**) con contenido principal y un aside donde se encuentra una nota al pie relacionada con el contenido principal pero que no es necesaria para entenderlo.

Es importante mencionar que una etiqueta **<aside>** no es necesaria para todos los contenidos y puede ser utilizado en conjunto con otras etiquetas como **<section>** y **<article>**.

4.3.6 <h1-6>

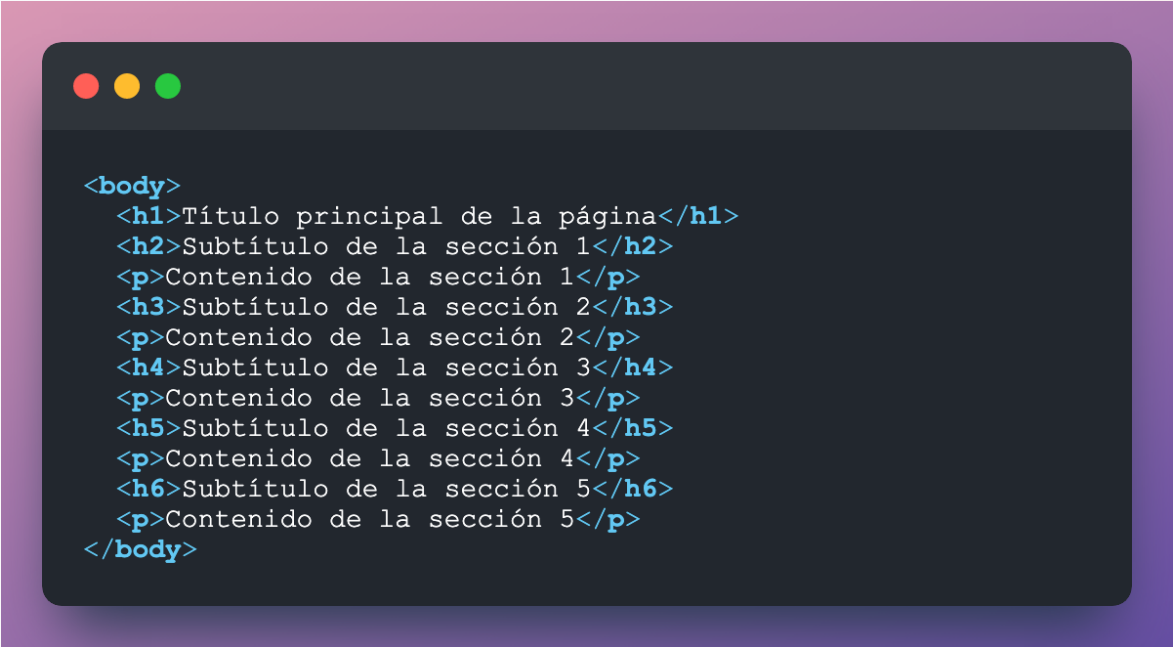
Las etiquetas **<h1>** a **<h6>** en HTML son etiquetas de encabezado que se utilizan para dar formato y estructura al contenido de una página web. Estas etiquetas son utilizadas para crear títulos y subtítulos dentro del contenido, y se utilizan para indicar la jerarquía de la información dentro de una página.

<h1> es el nivel de encabezado de mayor importancia y se utiliza para el título principal de una página, mientras que **<h2>** es el nivel de encabezado inmediatamente debajo de **<h1>** y así sucesivamente hasta llegar a **<h6>** que es el nivel de encabezado de menor importancia. Es importante recordar que el nivel de importancia de

4 ELEMENTOS

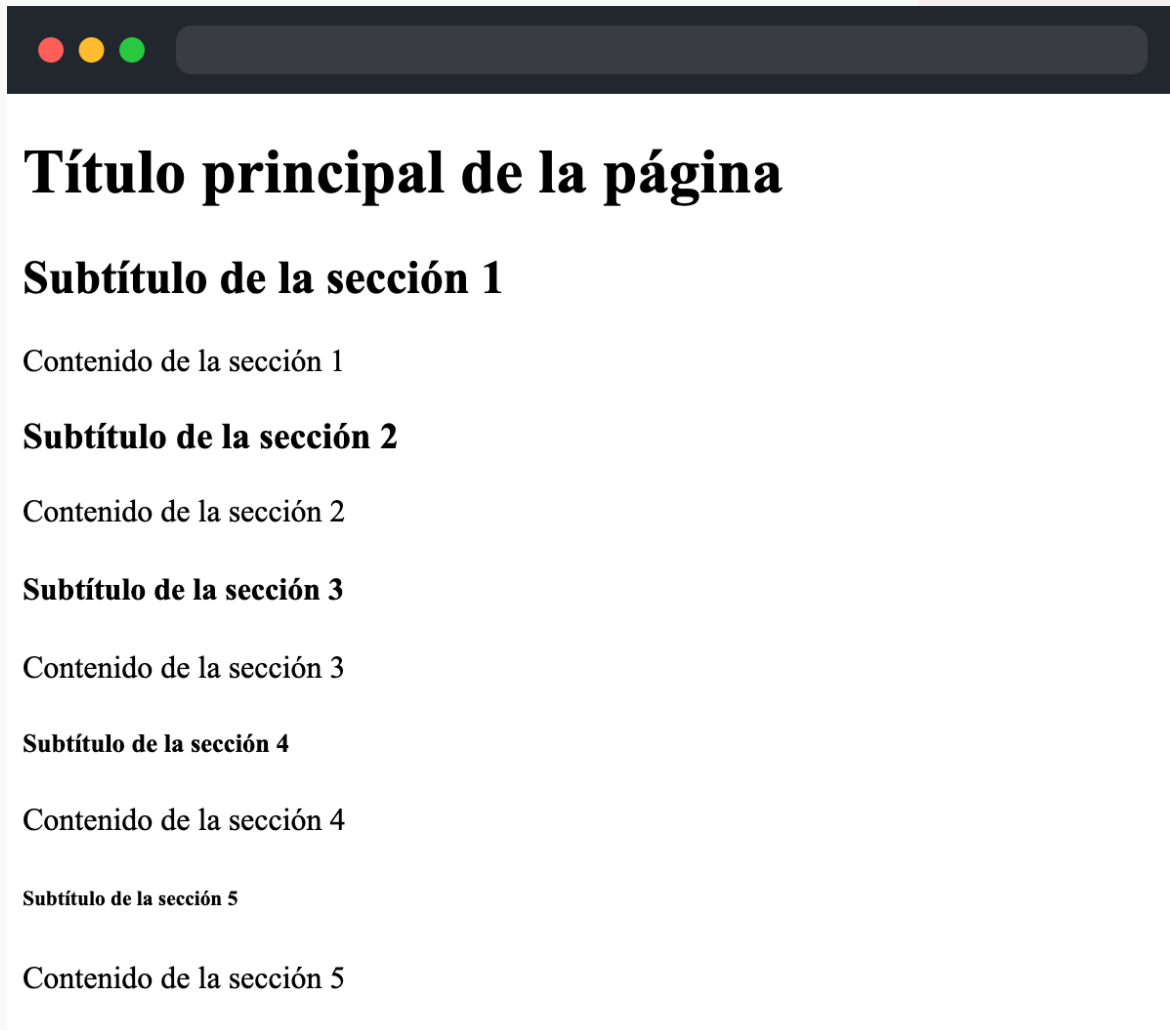
los encabezados debe seguir un orden lógico, **<h1>** es el principal y **<h6>** es el de menor importancia.

Aquí hay un ejemplo de cómo se utilizan las etiquetas de encabezado en un documento:

A code editor window with a dark background and light blue text. The code is HTML markup for a document body. It starts with an opening <body> tag, followed by a series of nested heading and paragraph tags. The headings are h1, h2, h3, h4, h5, and h6, each followed by a paragraph tag <p>. The code is as follows:

```
<body>
  <h1>Título principal de la página</h1>
  <h2>Subtítulo de la sección 1</h2>
  <p>Contenido de la sección 1</p>
  <h3>Subtítulo de la sección 2</h3>
  <p>Contenido de la sección 2</p>
  <h4>Subtítulo de la sección 3</h4>
  <p>Contenido de la sección 3</p>
  <h5>Subtítulo de la sección 4</h5>
  <p>Contenido de la sección 4</p>
  <h6>Subtítulo de la sección 5</h6>
  <p>Contenido de la sección 5</p>
</body>
```

El código se renderiza de esta manera al abrirlo en un navegador.



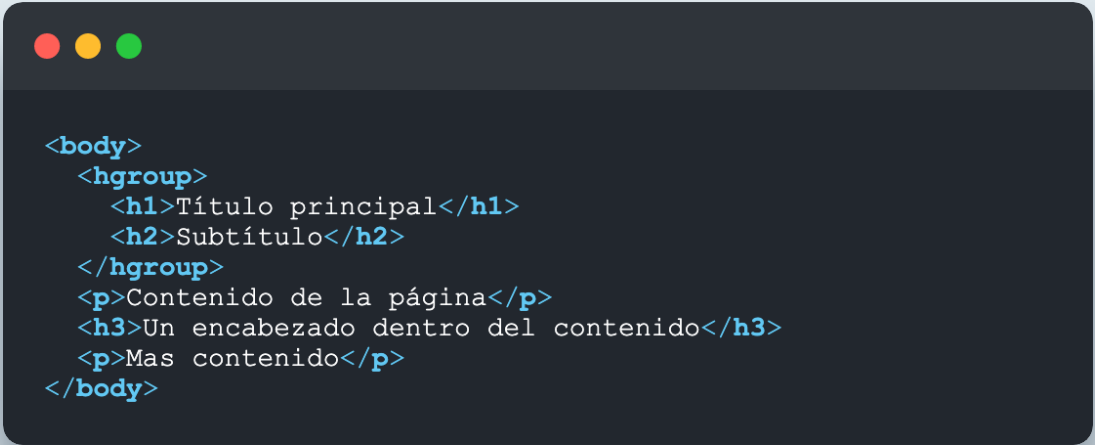
En este ejemplo se puede ver cómo se utilizan las etiquetas `<h1>` a `<h6>` para crear una jerarquía de títulos y subtítulos dentro del contenido de la página, donde `<h1>` es el título principal y `<h6>` es el de menor importancia.

4.3.7 <hgroup>

La etiqueta **<hgroup>** en HTML es una etiqueta de agrupamiento que se utiliza para agrupar varias etiquetas de encabezado (**<h1>** a **<h6>**) en un solo grupo. El objetivo de utilizar **<hgroup>** es establecer una jerarquía para los títulos dentro de un grupo de encabezados y asegurar que una página tenga una jerarquía lógica y consistente de encabezados.

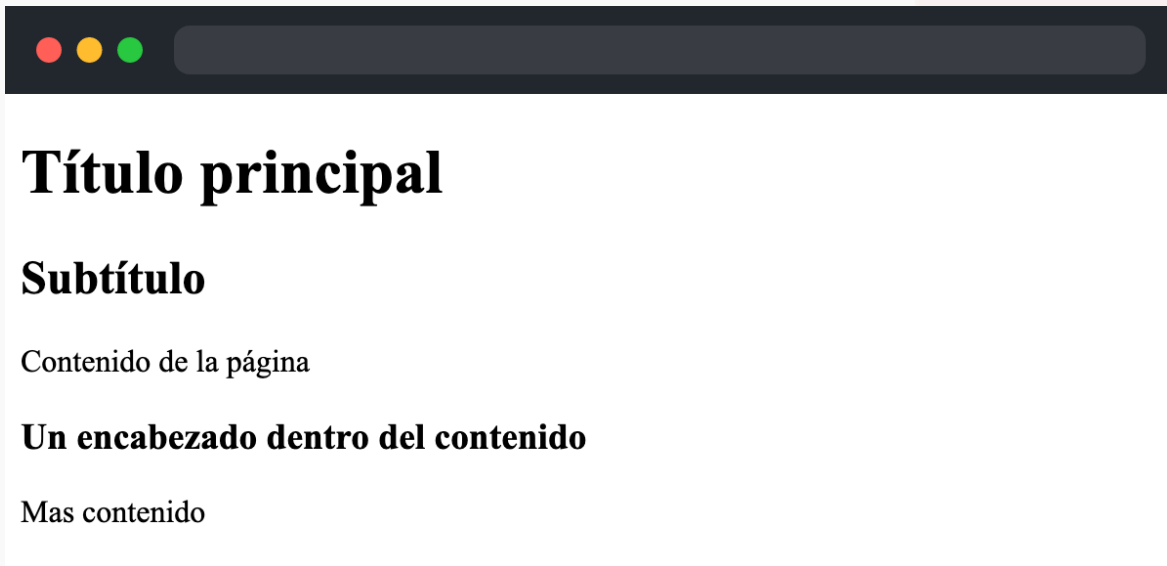
La etiqueta **<hgroup>** es utilizada para indicar que una serie de encabezados están relacionados entre sí y pertenecen al mismo contexto. Es importante mencionar que **<hgroup>** no es bastante utilizado, y es más común utilizar solo un **<h1>** para identificar el encabezado principal de una sección y utilizar las etiquetas **<h2>** a **<h6>** para subtítulos y encabezados secundarios.

Aquí hay un ejemplo de cómo se utiliza la etiqueta **<hgroup>** en un documento:



```
<body>
  <hgroup>
    <h1>Título principal</h1>
    <h2>Subtítulo</h2>
  </hgroup>
  <p>Contenido de la página</p>
  <h3>Un encabezado dentro del contenido</h3>
  <p>Mas contenido</p>
</body>
```

El código se renderiza de esta manera al abrirlo en un navegador.



En este ejemplo se puede ver como `<h1>` y `<h2>` son agrupados en un solo grupo mediante la etiqueta `<hgroup>`.

4.3.8 `<header>`

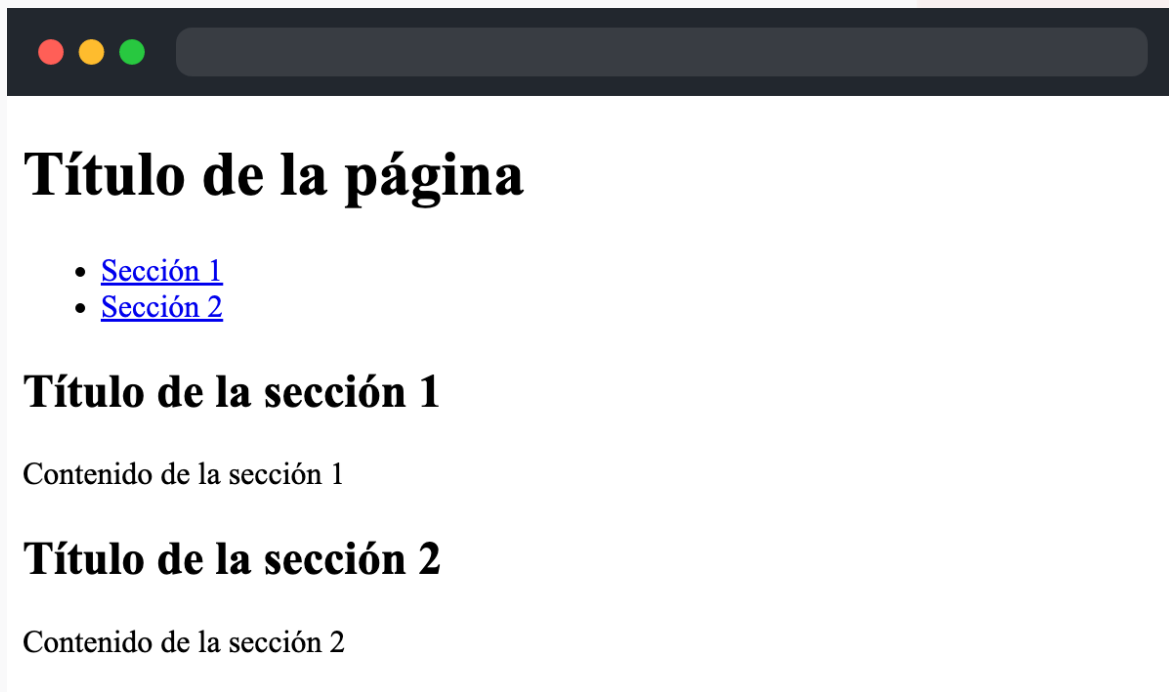
La etiqueta `<header>` en HTML es una etiqueta semántica que se utiliza para identificar la sección de encabezado de una página web. Esta sección puede incluir contenido como el título de la página, el logotipo, un menú de navegación, un buscador, entre otros elementos. El objetivo de utilizar `<header>` es marcar específicamente la sección de encabezado de una página para que los buscadores y navegadores puedan entender mejor el contenido de la página.

La etiqueta `<header>` no se limita a ser utilizada solo al principio de la página, sino que puede ser utilizada varias veces en la misma página, para secciones y artículos dentro de la página, permitiendo así tener una estructura jerárquica y organizada.

Aquí hay un ejemplo de cómo se utiliza la etiqueta `<header>` en un documento:

```
<body>
  <header>
    <h1>Título de la página</h1>
    <nav>
      <ul>
        <li><a href="#seccion1">Sección 1</a></li>
        <li><a href="#seccion2">Sección 2</a></li>
      </ul>
    </nav>
  </header>
  <main>
    <section id="seccion1">
      <h2>Título de la sección 1</h2>
      <p>Contenido de la sección 1</p>
    </section>
    <section id="seccion2">
      <h2>Título de la sección 2</h2>
      <p>Contenido de la sección 2</p>
    </section>
  </main>
</body>
```

El código se renderiza de esta manera al abrirlo en un navegador.



En este ejemplo, se puede ver cómo se utiliza la etiqueta **<header>** para contener el título de la página y un menú de navegación, indicando que estos elementos son parte del encabezado de la página. A su vez, se tiene un **<main>** que contiene dos **<section>** con contenido principal dentro de la página.

4.3.9 <footer>

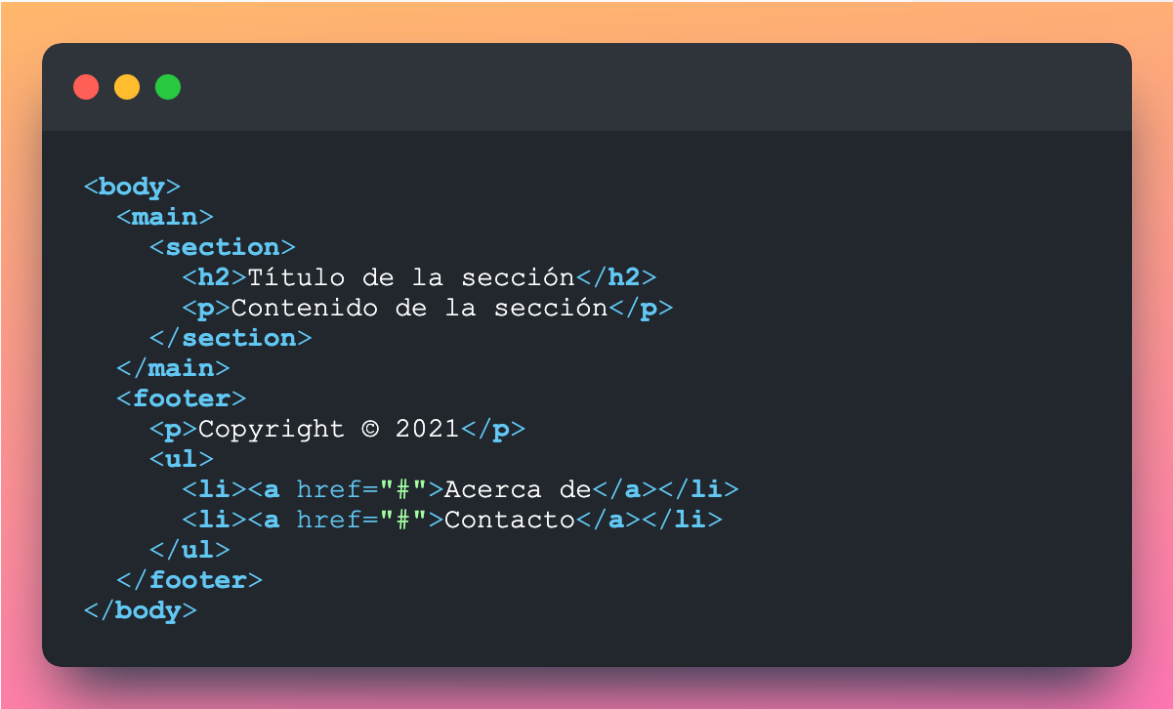
La etiqueta **<footer>** en HTML es una etiqueta semántica que se utiliza para identificar la sección de pie de página de una página web. Esta sección puede incluir contenido como el autor de la página, información de contacto, una lista de enlaces relacionados, entre otros elementos. El objetivo de utilizar **<footer>** es marcar específicamente la sección de pie de página de una página para que los buscadores y navegadores puedan entender mejor el contenido de la página.

Al igual que la etiqueta **<header>**, **<footer>** no se limita a ser utilizada solo al final de

4 ELEMENTOS

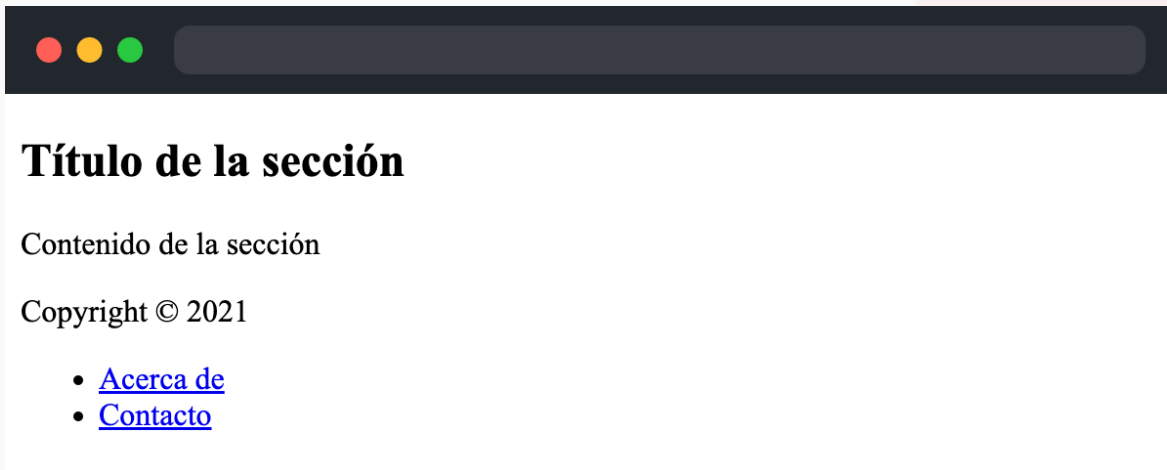
la página, sino que puede ser utilizada varias veces en la misma página, para secciones y artículos dentro de la página, permitiendo así tener una estructura jerárquica y organizada.

Aquí hay un ejemplo de cómo se utiliza la etiqueta **<footer>** en un documento:

A code editor window with a dark background and a light blue border. It contains the following HTML code:

```
<body>
  <main>
    <section>
      <h2>Título de la sección</h2>
      <p>Contenido de la sección</p>
    </section>
  </main>
  <footer>
    <p>Copyright © 2021</p>
    <ul>
      <li><a href="#">Acerca de</a></li>
      <li><a href="#">Contacto</a></li>
    </ul>
  </footer>
</body>
```

El código se renderiza de esta manera al abrirlo en un navegador.



En este ejemplo, se puede ver cómo se utiliza la etiqueta **<footer>** para contener el contenido del pie de página, que incluye una etiqueta **<p>** con información de Copyright y una lista de enlaces relacionados. Es importante mencionar que el contenido dentro del **<footer>** debe estar relacionado con el contenido de la página y no debe ser utilizado como un contenedor para elementos que no son relevantes al contexto.

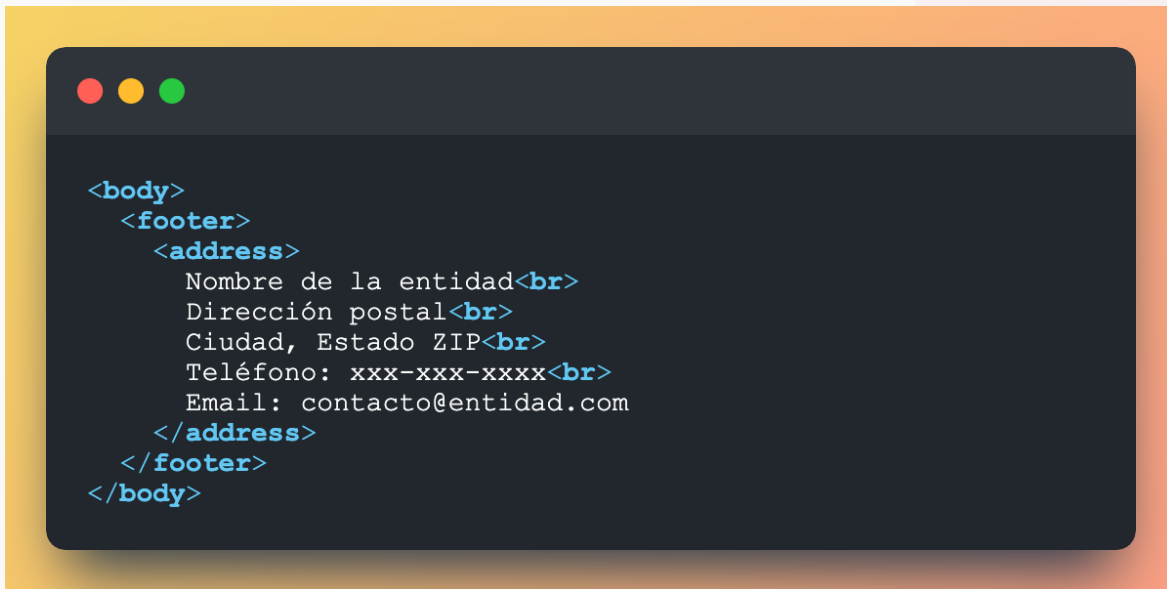
4.3.10 <address>

La etiqueta **<address>** en HTML es una etiqueta semántica que se utiliza para identificar la información de contacto de una entidad en una página web. Esta etiqueta se utiliza para contener información de contacto como nombre, dirección postal, teléfono, correo electrónico, entre otros. El objetivo de utilizar **<address>** es marcar específicamente la información de contacto de una entidad para que los buscadores y navegadores puedan entender mejor el contenido de la página.

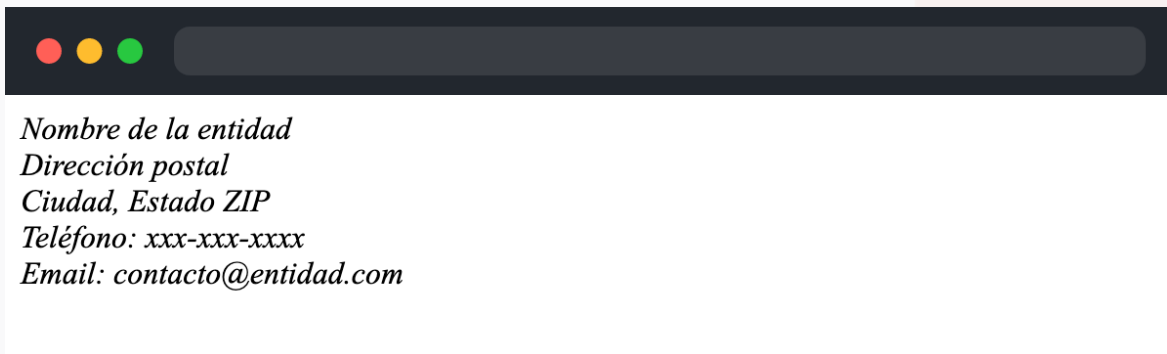
La etiqueta **<address>** se utiliza generalmente dentro del elemento **<footer>** de la página web, ya que es donde se suele encontrar la información de contacto.

Aquí hay un ejemplo de cómo se utiliza la etiqueta **<address>** en un documento:

4 ELEMENTOS



El código se renderiza de esta manera al abrirlo en un navegador.



En este ejemplo, se puede ver como se utiliza la etiqueta **<address>** para contener la información de contacto de una entidad, como el nombre, dirección, teléfono y correo electrónico. Es importante mencionar que no se requiere de atributos para la etiqueta **<address>** y su contenido debe estar en un formato legible para el usuario y no solo para los bots o buscadores.

4.4 Agrupaciones

Las etiquetas de agrupación de contenido en HTML (Grouping content) permiten agrupar elementos dentro de una etiqueta específica con el objetivo de dar un significado semántico y aplicar estilos y comportamientos a ese conjunto de elementos a través de hojas de estilo CSS.

Una etiqueta **<div>** es un ejemplo de contenedor genérico para agrupar contenido. Sin embargo, HTML5 introdujo varias etiquetas específicas para agrupar contenido con un significado semántico usando secciones. Las secciones en HTML son una forma de dividir el contenido de un documento HTML en bloques lógicos y significativos, cada uno con un significado específico. Algunos ejemplos de etiquetas comunes utilizadas para seccionar contenido son **<header>**, **<nav>**, **<main>**, **<article>**, **<aside>**, **<footer>**, entre otras.

La diferencia radica en que la agrupación de contenido se refiere principalmente a cómo se organiza y se asignan estilos a los elementos mientras que las secciones se refieren a cómo se divide el contenido en bloques lógicos. Ambas funciones son complementarias y se utilizan juntas para estructurar y presentar adecuadamente el contenido de un documento HTML.

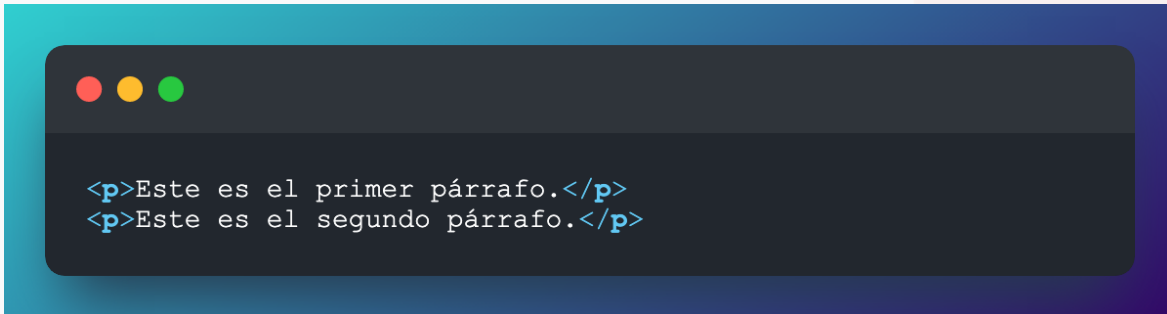
Estas son las etiquetas utilizadas para contenido agrupado a detalle:

4.4.1 <p>

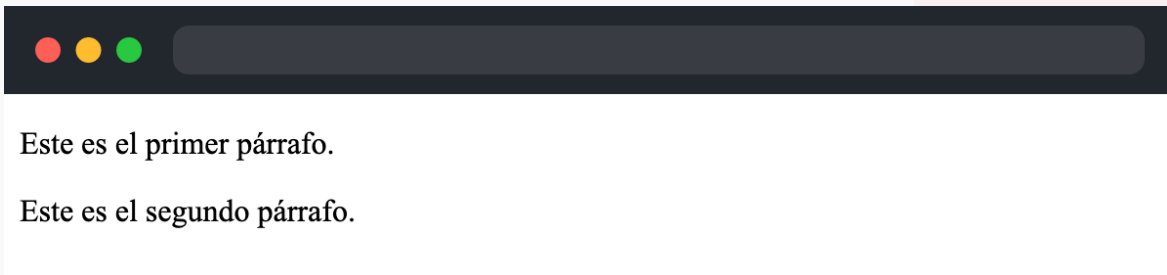
La etiqueta **<p>** en HTML es utilizada para indicar un párrafo de texto dentro de un documento. Cada vez que se utiliza una etiqueta **<p>**, se agrega un margen automático alrededor del texto contenido dentro de ella. El texto dentro de un párrafo es mostrado con una separación entre cada uno.

Aquí hay un ejemplo de cómo se utiliza la etiqueta **<p>** en un documento:

4 ELEMENTOS



El código se renderiza de esta manera al abrirlo en un navegador.



En este ejemplo, se puede ver cómo se utilizan dos etiquetas `<p>` para contener dos párrafos de texto diferentes. El resultado es que cada párrafo tendrá margen automáticamente y estará separado uno del otro.

4.4.2 `<hr>`

La etiqueta `<hr>` en HTML es utilizada para insertar una línea horizontal en un documento HTML. Esta línea se utiliza comúnmente para crear un separador visual entre secciones de una página web o para dividir un bloque de contenido en secciones. La etiqueta `<hr>` no tiene contenido, por lo que no es necesario cerrarla.

Aquí hay un ejemplo de cómo se utiliza la etiqueta `<hr>` en un documento:

```
<h1>Título de la página</h1>
<p>Contenido de la sección 1</p>
<hr>
<h2>Título de la sección 2</h2>
<p>Contenido de la sección 2</p>
```

El código se renderiza de esta manera al abrirlo en un navegador.

Título de la página

Contenido de la sección 1

Título de la sección 2

Contenido de la sección 2

En este ejemplo se puede ver que la etiqueta `<hr>` se utiliza para insertar una línea horizontal entre las dos secciones de contenido de la página. También es posible personalizar el diseño de la línea con CSS.

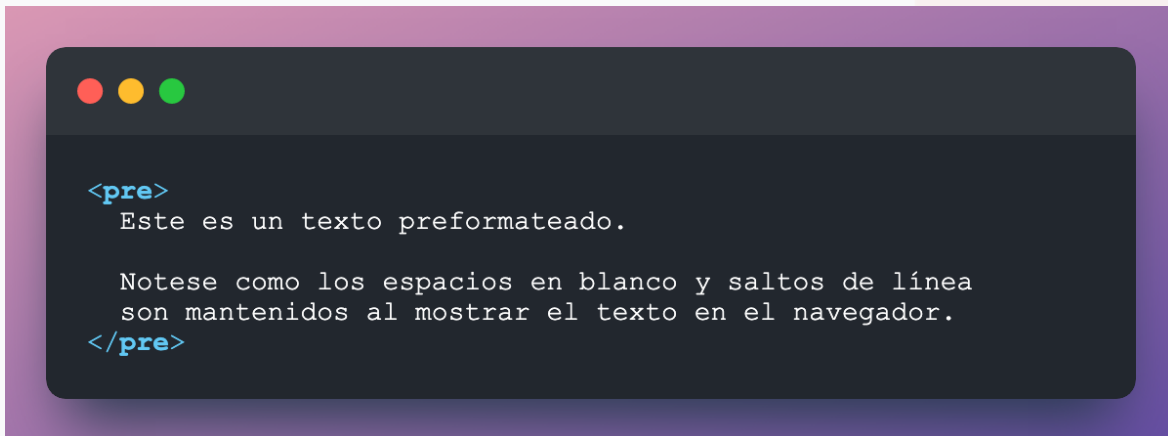
4.4.3 `<pre>`

La etiqueta `<pre>` en HTML es utilizada para indicar que el contenido dentro de ella debe ser preformateado. Esto significa que el navegador debe mostrar el contenido

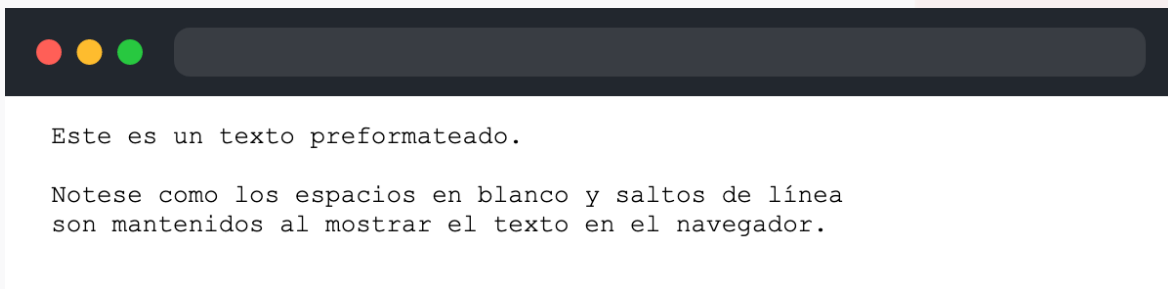
4 ELEMENTOS

tal y como es escrito, incluyendo espacios en blanco y saltos de línea. Esto es útil cuando se desea mostrar contenido que es sensible a la forma en que se ve como código fuente o texto con formato fijo.

Aquí hay un ejemplo de cómo se utiliza la etiqueta **<pre>** en un documento:



El código se renderiza de esta manera al abrirlo en un navegador.

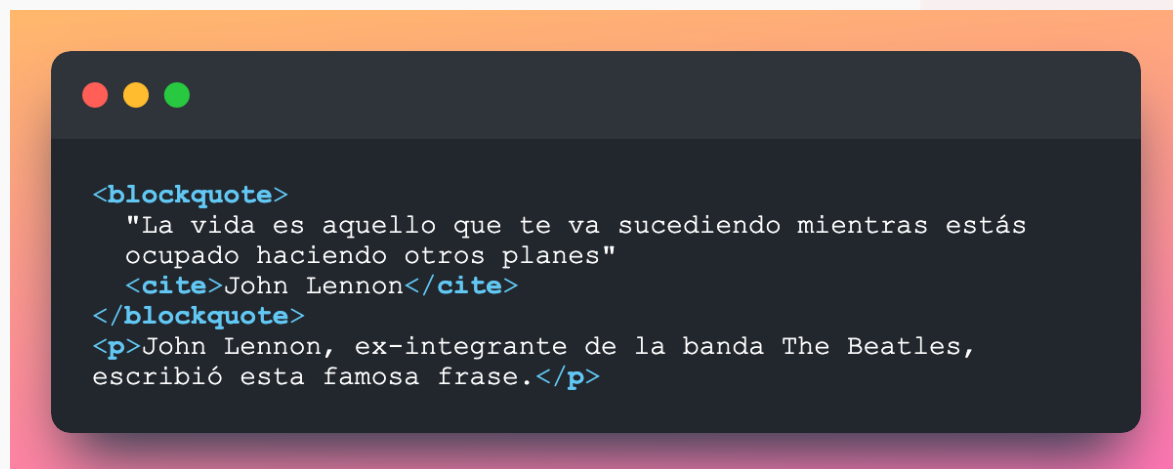


En este ejemplo, se puede ver como el texto dentro de la etiqueta **<pre>** es mostrado con los espacios en blanco y saltos de línea tal y como son escritos, en lugar de ser ignorados o eliminados como ocurre normalmente. Esto es útil para mostrar código fuente o texto con un formato específico.

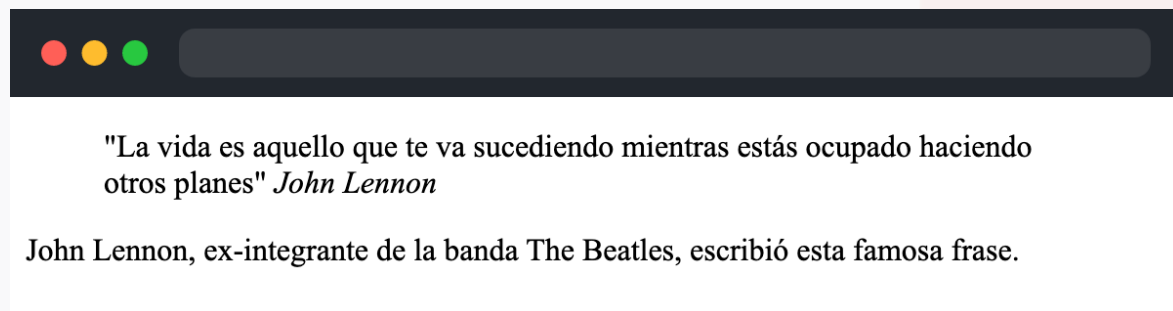
4.4.4 <blockquote>

La etiqueta **<blockquote>** en HTML es utilizada para indicar una cita de bloque dentro de un documento. El contenido dentro de una etiqueta **<blockquote>** se suele mostrar con un formato especial, como una sangría o un tipo de letra diferente, para indicar que se trata de una cita.

Aquí hay un ejemplo de cómo se utiliza la etiqueta **<blockquote>** en un documento:



El código se renderiza de esta manera al abrirlo en un navegador.



En este ejemplo, se puede ver como la etiqueta **<blockquote>** se utiliza para contener la cita y con **<cite>** se especifica la fuente de la cita. Esto es útil para mostrar citas dentro de una página web y darle un formato especial para distinguirlas del

4 ELEMENTOS

texto normal.

4.4.5

La etiqueta **** en HTML es utilizada para crear una lista ordenada. Una lista ordenada es una lista de elementos donde cada elemento está precedido por un número o letra indicando el orden. El navegador se encarga automáticamente de numerar los elementos de la lista.

Aquí hay un ejemplo de cómo se utiliza la etiqueta **** en un documento:

A screenshot of a code editor window with a dark background and a teal border. The code is written in a light blue font and shows the HTML structure for an ordered list with three items.

```
<ol>
  <li>Primer elemento</li>
  <li>Segundo elemento</li>
  <li>Tercer elemento</li>
</ol>
```

El código se renderiza de esta manera al abrirlo en un navegador.



En este ejemplo se puede ver como se utiliza la etiqueta **** para contener una lista ordenada, y como cada elemento de la lista es contenido dentro de una etiqueta **** (list item). El navegador se encarga de numerar automáticamente cada elemento de

4 ELEMENTOS

la lista (1. Primer elemento, 2. Segundo elemento, 3. Tercer elemento) También se pueden personalizar los números o letras que se usan para numerar los elementos utilizando css.

4.4.6

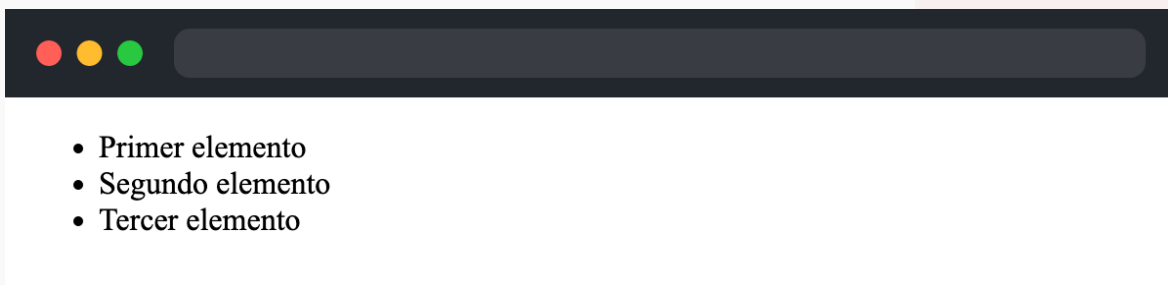
La etiqueta **** en HTML es utilizada para crear una lista desordenada. Una lista desordenada es una lista de elementos donde cada elemento es precedido por un símbolo de viñeta (como un círculo o un cuadrado) y no tiene un orden específico.

Aquí hay un ejemplo de cómo se utiliza la etiqueta **** en un documento:

A screenshot of a code editor window with a dark background and a blue-to-purple gradient border. The code is written in a light blue font and shows the HTML structure for an unordered list with three items.

```
<ul>
  <li>Primer elemento</li>
  <li>Segundo elemento</li>
  <li>Tercer elemento</li>
</ul>
```

El código se renderiza de esta manera al abrirlo en un navegador.



En este ejemplo se puede ver como se utiliza la etiqueta **** para contener una lista desordenada, y como cada elemento de la lista es contenido dentro de una

4 ELEMENTOS

etiqueta `<li>` (list item). El navegador se encarga de poner un símbolo de viñeta automáticamente antes de cada elemento de la lista.

Al igual que en la etiqueta `<ol>` es posible personalizar los símbolos de viñeta utilizando css.

4.4.7 `<menu>`

La etiqueta `<menu>` en HTML es utilizada para crear un menú en una página web. Un menú es un conjunto de elementos que proporcionan navegación o acceso a otras partes del sitio o a funciones específicas.

Sin embargo, es importante mencionar no se recomienda el uso de la etiqueta `<menu>` en proyectos profesionales. Es recomendable utilizar en su lugar las etiquetas `<ul>` o `<ol>` para crear una lista de elementos de menú, y darle estilos mediante css para que se vea como un menú.

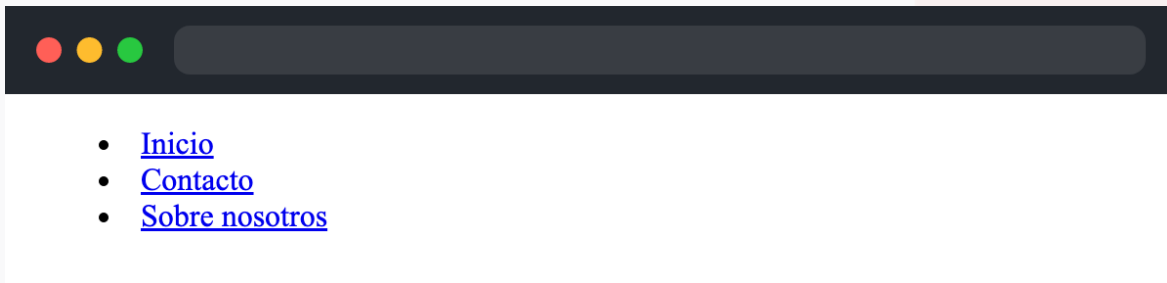
Sin embargo si deseas usarla, un ejemplo de como se podría usar la etiqueta `<menu>` en un documento es:

A screenshot of a code editor window with a dark background and a purple border. The code is written in a light blue font and shows the HTML structure for a menu using the <menu> tag. The code includes three list items, each with a link to a different page: 'Inicio', 'Contacto', and 'Sobre nosotros'.

```
<menu>
  <li><a href="inicio.html">Inicio</a></li>
  <li><a href="contacto.html">Contacto</a></li>
  <li><a href="sobre_nosotros.html">Sobre nosotros</a></li>
</menu>
```

El código se renderiza de esta manera al abrirlo en un navegador.

4 ELEMENTOS

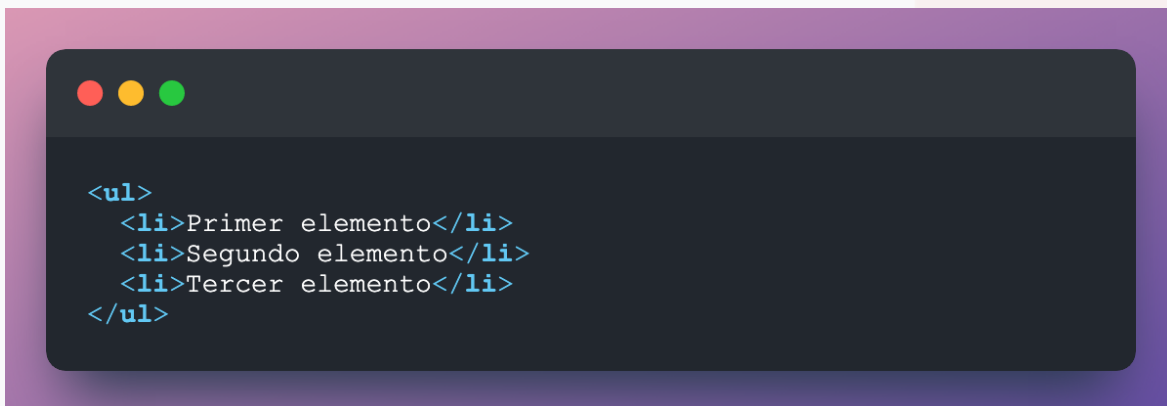


En este ejemplo se puede ver como se utiliza la etiqueta **<menu>** para contener un menú, y como cada elemento del menú es contenido dentro de una etiqueta **** (list item) y estos elementos están contenidos en una etiqueta **<a>** (anchor) para crear un enlace.

4.4.8

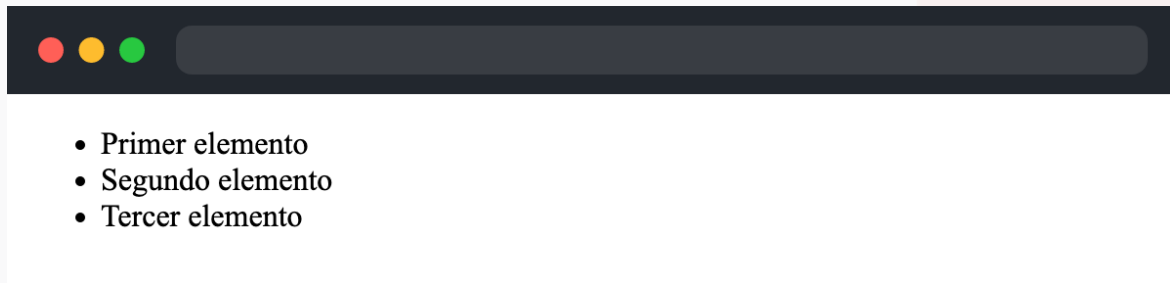
La etiqueta **** en HTML es utilizada para indicar un elemento dentro de una lista. Esta etiqueta se utiliza tanto para listas ordenadas (****) como para listas desordenadas (****) y para listas de menú (**<menu>**). Cada elemento de una lista debe estar contenido dentro de una etiqueta ****.

Aquí hay un ejemplo de cómo se utiliza la etiqueta **** en una lista desordenada:



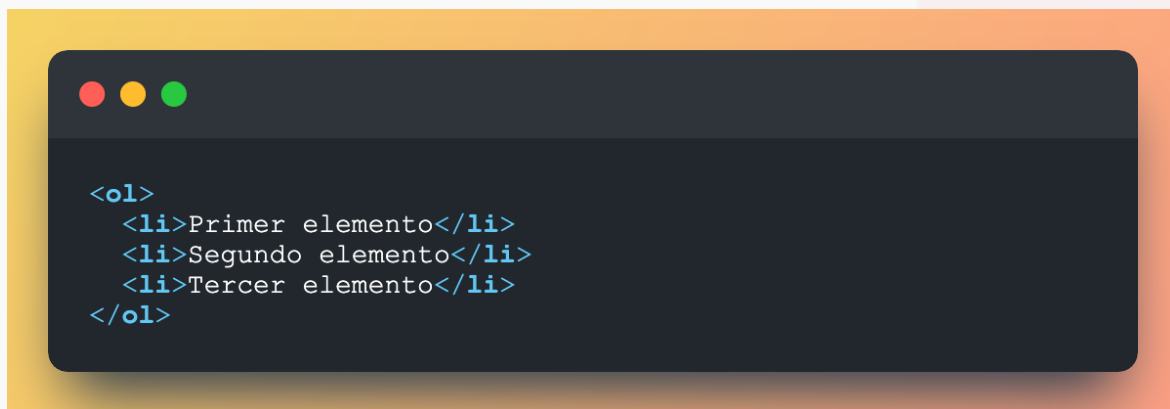
El código se renderiza de esta manera al abrirlo en un navegador.

4 ELEMENTOS

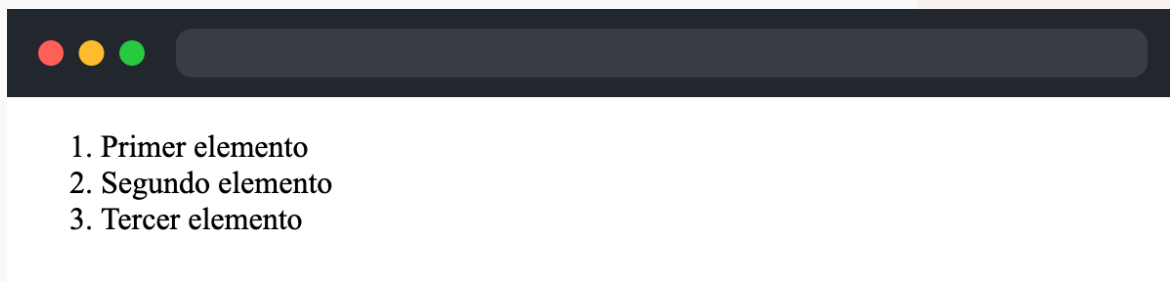


En este ejemplo se puede ver como se utiliza la etiqueta `<li>` para contener cada elemento de una lista desordenada. El navegador se encarga de poner un símbolo de viñeta automáticamente antes de cada elemento de la lista

Y un ejemplo de como se podría utilizar en una lista ordenada:



El código se renderiza de esta manera al abrirlo en un navegador.



En este ejemplo se puede ver como se utiliza la etiqueta `<li>` para contener cada elemento de una lista ordenada. El navegador se encarga de numerar automática-

4 ELEMENTOS

mente cada elemento de la lista.

La etiqueta `<li>` puede contener cualquier tipo de contenido, como texto, imágenes o enlaces.

4.4.9 `<dl>`

La etiqueta `<dl>` en HTML es utilizada para crear una lista de términos y definiciones. Cada término es representado por una etiqueta `<dt>` y cada definición es representada por una etiqueta `<dd>`.

Un ejemplo práctico podría ser un diccionario en línea. El siguiente código crea una lista de términos y definiciones para las palabras “perro” y “gato” :

```
<dl>
  <dt>Perro</dt>
  <dd>Animal de compañía de la familia de los cánidos.</dd>
  <dt>Gato</dt>
  <dd>Animales de compañía de la familia de los felinos.</dd>
</dl>
```

El código se renderiza de esta manera al abrirlo en un navegador.

Perro
Animal de compañía de la familia de los cánidos.

Gato
Animales de compañía de la familia de los felinos.

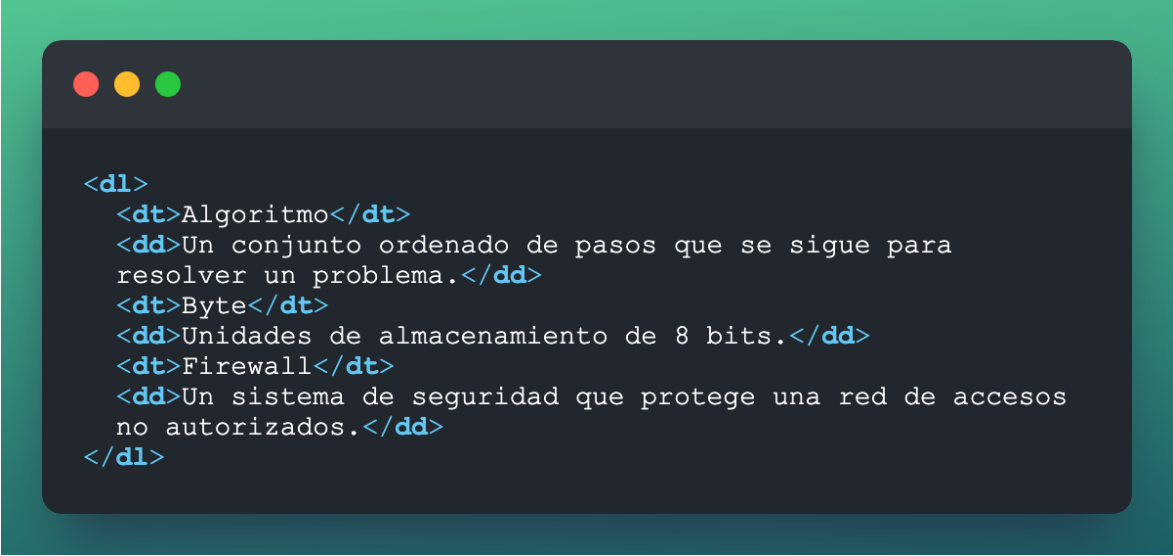
4 ELEMENTOS

Es importante recordar que esta etiqueta es para darle formato y significado semántico al contenido específico, en este caso la “lista de términos y definiciones”, no es para darle estilos o apariencia visual en específico ya que esa es la función de CSS.

4.4.10 <dt>

La etiqueta **<dt>** en HTML es utilizada en conjunto con la etiqueta **<dl>** para representar los términos en una lista de términos y definiciones. El contenido dentro de una etiqueta **<dt>** se considera el término que se va a definir, y estará ubicado inmediatamente antes de su correspondiente definición, marcada con una etiqueta **<dd>**.

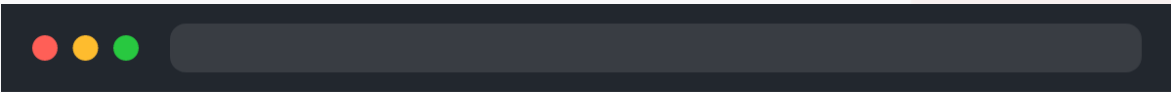
Un ejemplo práctico podría ser una página web que contiene un glosario de términos relacionados con un tema específico, como por ejemplo la informática. El siguiente código muestra cómo se podría utilizar las etiquetas **<dl>**, **<dt>** y **<dd>** para crear un glosario de términos informáticos:

A screenshot of a code editor window with a dark background and light-colored text. The code is HTML markup for a glossary. It starts with an opening <dl> tag, followed by a series of <dt> and <dd> pairs. The terms are 'Algoritmo', 'Byte', and 'Firewall'. Each term is followed by its definition. The code ends with a closing </dl> tag. The editor has three colored window control buttons (red, yellow, green) in the top left corner.

```
<dl>
  <dt>Algoritmo</dt>
  <dd>Un conjunto ordenado de pasos que se sigue para
  resolver un problema.</dd>
  <dt>Byte</dt>
  <dd>Unidades de almacenamiento de 8 bits.</dd>
  <dt>Firewall</dt>
  <dd>Un sistema de seguridad que protege una red de accesos
  no autorizados.</dd>
</dl>
```

El código se renderiza de esta manera al abrirlo en un navegador.

4 ELEMENTOS



Algoritmo	Un conjunto ordenado de pasos que se sigue para resolver un problema.
Byte	Unidades de almacenamiento de 8 bits.
Firewall	Un sistema de seguridad que protege una red de accesos no autorizados.

Es importante recordar que esta etiqueta es para darle formato y significado semántico al contenido específico, en este caso el “término” dentro de una lista de términos y definiciones, no es para darle estilos o apariencia visual en específico.

4.4.11 <dd>

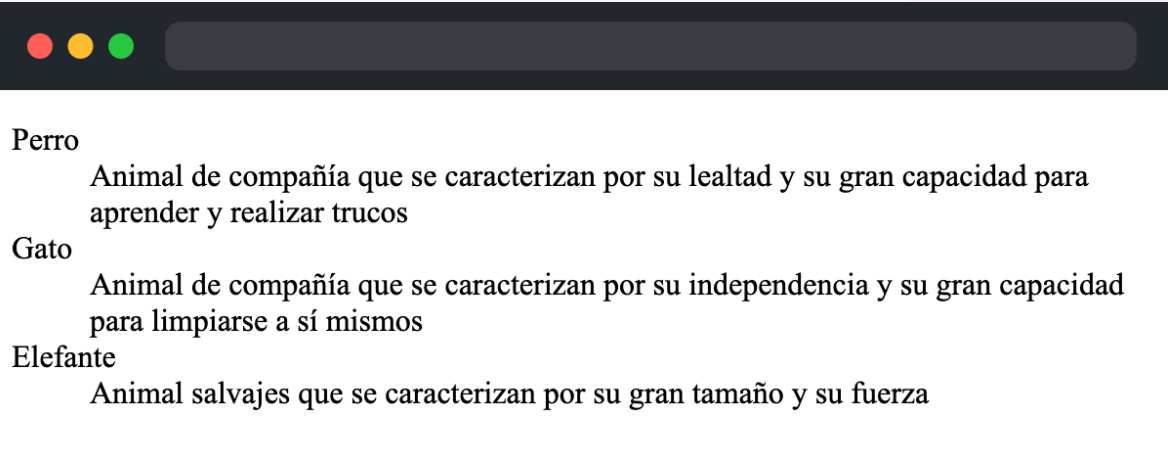
La etiqueta **<dd>** en HTML es utilizada para describir un elemento dentro de una lista de definiciones. Es utilizado junto con la etiqueta **<dt>**, que describe el término o la frase que se está definiendo.

Un ejemplo práctico de su uso podría ser el siguiente:

4 ELEMENTOS

```
<dl>
  <dt>Perro</dt>
  <dd>Animal de compañía que se caracterizan por su lealtad y su gran capacidad para aprender y realizar trucos</dd>
  <dt>Gato</dt>
  <dd>Animal de compañía que se caracterizan por su independencia y su gran capacidad para limpiarse a sí mismos</dd>
  <dt>Elefante</dt>
  <dd>Animal salvajes que se caracterizan por su gran tamaño y su fuerza</dd>
</dl>
```

El código se renderiza de esta manera al abrirlo en un navegador.



Perro
Animal de compañía que se caracterizan por su lealtad y su gran capacidad para aprender y realizar trucos

Gato
Animal de compañía que se caracterizan por su independencia y su gran capacidad para limpiarse a sí mismos

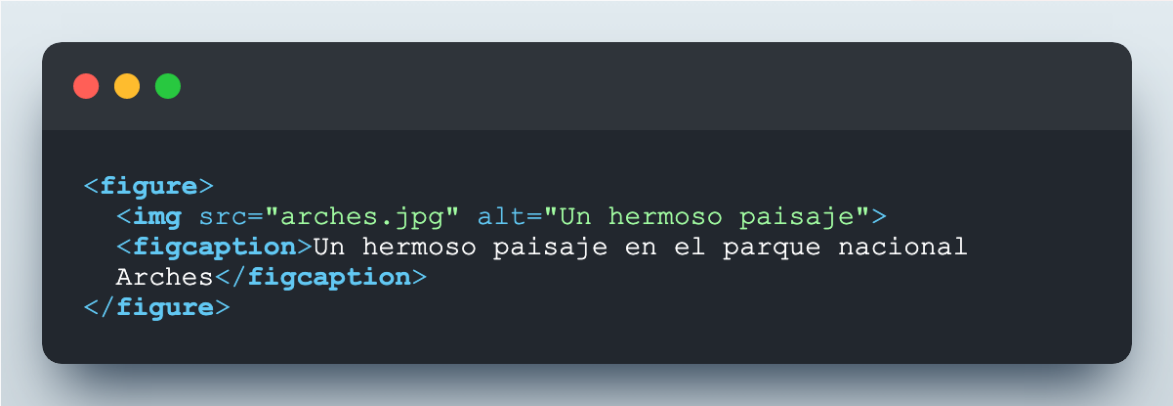
Elefante
Animal salvajes que se caracterizan por su gran tamaño y su fuerza

En este ejemplo, se están describiendo los términos “Perro”, “Gato”, y “Elefante” utilizando las etiquetas **<dt>** y **<dd>**. El resultado sería una lista de definiciones con los términos y sus respectivas descripciones.

4.4.12 <figure>

La etiqueta **<figure>** en HTML es utilizada para contener elementos multimedia como imágenes, gráficos, videos, e incluso código de ejemplo, en un documento HTML. La idea de esta etiqueta es que el contenido dentro de ella sea considerado una “cita” o una “figura” dentro del contexto de la página. La etiqueta **<figure>** también se utiliza para proporcionar una descripción alternativa para el contenido multimedia, a través de la etiqueta **<figcaption>**

Un ejemplo práctico podría ser el siguiente:

A code editor window with a dark background and three colored window control buttons (red, yellow, green) in the top left corner. It contains the following HTML code:

```
<figure>
  
  <figcaption>Un hermoso paisaje en el parque nacional
  Arches</figcaption>
</figure>
```

El código se renderiza de esta manera al abrirlo en un navegador.



Un hermoso paisaje en el parque nacional Arches

En este ejemplo se está usando la etiqueta **<figure>** para contener una imagen y su descripción alternativa. Es importante mencionar que **<figure>** no tiene un impacto visual, sino que está pensada para que sea utilizada como una forma de contener y etiquetar contenido dentro de una página HTML.

4.4.13 <figcaption>

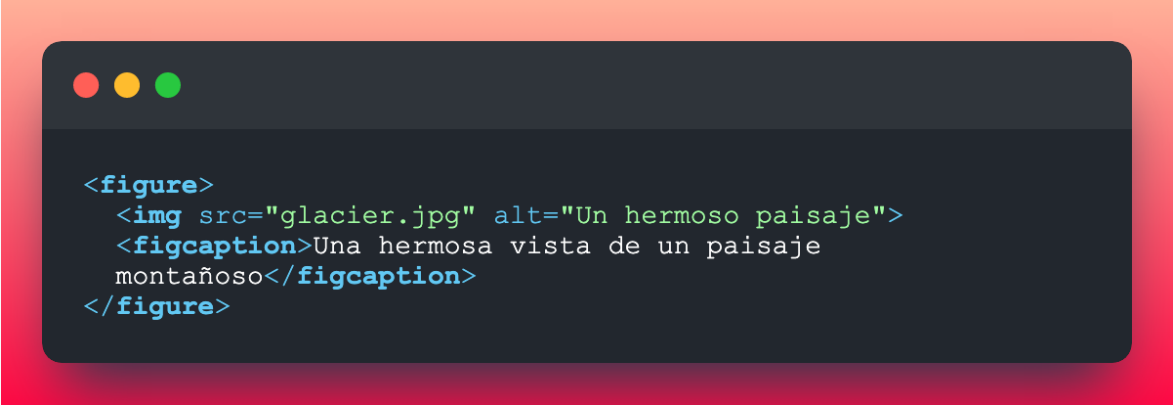
La etiqueta **<figcaption>** en HTML es utilizada para proporcionar una descripción o una leyenda para un elemento multimedia contenido dentro de una etiqueta **<figure>**.

El contenido dentro de una etiqueta **<figcaption>** se considera un subtítulo o una descripción de la figura contenida dentro de la etiqueta **<figure>**. Es un elemento

4 ELEMENTOS

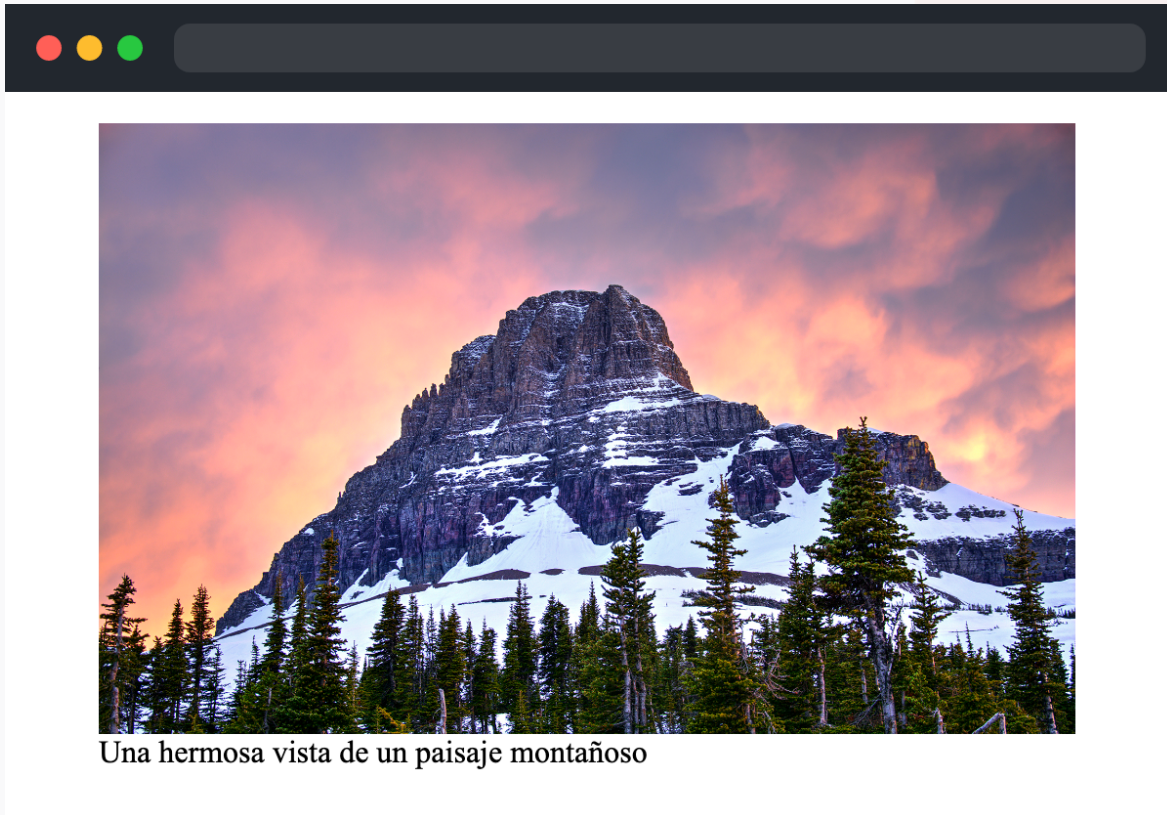
opcional, es decir no es necesario usarlo, pero si se usa es una buena practica para proporcionar información sobre el contenido de la figura.

Un ejemplo práctico podría ser el siguiente:



```
<figure>
  
  <figcaption>Una hermosa vista de un paisaje
montañoso</figcaption>
</figure>
```

El código se renderiza de esta manera al abrirlo en un navegador.



En este ejemplo, se está usando la etiqueta **<figcaption>** para proporcionar una descripción para una imagen contenida dentro de una etiqueta **<figure>**. La descripción de la imagen es “Una hermosa vista de un paisaje montañoso”, y se muestra en relación a la imagen, esto ayudaría a entender mejor el contexto de la imagen y a personas que utilizan tecnologías de lectura.

4.4.14 **<main>**

La etiqueta **<main>** en HTML se utiliza para indicar el contenido principal de un documento HTML. El contenido dentro de esta etiqueta debería ser único al resto del contenido en la página, y debería ser relevante para el contexto de la página.

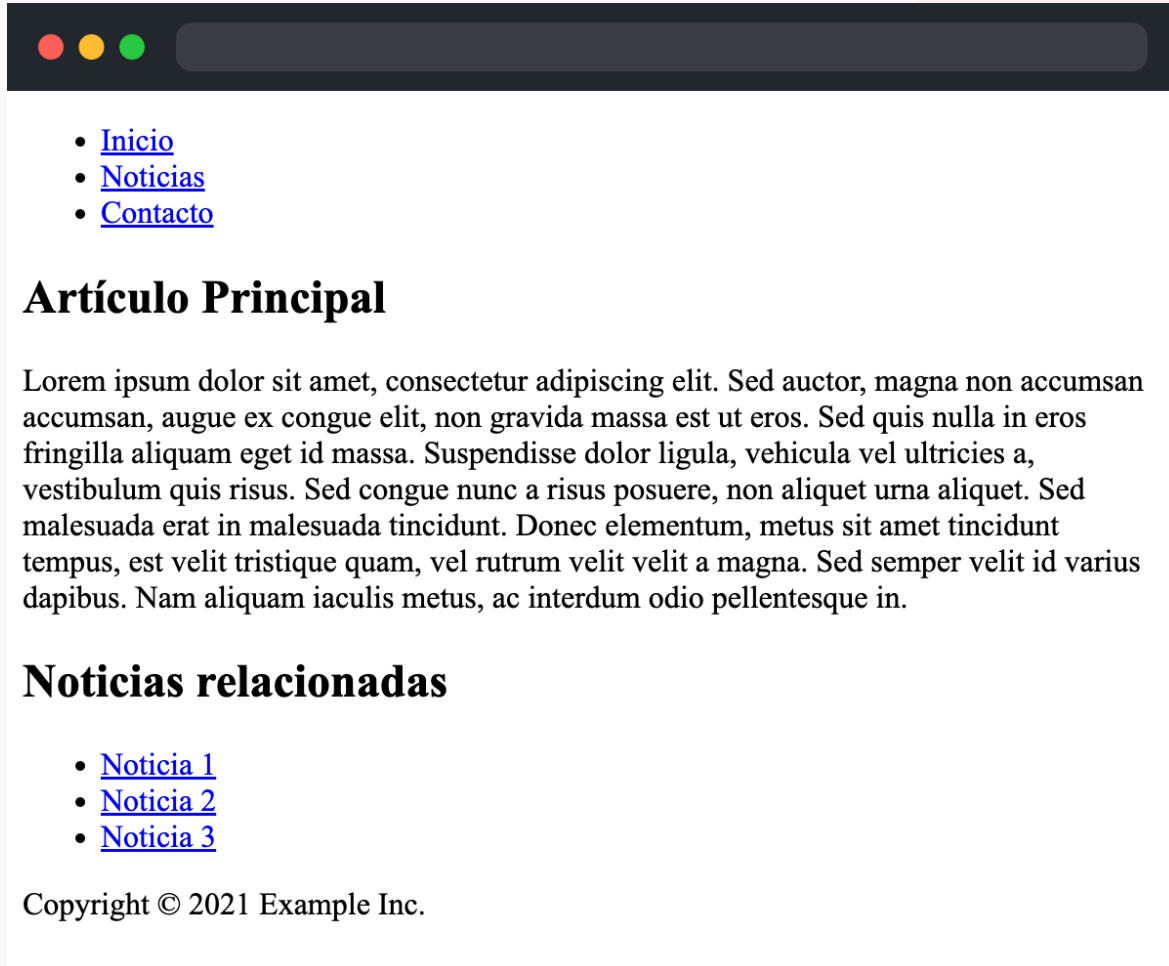
Por ejemplo, en una página de noticias, el contenido principal podría ser el artículo

4 ELEMENTOS

principal de la página, mientras que el contenido adicional como el menú de navegación, los anuncios y los enlaces relacionados podrían estar fuera de la etiqueta **<main>**.

```
<body>
  <header>
    <nav>
      <ul>
        <li><a href="#">Inicio</a></li>
        <li><a href="#">Noticias</a></li>
        <li><a href="#">Contacto</a></li>
      </ul>
    </nav>
  </header>
  <main>
    <article>
      <h1>Artículo Principal</h1>
      <p>Lorem ipsum dolor sit amet, consectetur adipiscing
elit. Sed auctor, magna non accumsan accumsan, augue ex
congue elit, non gravida massa est ut eros. Sed quis
nulla in eros fringilla aliquam eget id massa.
Suspendisse dolor ligula, vehicula vel ultricies a,
vestibulum quis risus. Sed congue nunc a risus posuere,
non aliquet urna aliquet. Sed malesuada erat in
malesuada tincidunt. Donec elementum, metus sit amet
tincidunt tempus, est velit tristique quam, vel rutrum
velit velit a magna. Sed semper velit id varius
dapibus. Nam aliquam iaculis metus, ac interdum odio
pellentesque in.</p>
    </article>
  </main>
  <aside>
    <h2>Noticias relacionadas</h2>
    <ul>
      <li><a href="#">Noticia 1</a></li>
      <li><a href="#">Noticia 2</a></li>
      <li><a href="#">Noticia 3</a></li>
    </ul>
  </aside>
  <footer>
    <p>Copyright © 2021 Example Inc.</p>
  </footer>
</body>
```

El código se renderiza de esta manera al abrirlo en un navegador.



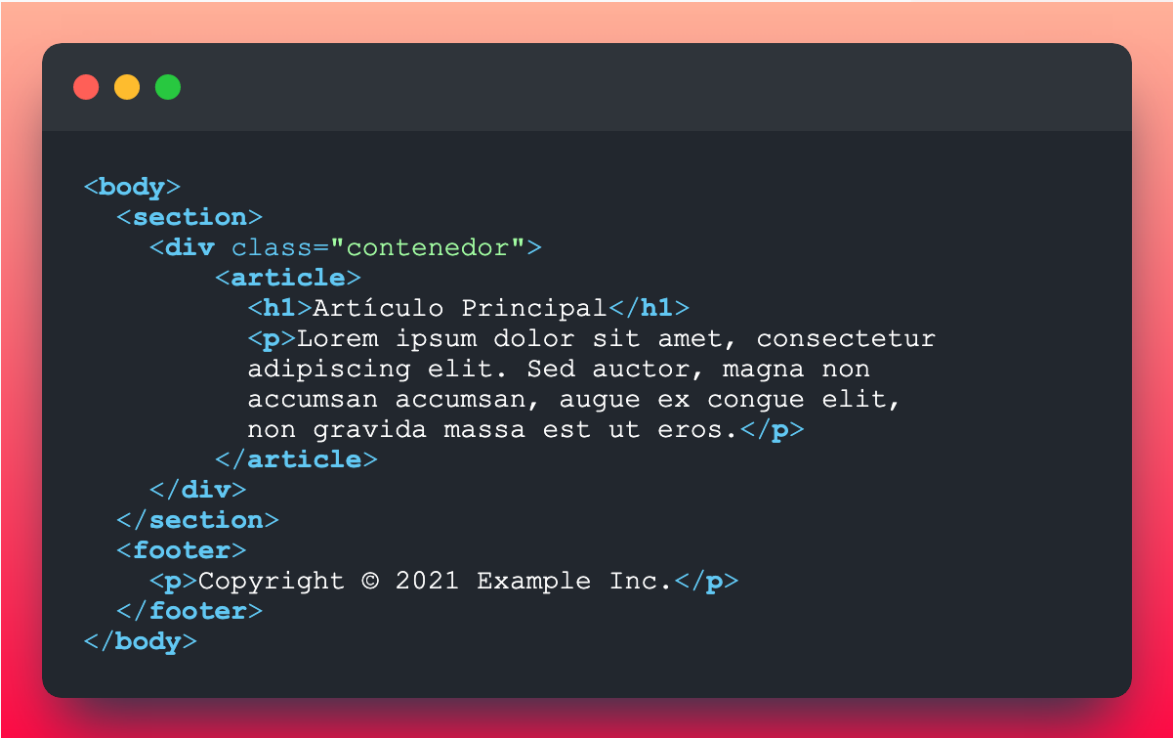
En el ejemplo anterior, el contenido principal de la página es el artículo “Artículo Principal” y se encuentra dentro de la etiqueta **<main>**. El menú de navegación, las noticias relacionadas y el pie de página se encuentran fuera de la etiqueta **<main>**. Esta separación ayuda a los motores de búsqueda y a las herramientas de accesibilidad a entender mejor la estructura de la página y el contenido más importante de la misma.

4.4.15 <div>

La etiqueta **<div>** en HTML es un contenedor genérico que se utiliza para agrupar contenido dentro de un documento HTML. El contenido dentro de una etiqueta **<div>** puede ser cualquier tipo de contenido, como texto, imágenes, enlaces, tablas, listas, etc.

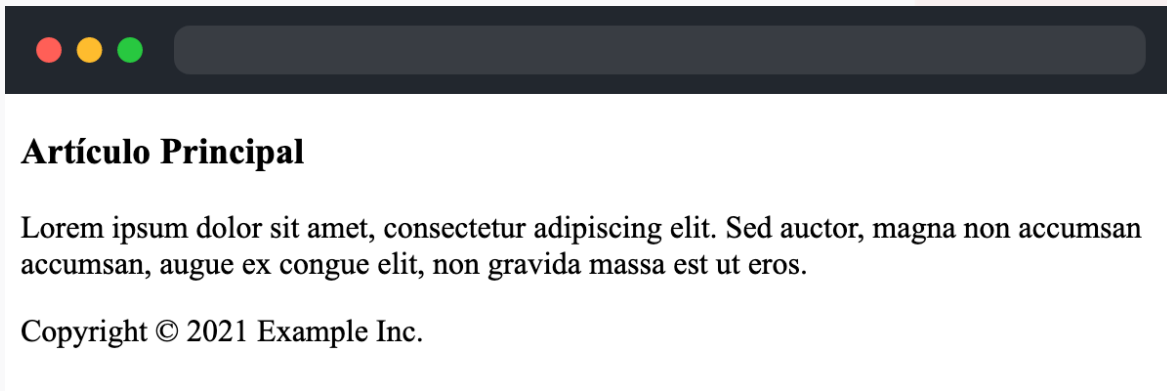
La principal ventaja de utilizar una etiqueta **<div>** es la capacidad de dar estilos y comportamientos al contenido agrupado mediante hojas de estilo CSS, ya que una vez asignado una clase o id específico se puede manipular todo el contenido dentro de esa etiqueta.

Un ejemplo práctico de utilización de **div** podría ser si queremos agrupar los elementos de una sección de una página para darles un estilo específico:

A screenshot of a code editor window with a dark background and a red-to-orange gradient border. The code is written in a light blue font and shows a nested HTML structure. It starts with a <body> tag, followed by a <section> tag. Inside the section is a <div> with a class attribute set to "contenedor". Inside this div is an <article> tag, which contains an <h1> tag with the text "Artículo Principal" and a <p> tag with a paragraph of Lorem Ipsum text. The structure is closed with </div>, </article>, </section>, and </body>. Below the section is a <footer> tag containing a <p> tag with the text "Copyright © 2021 Example Inc.".

```
<body>
  <section>
    <div class="contenedor">
      <article>
        <h1>Artículo Principal</h1>
        <p>Lorem ipsum dolor sit amet, consectetur
        adipiscing elit. Sed auctor, magna non
        accumsan accumsan, augue ex congue elit,
        non gravida massa est ut eros.</p>
      </article>
    </div>
  </section>
  <footer>
    <p>Copyright © 2021 Example Inc.</p>
  </footer>
</body>
```

El código se renderiza de esta manera al abrirlo en un navegador.



En este ejemplo podemos ver como los elementos `article` y `aside` están contenidos dentro de una etiqueta `<div>`, se le ha asignado una clase contenedor para poder darle estilos específicos a través de CSS, como por ejemplo, definir un margen, un ancho o un alineamiento para todos los elementos dentro de esa `div`. Esto permite una mayor flexibilidad y organización en la estructura de la página, y facilita el mantenimiento y actualización del diseño.

4.5 Texto

Las etiquetas de semántica de nivel de texto en HTML (Text-level semantics) dan significado semántico al texto contenido en una página web. Estas etiquetas indican al navegador y a los buscadores el tipo de contenido que se está mostrando. Algunos ejemplos comunes de etiquetas de semántica de nivel de texto incluyen: `<a>`, `<abbr>`, `<b>`, `<i>`, `<mark>`, `<strong>`, `<time>`, `<var>`, `<em>`, `<q>`, entre otros.

Un ejemplo práctico de uso de estas etiquetas podría ser en una página web de un artículo de noticias. El artículo podría contener fragmentos de texto resaltados con `<mark>`, una fecha con `<time>`, una palabra técnica con `<abbr>` y una cita con `<q>`:

4 ELEMENTOS

```
<p>El presidente <mark>anunció</mark> hoy que la situación económica del país mejorará en los próximos meses. La reunión fue realizada ayer <time> el 10 de marzo</time> y se esperan nuevos anuncios en cuanto a la <abbr title="Política económica">política económica</abbr>. El ministro de economía declaró: <q>Estamos trabajando duro para mejorar las condiciones económicas del país</q></p>
```

El código se renderiza de esta manera al abrirlo en un navegador.

El presidente **anunció** hoy que la situación económica del país mejorará en los próximos meses. La reunión fue realizada ayer el 10 de marzo y se esperan nuevos anuncios en cuanto a la política económica. El ministro de economía declaró: “Estamos trabajando duro para mejorar las condiciones económicas del país”

En este ejemplo se utilizan diferentes etiquetas de semántica de nivel de texto para dar un significado específico al contenido, permitiendo una mejor comprensión y estructuración del contenido. Estas etiquetas ayudan a los buscadores a entender el contenido y mejorar la accesibilidad para las personas con discapacidades y a los programas de lectura de pantalla.

Estas son las etiquetas utilizadas para texto semántico a detalle:

4.5.1 <a>

La etiqueta **<a>** en HTML es utilizada para crear un enlace hacia otro documento o recurso. El atributo “href” se utiliza para especificar la ubicación del enlace. Por

4 ELEMENTOS

ejemplo:

```
<a href="https://www.ejemplo.com">Este es un enlace</a>
```

El código se renderiza de esta manera al abrirlo en un navegador.

[Este es un enlace](https://www.ejemplo.com)

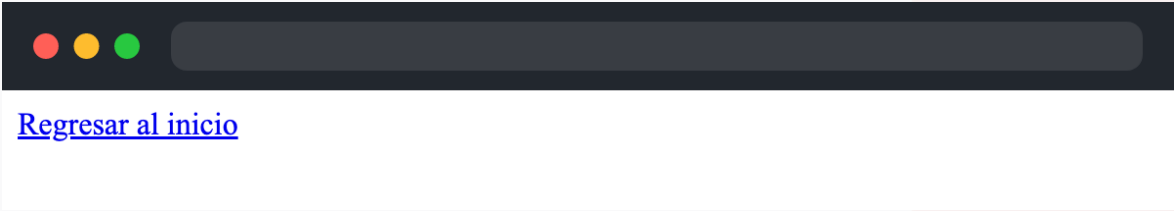
En este ejemplo, cuando un usuario hace clic en la palabra “enlace”, será redirigido a la página web “<https://www.ejemplo.com>” .

También se puede utilizar para crear un enlace a un lugar específico dentro del mismo documento, utilizando el atributo “name” o “id” en la sección del documento a la cual se desea enlazar, y luego haciendo referencia a ese atributo en el href. Por ejemplo:

```
<div id="top"></div>
<!-- contenido largo -->
<a href="#top">Regresar al inicio</a>
```

El código se renderiza de esta manera al abrirlo en un navegador.

4 ELEMENTOS



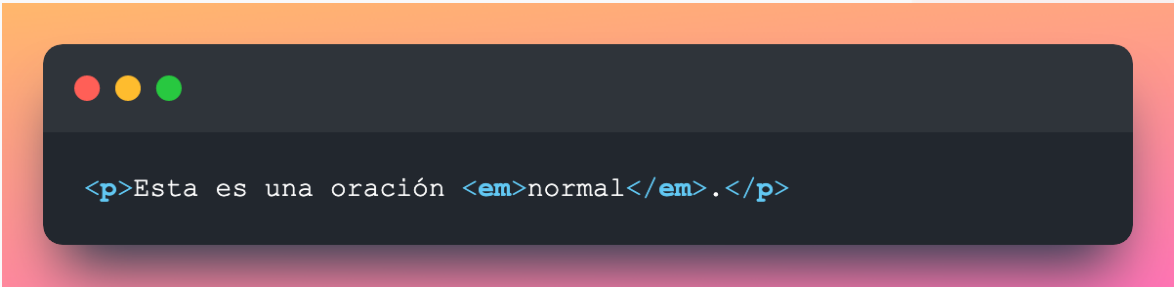
[Regresar al inicio](#)

En este ejemplo, cuando un usuario hace clic en la palabra “Regresar al inicio” , será llevado al inicio del documento, ya que el atributo “href” está apuntando a la sección del documento que tiene el atributo “id” con el valor “top” .

4.5.2

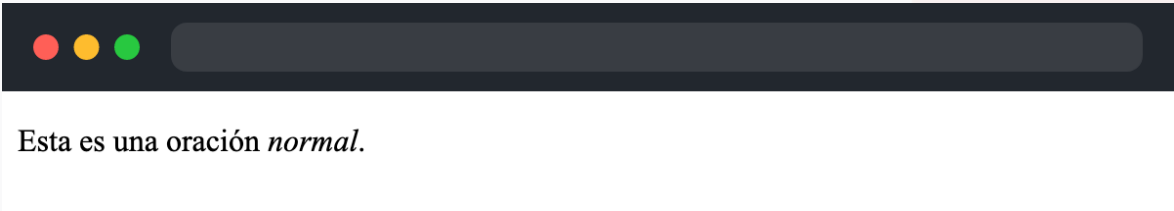
La etiqueta en HTML se utiliza para enfatizar el contenido. El navegador suele representar el contenido que se encuentra dentro de esta etiqueta con cursiva.

Por ejemplo:



```
<p>Esta es una oración <em>normal</em>.</p>
```

El código se renderiza de esta manera al abrirlo en un navegador.



Esta es una oración *normal*.

En este caso, el navegador mostraría “Esta es una oración normal” con la palabra “normal” en cursiva.

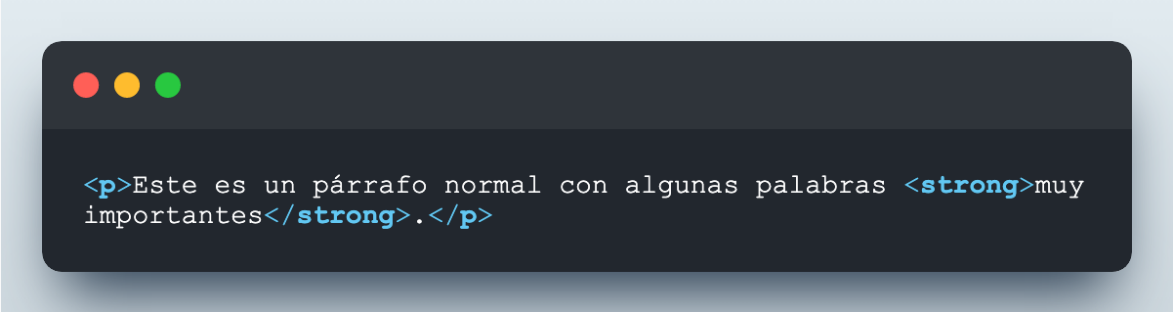
4 ELEMENTOS

Es importante notar que `<em>` y `'` son etiquetas semánticas, es decir, ayudan a describir el significado del contenido en lugar de describir su apariencia.

4.5.3 `<strong>`

La etiqueta `<strong>` en HTML se utiliza para indicar contenido importante dentro de un párrafo. A diferencia de la etiqueta `<em>`, la etiqueta `<strong>` indica que el contenido es particularmente importante y debe ser dado un peso adicional en la presentación. El navegador suele representar el contenido dentro de esta etiqueta con negrita.

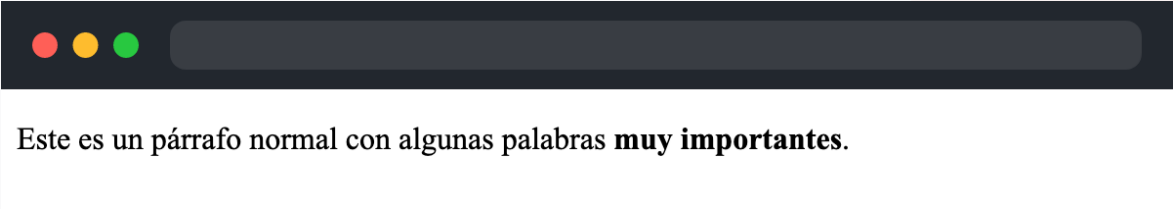
Por ejemplo:

A screenshot of a code editor window with a dark background and three colored window control buttons (red, yellow, green) in the top-left corner. The code inside the editor is:

```
<p>Este es un párrafo normal con algunas palabras <strong>muy importantes</strong>.</p>
```

```
<p>Este es un párrafo normal con algunas palabras <strong>muy importantes</strong>.</p>
```

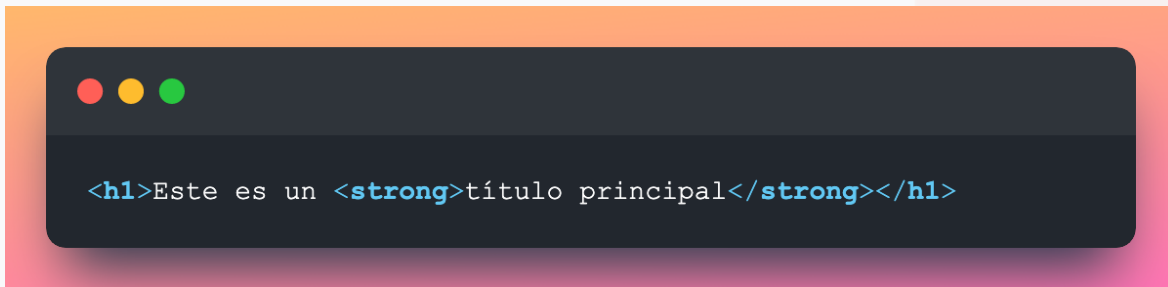
El código se renderiza de esta manera al abrirlo en un navegador.

A screenshot of a web browser window. The address bar is empty. The main content area displays the text: "Este es un párrafo normal con algunas palabras **muy importantes**." The words "muy importantes" are rendered in a bold font.

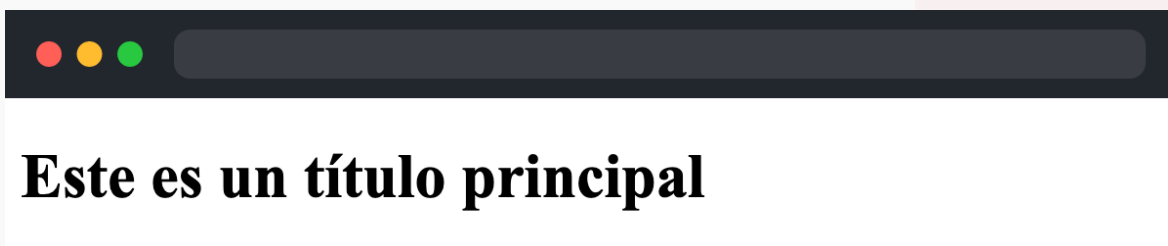
Este es un párrafo normal con algunas palabras **muy importantes**.

En este caso, el navegador mostraría “Este es un párrafo normal con algunas palabras muy importantes” con la palabra “muy importantes” en negrita.

También se puede utilizar para indicar una parte específica de un documento es particularmente importante.



El código se renderiza de esta manera al abrirlo en un navegador.



En este caso, el navegador mostraría “Este es un título principal” con la palabra “título principal” en negrita.

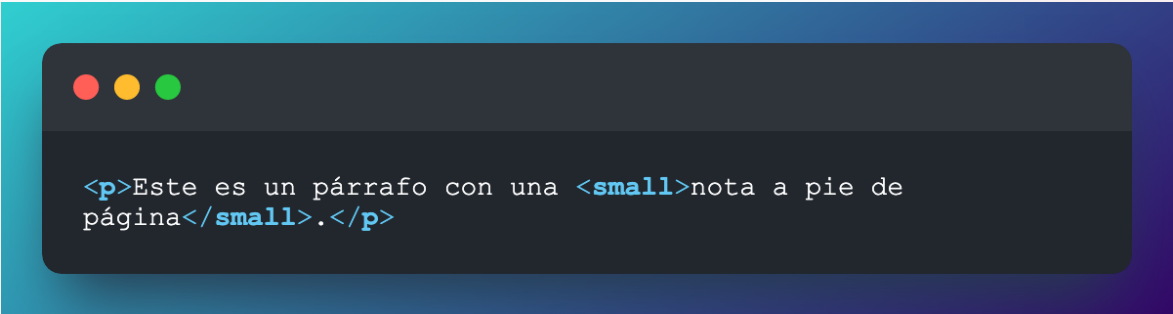
Al igual que `<em>`, `<strong>` es una etiqueta semántica que ayuda a describir el significado del contenido en lugar de describir su apariencia.

4.5.4 `<small>`

La etiqueta `<small>` en HTML se utiliza para indicar contenido menor o de menor importancia dentro de un párrafo. El navegador suele representar el contenido dentro de esta etiqueta con tamaño de letra reducido.

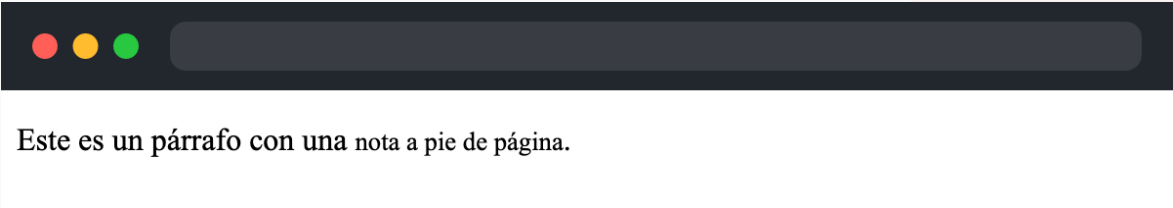
Por ejemplo:

4 ELEMENTOS



```
<p>Este es un párrafo con una <small>nota a pie de
página</small>.</p>
```

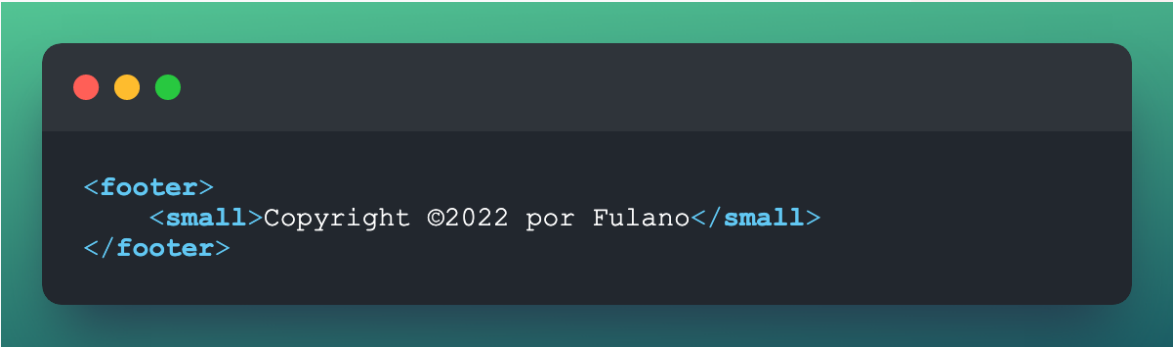
El código se renderiza de esta manera al abrirlo en un navegador.



Este es un párrafo con una nota a pie de página.

En este caso, el navegador mostraría “Este es un párrafo con una nota a pie de página” con la palabra “nota a pie de página” en tamaño de letra reducido.

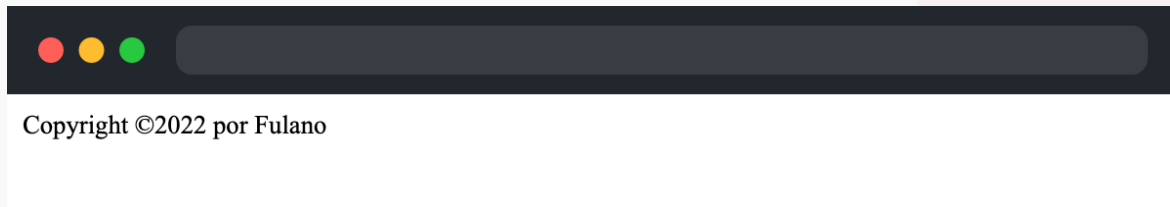
También se puede utilizar para dar créditos o para mostrar alguna información adicional pero no esencial en el contenido:



```
<footer>
  <small>Copyright ©2022 por Fulano</small>
</footer>
```

El código se renderiza de esta manera al abrirlo en un navegador.

4 ELEMENTOS



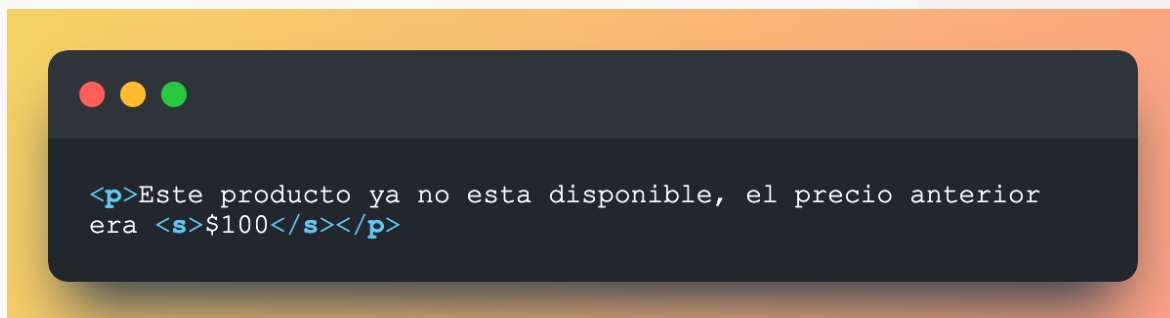
En este caso, el navegador mostraría “Copyright ©2022 por Fulano” con tamaño de letra reducido en la sección del pie de pagina.

Al igual que **** y ****, **<small>** es una etiqueta semántica que ayuda a describir el significado del contenido en lugar de describir su apariencia.

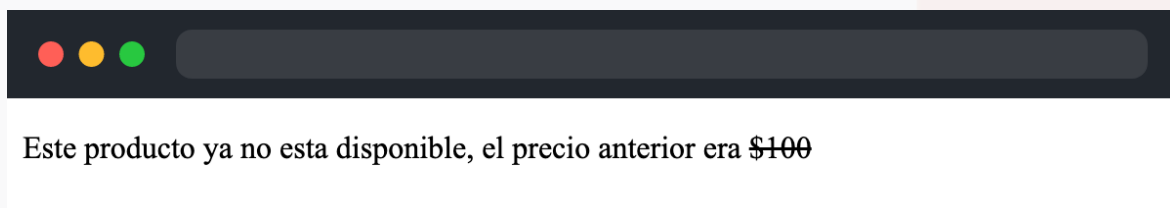
4.5.5 <s>

La etiqueta **<s>** en HTML se utiliza para indicar texto tachado o ya no válido. El navegador suele representar el contenido dentro de esta etiqueta con una línea tachando el texto.

Por ejemplo:



El código se renderiza de esta manera al abrirlo en un navegador.



4 ELEMENTOS

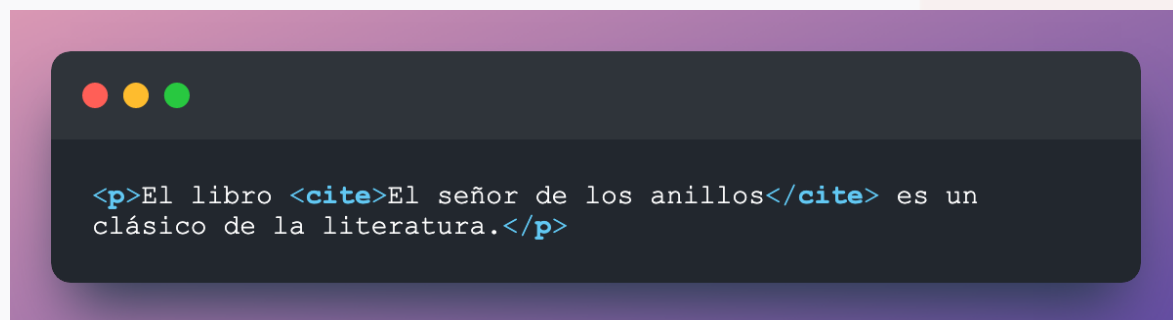
En este caso, el navegador mostraría “Este producto ya no esta disponible, el precio anterior era \$100” con la palabra “\$100” tachada.

Es importante notar que `<s>` es una etiqueta semántica que indica que el contenido dentro de ella ya no es válido o correcto, no solo que se ve tachado.

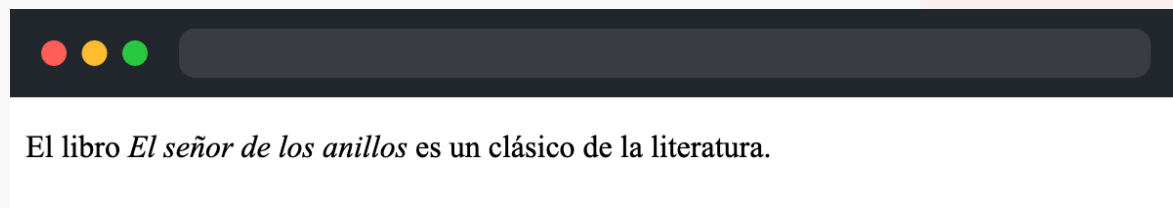
4.5.6 `<cite>`

La etiqueta `<cite>` en HTML se utiliza para indicar el título de una obra, como un libro, una película, una canción, una conferencia, entre otros. El navegador suele representar el contenido dentro de esta etiqueta con cursiva, pero esto puede variar dependiendo del estilo de la página.

Por ejemplo:



El código se renderiza de esta manera al abrirlo en un navegador.



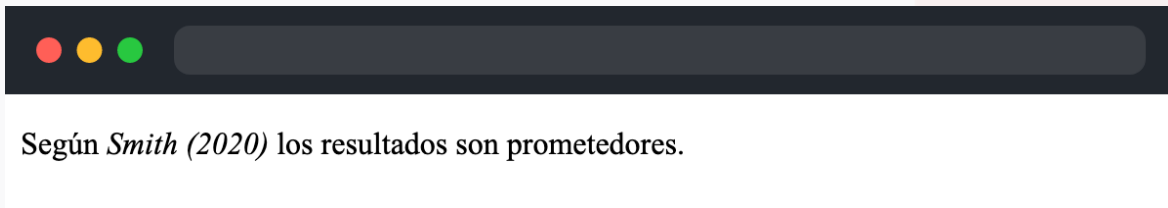
En este caso, el navegador mostraría “El libro” El señor de los anillos” es un clásico de la literatura.” con el titulo “El señor de los anillos” en cursiva.

4 ELEMENTOS

También se puede utilizar para indicar una referencia en un trabajo académico o científico:



El código se renderiza de esta manera al abrirlo en un navegador.



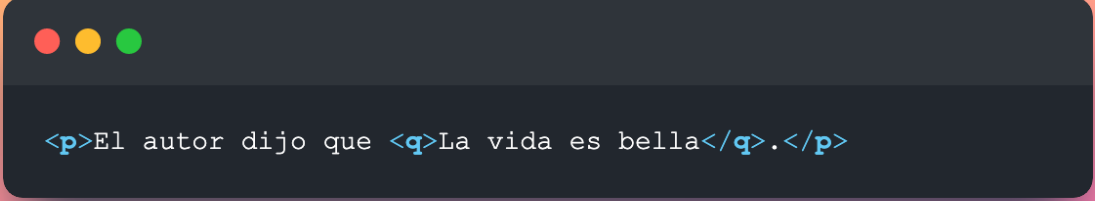
En este caso, el navegador mostraría “Según Smith (2020) los resultados son prometedores.” con el título de la referencia “Smith (2020)” en cursiva.

4.5.7 <q>

La etiqueta **<q>** en HTML se utiliza para indicar una cita corta o una cita incluida dentro de un párrafo. El navegador suele agregar comillas alrededor del contenido dentro de esta etiqueta, aunque esto puede variar dependiendo del navegador y de la configuración de estilo de la página.

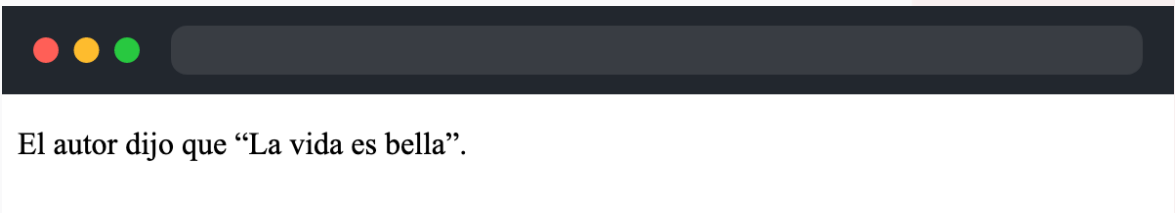
Por ejemplo:

4 ELEMENTOS

A dark-themed code editor window with three colored window control buttons (red, yellow, green) in the top-left corner. The text inside the editor is: `<p>El autor dijo que <q>La vida es bella</q>.</p>`

```
<p>El autor dijo que <q>La vida es bella</q>.</p>
```

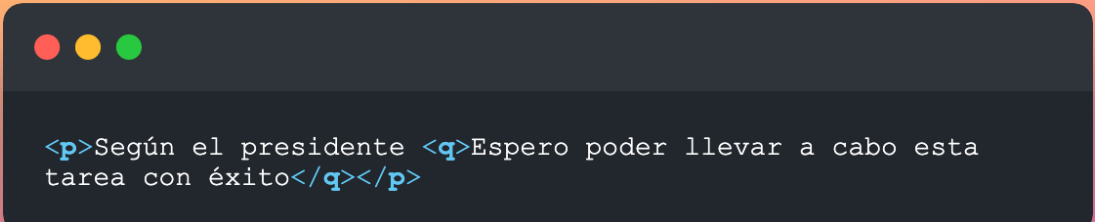
El código se renderiza de esta manera al abrirlo en un navegador.

A browser window with a dark header bar containing the same three window control buttons. Below the header is a white content area displaying the rendered text: "El autor dijo que "La vida es bella"."

El autor dijo que "La vida es bella".

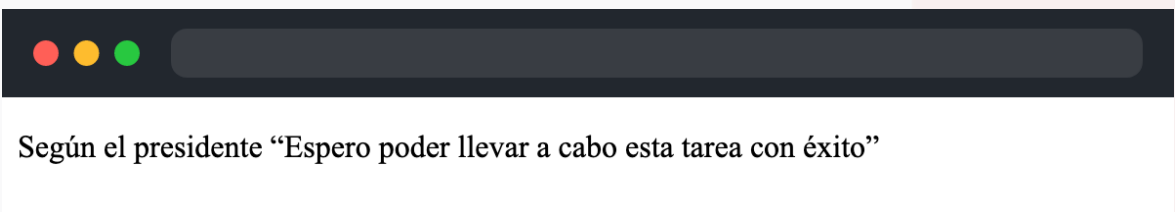
En este caso, el navegador mostraría "El autor dijo que" La vida es bella" ." con las comillas alrededor de la frase "La vida es bella"

También se puede utilizar para indicar una cita en una entrevista o para indicar una opinión o comentario:

A dark-themed code editor window with three colored window control buttons (red, yellow, green) in the top-left corner. The text inside the editor is: `<p>Según el presidente <q>Espero poder llevar a cabo esta tarea con éxito</q></p>`

```
<p>Según el presidente <q>Espero poder llevar a cabo esta  
tarea con éxito</q></p>
```

El código se renderiza de esta manera al abrirlo en un navegador.

A browser window with a dark header bar containing the same three window control buttons. Below the header is a white content area displaying the rendered text: "Según el presidente "Espero poder llevar a cabo esta tarea con éxito""

Según el presidente "Espero poder llevar a cabo esta tarea con éxito"

4 ELEMENTOS

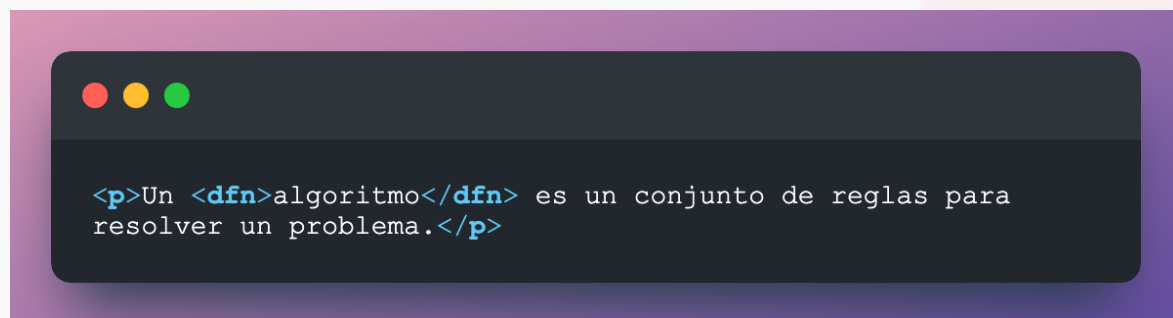
En este caso, el navegador mostraría “Según el presidente” Espero poder llevar a cabo esta tarea con éxito” ” con las comillas alrededor de la frase “Espero poder llevar a cabo esta tarea con éxito”

Es importante notar que a diferencia de **<cite>** esta etiqueta se utiliza para citas breves y dentro de un párrafo, y no para indicar el título de una obra.

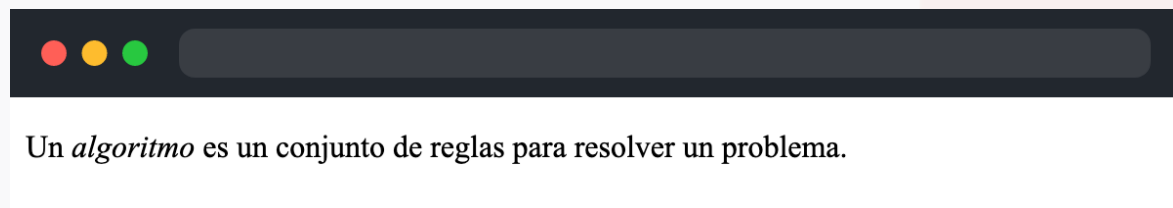
4.5.8 <dfn>

La etiqueta **<dfn>** en HTML se utiliza para indicar el primer uso o la definición de un término o una frase específica dentro de un párrafo. El navegador suele representar el contenido dentro de esta etiqueta con cursiva, pero esto puede variar dependiendo del estilo de la página. Es una etiqueta semántica que ayuda a describir el significado del contenido en lugar de describir su apariencia.

Por ejemplo:



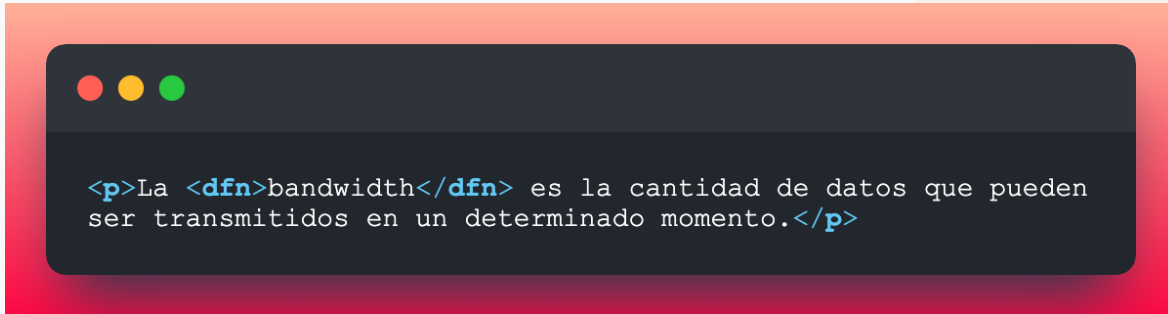
El código se renderiza de esta manera al abrirlo en un navegador.



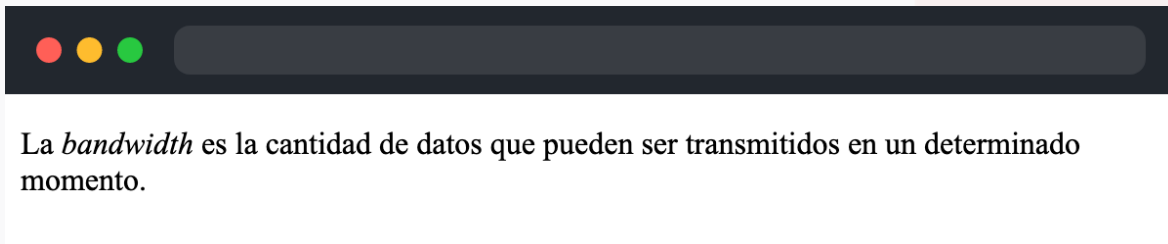
En este caso, el navegador mostraría “Un algoritmo es un conjunto de reglas para resolver un problema.” con la palabra “algoritmo” en cursiva.

4 ELEMENTOS

También se puede utilizar para definir una palabra técnica o un jergón específico en un documento.



El código se renderiza de esta manera al abrirlo en un navegador.



En este caso, el navegador mostraría “La bandwidth es la cantidad de datos que pueden ser transmitidos en un determinado momento.” con la palabra “bandwidth” en cursiva.

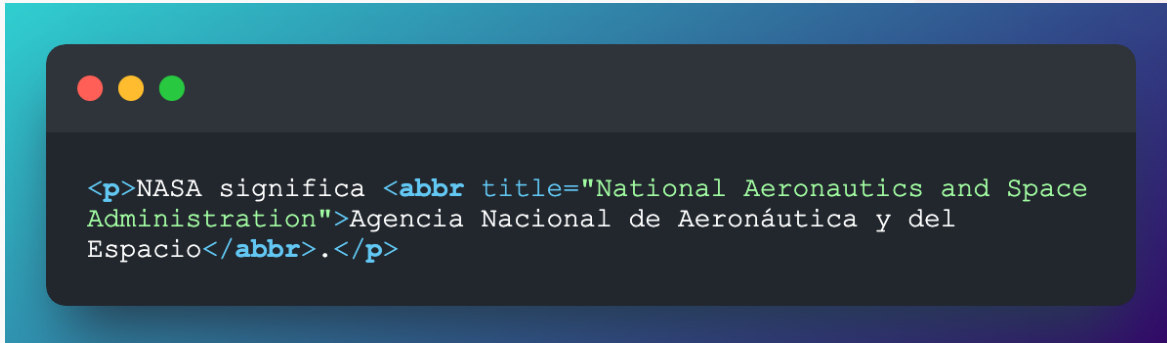
La idea detrás de esta etiqueta es que los navegadores o programas de lectura de pantalla podrían proporcionar una descripción adicional o una definición al usuario cuando el cursor se coloca sobre la palabra o frase marcada con dfn, o al utilizar ciertos atajos de teclado.

4.5.9 <abbr>

La etiqueta **<abbr>** en HTML se utiliza para indicar una abreviatura o un acrónimo dentro de un párrafo. Es una etiqueta semántica que ayuda a describir el significado del contenido en lugar de describir su apariencia.

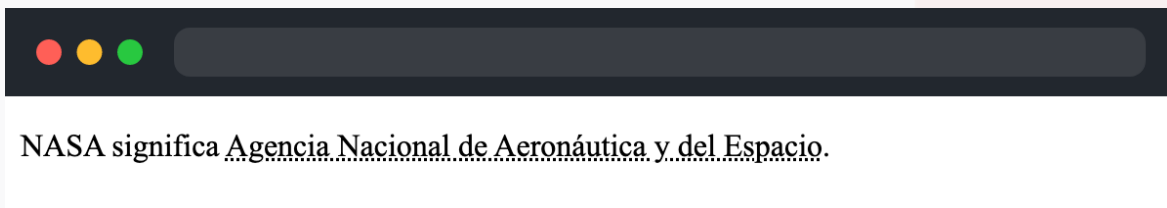
4 ELEMENTOS

Por ejemplo:



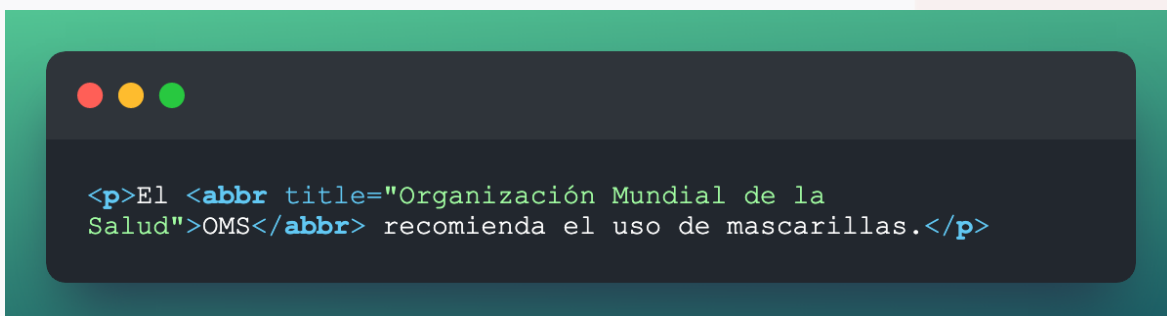
```
<p>NASA significa <abbr title="National Aeronautics and Space Administration">Agencia Nacional de Aeronáutica y del Espacio</abbr>.</p>
```

El código se renderiza de esta manera al abrirlo en un navegador.



En este caso, el navegador mostraría “NASA significa Agencia Nacional de Aeronáutica y del Espacio” con la palabra “NASA” podría ser mostrada en cursiva u otro formato dependiendo del estilo de la página y además si el usuario mueve el cursor sobre la abreviación NASA, el navegador mostrará la expansión de la abreviación “National Aeronautics and Space Administration”

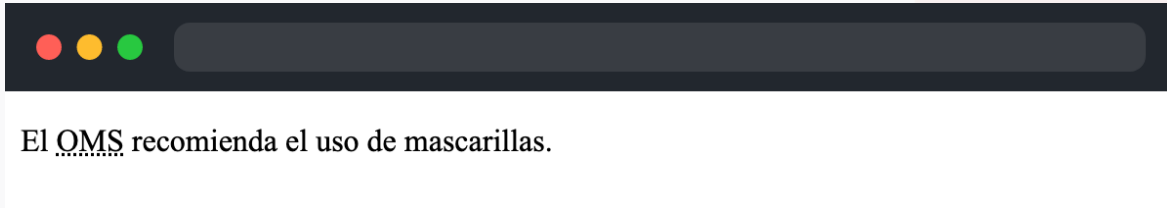
También se puede utilizar para indicar una abreviatura en un documento técnico o científico:



```
<p>El <abbr title="Organización Mundial de la Salud">OMS</abbr> recomienda el uso de mascarillas.</p>
```

4 ELEMENTOS

El código se renderiza de esta manera al abrirlo en un navegador.



En este caso, el navegador mostraría “El OMS recomienda el uso de mascarillas.” con la palabra “OMS” podría ser mostrada en cursiva u otro formato dependiendo del estilo de la página y además si el usuario mueve el cursor sobre la abreviación OMS, el navegador mostrará la expansión de la abreviación “Organización Mundial de la Salud”

Al igual que en `<dfn>`, la idea detrás de esta etiqueta es que los navegadores o programas de lectura de pantalla podrían proporcionar una descripción adicional o una definición al usuario cuando el cursor se coloca sobre la palabra o frase marcada con `abbr` o al utilizar ciertos atajos de teclado.

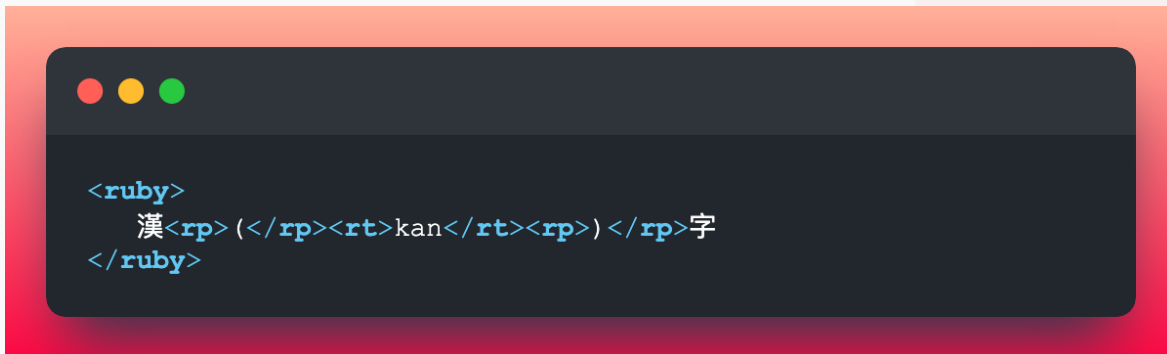
4.5.10 `<ruby>`

La etiqueta `<ruby>` en HTML es una extensión de HTML5 que se utiliza para indicar una notación de ruby o anotación de ruby, que son unas pequeñas anotaciones en caracteres japoneses que se utilizan para proporcionar una guía de pronunciación o un significado adicional a los caracteres. Es una etiqueta específica para manejar notación de ruby, que es común en lenguas con un sistema de escritura complejo como el japonés o el chino.

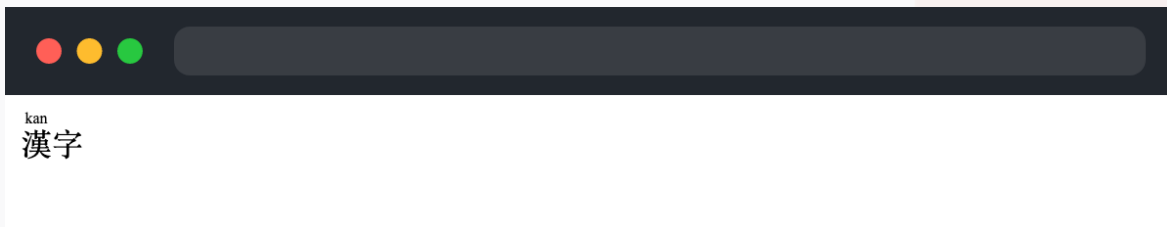
La etiqueta `<ruby>` se usa para contener el texto base y el texto de la ruby se incluye dentro de una etiqueta `<rt>` (ruby text) que se coloca dentro de la etiqueta `<ruby>`.

Por ejemplo:

4 ELEMENTOS



El código se renderiza de esta manera al abrirlo en un navegador.



En este caso, el navegador mostraría el carácter japonés “漢字” con el carácter “kan” en una notación ruby sobre el carácter “漢” y rp es usado para indicar los paréntesis que se usan para rodear la ruby.

En resumen, **<ruby>** es una extensión de HTML5 para agregar anotaciones de pronunciación o significado a caracteres, especialmente en idiomas como el chino o el japonés y su uso es muy específico y es para contenido en idiomas como el japonés o chino que tienen un sistema de escritura complejo.

4.5.11 <rt>

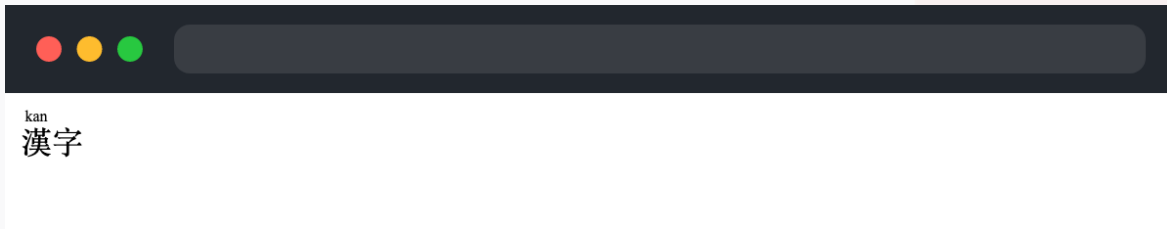
La etiqueta **<rt>** en HTML es una extensión de HTML5 que se utiliza en conjunto con la etiqueta **<ruby>** para indicar el texto de la anotación ruby. La etiqueta **<rt>** se coloca dentro de la etiqueta **<ruby>** para contener el texto de la anotación ruby. El contenido dentro de esta etiqueta se suele representar en un tamaño de letra reducido y alineado arriba o encima del texto base de la ruby.

4 ELEMENTOS

Por ejemplo:

```
<ruby>
  漢<rp>(</rp><rt>kan</rt><rp>)</rp>字
</ruby>
```

El código se renderiza de esta manera al abrirlo en un navegador.



En este caso, el navegador mostraría el caracter japonés “漢字” con el caracter “kan” en una notación ruby sobre el caracter “漢” y el contenido dentro de la etiqueta **<rt>** es “kan”

Es importante destacar que la etiqueta **<rt>** es parte de un conjunto de etiquetas específicas para manejar las notaciones de ruby, y su uso es exclusivo de estas anotaciones para idiomas como chino y japonés.

4.5.12 <rp>

La etiqueta **<rp>** en HTML es una extensión de HTML5 que se utiliza en conjunto con la etiqueta **<ruby>** para indicar el texto de paréntesis o el texto en lugar de una notación ruby, especialmente cuando los navegadores no son capaces de mostrar la ruby o los caracteres usados en la notación ruby. La etiqueta **<rp>** se coloca dentro

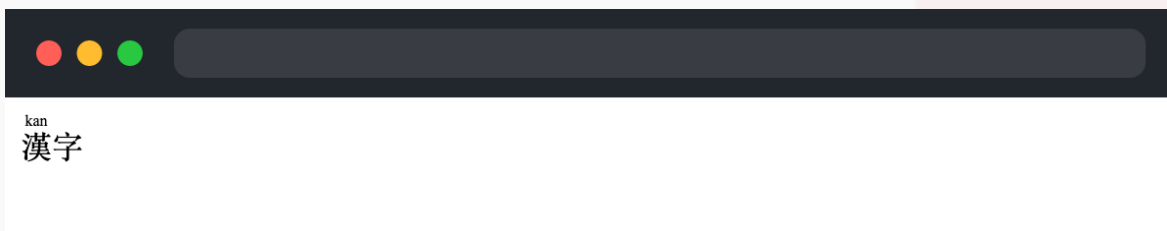
4 ELEMENTOS

de la etiqueta **<ruby>** para contener el texto de paréntesis o el texto en lugar de la ruby.

Por ejemplo:

```
<ruby>
  漢<rp>(</rp><rt>kan</rt><rp>)</rp>字
</ruby>
```

El código se renderiza de esta manera al abrirlo en un navegador.



En este caso, el navegador mostraría el carácter japonés “漢字” con el carácter “kan” en una notación ruby sobre el carácter “漢” y las etiquetas **<rp>** contienen los paréntesis que rodean la notación ruby “kan” .

Si el navegador no es capaz de mostrar la notación ruby, el contenido dentro de las etiquetas **<rp>** serán mostrados como una notación alternativa para ayudar a identificar la notación ruby y su pronunciación.

Es importante destacar que al igual que las etiquetas **<ruby>** y **<rt>**, **<rp>** es parte de un conjunto de etiquetas específicas para manejar las notaciones de ruby, y su uso es exclusivo de estas anotaciones para idiomas como chino y japonés.

4.5.13 <data>

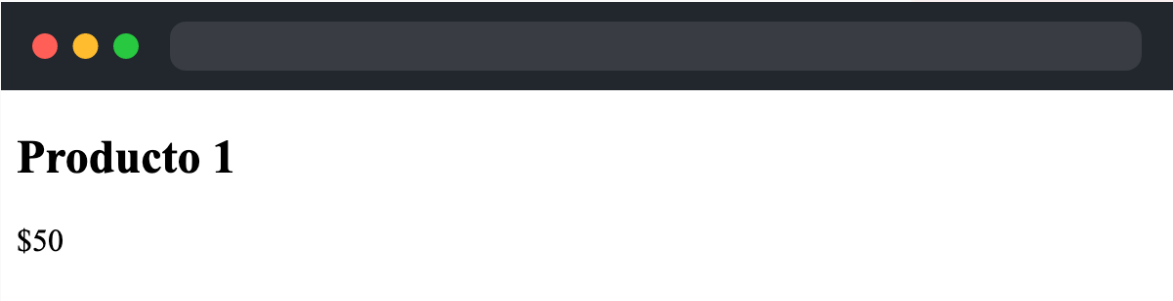
La etiqueta **<data>** en HTML es una extensión de HTML5 que se utiliza para almacenar datos privados dentro de un elemento del DOM (Document Object Model) para su uso posterior. El contenido dentro de esta etiqueta es invisible para el usuario final y no se representa en la página web. El valor del atributo “value” es el valor almacenado y accesible para su uso en JavaScript.

Por ejemplo, si deseas almacenar el precio de un producto en una página web para usarlo en una función de carrito de compras, podrías usar la etiqueta **<data>** de la siguiente manera:

A code editor window with a dark background and a red-to-orange gradient border. It contains the following HTML code:

```
<div id="producto1">
  <h2>Producto 1</h2>
  <p>$50</p>
  <data value="50" class="precio"></data>
</div>
```

El código se renderiza de esta manera al abrirlo en un navegador.

A browser window with a dark header bar containing three colored circles (red, yellow, green) and a search bar. The main content area is white and displays the rendered HTML code.

Producto 1

\$50

En este caso, el elemento **<data>** está contenido dentro del elemento **<div>** con id “producto1” y almacena el precio en el atributo “value” es accesible mediante

4 ELEMENTOS

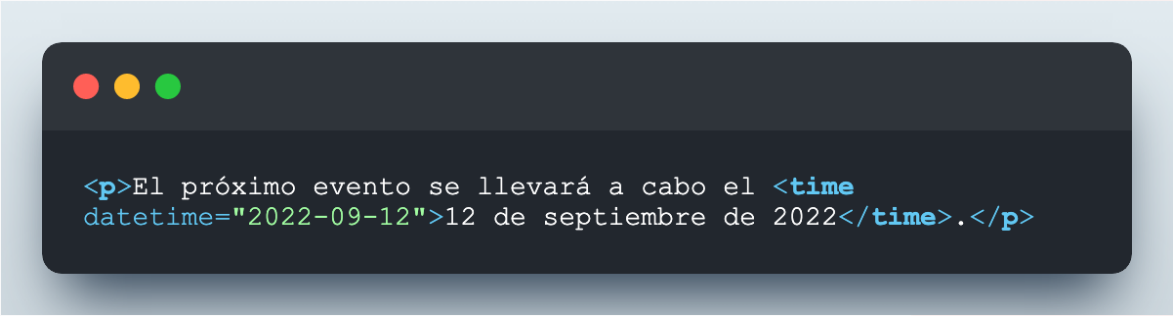
JavaScript para su uso en una función de carrito de compras.

Es importante destacar que la etiqueta **<data>** es solo para almacenar información y no tiene efectos visuales, para usar esa información, debes usar javascript y hacer uso del valor almacenado en el atributo “value” , para poder luego acceder a ese valor y usarlo en operaciones en el lado del cliente como calcular precios finales, guardar información en una sesión, o enviar información al servidor a través de una solicitud AJAX.

4.5.14 <time>

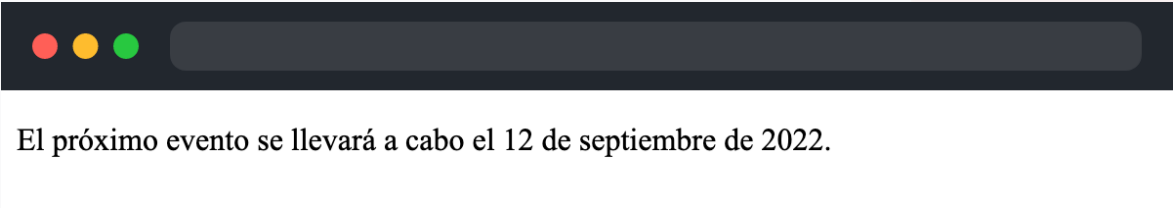
La etiqueta **<time>** en HTML es una extensión de HTML5 que se utiliza para indicar una fecha o una hora dentro de un párrafo. Es una etiqueta semántica que ayuda a describir el significado de una fecha o hora en lugar de describir su apariencia. La etiqueta **<time>** tiene un atributo “datetime” que contiene la fecha o hora en un formato estandarizado.

Por ejemplo:



```
<p>El próximo evento se llevará a cabo el <time  
datetime="2022-09-12">12 de septiembre de 2022</time>.</p>
```

El código se renderiza de esta manera al abrirlo en un navegador.



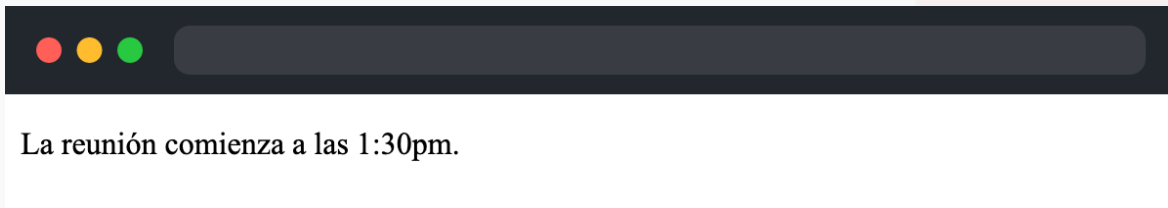
El próximo evento se llevará a cabo el 12 de septiembre de 2022.

4 ELEMENTOS

En este caso, el navegador mostraría “El próximo evento se llevará a cabo el 12 de septiembre de 2022.” y el contenido dentro de la etiqueta `time` es una representación visual de la fecha, mientras que el atributo `datetime` contiene la fecha en formato estandarizado “AAAA-MM-DD”

También se puede usar para mostrar un horario:

El código se renderiza de esta manera al abrirlo en un navegador.



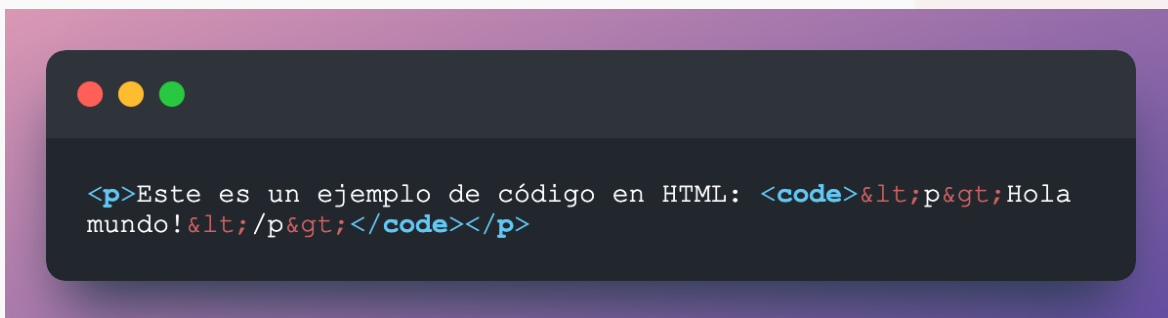
En este caso, el navegador mostraría “La reunión comienza a las 1:30pm.” y el contenido dentro de la etiqueta `<time>` es una representación visual del horario, mientras que el atributo `datetime` contiene la hora en formato estandarizado “HH:mm”

Es importante destacar que al igual que otras etiquetas semánticas en HTML, el objetivo de la etiqueta `<time>` es proporcionar significado y estructura al contenido y puede ser útil para el trabajo con script o estilos. Además, al usar el formato estandarizado en el atributo `datetime` puede ser procesado de manera más fácil por los programas o robots que consumen ese contenido.

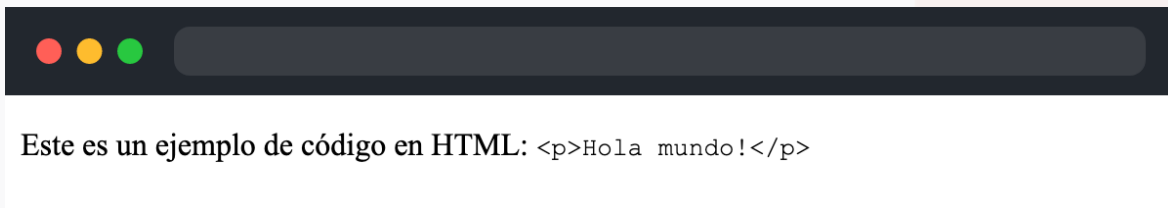
4.5.15 <code>

La etiqueta **<code>** en HTML se utiliza para mostrar contenido de código en una página web. El contenido dentro de esta etiqueta se presenta en una fuente mono-espaciada y generalmente se muestra con un aspecto distinto al del texto normal en una página.

Aquí tienes un ejemplo de cómo usar la etiqueta **<code>** en una página web:



El código se renderiza de esta manera al abrirlo en un navegador.



En este ejemplo, el texto dentro de las etiquetas **<code>** se mostrará como un fragmento de código. Al mostrarlo en el navegador, el código **<p>Hola mundo!</p>** sería mostrado con una apariencia distinta al resto del texto en la página.

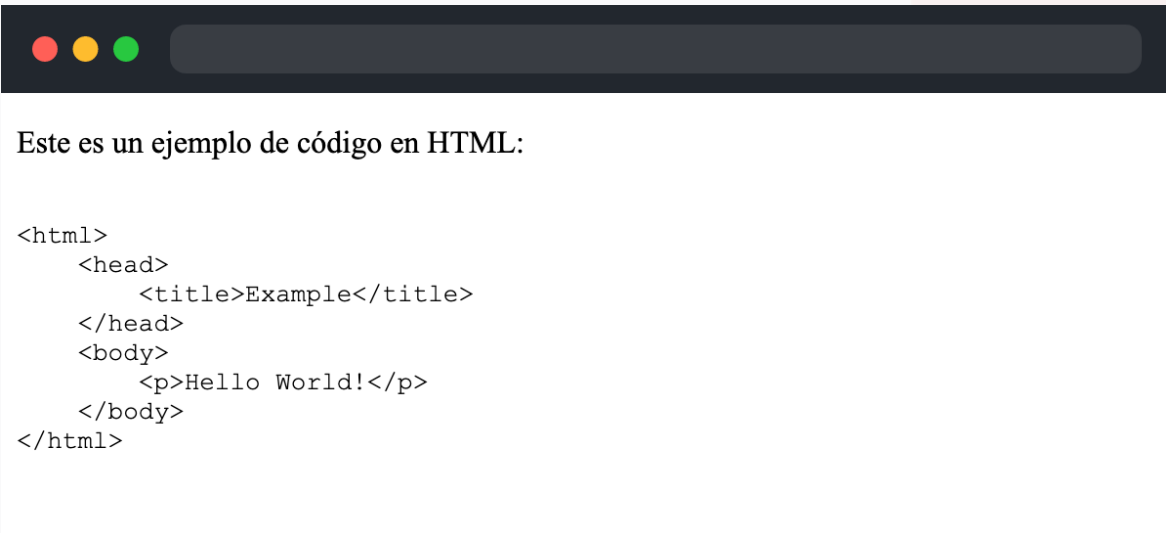
Además, hay una etiqueta **<pre>** que es similar pero los espacios y tabulaciones son mantenidos en su formato original y se presenta con una apariencia distinta para aprovechar mejor la estructura del código.

4 ELEMENTOS



```
<p>Este es un ejemplo de código en HTML:</p>
<pre>
<code>
<html>
  <head>
    <title>Example</title>
  </head>
  <body>
    <p>Hello World!</p>
  </body>
</html>
</code>
</pre>
```

El código se renderiza de esta manera al abrirlo en un navegador.



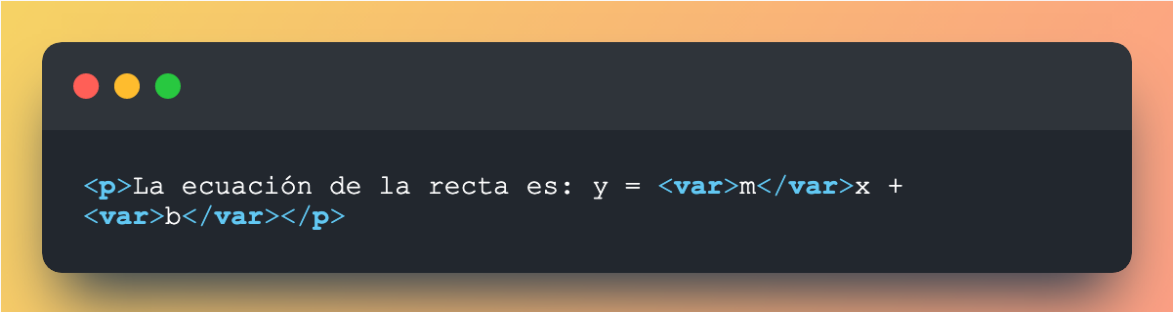
```
<html>
  <head>
    <title>Example</title>
  </head>
  <body>
    <p>Hello World!</p>
  </body>
</html>
```

En este caso se vería mejor estructurado el código y se puede apreciar mejor la indentación y estructura del mismo.

4.5.16 <var>

La etiqueta **<var>** en HTML se utiliza para indicar una variable en el contenido de una página web. Se utiliza principalmente en contenido matemático y científico, pero también puede utilizarse en otro tipo de contenido en el que se quiera destacar una variable específica.

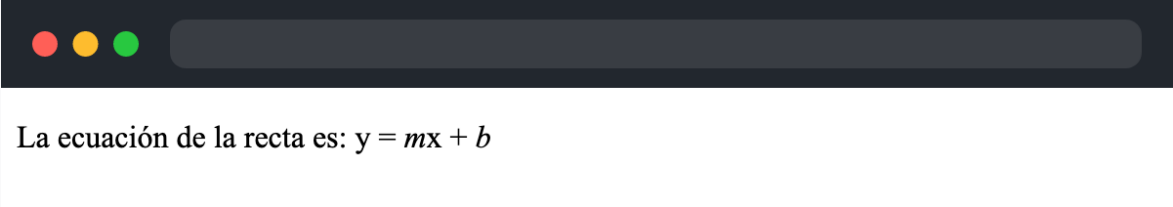
Aquí tienes un ejemplo de código HTML que utiliza la etiqueta **<var>** para resaltar una variable en una ecuación matemática:

A code editor window with a dark background and three colored window control buttons (red, yellow, green) in the top left corner. The code inside is:

```
<p>La ecuación de la recta es: y = <var>m</var>x +  
<var>b</var></p>
```

```
<p>La ecuación de la recta es: y = <var>m</var>x +  
<var>b</var></p>
```

El código se renderiza de esta manera al abrirlo en un navegador.

A browser window with a dark header bar containing three colored window control buttons (red, yellow, green) and a search bar. Below the header, the rendered HTML code is displayed:

La ecuación de la recta es: $y = mx + b$

La ecuación de la recta es: $y = mx + b$

En este ejemplo, la variable m y b son resaltadas utilizando la etiqueta **<var>** y se mostrara de manera diferente a lo que esta dentro de la etiqueta, dependiendo del navegador y configuraciones de estilo.

A pesar que este ejemplo es utilizando una ecuación matemática, la etiqueta **var** se puede usar en diferentes contextos, no necesariamente matemáticos.

4.5.17 <samp>

La etiqueta **<samp>** en HTML se utiliza para indicar una salida de un programa o código de ejemplo. Es similar a la etiqueta **<code>**, pero se utiliza específicamente para mostrar un ejemplo de código o salida generada por un programa.

Un ejemplo práctico de cómo utilizar la etiqueta **<samp>** podría ser en una página web que proporciona documentación técnica sobre una función de programación. La documentación podría mostrar el código de ejemplo para utilizar la función, seguido de una muestra de la salida generada cuando se ejecuta el código. El código de ejemplo y la salida generada podrían mostrarse como sigue:



```
<p>Código de ejemplo:</p>
<pre>
  <code>
    function greet(name) {
      console.log("Hello, " + name + "!");
    }
    greet("John");
  </code>
</pre>

<p>Salida generada:</p>
<samp>Hello, John!</samp>
```

El código se renderiza de esta manera al abrirlo en un navegador.

4 ELEMENTOS

Código de ejemplo:

```
function greet(name) {  
    console.log("Hello, " + name + "!");  
}  
greet("John");
```

Salida generada:

```
Hello, John!
```

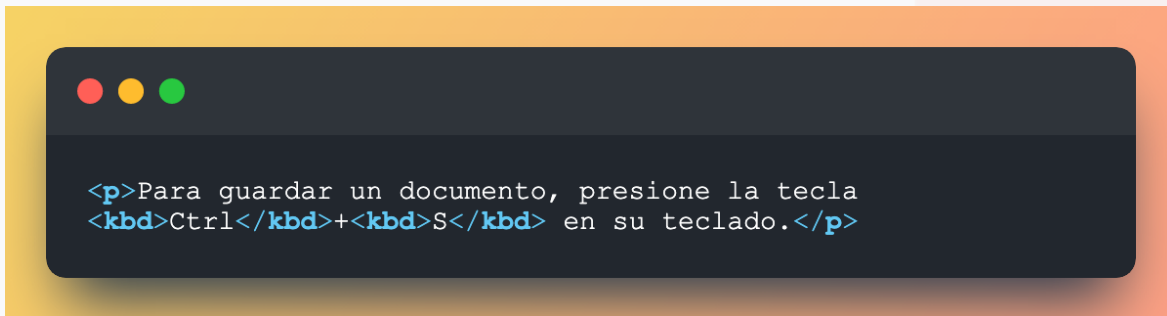
En este ejemplo, el código de ejemplo se muestra utilizando la etiqueta **<code>** y la salida generada se muestra utilizando la etiqueta **<samp>**, permitiendo así identificar de manera clara la salida que el programa genera.

4.5.18 <kbd>

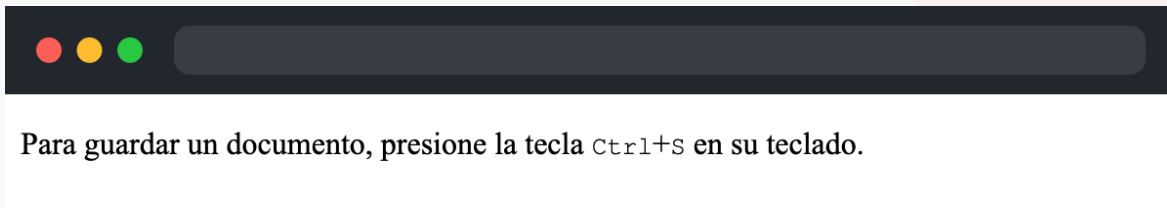
La etiqueta **<kbd>** en HTML se utiliza para indicar un texto que representa una entrada de teclado. Se utiliza para mostrar al usuario cómo introducir un comando o una secuencia de teclas específica en su sistema. Es comúnmente utilizado en documentación técnica para mostrar cómo utilizar comandos en línea de comando, atajos de teclado, etc.

Un ejemplo práctico de cómo utilizar la etiqueta **<kbd>** podría ser en una página web que proporciona instrucciones para guardar un documento en Microsoft Word. El documento podría mostrarse como:

4 ELEMENTOS



El código se renderiza de esta manera al abrirlo en un navegador.



En este ejemplo, las teclas Ctrl+S son mostradas en el navegador y se identifica como una secuencia de teclado que el usuario debe usar para guardar el documento

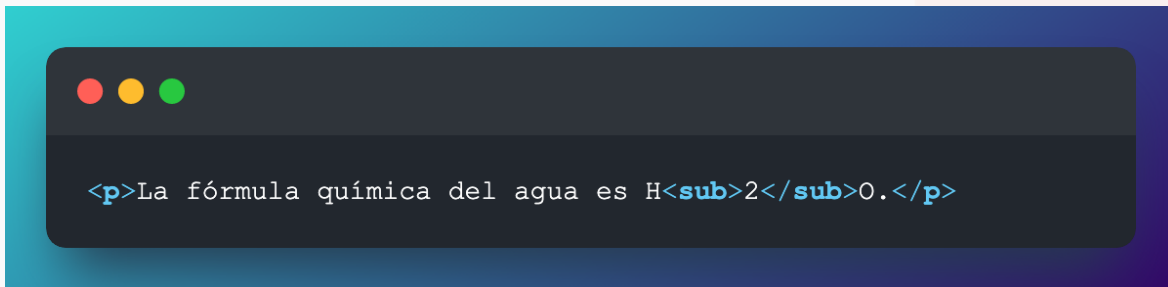
Es importante destacar que la apariencia de la etiqueta **<kbd>** dependerá de las configuraciones de estilo de cada navegador, pero en general se verá con un estilo de font similar al de la entrada de teclado.

4.5.19 <sub>

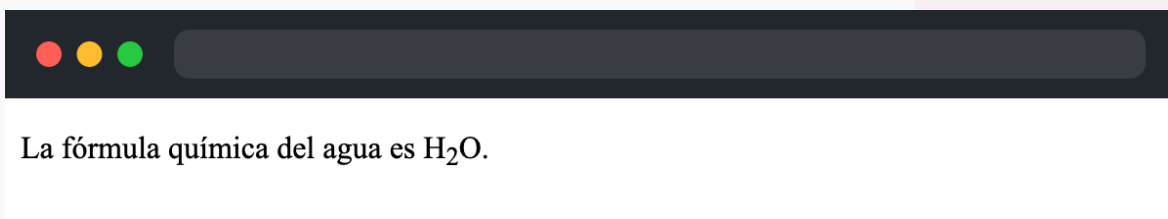
La etiqueta **<sub>** en HTML se utiliza para indicar un subíndice en el contenido de una página web. El texto dentro de esta etiqueta se mostrará como un subíndice en relación al texto que lo rodea. Esta etiqueta es comunmente utilizada en contenido matemático o científico, pero también puede utilizarse en otros contextos.

Un ejemplo práctico de cómo utilizar la etiqueta **<sub>** podría ser en una página web que muestra una fórmula química. La fórmula podría mostrarse como:

4 ELEMENTOS



El código se renderiza de esta manera al abrirlo en un navegador.



En este ejemplo se utiliza la etiqueta **<sub>** para mostrar el número 2 como un subíndice en relación al elemento Hidrogeno.

Es importante notar que para el uso de esta etiqueta es importante tener en cuenta el contexto y aplicarlo de manera adecuada para lograr una mejor comprensión en el contenido.

4.5.20 <sup>

La etiqueta **<sup>** en HTML se utiliza para indicar un superíndice en el contenido de una página web. El texto dentro de esta etiqueta se mostrará como un superíndice en relación al texto que lo rodea. Esta etiqueta es comunmente utilizada en contenido matemático o científico, pero también puede utilizarse en otros contextos.

Un ejemplo práctico de cómo utilizar la etiqueta **<sup>** podría ser en una página web que muestra una fórmula matemática. La fórmula podría mostrarse como:

4 ELEMENTOS

```
<p>El cuadrado de x es: x<sup>2</sup>.</p>
```

El código se renderiza de esta manera al abrirlo en un navegador.

El cuadrado de x es: x².

En este ejemplo se utiliza la etiqueta **<sup>** para mostrar el 2 como un superíndice en relación a la variable x. Es importante mencionar que al igual que la etiqueta **<sub>**, esta etiqueta también dependerá de la configuración de estilo de cada navegador para mostrar correctamente el superíndice.

Es importante destacar que estas etiquetas (**<sub>** y **<sup>**) son útiles para indicar matemáticamente un exponente o un índice en una fórmula, pero también se pueden usar para otro tipo de contenido en donde sea necesario indicar una relación de super o sub posicionamiento dentro de un texto.

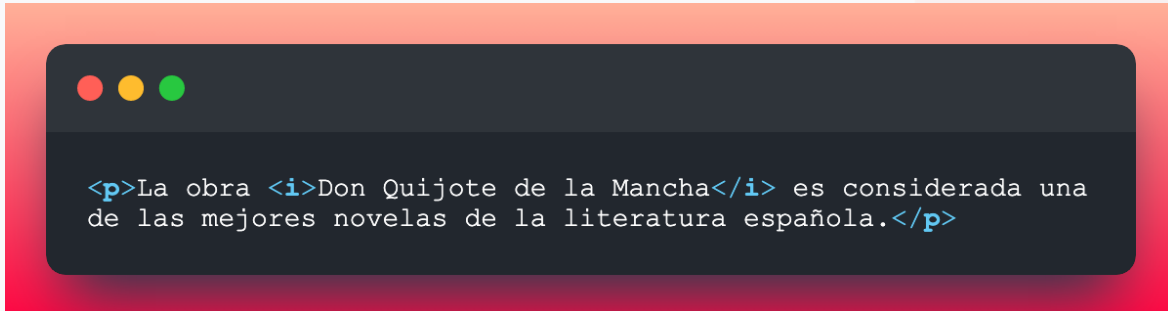
4.5.21 <i>

La etiqueta **<i>** en HTML se utiliza para indicar un texto en cursiva. Es utilizada para dar énfasis o destacar una palabra o frase dentro de un párrafo de texto. A menudo se utiliza para indicar un término técnico, una palabra en un idioma extranjero, una palabra figurativa, entre otros usos.

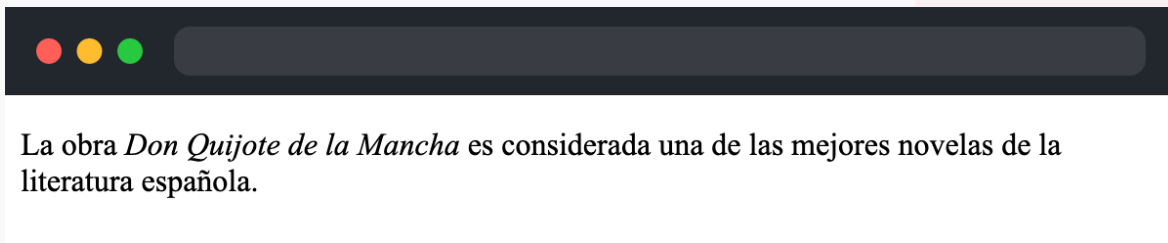
Un ejemplo práctico de cómo utilizar la etiqueta **<i>** podría ser en una página web

4 ELEMENTOS

que muestra un artículo sobre literatura. El artículo podría mostrarse como:



El código se renderiza de esta manera al abrirlo en un navegador.



En este ejemplo se utiliza la etiqueta `<i>` para mostrar el título de la obra en cursiva, identificando así que es una obra literaria y no una expresión común.

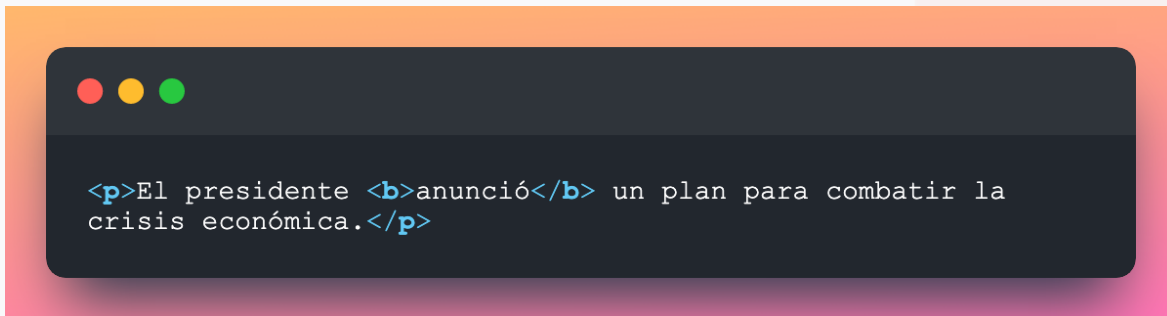
Es importante destacar que esta etiqueta solo cambia el estilo del texto a cursiva, pero no aporta ninguna otra funcionalidad adicional.

4.5.22 `<b>`

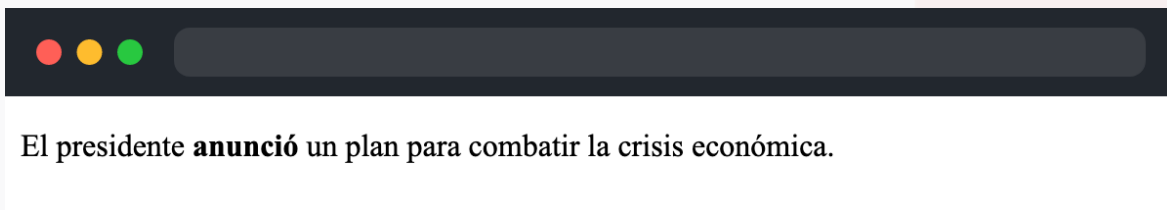
La etiqueta `<b>` en HTML se utiliza para indicar un texto en negrita. Es utilizada para dar énfasis o destacar una palabra o frase dentro de un párrafo de texto. A menudo se utiliza para indicar un término importante, una palabra clave, entre otros usos.

Un ejemplo práctico de cómo utilizar la etiqueta `<b>` podría ser en una página web que muestra un artículo sobre una noticia importante. La noticia podría mostrarse como:

4 ELEMENTOS



El código se renderiza de esta manera al abrirlo en un navegador.



En este ejemplo se utiliza la etiqueta `<b>` para mostrar la palabra “anunció” en negrita, identificando así que es una acción importante en la noticia.

Es importante destacar que esta etiqueta solo cambia el estilo del texto a negrita, pero no aporta ninguna otra funcionalidad adicional, existen otras etiquetas como `strong` que tiene un significado semántico similar pero se recomienda usar dependiendo del contexto.

4.5.23 `<u>`

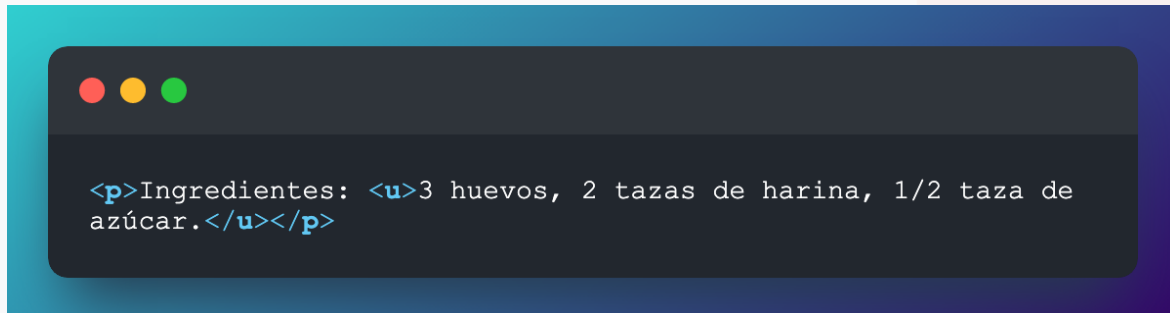
La etiqueta `<u>` en HTML se utiliza para indicar un texto subrayado. Es utilizada para dar énfasis o destacar una palabra o frase dentro de un párrafo de texto. A menudo se utiliza para indicar términos importantes, términos técnicos, entre otros usos.

Sin embargo, es importante tener en cuenta que el uso de esta etiqueta no es recomendado ya que el subrayado se usa en general para indicar hipervínculos en

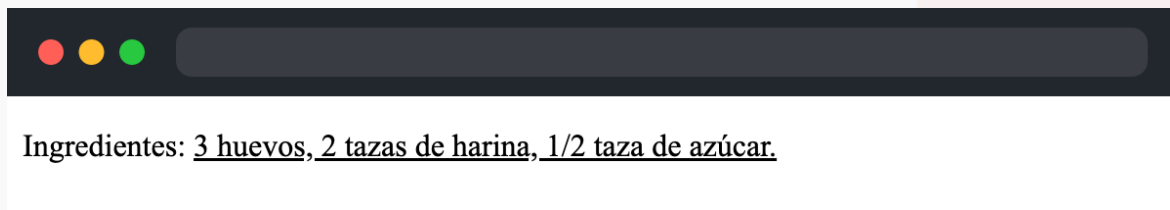
4 ELEMENTOS

una página web, por lo que su uso para otros propósitos puede causar confusión al usuario.

Un ejemplo práctico de cómo utilizar la etiqueta `<u>` podría ser en una página web que muestra una receta de cocina. La receta podría mostrarse como:



El código se renderiza de esta manera al abrirlo en un navegador.



En este ejemplo se utiliza la etiqueta `<u>` para mostrar los ingredientes en subrayado, identificando así de manera clara lo necesario para preparar la receta.

Es importante mencionar que existen otras etiquetas como `<em>` o `<strong>` que son mejor recomendadas para dar énfasis en un texto ya que no tienen las connotaciones de uso que tiene la etiqueta `<u>`.

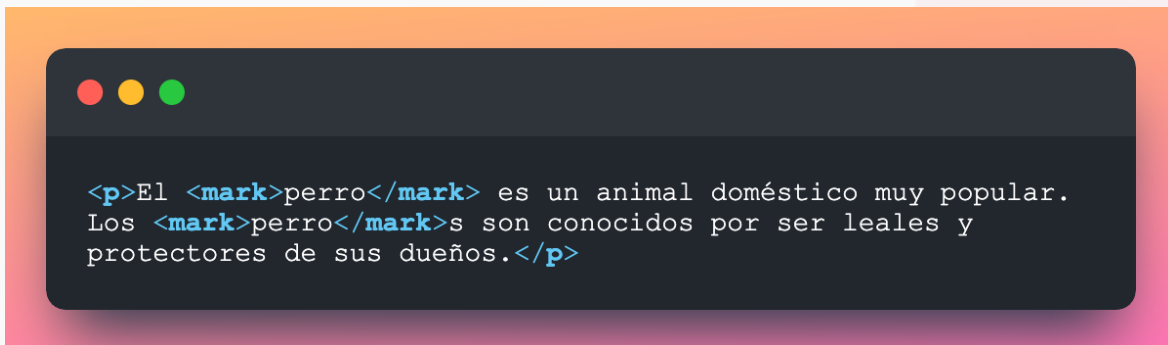
4.5.24 `<mark>`

La etiqueta `<mark>` en HTML se utiliza para destacar un texto específico dentro de un párrafo de contenido. Es similar a otras etiquetas como `<b>` y `<strong>`, pero tiene un enfoque más en el contexto de búsqueda. El texto dentro de esta etiqueta

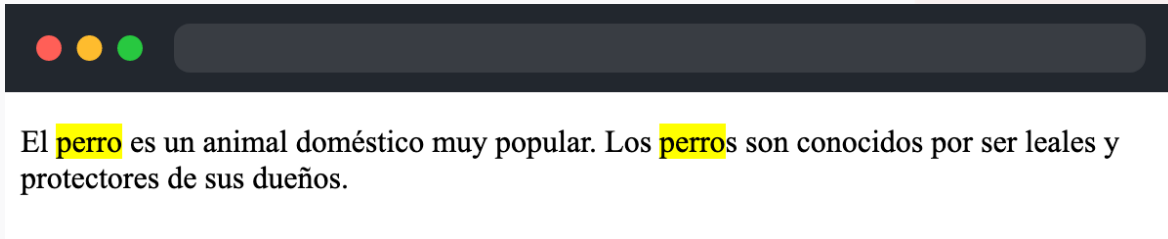
4 ELEMENTOS

se mostrará resaltado con un fondo diferente o con un borde, dependiendo de la configuración del navegador o del estilo CSS.

Un ejemplo práctico de cómo utilizar la etiqueta **<mark>** podría ser en una página web de búsqueda. Un usuario podría buscar la palabra “perro” y la página mostraría todas las ocurrencias de la palabra “perro” resaltadas con la etiqueta **<mark>** como:



El código se renderiza de esta manera al abrirlo en un navegador.



En este ejemplo se utiliza la etiqueta **<mark>** para resaltar la palabra “perro” que es el resultado de la búsqueda y de esta manera hacerlo más visible y evidente para el usuario. La apariencia del texto marcado puede variar dependiendo de las configuraciones de estilo de cada navegador y las hojas de estilo CSS.

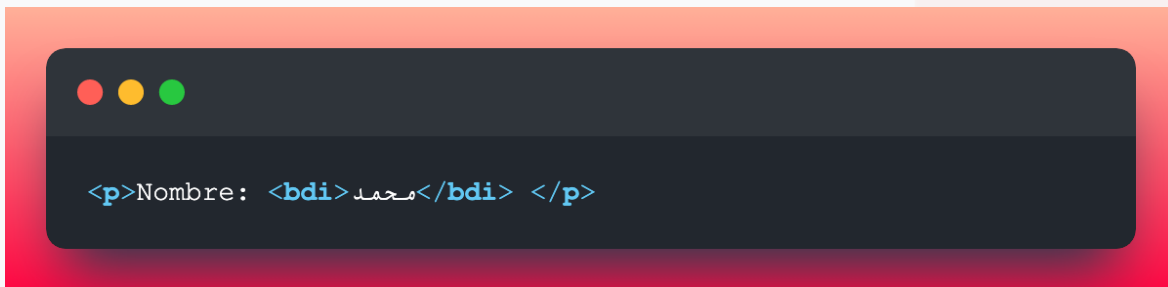
4.5.25 <bdi>

La etiqueta **<bdi>** (Bi-Directional Isolation) en HTML es una etiqueta semántica que se utiliza para indicar un fragmento de texto que se debe tratar de forma indepen-

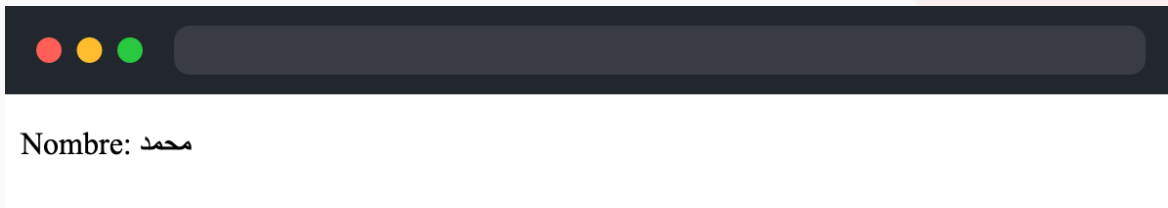
4 ELEMENTOS

diente en cuanto a su dirección de escritura. Esto es útil para lidiar con contenido que contiene texto escrito en varios idiomas o idiomas con dirección de escritura diferentes.

Un ejemplo práctico de cómo utilizar la etiqueta **<bdi>** podría ser en una página web que muestra un perfil de un usuario con un nombre escrito en árabe. La sección del perfil podría mostrarse como:



El código se renderiza de esta manera al abrirlo en un navegador.



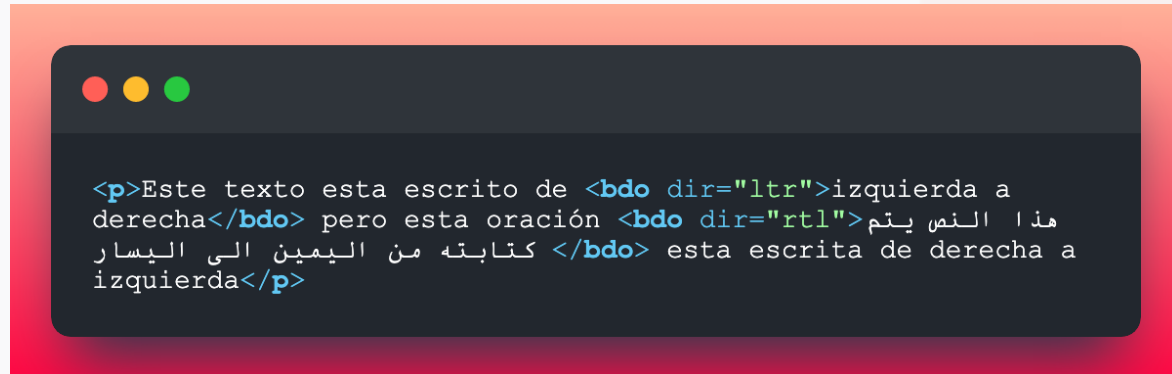
En este ejemplo se utiliza la etiqueta **<bdi>** para mostrar el nombre en árabe y permitir su correcta visualización e independencia de la dirección de escritura del resto del contenido de la página. Es importante mencionar que **<bdi>** sólo afecta a la dirección de escritura, pero no tiene un impacto estético o de formateo en el texto.

4.5.26 <bdo>

La etiqueta **<bdo>** (Bi-Directional Override) en HTML se utiliza para especificar la dirección de escritura de un texto de forma explícita. Esta etiqueta se utiliza para indicar cómo se deben mostrar los caracteres dentro de un elemento.

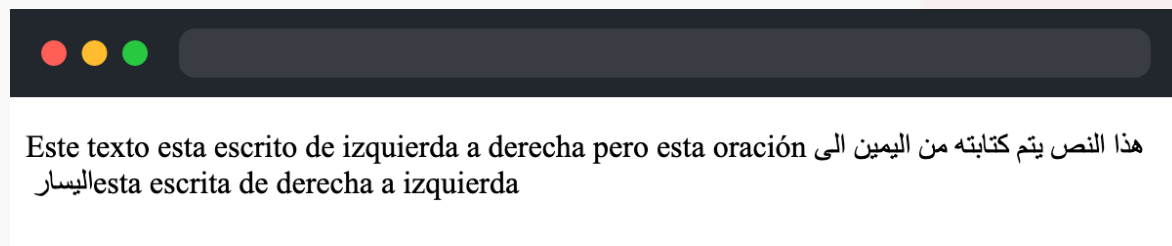
4 ELEMENTOS

Por ejemplo si una página tiene su dirección de escritura por defecto configurada como LTR (izquierda a derecha) pero en algún momento de la página se presenta una oración o una palabra en árabe se utilizaría `bdo` con la dirección RTL (derecha a izquierda).



```
<p>Este texto esta escrito de <bdo dir="ltr">izquierda a derecha</bdo> pero esta oración <bdo dir="rtl">هذا النص يتم كتابته من اليمين الى اليسار</bdo> esta escrita de derecha a izquierda</p>
```

El código se renderiza de esta manera al abrirlo en un navegador.



En este ejemplo se utiliza la etiqueta `<bdo>` con el atributo `dir` para indicar la dirección de escritura del texto “izquierda a derecha” y la oración en árabe “هذا النص يتم كتابته من اليمين الى اليسار” que es RTL y así garantizar su correcta visualización. Es importante mencionar que “bdo” es útil cuando se necesita cambiar la dirección de escritura de un texto, ya que si no se utiliza correctamente, puede generar problemas en la visualización del texto.

La diferencia entre `<bdi>` y `<bdo>` es que `bdi` aísla el contenido del contexto y permite que el navegador determine su dirección basándose en el contenido, mientras que `bdo` permite cambiar explícitamente la dirección del texto mediante el atributo “dir” . Ambas etiquetas son especialmente útiles en páginas web que tienen con-

4 ELEMENTOS

tenido multilingüe o con contenido que tiene una dirección de escritura diferente al contexto.

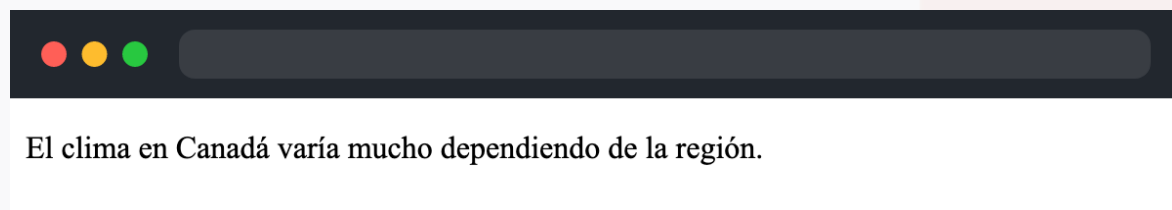
4.5.27

La etiqueta **** en HTML se utiliza para marcar un fragmento de texto dentro de un párrafo. Es una etiqueta de contenedor genérica que se utiliza para dar formato a un fragmento de texto dentro de una etiqueta de bloque. A menudo se utiliza en conjunto con hojas de estilo CSS para dar formato al contenido.

Un ejemplo práctico de cómo utilizar la etiqueta **** podría ser en una página web que muestra un artículo sobre un tema específico, y se quiere destacar algunas palabras con un formato diferente. El artículo podría mostrarse como:



El código se renderiza de esta manera al abrirlo en un navegador.



En este ejemplo se utiliza la etiqueta **** para encerrar la palabra “Canadá” y asignándole una clase CSS llamada “destacado”, esta clase puede tener ciertas propiedades de estilo para diferenciar visualmente la palabra “Canadá” del resto

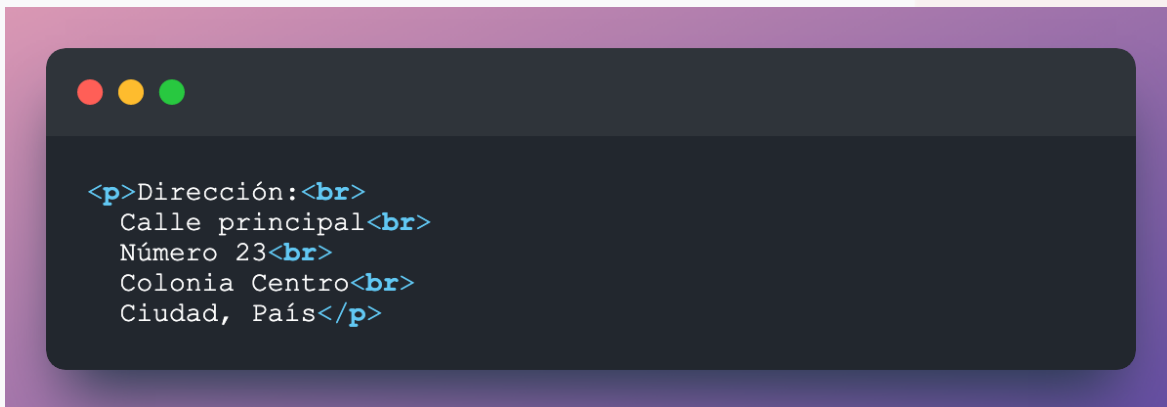
4 ELEMENTOS

del texto, como puede ser el color, tamaño de la fuente, entre otras cosas, de esta manera se logra un énfasis visual en esa palabra y se destaca para el usuario.

4.5.28

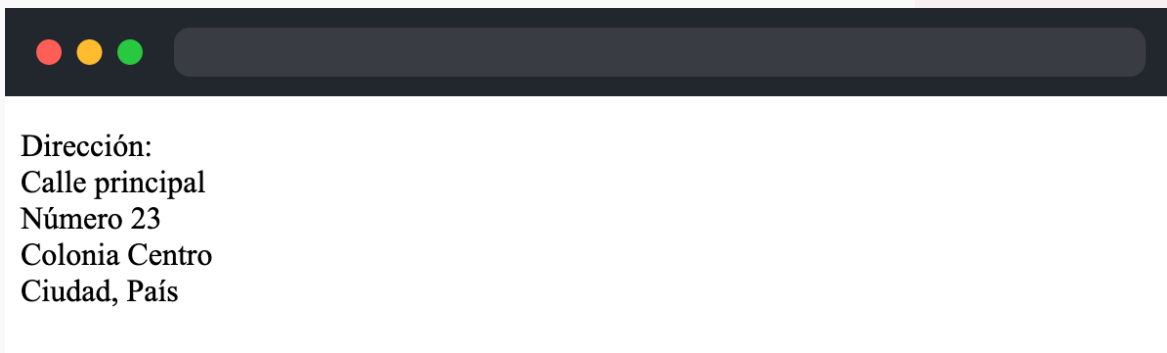
La etiqueta **
** en HTML se utiliza para insertar un salto de línea en un documento HTML. Es una etiqueta vacía, lo que significa que no tiene cierre y no contiene contenido. Simplemente inserta un salto de línea en el punto donde se coloca en el documento.

Un ejemplo práctico de cómo utilizar la etiqueta **
** podría ser en una página web que muestra una lista de direcciones. La lista podría mostrarse como:

A screenshot of a code editor window with a dark background and a purple border. The code is written in a light blue font and shows an HTML paragraph tag with several lines of text separated by line break tags. The code is:

```
<p>Dirección:<br>Calle principal<br>Número 23<br>Colonia Centro<br>Ciudad, País</p>
```

El código se renderiza de esta manera al abrirlo en un navegador.



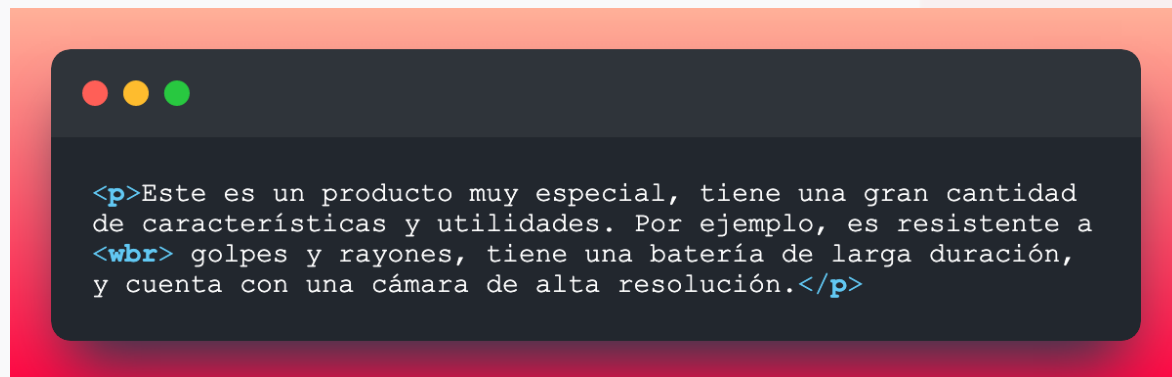
4 ELEMENTOS

En este ejemplo se utiliza la etiqueta **
** para separar cada línea de la dirección, generando un formato más claro y fácil de leer para el usuario. Es importante mencionar que **
** solo tiene un efecto visual, no tiene un significado semántico, hay alternativas como utilizar párrafos o listas para dar un estilo más estructurado al contenido.

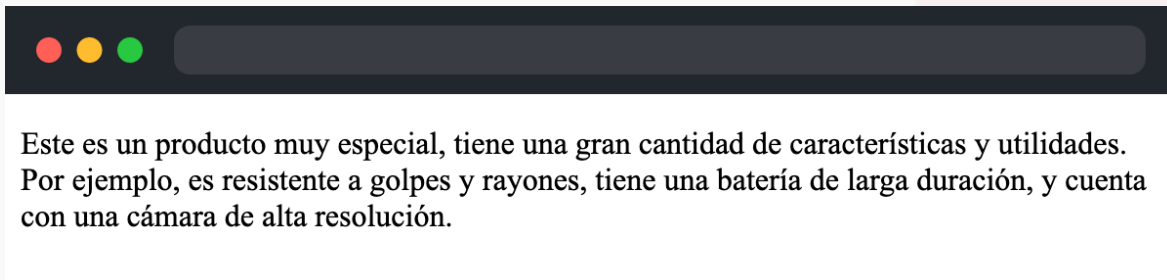
4.5.29 <wbr>

La etiqueta **<wbr>** en HTML es una etiqueta de corte de palabra, se utiliza para indicar un punto específico en el texto donde se permite que el navegador inserte un salto de línea si es necesario. Es especialmente útil cuando se tiene un texto largo que no se puede ajustar dentro del ancho de la pantalla.

Un ejemplo de uso práctico de **<wbr>** podría ser en una página web que muestra una descripción larga de un producto. El texto podría mostrarse de la siguiente manera:



El código se renderiza de esta manera al abrirlo en un navegador.



En este ejemplo se utiliza la etiqueta **<wbr>** para indicar al navegador que es posible insertar un salto de línea en la palabra “golpes” y “rayones” si es necesario, de esta forma se logra un formato más legible y estético para el usuario.

Es importante mencionar que **<wbr>** no garantiza que el navegador lo respete, ya que esto dependerá del ancho del contenedor y la configuración del navegador, pero proporciona una sugerencia para que el navegador pueda tomar una decisión.

4.6 Ediciones

Las etiquetas de ediciones en HTML (Edits) se utilizan para indicar inserciones, eliminaciones y correcciones en un documento. Estas etiquetas son **<ins>** y ****.

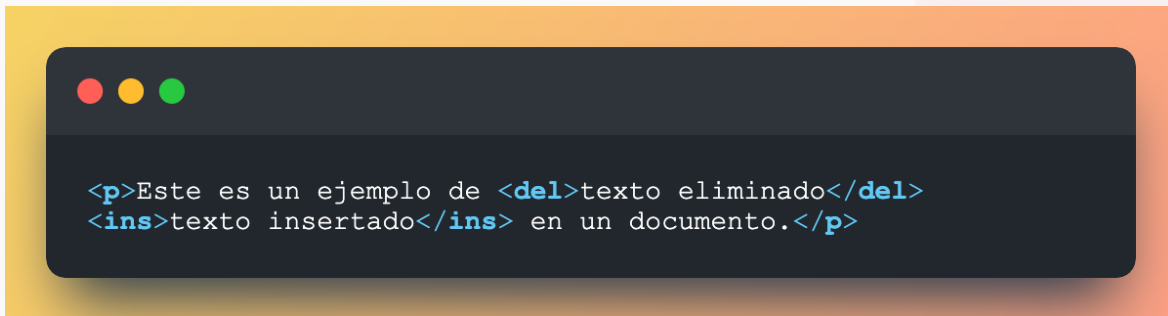
Estas son las etiquetas utilizadas para ediciones a detalle:

4.6.1 <ins>

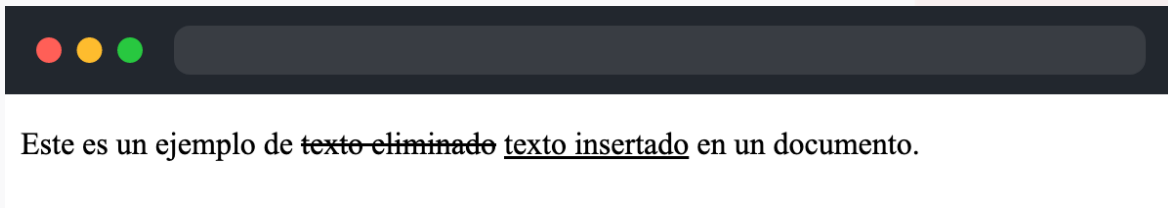
La etiqueta **<ins>** en HTML es utilizada para indicar una inserción en el contenido de un documento. Es decir, permite marcar un texto o una sección de un documento que ha sido agregado o insertado en una fecha específica. La etiqueta **<ins>** es similar a la etiqueta **** que se utiliza para indicar texto eliminado, pero mientras **** indica texto eliminado, **<ins>** indica texto insertado.

Ejemplo:

4 ELEMENTOS



El código se renderiza de esta manera al abrirlo en un navegador.

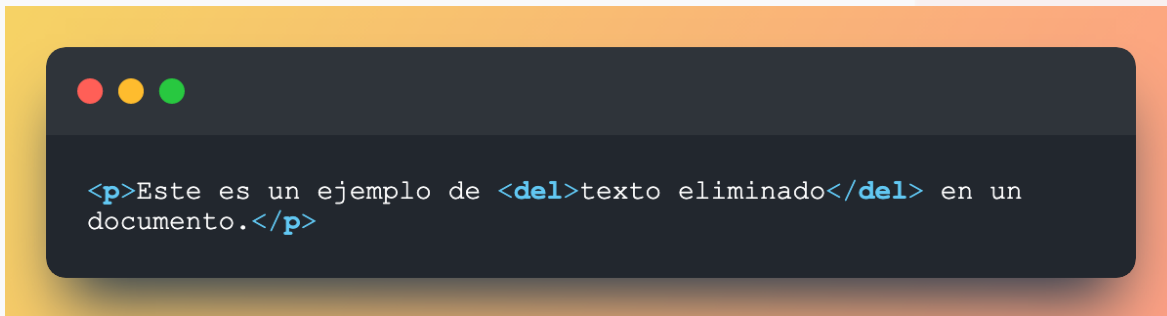


En este ejemplo se utilizan las etiquetas **** e **<ins>** para indicar que el texto “texto eliminado” fue borrado y el texto “texto insertado” fue agregado en una edición posterior, esto puede ser útil para llevar un registro de cambios en un documento o un sitio web, y es de gran utilidad en sistemas de colaboración y edición.

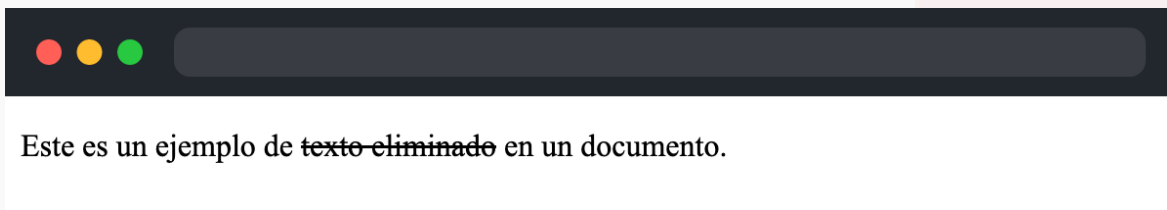
4.6.2

La etiqueta **** en HTML es utilizada para indicar contenido eliminado o tachado en un documento HTML. Esta etiqueta es útil para mostrar cambios o revisiones en un documento o sitio web, especialmente en sistemas de colaboración y edición en línea. Puede ser usada para indicar texto que ha sido eliminado y también se puede utilizar para indicar texto que ha sido eliminado y reemplazado con otro texto.

Ejemplo:



El código se renderiza de esta manera al abrirlo en un navegador.



En este ejemplo se utiliza la etiqueta **** para indicar que el texto “texto eliminado” fue eliminado en una edición posterior del documento. Al ver el ejemplo en el navegador este texto aparecера tachado para indicar su eliminación.

Además, la etiqueta **** también permite incluir un atributo “datetime” para especificar la fecha en la que el texto fue eliminado, y un atributo “cite” para especificar una URL que proporciona más contexto sobre la eliminación.

4.7 Incrustaciones

Las etiquetas de contenido incrustado en HTML (Embedded content) se utilizan para incluir contenido multimedia, como imágenes, audio, video, mapas, y aplicaciones, en una página web. Estas etiquetas permiten agregar elementos multimedia a una página sin tener que utilizar otras herramientas o scripts. Algunos ejemplos de etiquetas de contenido incrustado son: ****, **<audio>**, **<video>**, **<iframe>**, **<embed>**, **<object>**, entre otras.

4 ELEMENTOS

Al utilizar estas etiquetas, se puede integrar contenido multimedia en una página web sin tener que escribir código adicional, lo que facilita la creación de contenido dinámico y atractivo.

Un ejemplo es:



El código se renderiza de esta manera al abrirlo en un navegador.



Este ejemplo muestra cómo utilizar la etiqueta **<video>** para incluir un video en una página web, y utilizar la etiqueta **<source>** para especificar diferentes formatos del video (mp4, ogg) para diferentes navegadores.

Estas son las etiquetas utilizadas para contenido incrustado a detalle:

4.7.1 <picture>

La etiqueta **<picture>** en HTML es utilizada para mostrar una imagen adaptable a diferentes dispositivos y pantallas con diferentes resoluciones. La etiqueta **<picture>** es utilizada para especificar diferentes versiones de una imagen para diferentes tamaños de pantalla, mediante el uso de los atributos “srcset” y “sizes” . Con ella se puede especificar diferentes versiones de una imagen para diferentes resoluciones de pantalla, lo que mejora el rendimiento en dispositivos con pantallas más pequeñas o con conexiones más lentas, ya que se carga una imagen más pequeña, en vez

4 ELEMENTOS

de una imagen de mayor resolución que puede ser innecesaria para ese dispositivo o pantalla.

Ejemplo:

A code editor window with a dark background and orange border. It contains HTML code for the <picture> element. The code is as follows:

```
<picture>
  <source
    srcset="imagen-300.jpg"
    media="(max-width: 300px)">
  <source
    srcset="imagen-600.jpg"
    media="(max-width: 600px)">
  <source srcset="imagen-1200.jpg">
  
</picture>
```

El código se renderiza de esta manera al abrirlo en un navegador.



En este ejemplo se utiliza la etiqueta **<picture>** con las etiquetas **<source>** para especificar diferentes versiones de la imagen “image-300.jpg” , “image-600.jpg” y “image-1200.jpg” para diferentes tamaños de pantalla, con los atributos “srcset” y “media” se especifica cual imagen se cargará en función del tamaño de la pantalla. Finalmente se utiliza la etiqueta **** para mostrar la imagen, esta etiqueta se utiliza como “fallback” en caso de que alguno de los navegadores no soporte la etiqueta **<picture>**

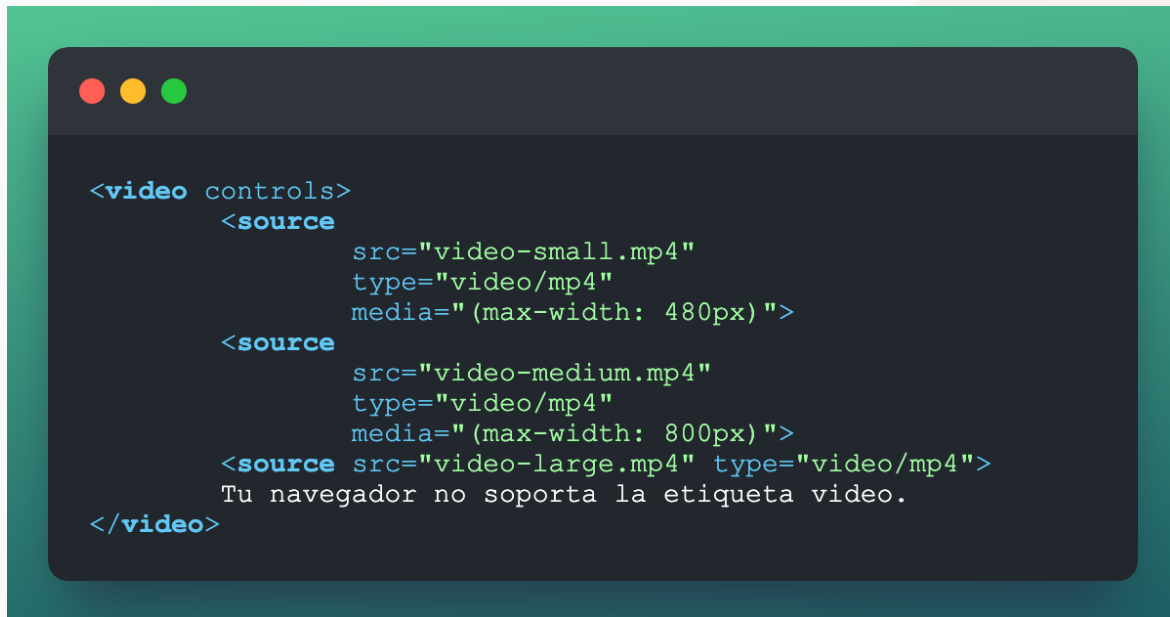
4.7.2 **<source>**

La etiqueta **<source>** en HTML es utilizada para especificar diferentes versiones de un recurso multimedia, como una imagen, audio o video, que se pueden utilizar en un elemento contenedor como “picture” , “audio” o “video” . El atributo “src”

4 ELEMENTOS

especifica la dirección URL del recurso, mientras que otros atributos como “type” y “media” permiten especificar qué tipo de recurso se está proporcionando y en qué dispositivos o pantallas debería ser utilizado.

Ejemplo:



```
<video controls>
  <source
    src="video-small.mp4"
    type="video/mp4"
    media="(max-width: 480px)">
  <source
    src="video-medium.mp4"
    type="video/mp4"
    media="(max-width: 800px)">
  <source src="video-large.mp4" type="video/mp4">
  Tu navegador no soporta la etiqueta video.
</video>
```

El código se renderiza de esta manera al abrirlo en un navegador.



En este ejemplo se utiliza la etiqueta **<source>** dentro de un elemento de **<video>** con el atributo “controls” para mostrar controles en el reproductor, se especifican diferentes versiones del video “video-small.mp4” , “video-medium.mp4” y “video-large.mp4” para diferentes tamaños de pantalla con los atributos “src” y “media” , así como el tipo de video con el atributo “type” el navegador escoge cual video cargar dependiendo del tamaño de pantalla.

En resumen, la etiqueta **<source>** permite especificar diferentes versiones de un recurso multimedia para diferentes dispositivos y pantallas, mejorando el rendimiento y la experiencia del usuario.

4.7.3

La etiqueta **** en HTML es utilizada para insertar imágenes en un documento HTML. El atributo “src” especifica la dirección URL de la imagen, mientras que otros

4 ELEMENTOS

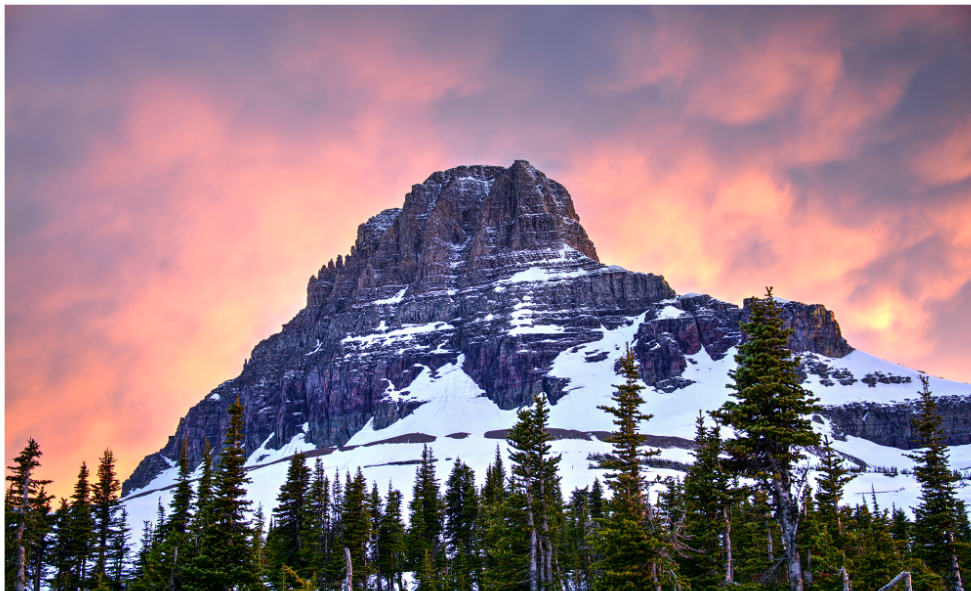
atributos como “alt” y “width” permiten especificar un texto alternativo y el tamaño de la imagen, respectivamente.

Ejemplo:

```

```

El código se renderiza de esta manera al abrirlo en un navegador.



En este ejemplo se utiliza la etiqueta **** para insertar una imagen en un docu-

4 ELEMENTOS

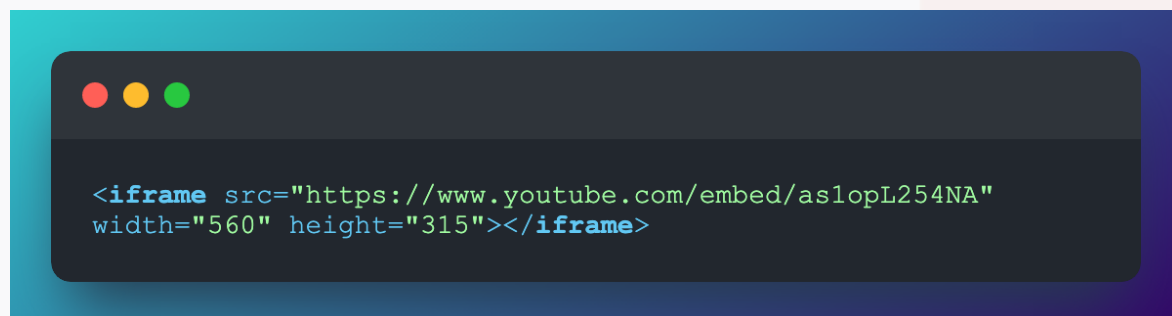
mento HTML. La dirección URL de la imagen se especifica en el atributo “src” y se especifica el texto alternativo “A beautiful landscape” en el atributo “alt” . El atributo “width” y “height” especifica el tamaño de la imagen, en este caso es de 500x300 pixels.

Con la etiqueta **** se puede insertar una imagen en cualquier parte de un documento HTML, ya sea como parte de un contenido de texto, una tabla, un enlace, un formulario, etc. Es una de las etiquetas más utilizadas para incorporar imágenes en una pagina web, es esencial que la imagen exista en el src especificado ya que si no existe se mostrará un icono de imagen no encontrada.

4.7.4 <iframe>

La etiqueta **<iframe>** en HTML es utilizada para insertar un documento HTML dentro de otro documento HTML. Es una forma de incluir contenido de otras páginas web o aplicaciones dentro de una página. El atributo “src” especifica la dirección URL del documento que se va a incluir, mientras que otros atributos como “width” y “height” permiten especificar el tamaño del iframe.

Ejemplo:



El código se renderiza de esta manera al abrirlo en un navegador.



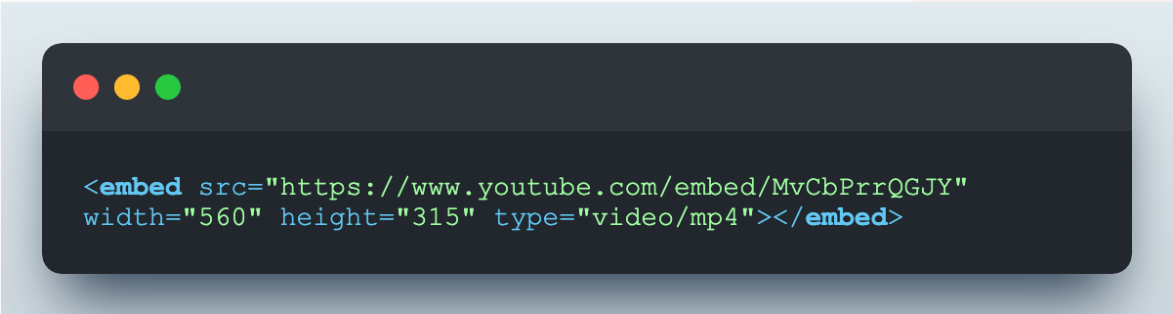
En este ejemplo se utiliza la etiqueta **<iframe>** para insertar un mapa en una página web. La dirección URL del mapa se especifica en el atributo “src” y se especifica el tamaño del iframe con los atributos “width” y “height” . En este caso es de 600x450 pixels.

Es importante mencionar que algunos navegadores bloquean o limitan el contenido de un **<iframe>** por seguridad, por ejemplo, algunos navegadores no permiten scripts ejecutarse dentro de un iframe que proviene de un dominio diferente al del sitio que lo contiene. A esto se lo llama CORS (Cross-Origin Resource Sharing).

4.7.5 <embed>

La etiqueta **<embed>** en HTML es utilizada para insertar contenido multimedia de terceros, como videos, audio, mapas, etc. Es similar a la etiqueta **<iframe>** pero en vez de mostrar una pagina completa, solo muestra el contenido multimedia específico. Es compatible con diferentes tipos de medios, como Flash, Quicktime, etc. El atributo “src” especifica la dirección URL del contenido multimedia a insertar, mientras que otros atributos como “width” y “height” permiten especificar el tamaño del embed.

Ejemplo:

A code editor window with a dark background and three colored window control buttons (red, yellow, green) in the top left corner. It displays an HTML code snippet for embedding a YouTube video.

```
<embed src="https://www.youtube.com/embed/MvCbPrrQGJY"
width="560" height="315" type="video/mp4"></embed>
```

El código se renderiza de esta manera al abrirlo en un navegador.



En este ejemplo se utiliza la etiqueta **<embed>** para insertar un video en una página web. La dirección URL del video se especifica en el atributo “src” y se especifica el tamaño del video con los atributos “width” y “height” , además se especifica el tipo de video con el atributo “type” .

Es importante mencionar que a diferencia de **<iframe>**, el navegador debe tener el plugin necesario para poder mostrar el contenido especificado en **<embed>**, si no lo tiene el navegador no podrá mostrar el contenido.

4.7.6 <object>

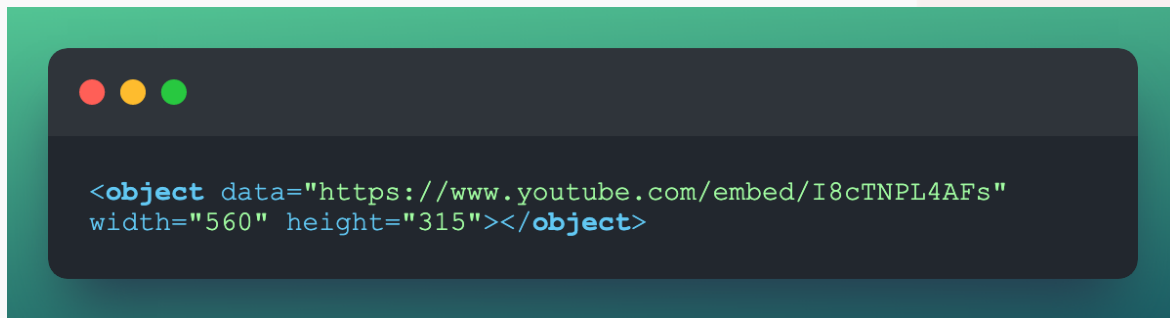
La etiqueta **<object>** en HTML es utilizada para insertar un objeto en un documento HTML. Los objetos pueden incluir cualquier tipo de contenido, como imágenes,

4 ELEMENTOS

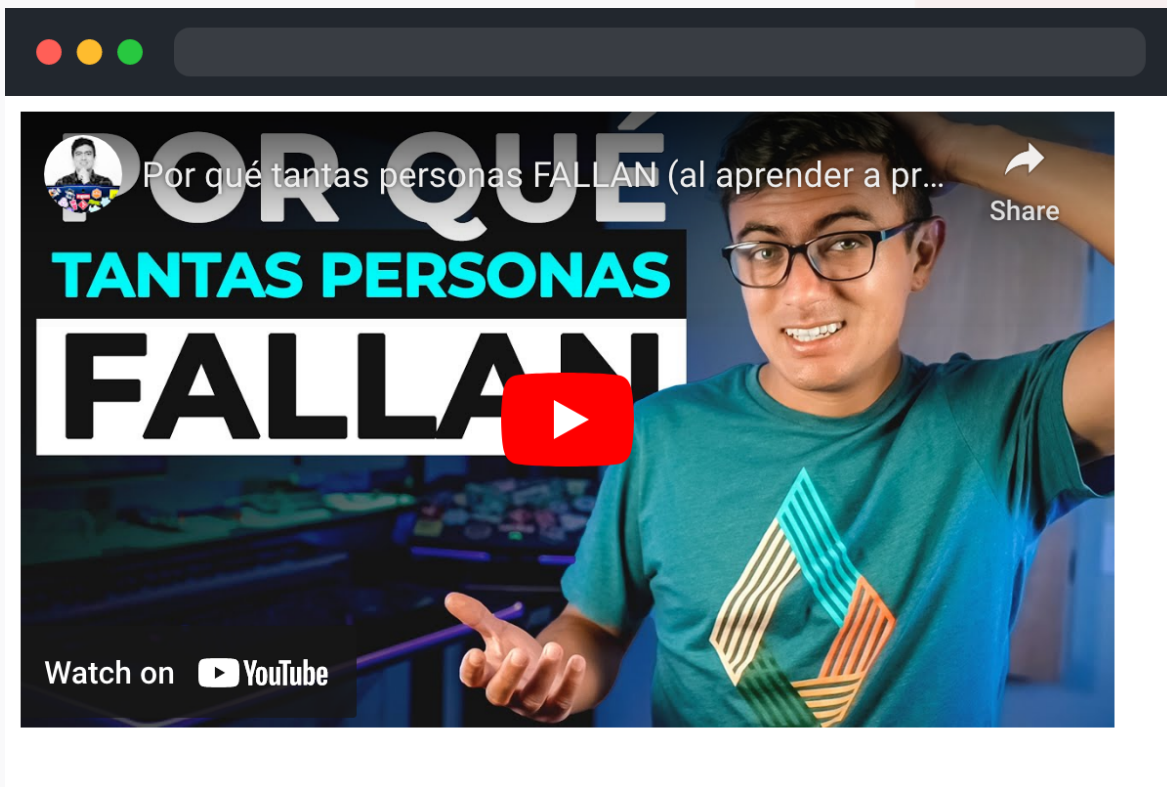
videos, audio, aplicaciones, etc. La etiqueta **<object>** es similar a “embed” , pero tiene algunas funciones adicionales.

El atributo “data” especifica la dirección URL del objeto, mientras que otros atributos como “width” y “height” permiten especificar el tamaño del objeto.

Ejemplo:



El código se renderiza de esta manera al abrirlo en un navegador.



En este ejemplo se utiliza la etiqueta **<object>** para insertar una aplicación de Flash en una página web. La dirección URL de la aplicación se especifica en el atributo “data” y se especifica el tamaño de la aplicación con los atributos “width” y “height” .

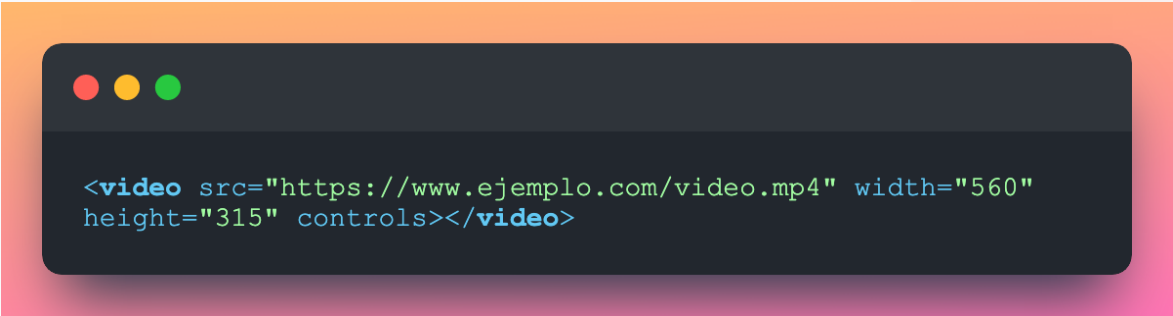
La etiqueta **<object>** es una forma más versátil de insertar contenido multimedia en una página web, ya que permite especificar una variedad de datos alternativos para los navegadores que no soportan el tipo de contenido especificado. Por ejemplo, si el navegador no soporta Flash, se puede especificar una imagen alternativa para que el navegador muestre.

4.7.7 <video>

La etiqueta **<video>** en HTML es utilizada para insertar videos en un documento HTML. Es una forma sencilla de incluir videos en una página web sin necesidad de utilizar etiquetas más complejas como **<object>** o **<embed>**. La etiqueta **<video>** es compatible con los formatos de video modernos como MP4, WebM, Ogg, etc.

El atributo “src” especifica la dirección URL del video, mientras que otros atributos como “width” y “height” permiten especificar el tamaño del video.

Ejemplo:

A code editor window with a dark background and a light blue border. It contains the following HTML code:

```
<video src="https://www.ejemplo.com/video.mp4" width="560" height="315" controls></video>
```

```
<video src="https://www.ejemplo.com/video.mp4" width="560" height="315" controls></video>
```

En este ejemplo se utiliza la etiqueta **<video>** para insertar un video en una página web. La dirección URL del video se especifica en el atributo “src” y se especifica el tamaño del video con los atributos “width” y “height”. El atributo “controls” permite al usuario controlar el video, como play, pause, volumen, entre otros.

La etiqueta **<video>** también acepta varios formatos de video dentro de ella, para que si el navegador no soporta el formato especificado en “src” utilice otro soportado.


```
<video controls>
  <source src="pelicula.mp4" type="video/mp4">
  <source src="pelicula.ogg" type="video/ogg">
  Tu navegador no soporta la etiqueta de video.
</video>
```

El código se renderiza de esta manera al abrirlo en un navegador.

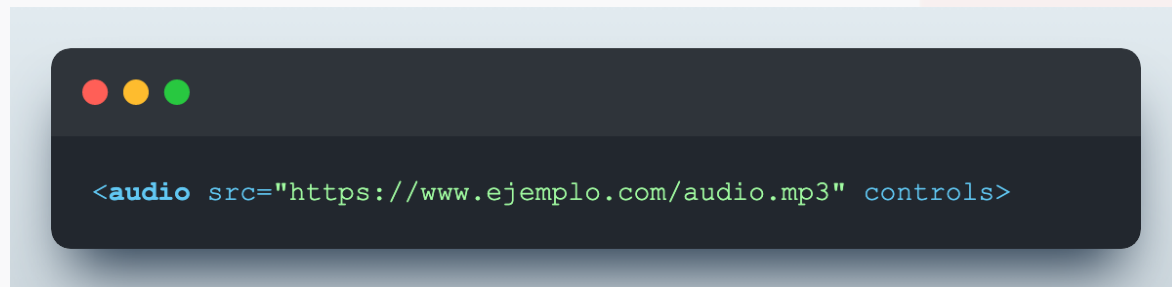


En este caso se está especificando 2 formatos de video diferentes, mp4 y ogg y si el navegador no soporta el primero utilizará el segundo.

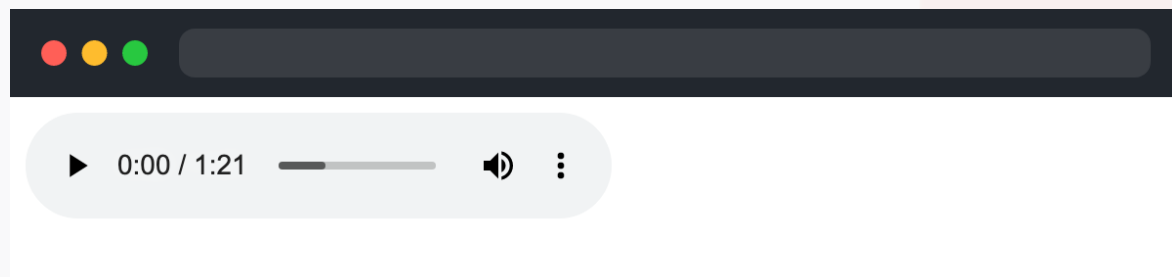
4.7.8 <audio>

La etiqueta **<audio>** en HTML es utilizada para insertar audio en un documento HTML. Es una forma sencilla de incluir audio en una página web sin necesidad de utilizar etiquetas más complejas como **<object>** o **<embed>**. La etiqueta **<audio>** es compatible con los formatos de audio modernos como MP3, Ogg, AAC, entre otros. El atributo “src” especifica la dirección URL del audio.

Ejemplo:

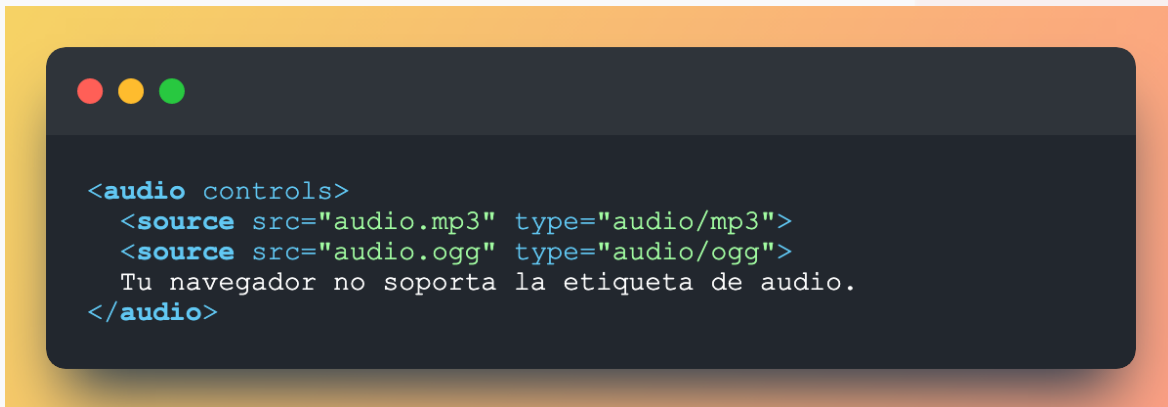


El código se renderiza de esta manera al abrirlo en un navegador.

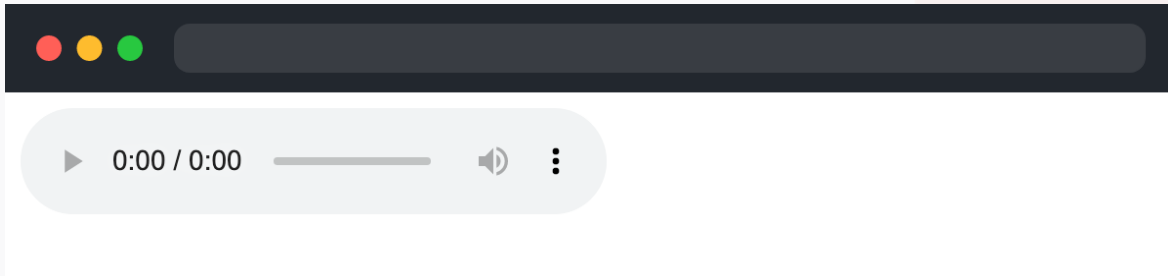


En este ejemplo se utiliza la etiqueta **<audio>** para insertar un audio en una página web. La dirección URL del audio se especifica en el atributo “src” y se le agrega el atributo “controls” para permitir que el usuario pueda controlar el audio como play, pause, volumen, entre otros.

También se pueden especificar varios formatos de audio dentro de la etiqueta para que el navegador utilice el que soporte.



El código se renderiza de esta manera al abrirlo en un navegador.

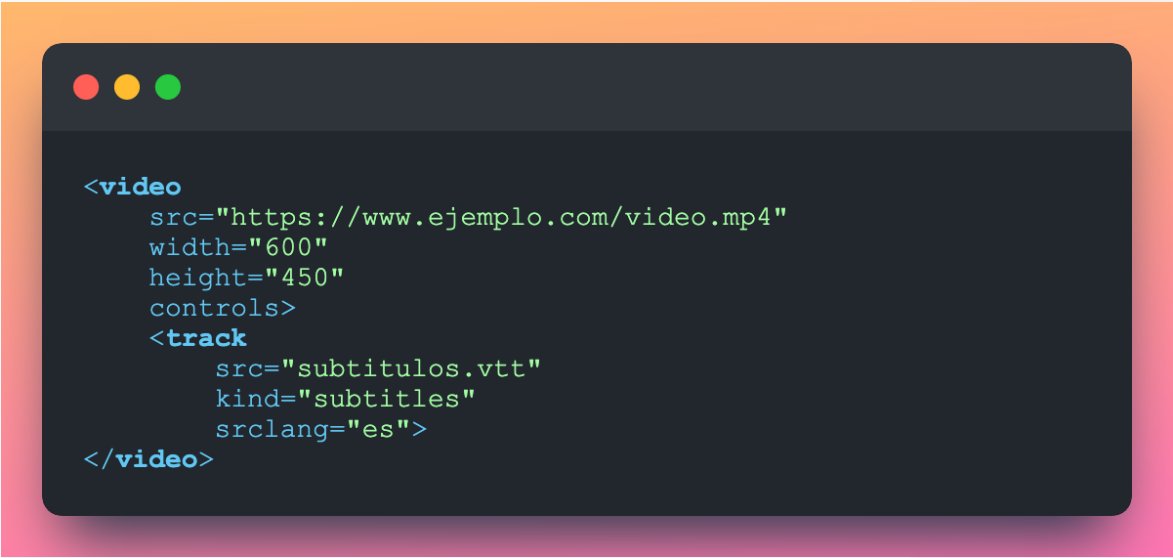


En este caso se está especificando 2 formatos de audio diferentes, mp3 y ogg y si el navegador no soporta el primero utilizará el segundo.

4.7.9 <track>

La etiqueta **<track>** en HTML es utilizada para añadir subtítulos, descripciones de audio o metadatos a un elemento multimedia como un video o un audio. Esta etiqueta se utiliza dentro de elementos como **<audio>** o **<video>** y su atributo “src” especifica la dirección URL del archivo de subtítulos o metadatos. Además, se puede especificar el idioma y el tipo de pista con los atributos “srclang” y “kind” respectivamente.

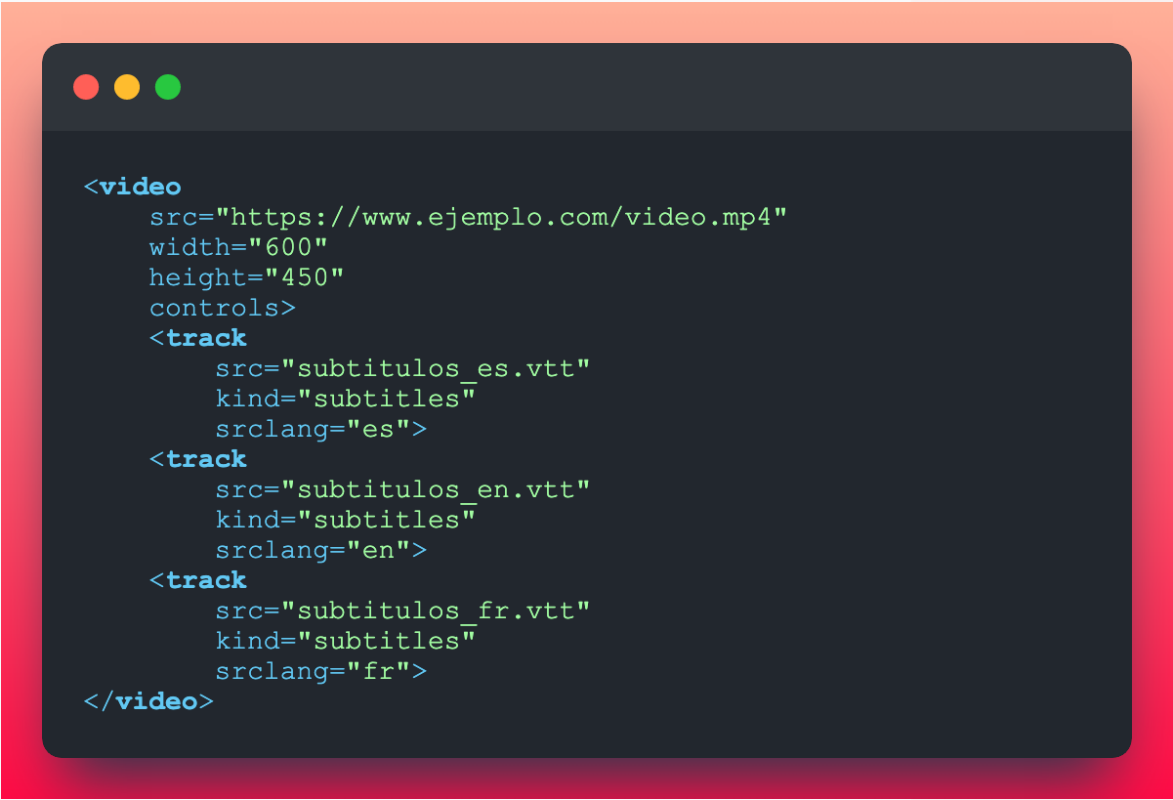
Ejemplo de uso de la etiqueta “track” para subtítulos:



```
<video
  src="https://www.ejemplo.com/video.mp4"
  width="600"
  height="450"
  controls>
  <track
    src="subtitulos.vtt"
    kind="subtitles"
    srclang="es">
</video>
```

En este ejemplo, la etiqueta **<track>** se utiliza para añadir subtítulos en español al video, El archivo de subtítulos es especificado en el atributo “src” y los atributos “kind” y “srclang” especifica el tipo de pista como subtítulos y el idioma como español respectivamente.

En caso de varios idiomas, se podrían agregar varios elementos **<track>** con distinto idioma:



```
<video
  src="https://www.ejemplo.com/video.mp4"
  width="600"
  height="450"
  controls>
  <track
    src="subtitulos_es.vtt"
    kind="subtitles"
    srclang="es">
  <track
    src="subtitulos_en.vtt"
    kind="subtitles"
    srclang="en">
  <track
    src="subtitulos_fr.vtt"
    kind="subtitles"
    srclang="fr">
</video>
```

De esa forma, el navegador seleccionará automáticamente el idioma que tenga el usuario configurado en su navegador.

Ejemplo de uso de la etiqueta **<track>** para descripciones de audio:



```
<audio
  src="https://www.ejemplo.com/audio.mp3"
  controls>
  <track
    src="descripcion_audio.vtt"
    kind="descriptions"
    srclang="es">
</audio>
```

En este ejemplo se utiliza la etiqueta **<track>** para añadir descripciones de audio al audio. El archivo de descripciones es especificado en el atributo “src” y los atributos “kind” y “srclang” especifica el tipo de pista como descripciones y el idioma como español respectivamente.

4.7.10 <map>

La etiqueta **<map>** en HTML es utilizada para crear una “mapa de imagen” o un mapa de enlace. Un mapa de imagen es una imagen con áreas específicas que se pueden hacer clic, y cada una de estas áreas se puede asociar con un enlace. El mapa se define en la etiqueta **<map>** y se asocia con la imagen usando el atributo “name” , y luego se define las áreas activas dentro de la etiqueta **<map>** utilizando la etiqueta **<area>**.

4.7.11 <area>

La etiqueta **<area>** en HTML se utiliza junto con la etiqueta **<map>** para definir las áreas activas dentro de un mapa de imagen. Una etiqueta **<area>** define una región

4 ELEMENTOS

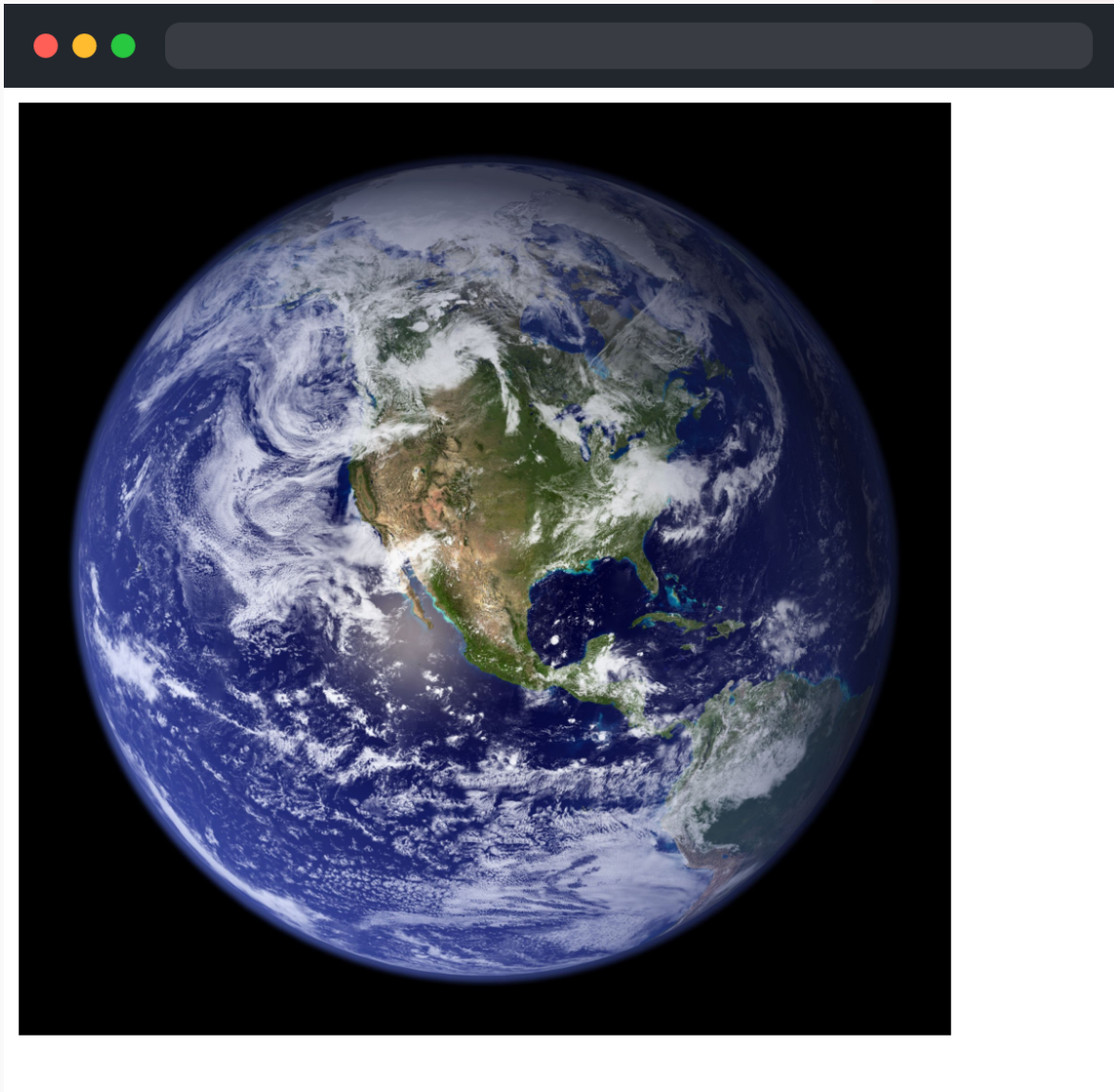
dentro de una imagen que se puede hacer clic y se puede asociar con un enlace. Cada etiqueta **<area>** debe tener un atributo “shape” que especifica la forma de la área activa, como rectángulos o círculos, y un atributo “coords” que especifica las coordenadas exactas de la área activa. También puede incluir un atributo “href” que especifica el enlace a seguir cuando se hace clic en la área activa, y un atributo “alt” que proporciona una descripción alternativa de la área activa para los usuarios de dispositivos que no pueden ver el mapa.

Un ejemplo de uso de las etiquetas **<map>** y **<area>** podría ser:

```

<map name="planeta">
  <area
    alt="Canada"
    title="Canada"
    href="https://www.state.gov/countries-areas/canada/"
    coords="707,796,1467,465"
    shape="rect">
  <area
    alt="Cuba"
    title="Cuba"
    href="https://www.state.gov/countries-areas/cuba/"
    coords="1443,1142,90"
    shape="circle">
  <area
    alt="Ecuador"
    title="Ecuador"
    href="https://www.state.gov/countries-areas/ecuador/"
    coords="1447,1491,1411,1554,1420,1616,1467,1632,1502"
    shape="poly">
</map>
```

El código se renderiza de esta manera al abrirlo en un navegador.



En este ejemplo se tiene una imagen de un mapa de un planeta con tres áreas específicas (Canada, Cuba, Ecuador) que se pueden hacer clic, y cada una de estas áreas se asocia con un enlace diferente a través del atributo “href” en la etiqueta **<area>**. El atributo “shape” especifica la forma de la área activa, como rectángulos o círculos, y el atributo “coords” especifica las coordenadas exactas de la área

activa.

Es importante mencionar que para las imagenes con mapa de enlace, es recomendable proporcionar una alternativa en el atributo “alt” ya que muchos dispositivos no soportan el mapa de enlace y los usuarios de estos dispositivos podrían no poder acceder a la imagen.

4.7.12 Tablas

Las etiquetas de datos tabulares en HTML (Tabular data) se utilizan para crear tablas de datos en una página web. Entre las etiquetas más comunes utilizadas para crear tablas se encuentran: **<table>**, **<thead>**, **<tbody>**, **<tfoot>**, **<tr>**, **<td>**, **<th>**, ente otras.

Este es un ejemplo de uso:


```
<table>
  <thead>
    <tr>
      <th>Encabezamiento 1</th>
      <th>Encabezamiento 2</th>
      <th>Encabezamiento 3</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>Datos 1</td>
      <td>Datos 2</td>
      <td>Datos 3</td>
    </tr>
    <tr>
      <td>Datos 4</td>
      <td>Datos 5</td>
      <td>Datos 6</td>
    </tr>
  </tbody>
</table>
```

El código se renderiza de esta manera al abrirlo en un navegador.

Encabezamiento 1	Encabezamiento 2	Encabezamiento 3
Datos 1	Datos 2	Datos 3
Datos 4	Datos 5	Datos 6

Este ejemplo crea una tabla con un encabezado y dos filas de datos. El encabezado está contenido dentro de la etiqueta **<thead>**, el cuerpo de la tabla dentro

4 ELEMENTOS

de **<tbody>** y cada celda de la tabla está contenida dentro de una etiqueta **<td>** o **<th>**.

Estas son las etiquetas utilizadas para tablas a detalle:

4.7.13 <table>

La etiqueta **<table>** en HTML se utiliza para crear tablas. Una tabla se compone de filas y columnas que contienen celdas. Cada celda puede contener texto, imágenes u otros elementos. Con las etiquetas **<thead>**, **<tbody>** y **<tfoot>** se pueden agrupar las filas de una tabla para facilitar su estilado y navegación.

Un ejemplo de uso de la etiqueta **<table>** podría ser:

```
<table>
  <thead>
    <tr>
      <th>Nombre</th>
      <th>Apellido</th>
      <th>Edad</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>Juan</td>
      <td>Pérez</td>
      <td>30</td>
    </tr>
    <tr>
      <td>Maria</td>
      <td>Garcia</td>
      <td>25</td>
    </tr>
  </tbody>
</table>
```

El código se renderiza de esta manera al abrirlo en un navegador.

Nombre	Apellido	Edad
Juan	Pérez	30
Maria	Garcia	25

En este ejemplo se tiene una tabla con tres columnas, “Nombre” , “Apellido” y “Edad” y dos filas con datos de dos personas.

4 ELEMENTOS

Notese que dentro de la tabla se utilizan las etiquetas **<thead>** para definir el encabezado de la tabla, **<tbody>** para el cuerpo de la tabla y **<tfoot>** para el pie de la tabla. Las etiquetas **<th>** se usan para los encabezados de las columnas y las etiquetas **<td>** para los datos de las celdas.

Es importante mencionar que las tablas pueden tener clases, id, atributos y estilos CSS especificos para mejorar la estetica y usabilidad.

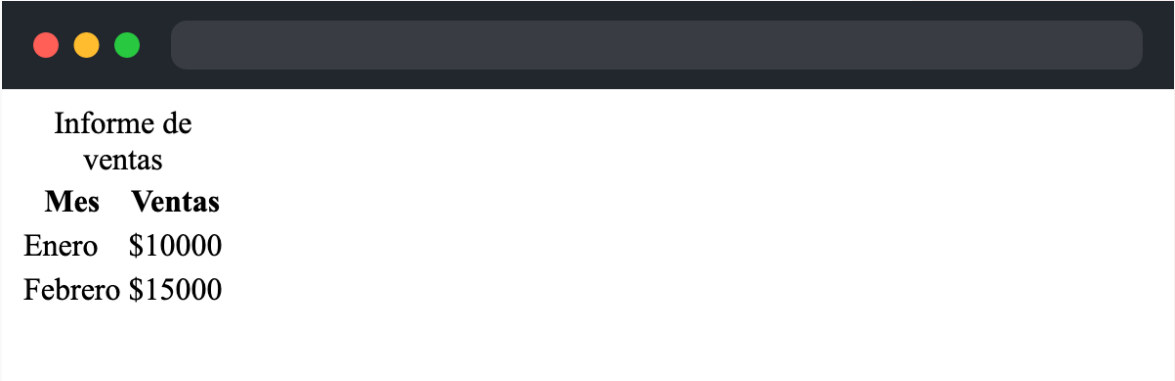
4.7.14 <caption>

La etiqueta **<caption>** en HTML se utiliza para agregar un título o descripción a una tabla. Esta etiqueta debe contenerse dentro de la etiqueta **<table>** y su contenido aparecerá en la parte superior o inferior de la tabla, dependiendo de la configuración de estilos CSS.

Un ejemplo de uso de la etiqueta **<caption>** podría ser:

```
<table>
  <caption>Informe de ventas</caption>
  <thead>
    <tr>
      <th>Mes</th>
      <th>Ventas</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>Enero</td>
      <td>$10000</td>
    </tr>
    <tr>
      <td>Febrero</td>
      <td>$15000</td>
    </tr>
  </tbody>
</table>
```

El código se renderiza de esta manera al abrirlo en un navegador.



Informe de
ventas

Mes	Ventas
Enero	\$10000
Febrero	\$15000

En este ejemplo se tiene una tabla con dos columnas, “Mes” y “Ventas” y dos filas con datos de las ventas mensuales. La etiqueta **<caption>** contiene el título de la

4 ELEMENTOS

tabla “Informe de ventas”

La etiqueta **<caption>** es opcional pero recomendada ya que permite especificar un título para una tabla, siendo una buena práctica para la accesibilidad.

4.7.15 <colgroup>

La etiqueta **<colgroup>** en HTML se utiliza para agrupar un conjunto de columnas en una tabla y aplicarles estilos o atributos comunes de forma simultánea. Esta etiqueta debe contenerse dentro de la etiqueta **<table>** y se utiliza en conjunto con las etiquetas **<col>** que indican las columnas a las que se aplican las configuraciones.

Un ejemplo de uso de la etiqueta **<colgroup>** podría ser:

```
<table>
  <colgroup span="2" style="background-color:lightgray">
</colgroup>
  <thead>
    <tr>
      <th>Mes</th>
      <th>Ventas</th>
      <th>Gastos</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>Enero</td>
      <td>$10000</td>
      <td>$5000</td>
    </tr>
    <tr>
      <td>Febrero</td>
      <td>$15000</td>
      <td>$8000</td>
    </tr>
  </tbody>
</table>
```

El código se renderiza de esta manera al abrirlo en un navegador.

Mes	Ventas	Gastos
Enero	\$10000	\$5000
Febrero	\$15000	\$8000

En este ejemplo se tiene una tabla con tres columnas, “Mes”, “Ventas” y “Gastos” y

4 ELEMENTOS

dos filas con datos de las ventas mensuales y los gastos. La etiqueta **<colgroup>** indica que las primeras dos columnas tendrán el estilo de fondo gris claro (lightgray)

Notese que si se utiliza la etiqueta **<colgroup>** y se le da un atributo span, no se necesita utilizar la etiqueta **<col>**, ya que **<colgroup>** tiene un atributo span para indicar cuantas columnas se estan agrupando.

En resumen, el uso de etiqueta **<colgroup>** permite agrupar columnas en una tabla para aplicar configuraciones a varias columnas de forma simultánea, facilitando la gestión de estilos y mejorando la legibilidad del código.

4.7.16 <col>

La etiqueta **<col>** en HTML se utiliza para especificar propiedades para una o varias columnas de una tabla. A diferencia de la etiqueta **<colgroup>** que agrupa un conjunto de columnas, la etiqueta **<col>** se utiliza para configurar una o varias columnas específicas de una tabla. Esta etiqueta debe contenerse dentro de la etiqueta **<table>** y se utiliza en conjunto con las etiquetas **<colgroup>** para aplicar estilos o atributos comunes a varias columnas de forma simultánea.

Un ejemplo de uso de la etiqueta **<col>** podría ser:


```
<table>
  <colgroup>
    <col span="1" style="background-color:lightgray"></col>
    <col span="2"></col>
  </colgroup>
  <thead>
    <tr>
      <th>Mes</th>
      <th>Ventas</th>
      <th>Gastos</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>Enero</td>
      <td>$10000</td>
      <td>$5000</td>
    </tr>
    <tr>
      <td>Febrero</td>
      <td>$15000</td>
      <td>$8000</td>
    </tr>
  </tbody>
</table>
```

El código se renderiza de esta manera al abrirlo en un navegador.

Mes	Ventas	Gastos
Enero	\$10000	\$5000
Febrero	\$15000	\$8000

4 ELEMENTOS

En este ejemplo se tiene una tabla con tres columnas, “Mes”, “Ventas” y “Gastos” y dos filas con datos de las ventas mensuales y los gastos. La etiqueta **<colgroup>** indica que las primeras dos columnas no tendrán estilo y la primera columna tendrá fondo gris claro (lightgray)

Notese que si se utiliza la etiqueta **<colgroup>** y se le da un atributo span, no se necesita utilizar la etiqueta **<col>**, ya que **<colgroup>** tiene un atributo span para indicar cuantas columnas se están agrupando.

En resumen, El uso de etiqueta **<col>** permite especificar configuraciones para una o varias columnas específicas en una tabla, facilitando la gestión de estilos y mejorando la legibilidad del código.

4.7.17 <tbody>

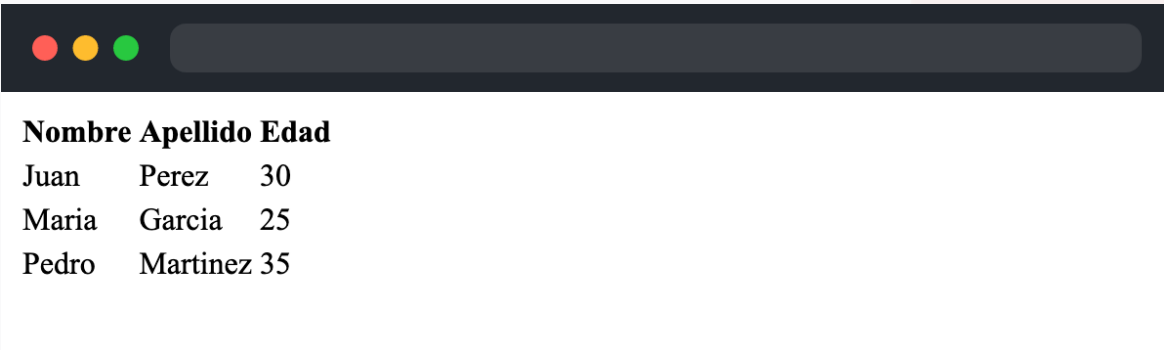
La etiqueta **<tbody>** en HTML se utiliza para agrupar el contenido de las filas de una tabla que representan el cuerpo de la misma. Es recomendado utilizar esta etiqueta junto con la etiqueta **<thead>** para separar el encabezado de la tabla del cuerpo de la misma. El uso de la etiqueta **<tbody>** ayuda a mejorar la estructura y legibilidad del código, y permite aplicar estilos o atributos específicos a las filas que contienen el cuerpo de la tabla.

Un ejemplo de uso de la etiqueta **<tbody>** podría ser:

```
<table>
  <thead>
    <tr>
      <th>Nombre</th>
      <th>Apellido</th>
      <th>Edad</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>Juan</td>
      <td>Perez</td>
      <td>30</td>
    </tr>
    <tr>
      <td>Maria</td>
      <td>Garcia</td>
      <td>25</td>
    </tr>
    <tr>
      <td>Pedro</td>
      <td>Martinez</td>
      <td>35</td>
    </tr>
  </tbody>
</table>
```

El código se renderiza de esta manera al abrirlo en un navegador.

4 ELEMENTOS



Nombre	Apellido	Edad
Juan	Perez	30
Maria	Garcia	25
Pedro	Martinez	35

En este ejemplo se tiene una tabla con tres columnas “Nombre”, “Apellido” y “Edad” y tres filas con datos de personas. La etiqueta **<thead>** indica que la primera fila es el encabezado de la tabla y la etiqueta **<tbody>** indica que las siguientes filas son el cuerpo de la tabla, es decir, contienen los datos.

En resumen, la etiqueta **<tbody>** es una etiqueta opcional pero recomendada para agrupar el contenido de las filas de una tabla que representan el cuerpo de la misma, lo que ayuda a mejorar la estructura y legibilidad del código.

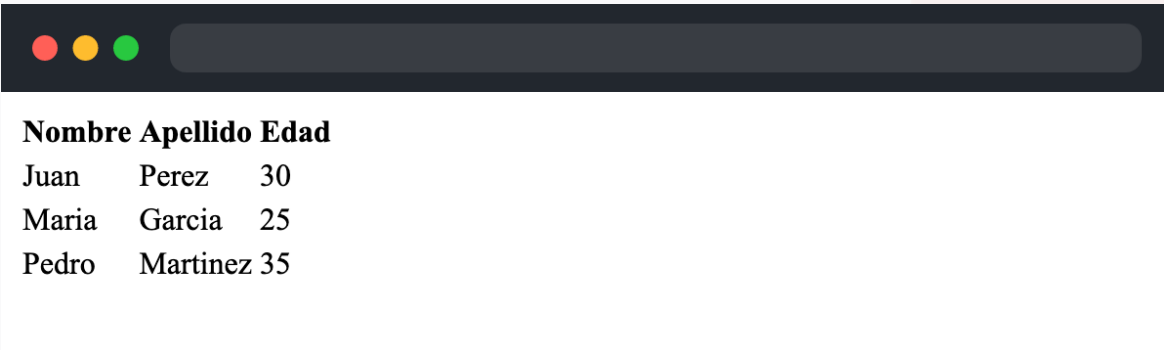
4.7.18 <thead>

La etiqueta **<thead>** en HTML se utiliza para agrupar el contenido de las filas de una tabla que representan el encabezado de la misma. Es recomendado utilizar esta etiqueta junto con la etiqueta **<tbody>** para separar el encabezado de la tabla del cuerpo de la misma. El uso de la etiqueta **<thead>** ayuda a mejorar la estructura y legibilidad del código, y permite aplicar estilos o atributos específicos a las filas que contienen el encabezado de la tabla.

Un ejemplo de uso de la etiqueta **<thead>** podría ser:

```
<table>
  <thead>
    <tr>
      <th>Nombre</th>
      <th>Apellido</th>
      <th>Edad</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>Juan</td>
      <td>Perez</td>
      <td>30</td>
    </tr>
    <tr>
      <td>Maria</td>
      <td>Garcia</td>
      <td>25</td>
    </tr>
    <tr>
      <td>Pedro</td>
      <td>Martinez</td>
      <td>35</td>
    </tr>
  </tbody>
</table>
```

El código se renderiza de esta manera al abrirlo en un navegador.



Nombre	Apellido	Edad
Juan	Perez	30
Maria	Garcia	25
Pedro	Martinez	35

En este ejemplo se tiene una tabla con tres columnas “Nombre”, “Apellido” y “Edad” y tres filas con datos de personas. La etiqueta **<thead>** indica que la primera fila es el encabezado de la tabla y la etiqueta **<tbody>** indica que las siguientes filas son el cuerpo de la tabla, es decir, contienen los datos.

En resumen, la etiqueta **<thead>** es una etiqueta opcional pero recomendada para agrupar el contenido de las filas de una tabla que representan el encabezado de la misma, lo que ayuda a mejorar la estructura y legibilidad del código.

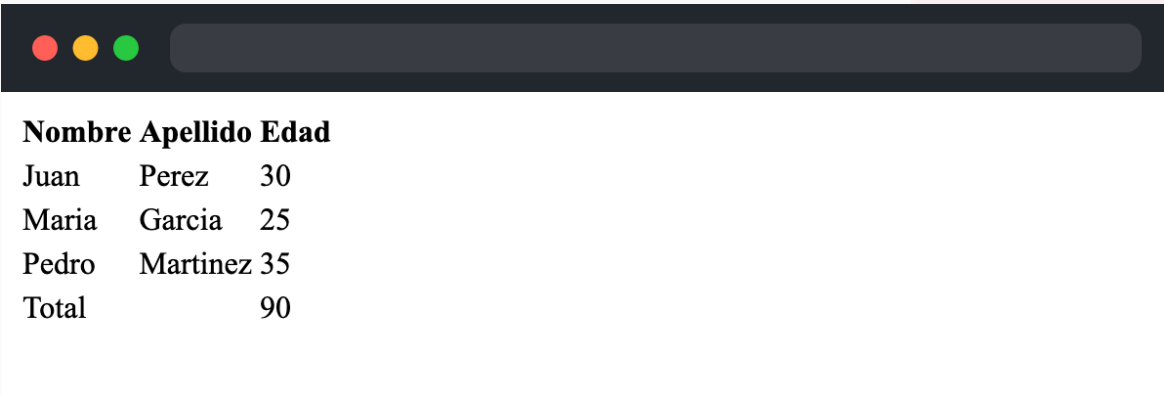
4.7.19 <tfoot>

La etiqueta **<tfoot>** en HTML se utiliza para agrupar el contenido de las filas de una tabla que representan el pie de la misma. Es recomendado utilizar esta etiqueta junto con la etiqueta **<thead>** y **<tbody>** para separar el encabezado, cuerpo y pie de una tabla. El uso de la etiqueta **<tfoot>** ayuda a mejorar la estructura y legibilidad del código, y permite aplicar estilos o atributos específicos a las filas que contienen el pie de la tabla.

Un ejemplo de uso de la etiqueta **<tfoot>** podría ser:

```
<table>
  <thead>
    <tr>
      <th>Nombre</th>
      <th>Apellido</th>
      <th>Edad</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>Juan</td>
      <td>Perez</td>
      <td>30</td>
    </tr>
    <tr>
      <td>Maria</td>
      <td>Garcia</td>
      <td>25</td>
    </tr>
    <tr>
      <td>Pedro</td>
      <td>Martinez</td>
      <td>35</td>
    </tr>
  </tbody>
  <tfoot>
    <tr>
      <td>Total</td>
      <td></td>
      <td>90</td>
    </tr>
  </tfoot>
</table>
```

El código se renderiza de esta manera al abrirlo en un navegador.



Nombre	Apellido	Edad
Juan	Perez	30
Maria	Garcia	25
Pedro	Martinez	35
Total		90

En este ejemplo se tiene una tabla con tres columnas “Nombre”, “Apellido” y “Edad” y tres filas con datos de personas. La etiqueta **<thead>** indica que la primera fila es el encabezado de la tabla, la etiqueta **<tbody>** indica que las siguientes filas son el cuerpo de la tabla, es decir, contienen los datos y la etiqueta **<tfoot>** indica que la ultima fila es el pie de la tabla, en este caso, el total.

En resumen, la etiqueta “**<tfoot>**” es una etiqueta opcional pero recomendada para agrupar el contenido de las filas de una tabla que representan el pie de la misma, lo que ayuda a mejorar la estructura y legibilidad del código.

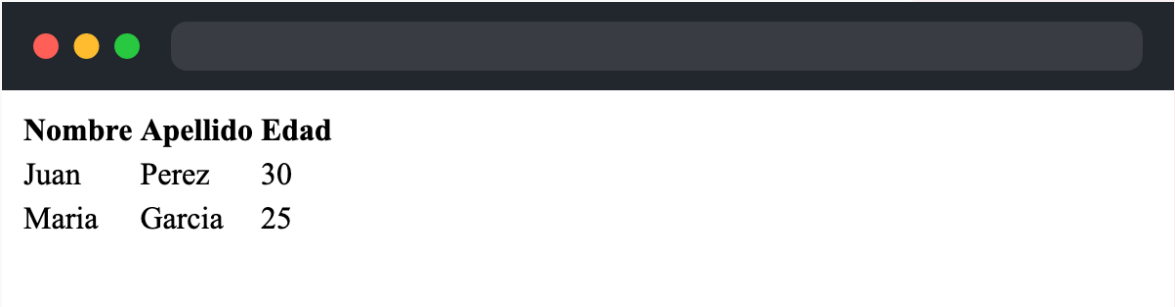
4.7.20 <tr>

La etiqueta **<tr>** en HTML se utiliza para definir una fila en una tabla. Dentro de esta etiqueta se deben agregar las etiquetas **<td>** o **<th>** que representan cada una de las celdas de la fila. La etiqueta **<tr>** es una etiqueta obligatoria para crear una tabla válida en HTML.

Un ejemplo de uso de la etiqueta **<tr>** podría ser:


```
<table>
  <tr>
    <th>Nombre</th>
    <th>Apellido</th>
    <th>Edad</th>
  </tr>
  <tr>
    <td>Juan</td>
    <td>Perez</td>
    <td>30</td>
  </tr>
  <tr>
    <td>Maria</td>
    <td>Garcia</td>
    <td>25</td>
  </tr>
</table>
```

El código se renderiza de esta manera al abrirlo en un navegador.



Nombre	Apellido	Edad
Juan	Perez	30
Maria	Garcia	25

En este ejemplo se tiene una tabla con tres columnas “Nombre”, “Apellido” y “Edad” y dos filas con datos de personas. La primera fila se define utilizando la etiqueta **<tr>** junto con las etiquetas **<th>** que indica que son celdas de encabezado, mientras que las siguientes filas utilizan etiquetas **<tr>** junto con las etiquetas **<td>** que indican celdas de datos regulares.

4 ELEMENTOS

En resumen, la etiqueta **<tr>** es una etiqueta obligatoria para definir una fila en una tabla y dentro de ella se deben agregar las etiquetas **<td>** o **<th>** que representan cada una de las celdas de la fila.

4.7.21 <td>

La etiqueta **<td>** en HTML se utiliza para definir una celda de datos en una tabla. Una celda es una unidad de contenido dentro de una fila. Cada celda puede contener diferentes tipos de contenido como texto, imágenes, enlaces, etc. La etiqueta **<td>** es una etiqueta obligatoria para crear una tabla válida en HTML y debe estar dentro de una etiqueta **<tr>** que define la fila en la que se encuentra la celda.

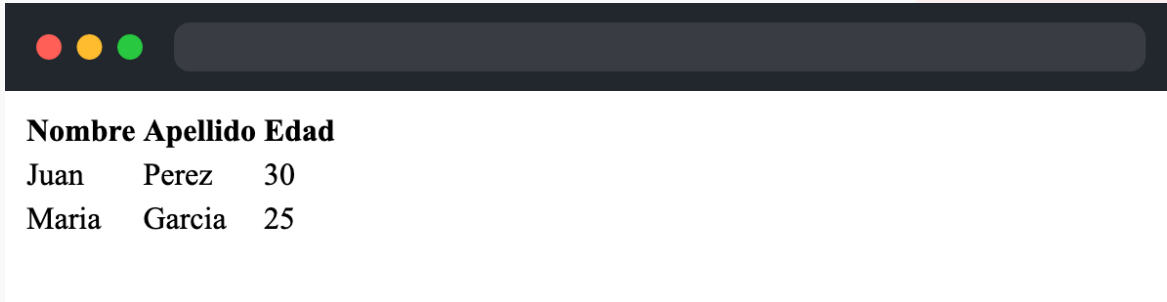
Un ejemplo de uso de la etiqueta **<td>** podría ser:



```
<table>
  <tr>
    <th>Nombre</th>
    <th>Apellido</th>
    <th>Edad</th>
  </tr>
  <tr>
    <td>Juan</td>
    <td>Perez</td>
    <td>30</td>
  </tr>
  <tr>
    <td>Maria</td>
    <td>Garcia</td>
    <td>25</td>
  </tr>
</table>
```

4 ELEMENTOS

El código se renderiza de esta manera al abrirlo en un navegador.



Nombre	Apellido	Edad
Juan	Perez	30
Maria	Garcia	25

En este ejemplo se tiene una tabla con tres columnas “Nombre”, “Apellido” y “Edad” y dos filas con datos de personas. Dentro de cada fila se encuentran las celdas de datos cada una con un valor diferente, las cuales están definidas utilizando la etiqueta **<td>**.

En resumen, la etiqueta **<td>** es una etiqueta obligatoria para definir una celda de datos en una tabla y debe estar dentro de una etiqueta **<tr>** que define la fila en la que se encuentra la celda. Esta contiene el contenido de cada celda de la tabla, puede contener diferentes tipos de contenido como texto, imágenes, enlaces, etc.

4.7.22 <th>

La etiqueta **<th>** en HTML se utiliza para definir una celda de encabezado en una tabla. Una celda de encabezado es una unidad de contenido que se utiliza para proporcionar un título o descripción para una columna o una fila de una tabla. La etiqueta **<th>** se utiliza normalmente para definir el contenido de la primera fila o primera columna de una tabla, y suele tener un diseño diferente al de las celdas de datos normales (td) para destacar su función especial.

Un ejemplo de uso de la etiqueta **<th>** podría ser:

```
<table>
  <tr>
    <th>Nombre</th>
    <th>Apellido</th>
    <th>Edad</th>
  </tr>
  <tr>
    <td>Juan</td>
    <td>Perez</td>
    <td>30</td>
  </tr>
  <tr>
    <td>Maria</td>
    <td>Garcia</td>
    <td>25</td>
  </tr>
</table>
```

El código se renderiza de esta manera al abrirlo en un navegador.

Nombre	Apellido	Edad
Juan	Perez	30
Maria	Garcia	25

En este ejemplo se tiene una tabla con tres columnas “Nombre”, “Apellido” y “Edad” y dos filas con datos de personas. Dentro de la primera fila se encuentran las celdas de encabezado cada una con un valor diferente, las cuales están definidas utilizando la etiqueta `<th>`.

En resumen, La etiqueta **<th>** es utilizada para definir una celda de encabezado en una tabla. Proporciona un título o descripción para una columna o una fila de una tabla. Es utilizado normalmente para definir el contenido de la primera fila o primera columna de una tabla, y suele tener un diseño diferente al de las celdas de datos normales (td) para destacar su función especial.

4.8 Formularios

Las etiquetas de formularios en HTML (Forms) se utilizan para crear formularios en una página web. Los formularios son una herramienta esencial para permitir a los usuarios interactuar con un sitio web, ya sea para enviar información, buscar contenido o realizar compras. Entre las etiquetas más comunes utilizadas para crear formularios se encuentran: **<table>**, **<input>**, **<label>**, **<select>**, **<option>**, **<textarea>**, **<button>**, **<fieldset>**.

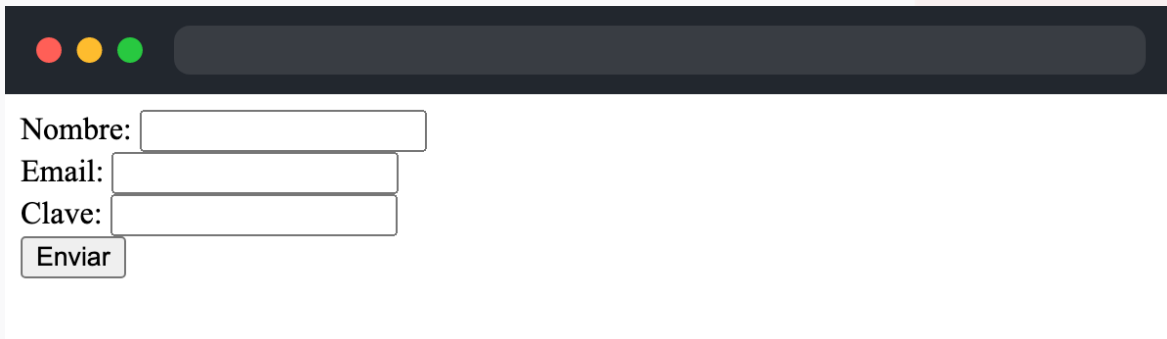
Ejemplo:

A screenshot of a code editor window with a dark background and light-colored text. The code is HTML for a form with three input fields: a text field for 'Nombre', an email field for 'Email', and a password field for 'Clave'. Each field is preceded by a label. There is a submit button labeled 'Enviar'. The code is enclosed in a <form> tag.

```
<form>
  <label for="nombre">Nombre:</label>
  <input type="text" id="nombre" name="nombre">
  <br>
  <label for="email">Email:</label>
  <input type="email" id="email" name="email">
  <br>
  <label for="clave">Clave:</label>
  <input type="password" id="clave" name="clave">
  <br>
  <input type="submit" value="Enviar">
</form>
```

El código se renderiza de esta manera al abrirlo en un navegador.

4 ELEMENTOS

A screenshot of a web browser window showing a simple form. The browser's title bar is dark with three colored window control buttons (red, yellow, green) on the left. The form itself is white and contains three stacked text input fields. The first field is preceded by the label 'Nombre:', the second by 'Email:', and the third by 'Clave:'. Below the third field is a button with the text 'Enviar'.

Este ejemplo muestra como crear un formulario simple con tres campos de entrada para introducir nombre, email y contraseña y un botón de enviar. Cada campo de entrada está precedido por una etiqueta **<label>** para indicar el propósito del campo. La etiqueta **<form>** es la etiqueta principal del formulario.

Estas son las etiquetas utilizadas para formularios a detalle:

4.8.1 **<form>**

La etiqueta **<form>** en HTML se utiliza para crear un formulario web. Los formularios permiten a los usuarios ingresar información en un sitio web. El formulario se compone de campos de entrada (como cajas de texto, botones de opción, etc.) y elementos de acción (como botones de envío o de restablecimiento). Los datos ingresados en el formulario se envían al servidor para su procesamiento.

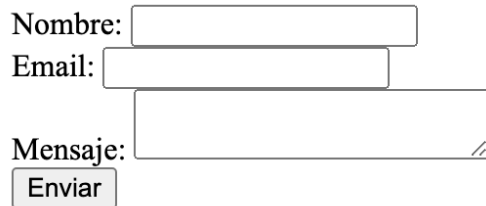
Un ejemplo de uso práctico de un formulario podría ser un formulario de contacto en una página web de negocios, donde los visitantes pueden ingresar su nombre, correo electrónico y mensaje, y luego hacer clic en un botón “Enviar” para enviar la información al propietario del sitio.

Ejemplo:

4 ELEMENTOS

```
<form action="enviar-contacto.php" method="post">
  <label for="nombre">Nombre:</label>
  <input type="text" id="nombre" name="nombre">
  <br>
  <label for="email">Email:</label>
  <input type="email" id="email" name="email">
  <br>
  <label for="mensaje">Mensaje:</label>
  <textarea id="mensaje" name="mensaje"></textarea>
  <br>
  <input type="submit" value="Enviar">
</form>
```

El código se renderiza de esta manera al abrirlo en un navegador.



Nombre:

Email:

Mensaje:

La forma en que se envía la información del formulario es a través de la atributo “action” y “method”, en este ejemplo se esta enviando la información a través de un archivo PHP llamado “send-contact.php” y el método de envío es “post” .

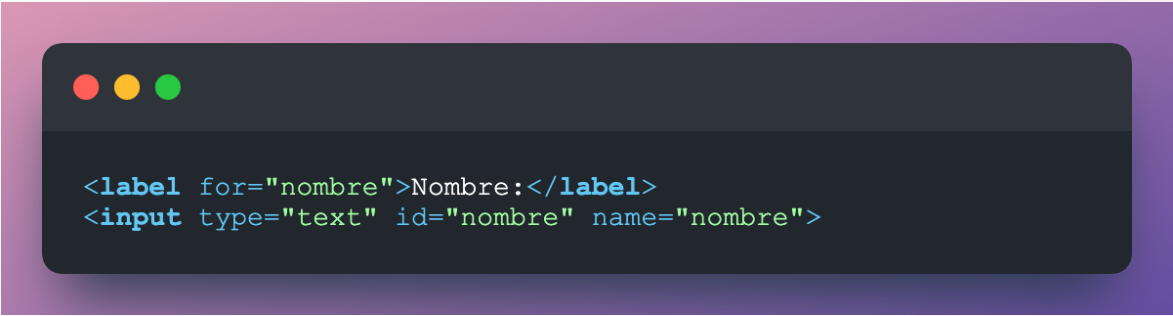
4 ELEMENTOS

4.8.2 <label>

La etiqueta **<label>** en HTML se utiliza para describir el propósito de un elemento de formulario. El texto dentro de la etiqueta se muestra como una etiqueta para el elemento de formulario, y hace que sea más fácil para los usuarios entender qué información deben ingresar en un campo específico.

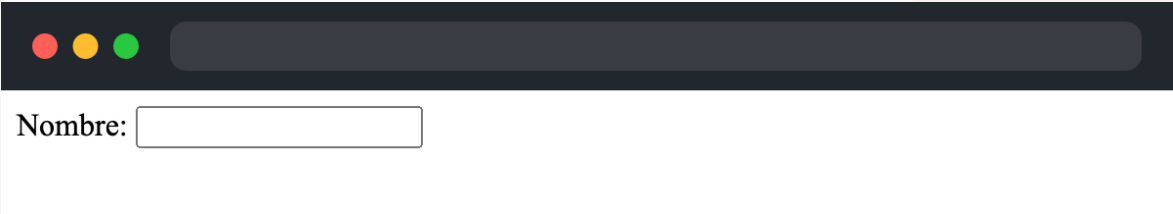
Una etiqueta puede ser asociada a un elemento de formulario utilizando el atributo “for” , donde el valor del atributo es el valor del atributo “id” del elemento de formulario que se desea etiquetar. Esto permite que el navegador asocie el elemento de formulario con su etiqueta, lo que mejora la accesibilidad y la usabilidad.

Un ejemplo práctico de una etiqueta seria en un formulario de una pagina de compra donde se pide el nombre del cliente

A screenshot of a code editor with a dark background and a purple gradient border. It shows two lines of HTML code: `<label for="nombre">Nombre:</label>` and `<input type="text" id="nombre" name="nombre">`.

```
<label for="nombre">Nombre:</label>
<input type="text" id="nombre" name="nombre">
```

El código se renderiza de esta manera al abrirlo en un navegador.

A screenshot of a web browser window. The address bar is empty. Below it, the text "Nombre:" is followed by a text input field.

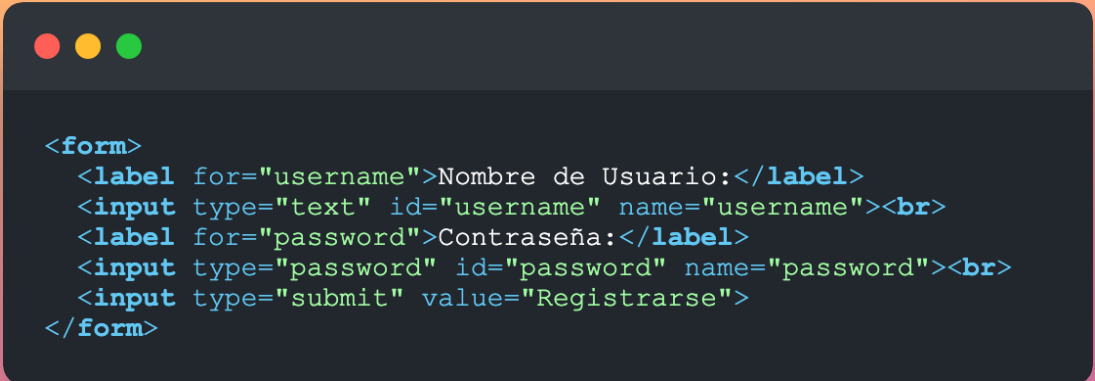
Nombre:

En este caso el texto “Nombre:” es la etiqueta que se le asocia al campo de texto para que el usuario entienda que allí debe ingresar su nombre.

4.8.3 <input>

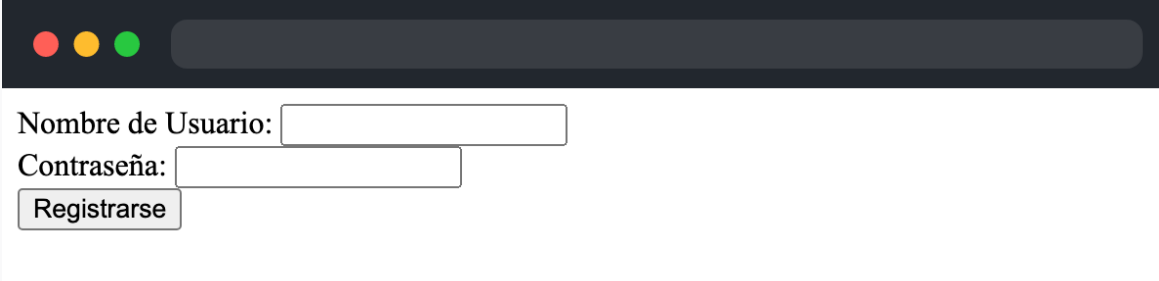
La etiqueta **<input>** en HTML se utiliza para crear un campo de entrada para un formulario web. Puede ser utilizado para crear diferentes tipos de campos de entrada, como cajas de texto, botones de opción, casillas de verificación, deslizadores, y más. El tipo de campo de entrada se especifica mediante el atributo “type”, y los valores válidos para este atributo incluyen “text”, “password”, “radio”, “checkbox”, “submit”, entre otros.

Un ejemplo de uso práctico podría ser en un formulario de registro de usuario:



```
<form>
  <label for="username">Nombre de Usuario:</label>
  <input type="text" id="username" name="username"><br>
  <label for="password">Contraseña:</label>
  <input type="password" id="password" name="password"><br>
  <input type="submit" value="Registrarse">
</form>
```

El código se renderiza de esta manera al abrirlo en un navegador.



Nombre de Usuario:

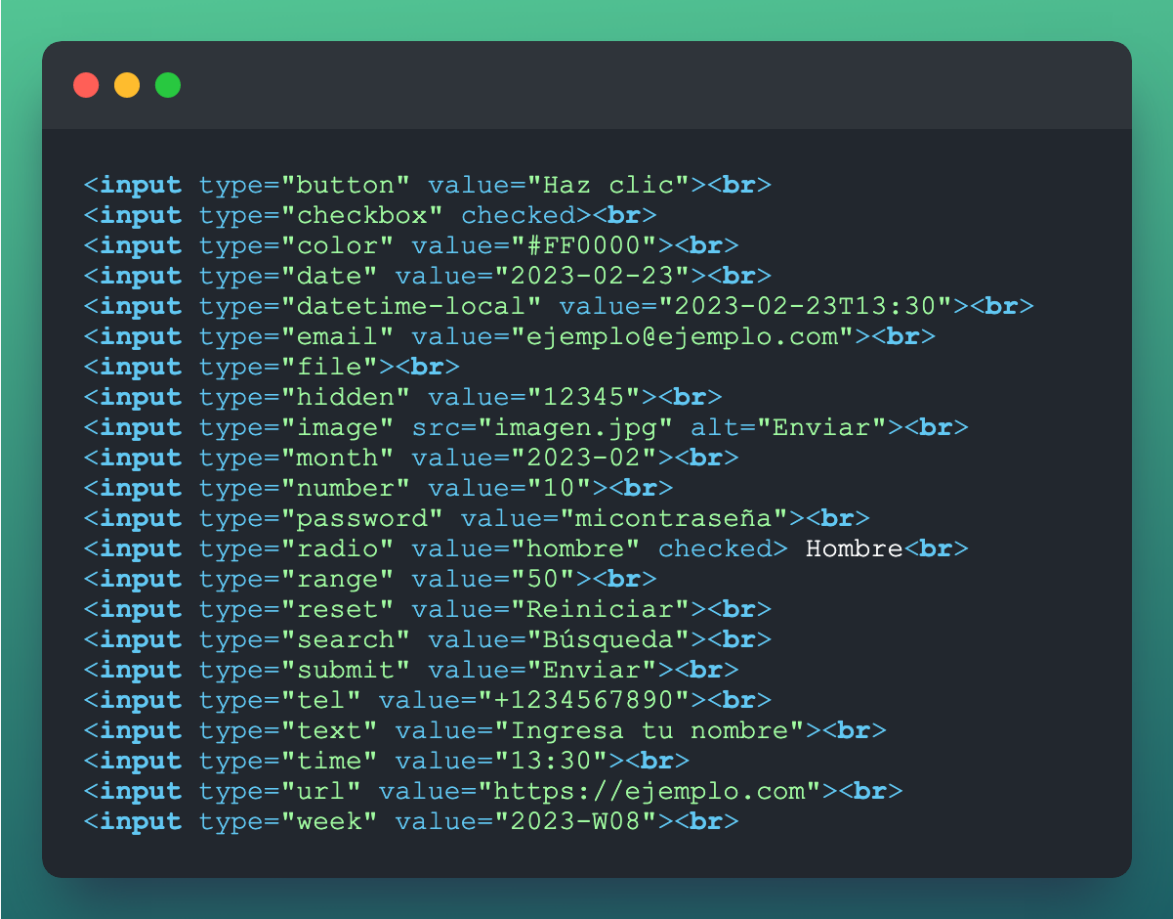
Contraseña:

En este ejemplo se utiliza el input para crear dos campos de texto, uno para ingresar el nombre de usuario y otro para ingresar la contraseña, y un botón de tipo “submit”

4 ELEMENTOS

para enviar el formulario. El type “password” ayuda a ocultar la información ingresada por el usuario mientras lo escribe.

Estos son los tipos de input disponibles en HTML modificando el atributo “type” :



```
<input type="button" value="Haz clic"><br>
<input type="checkbox" checked><br>
<input type="color" value="#FF0000"><br>
<input type="date" value="2023-02-23"><br>
<input type="datetime-local" value="2023-02-23T13:30"><br>
<input type="email" value="ejemplo@ejemplo.com"><br>
<input type="file"><br>
<input type="hidden" value="12345"><br>
<input type="image" src="imagen.jpg" alt="Enviar"><br>
<input type="month" value="2023-02"><br>
<input type="number" value="10"><br>
<input type="password" value="micontraseña"><br>
<input type="radio" value="hombre" checked> Hombre<br>
<input type="range" value="50"><br>
<input type="reset" value="Reiniciar"><br>
<input type="search" value="Búsqueda"><br>
<input type="submit" value="Enviar"><br>
<input type="tel" value="+1234567890"><br>
<input type="text" value="Ingresa tu nombre"><br>
<input type="time" value="13:30"><br>
<input type="url" value="https://ejemplo.com"><br>
<input type="week" value="2023-W08"><br>
```

El código se renderiza de esta manera al abrirlo en un navegador.

4 ELEMENTOS

Haz clic

☒

02/23/2023

02/23/2023, 01:30 PM

ejemplo@ejemplo.com

Choose File

No file chosen



February 2023

10

.....

☒ Hombre

Reiniciar

Búsqueda

Enviar

+1234567890

Ingresa tu nombre

01:30 PM

https://ejemplo.com

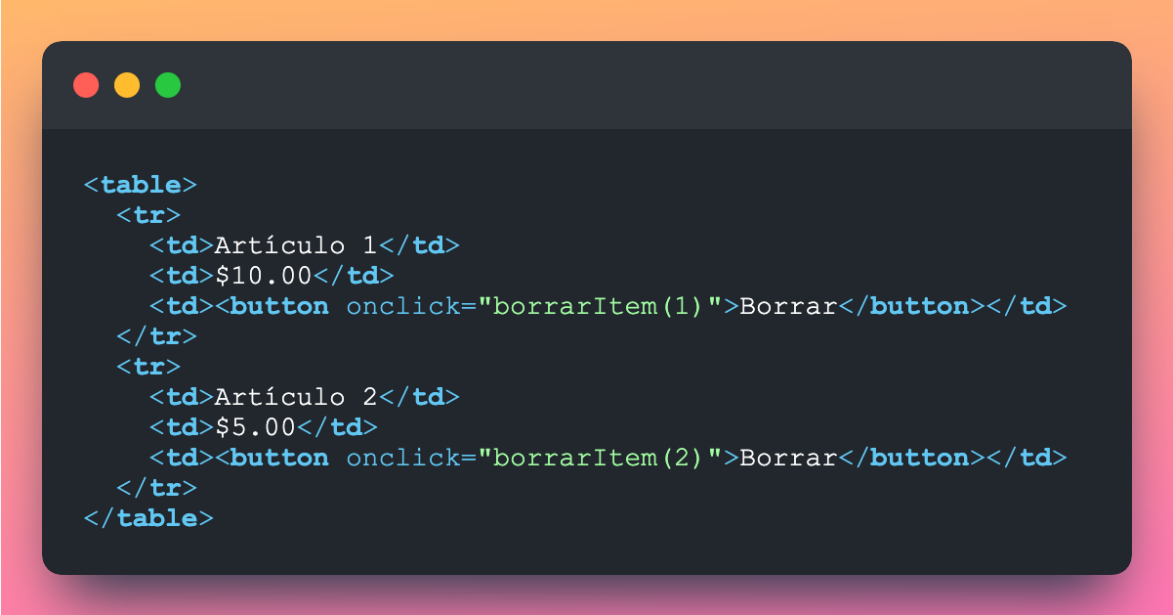
Week 08, 2023

4.8.4 <button>

La etiqueta **<button>** en HTML se utiliza para crear un botón de interfaz de usuario en una página web. Los botones pueden ser utilizados para realizar una variedad de acciones, como enviar un formulario, ejecutar una función JavaScript, o navegar a otra página web.

El contenido del botón se puede especificar utilizando el contenido entre las etiquetas de apertura y cierre **<button>** . Por ejemplo, un botón puede tener el texto “Enviar” entre las etiquetas, lo que significa que el texto “Enviar” se mostrará en el botón.

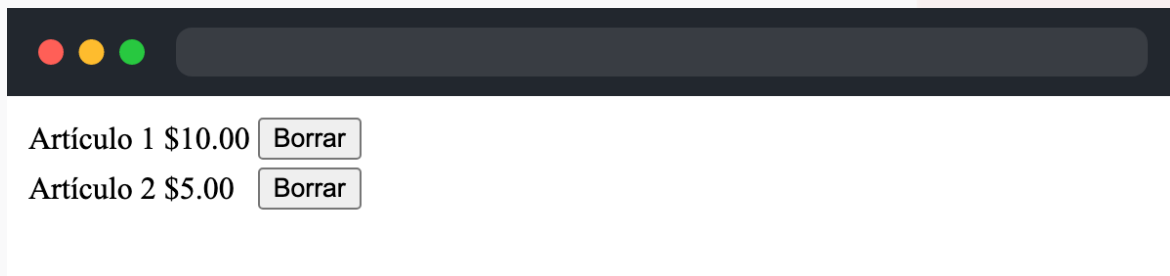
Un ejemplo de uso práctico podría ser un botón de “borrar” en una tabla:



```
<table>
  <tr>
    <td>Artículo 1</td>
    <td>$10.00</td>
    <td><button onclick="borrarItem(1) ">Borrar</button></td>
  </tr>
  <tr>
    <td>Artículo 2</td>
    <td>$5.00</td>
    <td><button onclick="borrarItem(2) ">Borrar</button></td>
  </tr>
</table>
```

El código se renderiza de esta manera al abrirlo en un navegador.

4 ELEMENTOS



En este ejemplo cada botón tiene la acción “onclick” que ejecuta una función javascript llamada “borrarItem” y pasa como parametro el identificador del item que se desea borrar.

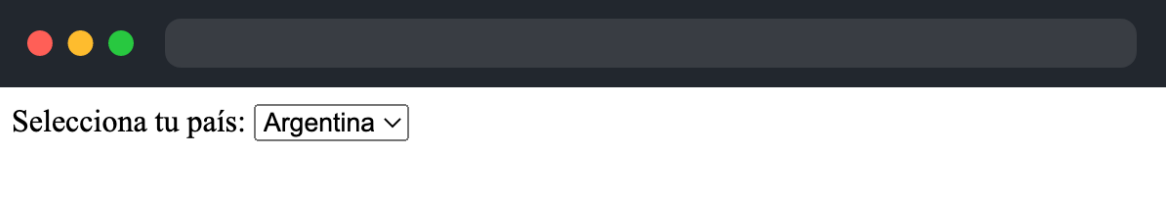
4.8.5 <select>

La etiqueta **<select>** en HTML se utiliza para crear un menú desplegable o “lista desplegable” en un formulario web. Un menú desplegable permite a los usuarios elegir una opción de varias opciones proporcionadas. La etiqueta **<select>** es el contenedor para las opciones disponibles, y cada opción se especifica utilizando la etiqueta **<option>**. El valor seleccionado por defecto se especifica utilizando el atributo “selected” en la etiqueta **<option>**.

Un ejemplo de uso práctico podría ser un menú desplegable para elegir un país:

```
<label for="pais">Selecciona tu país:</label>
<select id="pais" name="pais">
  <option value="mx">México</option>
  <option value="col">Colombia</option>
  <option value="arg" selected>Argentina</option>
  <option value="per">Perú</option>
  <option value="chi">Chile</option>
</select>
```

El código se renderiza de esta manera al abrirlo en un navegador.



Selecciona tu país:

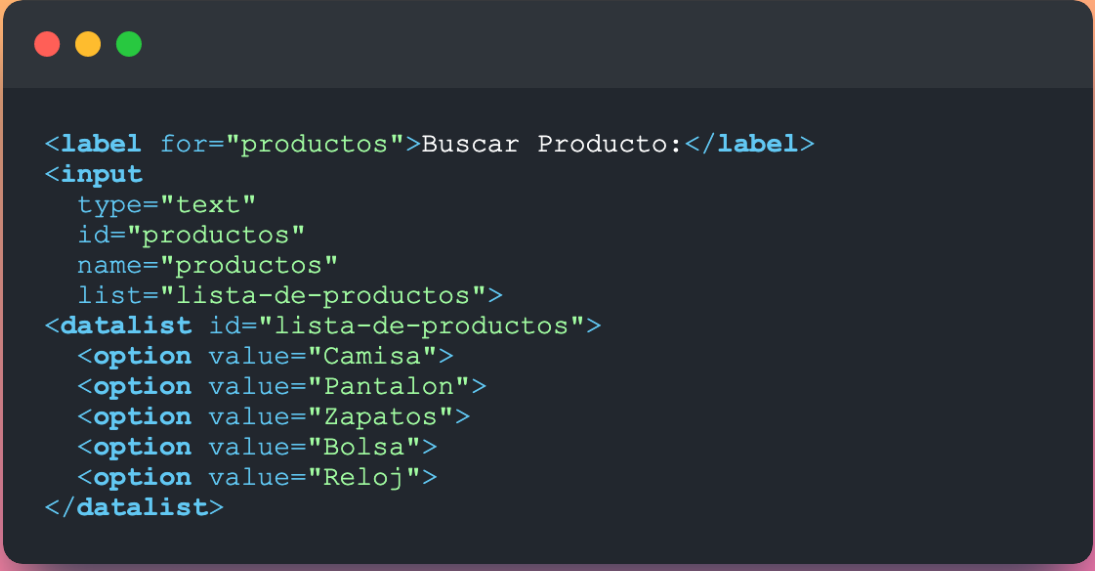
En este ejemplo se tiene un select con varias opciones de países, y Argentina es la opción seleccionada por defecto. Al elegir un país, su valor se enviará en el formulario.

4.8.6 <datalist>

La etiqueta **<datalist>** en HTML es utilizada para crear una lista de opciones para un campo de entrada en un formulario. La lista se muestra como una lista desplegable cuando el usuario comienza a escribir en el campo de entrada asociado. Las opciones son especificadas utilizando la etiqueta **<option>** dentro de la etiqueta **<datalist>**. El campo de entrada se asocia a la lista utilizando el atributo "list", cuyo valor es el valor del atributo "id" de la etiqueta **<datalist>**.

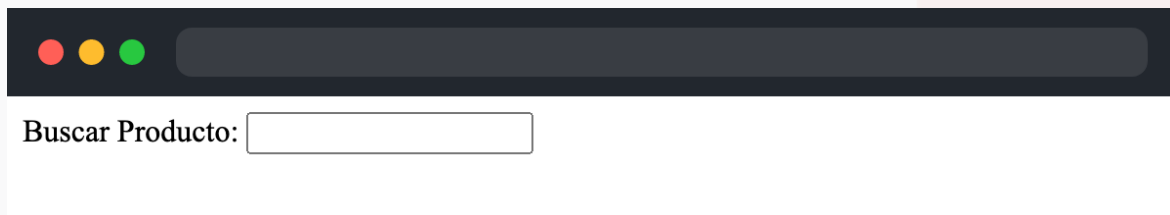
4 ELEMENTOS

Un ejemplo de uso práctico podría ser un campo de entrada para buscar un producto en una tienda en línea:



```
<label for="productos">Buscar Producto:</label>
<input
  type="text"
  id="productos"
  name="productos"
  list="lista-de-productos">
<datalist id="lista-de-productos">
  <option value="Camisa">
  <option value="Pantalon">
  <option value="Zapatos">
  <option value="Bolsa">
  <option value="Reloj">
</datalist>
```

El código se renderiza de esta manera al abrirlo en un navegador.



En este ejemplo se tiene un input para buscar producto, cuando el usuario comienza a escribir, se le muestra una lista desplegable con opciones como camisa, pantalon, zapato, entre otros. Esto ayuda a los usuarios a elegir productos de manera mas fácil y rápida.

4.8.7 <optgroup>

La etiqueta **<optgroup>** en HTML se utiliza para agrupar opciones relacionadas en una lista desplegable (**<select>**). Una etiqueta **<optgroup>** es un contenedor para varias etiquetas **<option>** y tiene un atributo "label" que se utiliza para especificar el título del grupo de opciones.

Un ejemplo de uso práctico podría ser una lista desplegable de países donde se agrupan las opciones por continentes:

```
<label for="pais">Selecciona tu país:</label>
<select id="pais" name="pais">
  <optgroup label="América">
    <option value="mx">México</option>
    <option value="col">Colombia</option>
    <option value="arg">Argentina</option>
    <option value="per">Perú</option>
    <option value="chi">Chile</option>
  </optgroup>
  <optgroup label="Europa">
    <option value="esp">España</option>
    <option value="ita">Italia</option>
    <option value="ger">Alemania</option>
  </optgroup>
</select>
```

El código se renderiza de esta manera al abrirlo en un navegador.

Selecciona tu país:

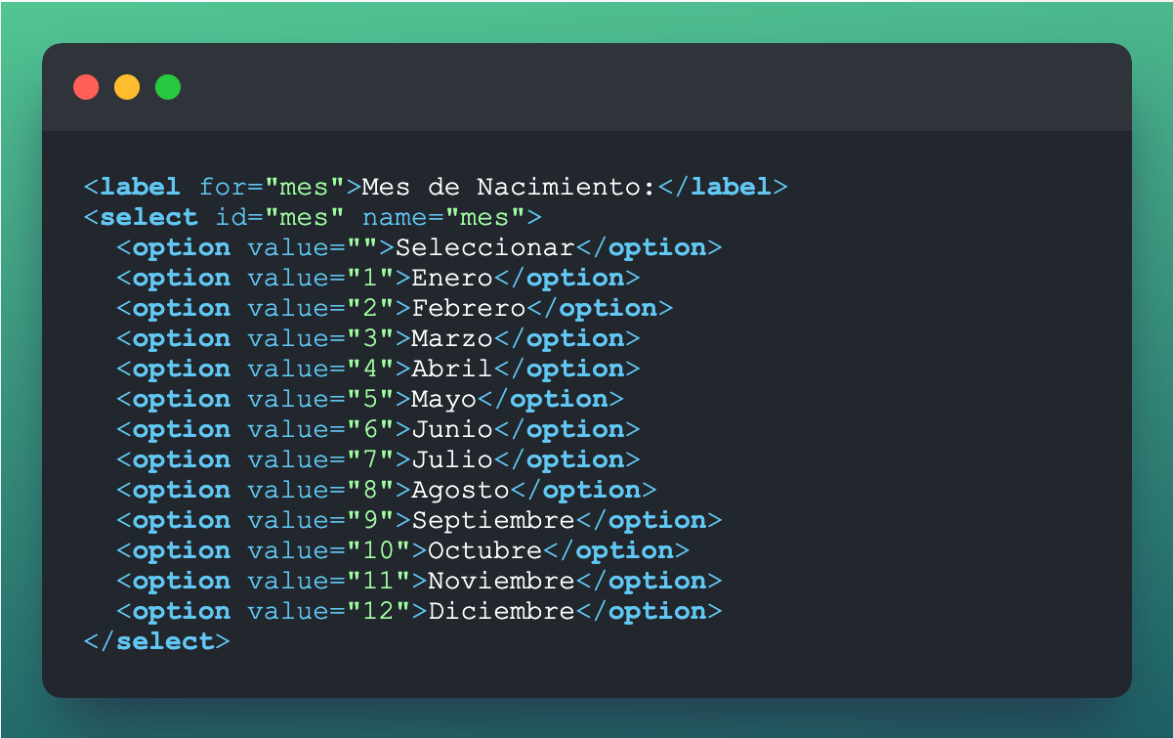
4 ELEMENTOS

En este ejemplo se tiene un select con varias opciones de países, agrupados por continentes, América y Europa, lo que permite una mejor organización y facilidad para buscar un país específico.

4.8.8 <option>

La etiqueta **<option>** en HTML se utiliza para definir una opción en un menú desplegable (**<select>**) o en una lista de opciones (**<datalist>**). Cada etiqueta **<option>** representa una posible opción para seleccionar. El valor seleccionado se especifica utilizando el atributo "selected". El valor de la opción se especifica utilizando el atributo "value" y el texto mostrado en la lista desplegable se especifica entre las etiquetas de apertura y cierre **<option>**.

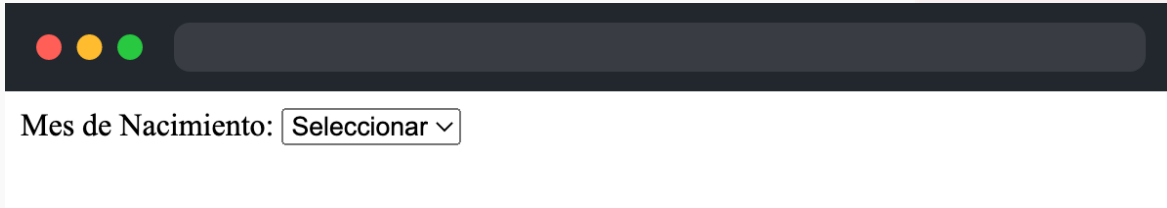
Un ejemplo de uso práctico podría ser un menú desplegable para elegir el mes de nacimiento:

A screenshot of a code editor with a dark background and light green text. The code defines a label and a select element with twelve options representing the months of the year. The code is as follows:

```
<label for="mes">Mes de Nacimiento:</label>
<select id="mes" name="mes">
  <option value="">Seleccionar</option>
  <option value="1">Enero</option>
  <option value="2">Febrero</option>
  <option value="3">Marzo</option>
  <option value="4">Abril</option>
  <option value="5">Mayo</option>
  <option value="6">Junio</option>
  <option value="7">Julio</option>
  <option value="8">Agosto</option>
  <option value="9">Septiembre</option>
  <option value="10">Octubre</option>
  <option value="11">Noviembre</option>
  <option value="12">Diciembre</option>
</select>
```

4 ELEMENTOS

El código se renderiza de esta manera al abrirlo en un navegador.



En este ejemplo se tiene un select con una lista de meses del año donde el usuario puede elegir el mes de su nacimiento, el valor de cada opción es el número del mes y el texto mostrado en el select es el nombre del mes.

4.8.9 <textarea>

La etiqueta **<textarea>** en HTML se utiliza para crear un área de texto en un formulario web. Es similar a un campo de entrada de texto (**<input type="text">**), pero permite a los usuarios ingresar varias líneas de texto en lugar de solo una. Esto lo hace útil para recopilar información como comentarios, descripciones, direcciones, entre otros.

La etiqueta **<textarea>** tiene atributos como “name”, “id”, “cols” y “rows” que se utilizan para especificar el nombre del campo, el número de columnas y filas que se mostrarán en el área de texto. También se puede establecer un valor predeterminado utilizando el contenido entre las etiquetas de apertura y cierre de textarea.

Un ejemplo de uso práctico podría ser un formulario de contacto para recibir comentarios:

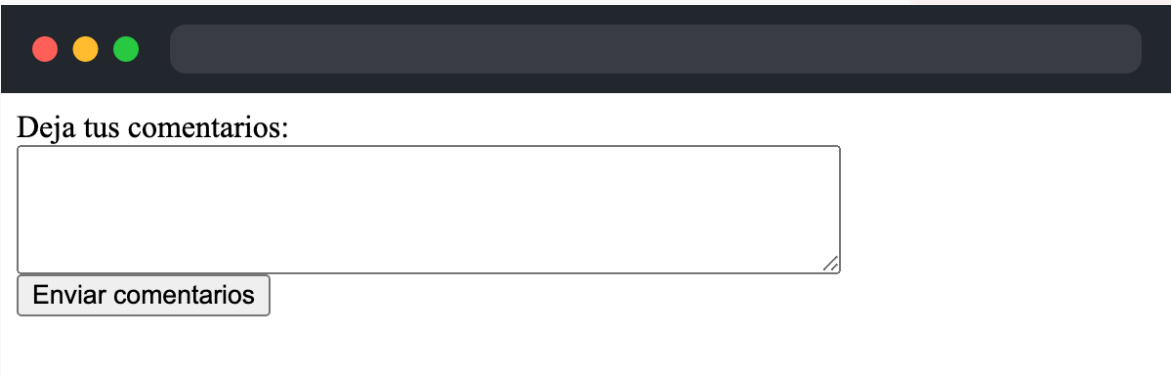
Mis disculpas, aquí esta el final del ejemplo:

4 ELEMENTOS



```
<form>
  <label for="comentarios">Deja tus comentarios:</label>
  <br>
  <textarea
    id="comentarios"
    name="comentarios"
    rows="4"
    cols="50"></textarea>
  <br>
  <input type="submit" value="Enviar comentarios">
</form>
```

El código se renderiza de esta manera al abrirlo en un navegador.



Deja tus comentarios:

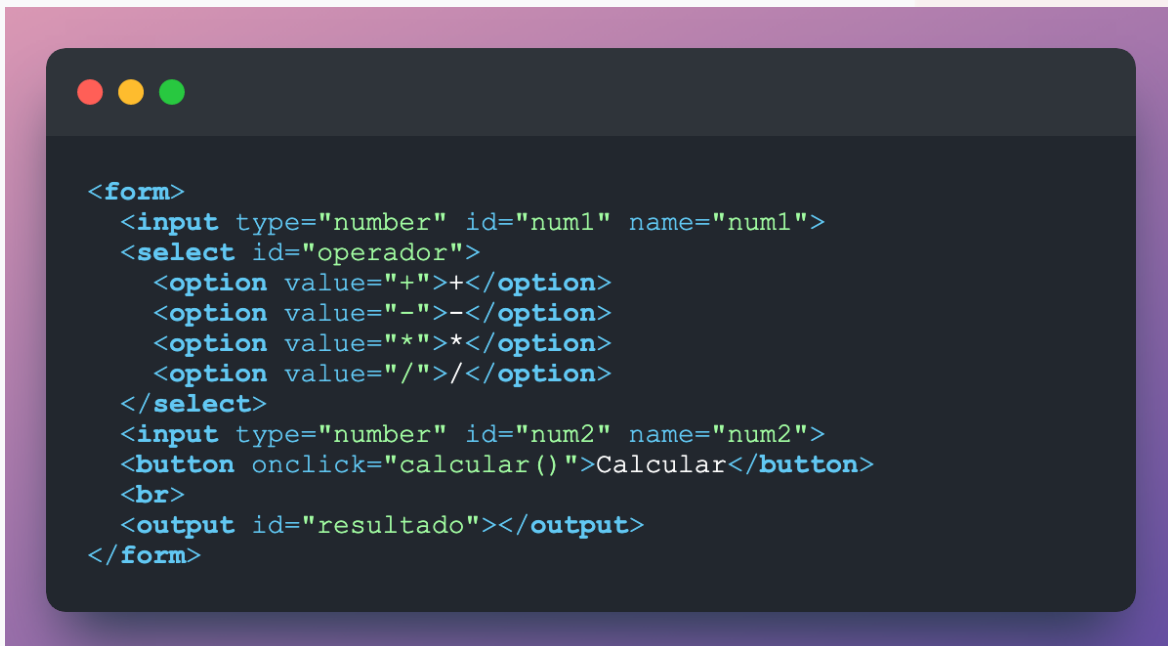
Enviar comentarios

En este ejemplo se tiene un textarea donde el usuario puede escribir sus comentarios, el textarea se caracteriza por tener un nombre y un id “comments” , además de tener una altura de 4 filas y un ancho de 50 columnas. El texto ingresado en el textarea será enviado junto con el formulario al enviarlo.

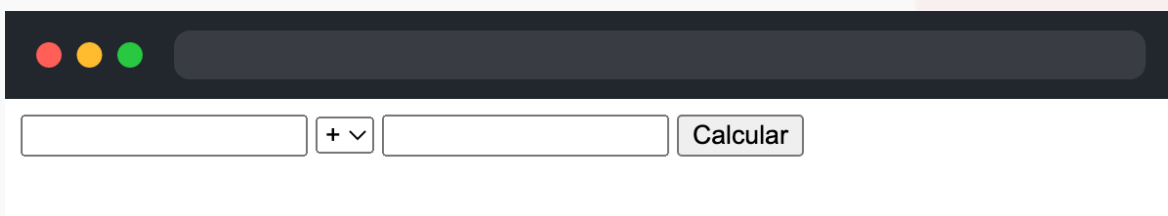
4.8.10 <output>

La etiqueta **<output>** en HTML se utiliza para mostrar el resultado de un cálculo o script en una página web. El valor de una etiqueta **<output>** se puede establecer utilizando JavaScript y puede ser enlazado a elementos de formulario como input o select para actualizar automáticamente el resultado en función de los datos ingresados.

Un ejemplo de uso práctico podría ser una calculadora en una página web:



El código se renderiza de esta manera al abrirlo en un navegador.



En este ejemplo se tiene una simple calculadora con dos campos para ingresar

4 ELEMENTOS

números y un select para seleccionar la operación a realizar, cuando se da click al botón Calcular se ejecuta una función javascript “calcular” que realiza el calculo y asigna el resultado al elemento output con el id “resultado” mostrando el resultado en la pagina.

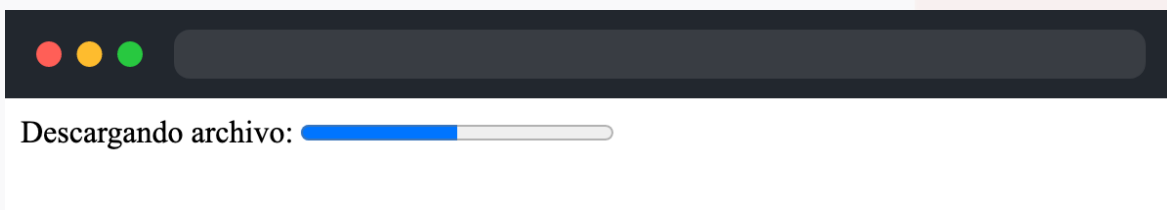
4.8.11 <progress>

La etiqueta **<progress>** en HTML se utiliza para representar el progreso de una tarea en curso. Es similar a una barra de progreso que se utiliza para mostrar el progreso de una tarea o proceso. La etiqueta **<progress>** tiene atributos como “value” y “max” para especificar el progreso actual y el progreso máximo.

Un ejemplo de uso práctico podría ser mostrar el progreso de una descarga:



El código se renderiza de esta manera al abrirlo en un navegador.

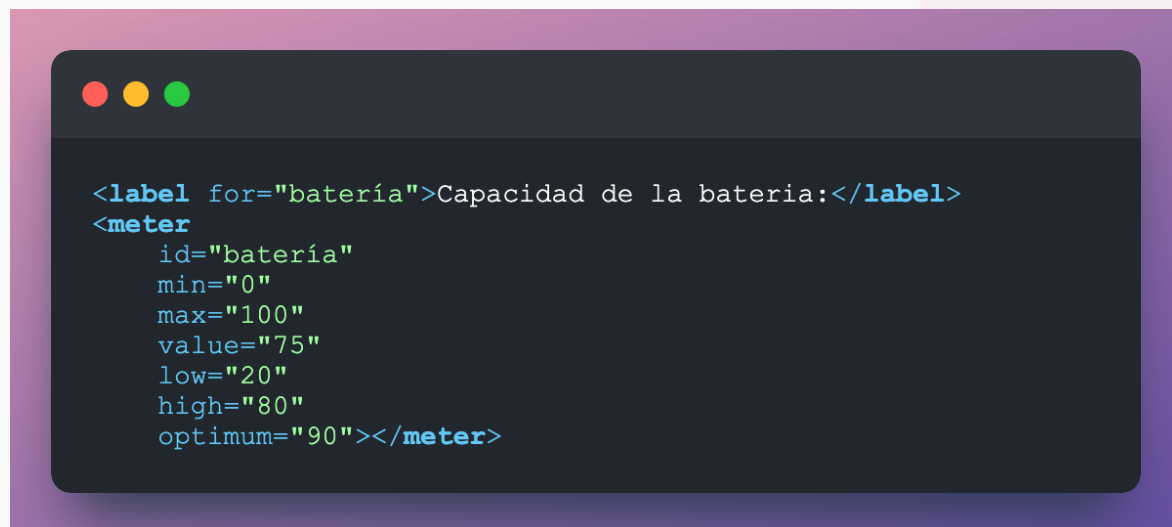


En este ejemplo se tiene una barra de progreso para mostrar el progreso de una descarga, el valor actual es 50 y el valor máximo es 100, indicando que el progreso de descarga es del 50%. En este caso se podría utilizar JavaScript para actualizar el valor de value en tiempo real.

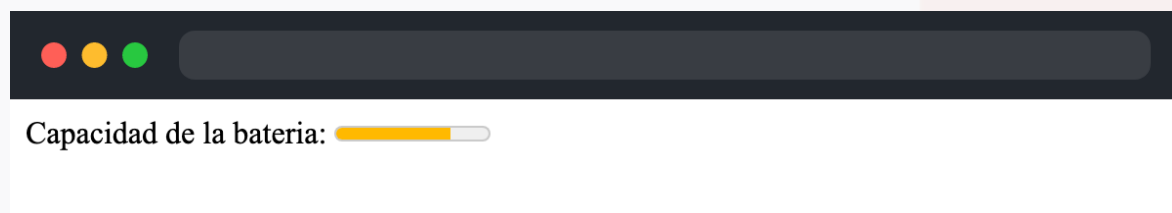
4.8.12 <meter>

La etiqueta **<meter>** en HTML se utiliza para representar un valor numérico dentro de un rango específico en una página web. Es similar a una barra de progreso, pero en lugar de mostrar un progreso hacia un objetivo, se utiliza para mostrar un valor numérico dentro de un rango. La etiqueta **<meter>** tiene atributos como “min” , “max” , “value” y “low” , “high” , “optimum” para especificar el valor mínimo, máximo y actual y los valores críticos de la medida.

Un ejemplo de uso práctico podría ser un indicador de capacidad de un dispositivo:



El código se renderiza de esta manera al abrirlo en un navegador.



En este ejemplo se tiene un indicador de capacidad de una batería, se utiliza el valor actual 75 que se encuentra en un rango de valores 0-100, el valor low indica que el dispositivo necesita carga cuando el valor baja a 20, el valor high indica que

esta cargando cuando llega a 80 y el valor optimum indica que esta completamente cargado cuando llega a 90, esto ayuda a los usuarios a conocer el estado de su batería.

4.8.13 <fieldset>

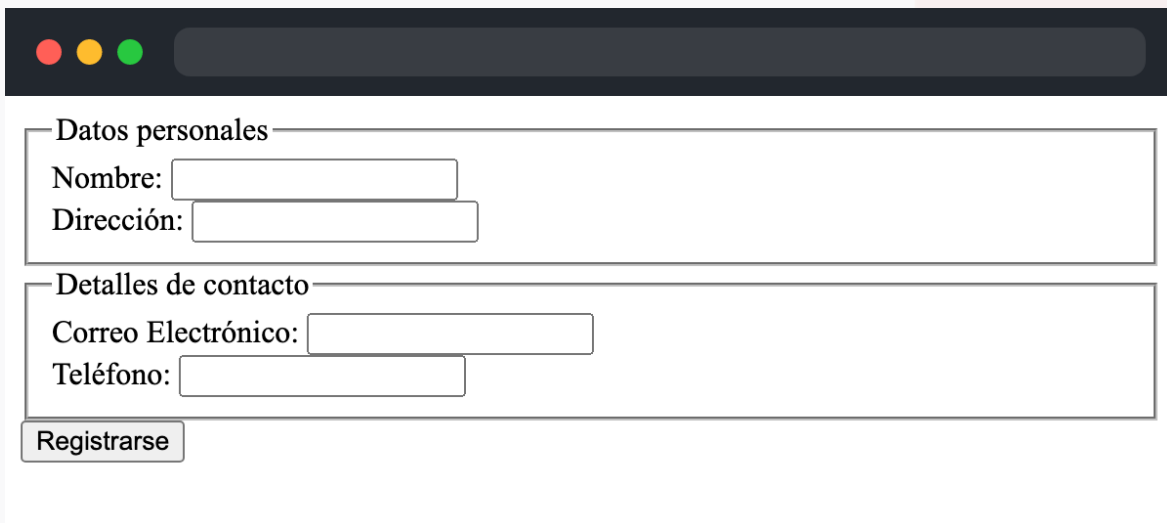
La etiqueta **<fieldset>** en HTML se utiliza para agrupar elementos relacionados en un formulario web. Un **<fieldset>** es un contenedor para varios elementos de formulario, como inputs, labels, selects, etc. También puede tener un **<legend>** para especificar un título para el grupo de elementos. El **<legend>** se muestra como un encabezado para el **<fieldset>**.

Un ejemplo de uso práctico podría ser un formulario de registro de usuario, en donde se agrupen elementos relacionados como nombre, dirección y detalles de contacto:

```
<form>
  <fieldset>
    <legend>Datos personales</legend>
    <label for="nombre">Nombre:</label>
    <input type="text" id="nombre" name="nombre">
    <br>
    <label for="dirección">Dirección:</label>
    <input type="text" id="dirección" name="dirección">
  </fieldset>
  <fieldset>
    <legend>Detalles de contacto</legend>
    <label for="email">Correo Electrónico:</label>
    <input type="email" id="email" name="email">
    <br>
    <label for="teléfono">Teléfono:</label>
    <input type="tel" id="teléfono" name="teléfono">
  </fieldset>
  <input type="submit" value="Registrarse">
</form>
```

El código se renderiza de esta manera al abrirlo en un navegador.

4 ELEMENTOS



The image shows a web browser window with a dark title bar. Inside, there is a form with two distinct sections. The first section is titled 'Datos personales' and contains two input fields: 'Nombre:' and 'Dirección:'. The second section is titled 'Detalles de contacto' and contains two input fields: 'Correo Electrónico:' and 'Teléfono:'. Below these sections is a button labeled 'Registrarse'.

En este ejemplo se tiene dos fieldset, el primero agrupa los elementos relacionados a los datos personales y el segundo agrupa los detalles de contacto, cada fieldset tiene un legend con un título describiendo su contenido. Esto ayuda a organizar y estructurar el formulario de manera clara para los usuarios.

4.8.14 <legend>

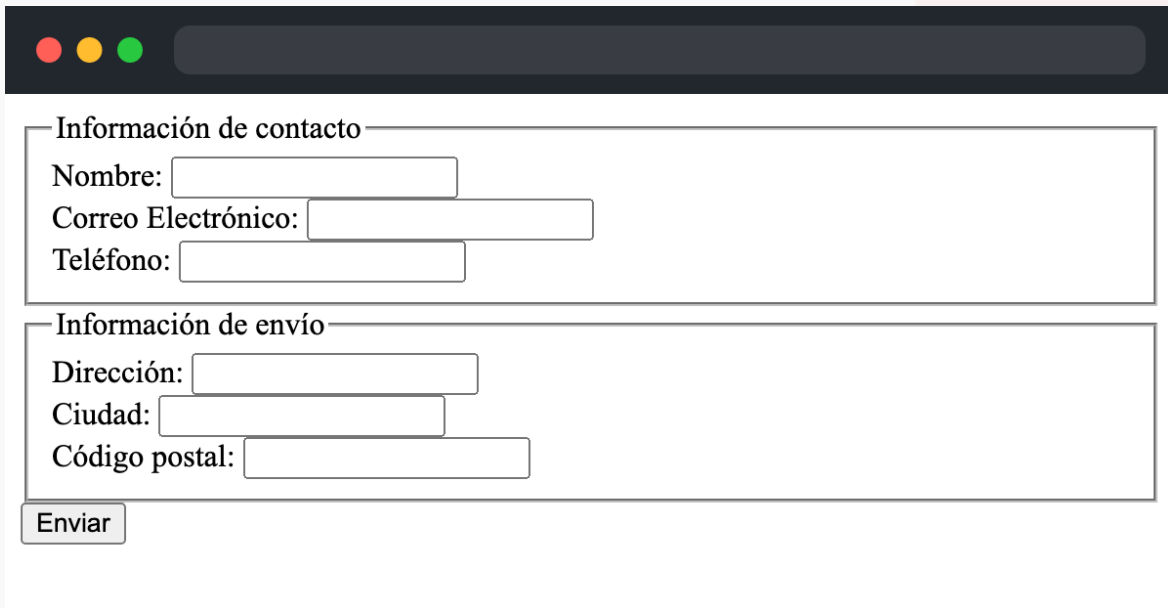
La etiqueta **<legend>** en HTML se utiliza para proporcionar un título o una descripción para un **fieldset**. El texto contenido dentro de la etiqueta **<legend>** se muestra como un encabezado para el **fieldset** y proporciona información adicional sobre el grupo de elementos dentro del **fieldset**.

Un ejemplo de uso práctico es en un formulario, donde se quiere describir las secciones o grupos de inputs para facilitar al usuario la lectura y comprensión de los datos que debe de ingresar:

```
<form>
  <fieldset>
    <legend>Información de contacto</legend>
    <label for="nombre">Nombre:</label>
    <input type="text" id="nombre" name="nombre">
    <br>
    <label for="email">Correo Electrónico:</label>
    <input type="email" id="email" name="email">
    <br>
    <label for="teléfono">Teléfono:</label>
    <input type="tel" id="teléfono" name="teléfono">
  </fieldset>
  <fieldset>
    <legend>Información de envío</legend>
    <label for="dirección">Dirección:</label>
    <input type="text" id="dirección" name="dirección">
    <br>
    <label for="ciudad">Ciudad:</label>
    <input type="text" id="ciudad" name="ciudad">
    <br>
    <label for="zip">Código postal:</label>
    <input type="text" id="zip" name="zip">
  </fieldset>
  <input type="submit" value="Enviar">
</form>
```

El código se renderiza de esta manera al abrirlo en un navegador.

4 ELEMENTOS



The image shows a web browser window with a dark title bar. Inside the browser, there is a form with two distinct sections, each enclosed in a box with a legend at the top. The first section, titled 'Información de contacto', contains three input fields: 'Nombre:', 'Correo Electrónico:', and 'Teléfono:'. The second section, titled 'Información de envío', contains three input fields: 'Dirección:', 'Ciudad:', and 'Código postal:'. Below these sections is a button labeled 'Enviar'.

En este ejemplo se tiene un formulario de compra, donde se han agrupado los elementos relacionados a la información de contacto en un primer fieldset y los elementos relacionados a la información de envío en un segundo fieldset, cada fieldset tiene un legend describiendo el contenido del grupo de elementos. Esto ayuda a organizar y estructurar el formulario de manera clara para los usuarios, facilitando la lectura y comprensión del mismo.

4.9 Interactividad

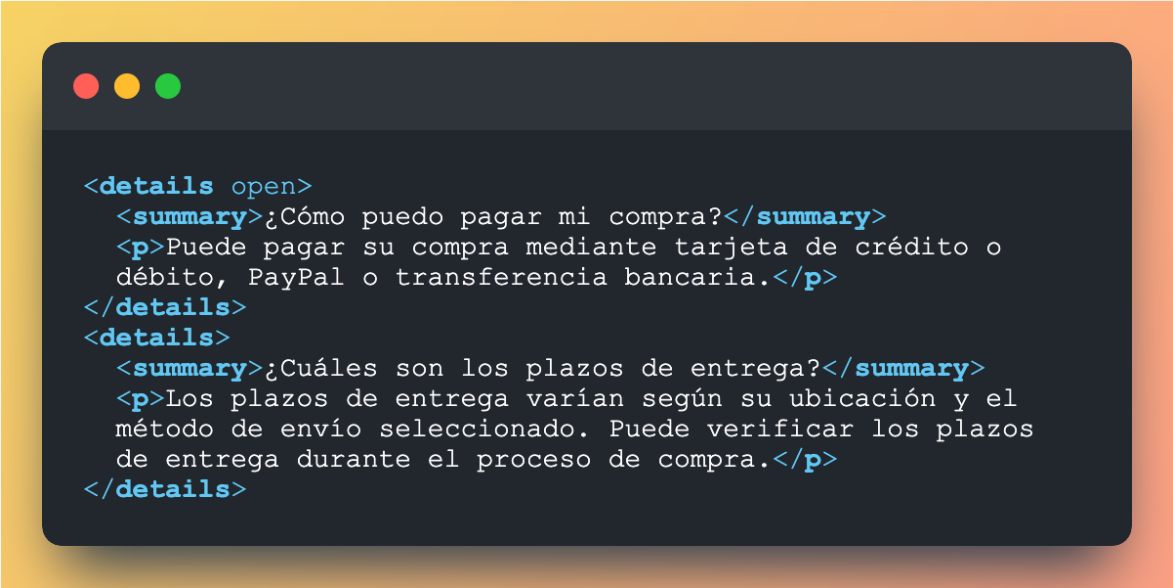
Los elementos interactivos en HTML (Interactive elements) se utilizan para crear elementos reactivos en una página web, los cuales permiten al usuario interactuar con la página de manera dinámica.

Estas son las etiquetas utilizadas para interactividad a detalle:

4.9.1 <details>

La etiqueta **<details>** en HTML se utiliza para proporcionar una forma de desplegar y ocultar contenido adicional en una página web. El contenido está oculto por defecto y se muestra al hacer clic en un botón de control de detalles. El botón de control de detalles es el primer elemento dentro de la etiqueta **<details>**. Puede contener cualquier elemento HTML y se utiliza a menudo junto con la etiqueta **<summary>** para proporcionar un título o una descripción del contenido desplegable.

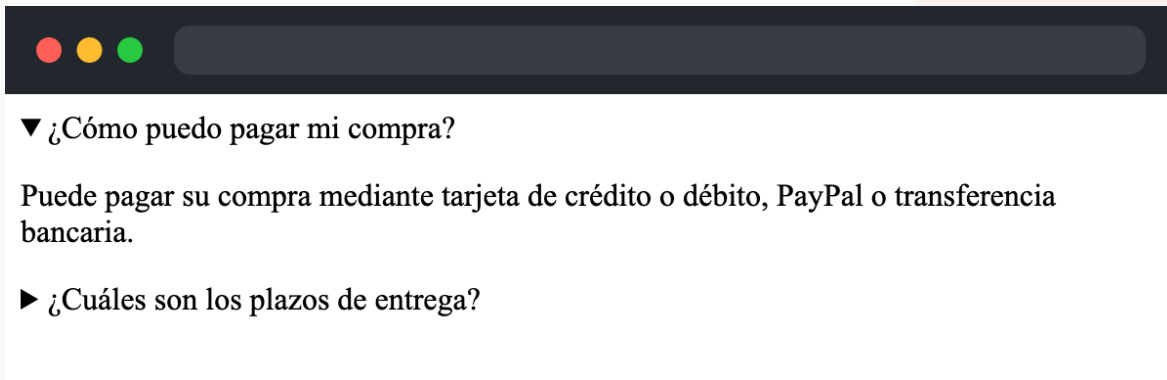
Un ejemplo de uso práctico podría ser un acordeón de preguntas frecuentes:



```
<details open>
  <summary>¿Cómo puedo pagar mi compra?</summary>
  <p>Puede pagar su compra mediante tarjeta de crédito o
  débito, PayPal o transferencia bancaria.</p>
</details>
<details>
  <summary>¿Cuáles son los plazos de entrega?</summary>
  <p>Los plazos de entrega varían según su ubicación y el
  método de envío seleccionado. Puede verificar los plazos
  de entrega durante el proceso de compra.</p>
</details>
```

El código se renderiza de esta manera al abrirlo en un navegador.

4 ELEMENTOS



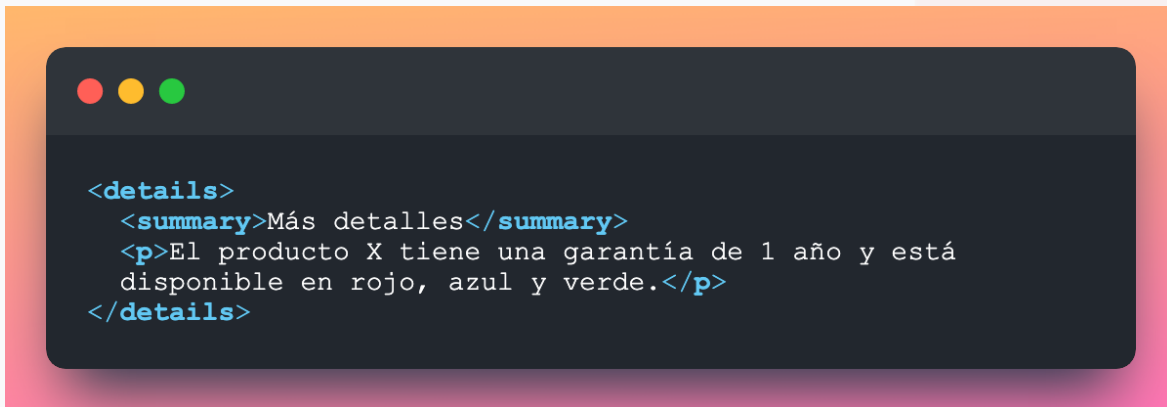
En este ejemplo se tiene una lista de preguntas y respuestas, cada una tiene un summary con el texto de pregunta y un párrafo con la respuesta correspondiente, al hacer click en cada summary se desplegara el contenido del párrafo correspondiente, permitiendo al usuario ver solo las respuestas de las preguntas que le interesen sin tener que ver todo el contenido de la página, esto ayuda a organizar y estructurar la información de manera clara para el usuario.

4.9.2 <summary>

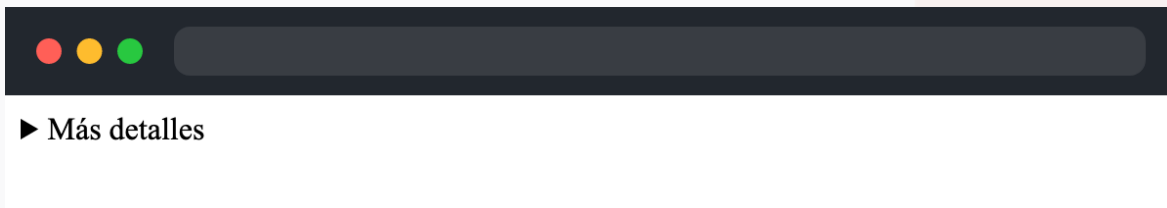
La etiqueta **<summary>** en HTML se utiliza para proporcionar un título o una descripción para un elemento **<details>**. El texto contenido dentro de la etiqueta **<summary>** se muestra como un encabezado o título para el contenido desplegable dentro del elemento **<details>**, es el primer elemento dentro de la etiqueta **<details>**. Es esencialmente un botón que permite al usuario mostrar u ocultar el contenido desplegable.

Un ejemplo de uso práctico podría ser una sección de Información adicional en un sitio web de un producto, donde se despliega detalles adicionales sobre el producto al hacer clic en el botón “Más detalles” :

4 ELEMENTOS



El código se renderiza de esta manera al abrirlo en un navegador.



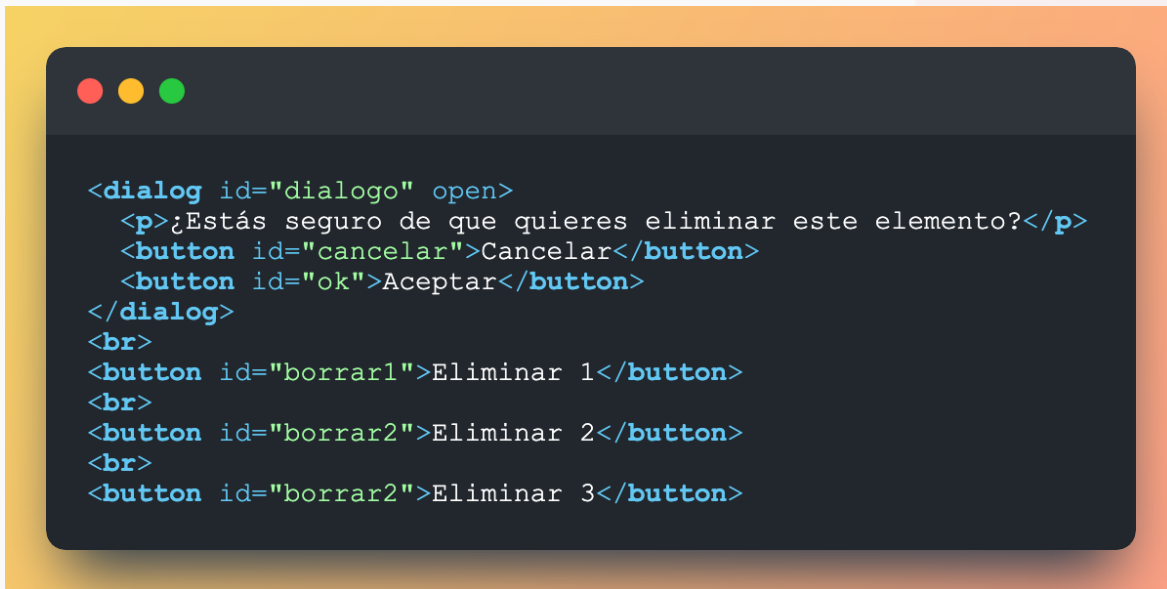
Al hacer click en “Más detalles” , el usuario puede ver información adicional sobre el producto.

Es importante notar que el contenido dentro del elemento **<summary>** no se mostrará al hacer click, es solo para darle un título al contenido dentro del **<details>**

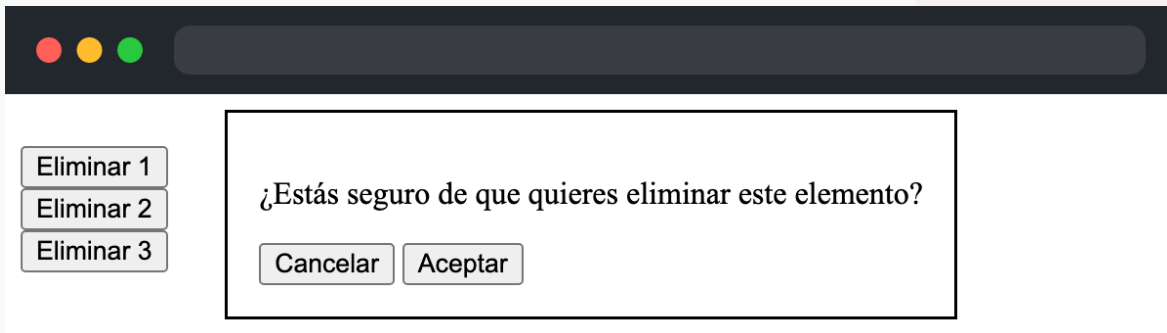
4.9.3 <dialog>

La etiqueta **<dialog>** es una etiqueta experimental en HTML que se utiliza para crear una ventana emergente o un cuadro de diálogo modal. El contenido dentro de la etiqueta **<dialog>** se muestra en una ventana emergente y el usuario debe interactuar con él antes de poder volver a interactuar con la página principal.

Un ejemplo de uso práctico podría ser un cuadro de diálogo de confirmación al eliminar un elemento:



El código se renderiza de esta manera al abrirlo en un navegador.



En este ejemplo al hacer clic en el botón “Eliminar 1” , se mostrara el cuadro de dialogo y el usuario deberá interactuar con el (aceptar o cancelar) antes de volver a interactuar con la página principal.

4.10 Programación

Las etiquetas de código de programación en HTML (Scripting) se utilizan para agregar scripts a una página web, los cuales permiten al sitio web tener funcionalidades

4 ELEMENTOS

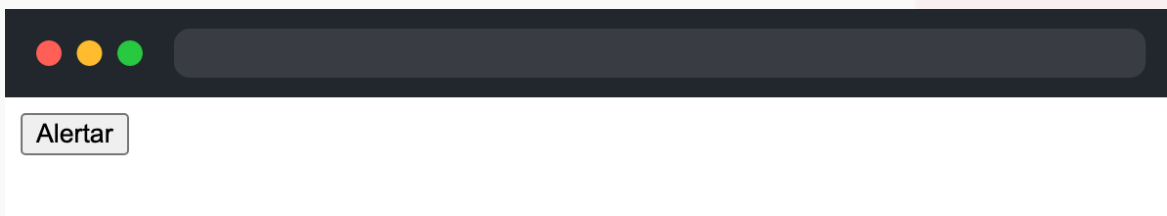
más complejas. Entre las etiquetas más comunes utilizadas para agregar scripts se encuentran: **<script>**, **<noscript>**, **<canvas>** , entre otras.

Ejemplo:

A code editor window with a dark background and a red-to-orange gradient border. It contains the following code:

```
<script>
  function alertar() {
    alert("Hola, Mundo!");
  }
</script>
<button onclick="alertar()">Alertar</button>
```

El código se renderiza de esta manera al abrirlo en un navegador.



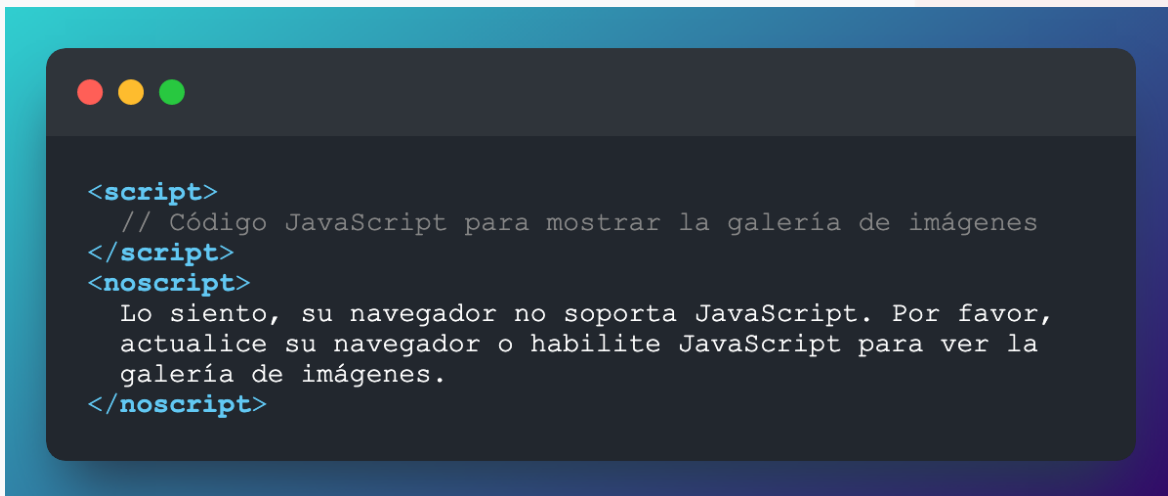
Este ejemplo muestra como se agrega un script en JavaScript para mostrar una alerta al presionar el boton “Alertar”. El script esta envuelto en las etiquetas **<script>**, esto indica que se está agregando contenido JavaScript al documento. El código dentro de las etiquetas **<script>** es la función “alertar” que se invoca al presionar el botón “Alertar” gracias al atributo “onclick” .

Estas son las etiquetas utilizadas para programación a detalle:

4.10.1 <noscript>

La etiqueta **<noscript>** en HTML se utiliza para proporcionar contenido alternativo para los usuarios cuya navegador no soporta o tiene deshabilitado JavaScript. El contenido dentro de la etiqueta **<noscript>** solo se mostrará si el navegador no soporta JavaScript.

Un ejemplo de uso práctico podría ser una página web que utiliza JavaScript para mostrar una galería de imágenes:



En este ejemplo, si el navegador no soporta JavaScript o lo tiene deshabilitado, se mostrará un mensaje de error indicando que es necesario habilitar JavaScript para ver la galería de imágenes.

Es importante mencionar que cada vez es menos común la necesidad de usar la etiqueta **<noscript>** ya que la mayoría de los navegadores modernos soportan JavaScript y también a medida que las aplicaciones web se vuelven más complejas, es importante tener en cuenta de que los usuarios podrían deshabilitar javascript en su navegador, es recomendable utilizar técnicas de Accesibilidad y degradación graciosa en lugar de depender exclusivamente de javascript.

4.10.2 <template>

Para entender el uso de esta etiqueta necesitamos hablar de **Web Components**. Los Web Components son una tecnología de desarrollo web que permite a los desarrolladores crear componentes personalizados y reutilizables para la web. Es decir, etiquetas personalizadas con atributos personalizados como **<mi-etiqueta mi-atributo="valor" ></mi-etiqueta>**. Estos componentes pueden ser utilizados en diferentes proyectos y aplicaciones web sin la necesidad de volver a escribir el código pero requieren conocimientos de CSS y JavaScript.

Los Web Components están compuestos por tres tecnologías principales:

- Custom Elements (Componentes personalizados): Permite a los desarrolladores crear sus propias etiquetas HTML personalizadas, con su propio comportamiento y apariencia.
- Shadow DOM (Sombra del DOM): Proporciona un ámbito de encapsulación para los estilos, el HTML y el JavaScript de los componentes, lo que evita que afecten al resto de la página y reduce los conflictos con otros componentes. Es como un mini DOM dentro del DOM.
- Templates (Plantillas): Proporciona una forma de definir un contenido HTML que puede ser reutilizado en diferentes componentes y aplicaciones.

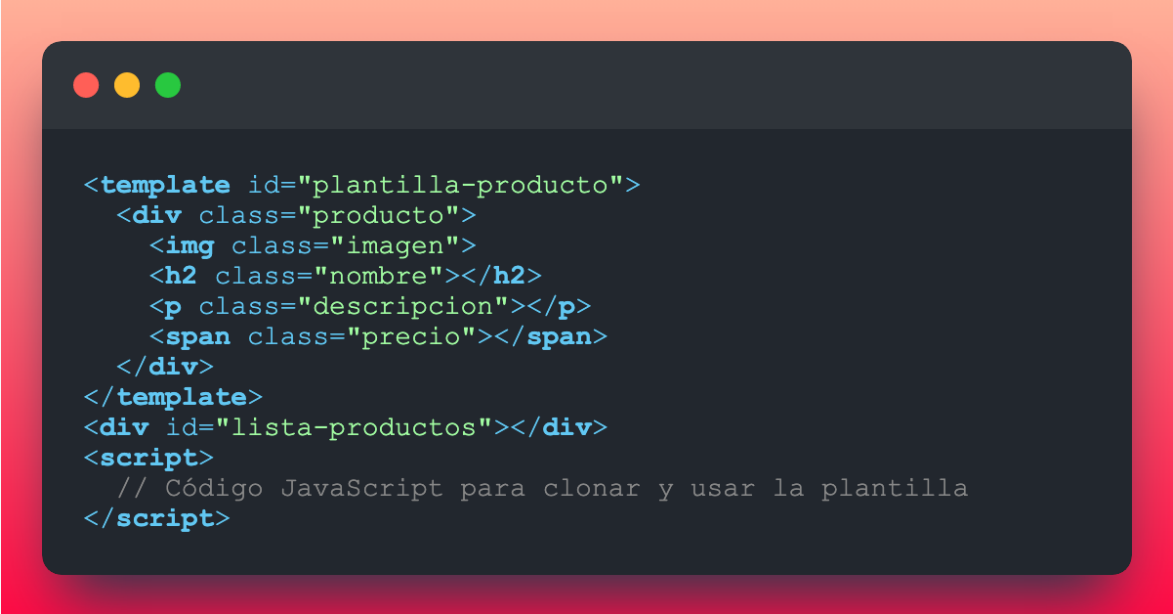
En conjunto, estas tecnologías permiten a los desarrolladores crear componentes personalizados que son fáciles de usar y mantener. Los Web Components también fomentan la modularidad y la reutilización de código, lo que puede reducir el tiempo y los costos de desarrollo de aplicaciones web complejas. Además, los Web Components son compatibles con todos los navegadores modernos, lo que los hace una tecnología muy accesible para los desarrolladores web.

La etiqueta **<template>** en HTML es una etiqueta de contenedor que se utiliza para definir un contenido no visible que se puede utilizar como plantilla para generar contenido dinámico en tiempo de ejecución. El contenido dentro de la etiqueta **<template>** no se renderiza en la página y solo se utiliza como una especie de

4 ELEMENTOS

“molde” que se puede clonar y modificar en tiempo de ejecución.

Un ejemplo de uso práctico podría ser una lista de elementos, donde cada elemento tiene una estructura similar, pero con diferentes valores:



```
<template id="plantilla-producto">
  <div class="producto">
    <img class="imagen">
    <h2 class="nombre"></h2>
    <p class="descripcion"></p>
    <span class="precio"></span>
  </div>
</template>
<div id="lista-productos"></div>
<script>
  // Código JavaScript para clonar y usar la plantilla
</script>
```

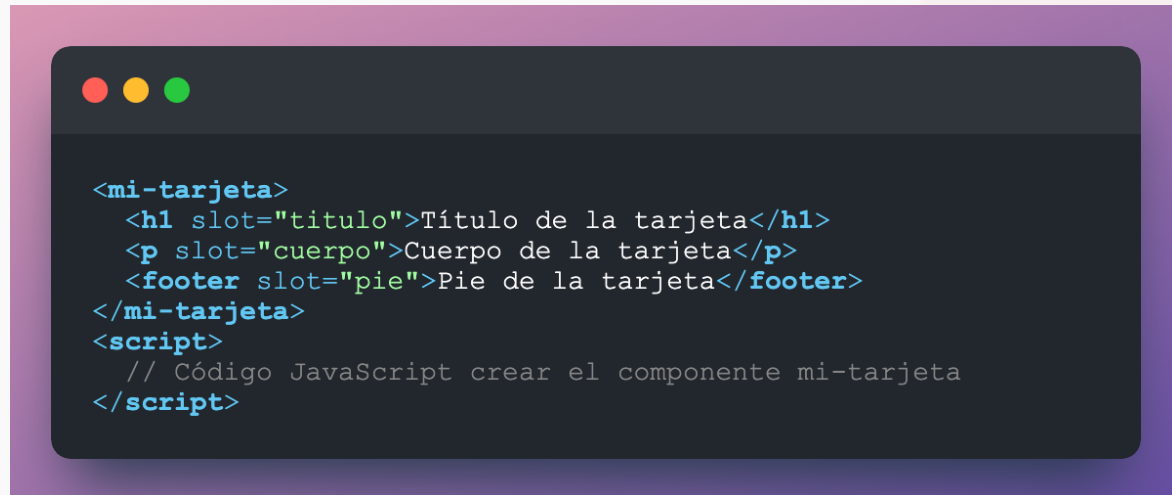
En este ejemplo, se tiene una plantilla de un elemento con una estructura fija pero vacía, que será utilizada para generar elementos de una lista, se llenará con los valores correspondientes en tiempo de ejecución, ahorrando código y haciéndolo mas legible y estructurado.

4.10.3 <slot>

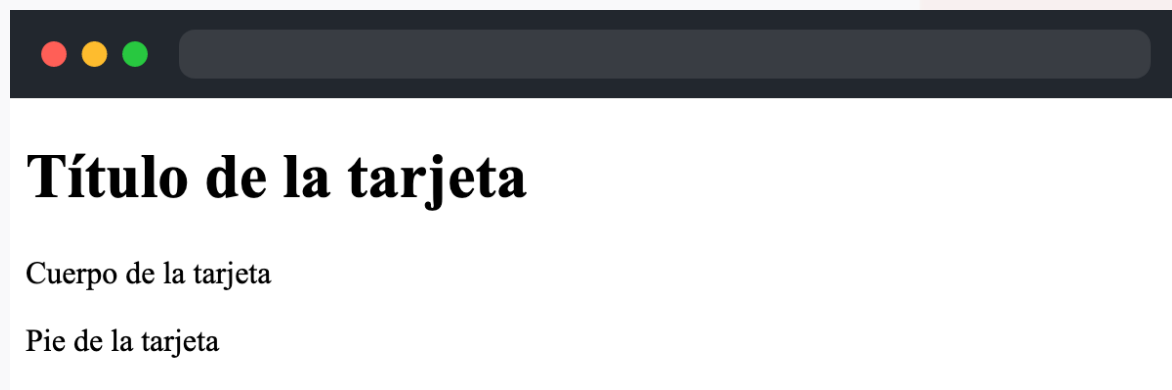
La etiqueta **<slot>** en HTML es una forma de personalizar contenido en una web component, es un contenedor dentro de un web component que espera recibir contenido externo, el contenido que se inserta en el slot se mostrara en la posicion donde se encuentra el slot en el web component.

Un ejemplo de uso práctico podría ser un componente de tarjeta que tiene un slot

para el título, uno para el cuerpo y otro para el pie de la tarjeta:



El código se renderiza de esta manera al abrirlo en un navegador.



En este ejemplo, el componente **mi-tarjeta** tiene tres slots, uno para el título, uno para el cuerpo y uno para el pie de la tarjeta. El contenido que se inserta en los slots se mostrará en el lugar correspondiente dentro del componente **mi-tarjeta**.

Es importante mencionar que los web components son una tecnología experimental y su uso no está totalmente soportado por todos los navegadores y su comportamiento puede variar. Es recomendable usar una librería como lit-element o stencil.js para poder trabajar con facilidad con esta tecnología.

4.10.4 <canvas>

La etiqueta **<canvas>** en HTML es un contenedor para dibujar gráficos mediante scripts, como JavaScript. El área de dibujo se especifica mediante atributos `width` y `height` del elemento `canvas`. Los scripts pueden dibujar en el `canvas` mediante métodos como `getContext()` y `drawImage()`. Un `canvas` es como un lienzo en blanco en el que se puede dibujar mediante scripts en tiempo de ejecución.

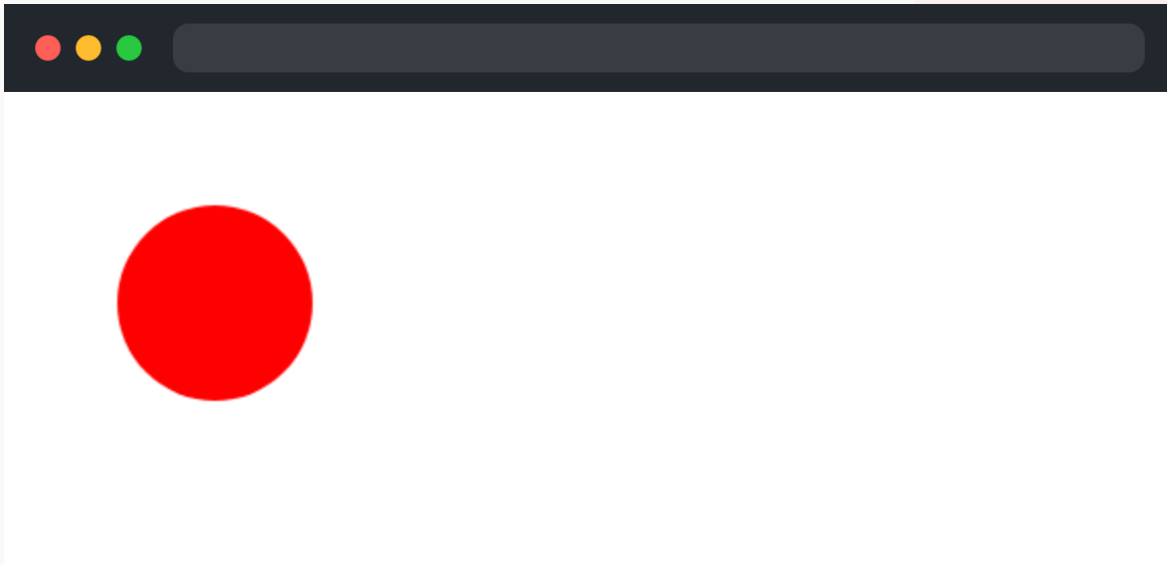
Supongamos que queremos dibujar un círculo rojo en un lienzo (`canvas`) en nuestro sitio web:



```
<canvas id="miCanvas" width="200" height="200"></canvas>
<script>
  // Obtener el contexto del canvas
  let canvas = document.getElementById("miCanvas");
  let ctx = canvas.getContext("2d");

  // Dibujar un círculo rojo
  ctx.beginPath();
  ctx.arc(100, 100, 50, 0, 2 * Math.PI);
  ctx.fillStyle = "red";
  ctx.fill();
</script>
```

El código se renderiza de esta manera al abrirlo en un navegador.



En este ejemplo, estamos creando un lienzo (canvas) con un identificador único “miCanvas” y un tamaño de 200 píxeles por 200 píxeles. Luego, en el bloque de script, estamos obteniendo el contexto del canvas usando el método `getContext()`, que nos permite dibujar en el canvas.

Luego, estamos usando el método `beginPath()` para comenzar un nuevo trazo, y `arc()` para dibujar un círculo en las coordenadas $x = 100$, $y = 100$, con un radio de 50 píxeles. Finalmente, estamos llenando el círculo con color rojo usando el método `fillStyle` y `fill()`. Al cargar la página, deberíamos ver un círculo rojo dibujado en el canvas.

Es importante mencionar que también existen otras librerías y frameworks que pueden usarse para dibujar en un canvas, como por ejemplo `p5.js` o `Three.js`.

4.11 Desuso

Algunas etiquetas son consideradas obsoletas y ya no se recomienda su uso en nuevos desarrollos.

Entre ellas se encuentran **<big>**, **<center>**, ****, **<s>**, **<strike>** entre otras.

Estas etiquetas se utilizaron en versiones antiguas de HTML, pero con el avance de las especificaciones y la mejora en la semántica del lenguaje, se han descontinuado.

Existen alternativas para estas etiquetas que brindan mejor soporte y accesibilidad para los navegadores y los programas de lectura de pantalla, como el uso de CSS en lugar de **<big>** y **** para controlar el tamaño del texto, **** en lugar de **<i>** y **** en lugar de ****, “del” en lugar de “strike” .

Es importante mencionar que si se está desarrollando una nueva página web, se recomienda evitar el uso de etiquetas obsoletas y utilizar alternativas actuales para garantizar la compatibilidad y accesibilidad en un futuro.

5 Atributos

5.1 Globales

Los atributos globales en HTML son atributos que se pueden utilizar en cualquier etiqueta HTML. Estos atributos proporcionan información adicional sobre los elementos de la página web, como su accesibilidad, estilo y eventos. Algunos ejemplos de atributos globales son:

- **class**: Este atributo se utiliza para asignar una o varias clases CSS a un elemento. Por ejemplo: **<p class="error">Contenido de error</p>**
- **id**: Este atributo se utiliza para identificar de manera única un elemento en una página web. Por ejemplo: **<div id="encabezado">Encabezado</div>**
- **style**: Este atributo se utiliza para aplicar estilos CSS directamente a un elemento. Por ejemplo: **<h1 style="color: red;">Título en rojo</h1>**
- **title**: Este atributo se utiliza para proporcionar una descripción adicional de un

5 ATRIBUTOS

elemento, y se muestra como una sugerencia al pasar el cursor sobre el elemento. Por ejemplo: ``

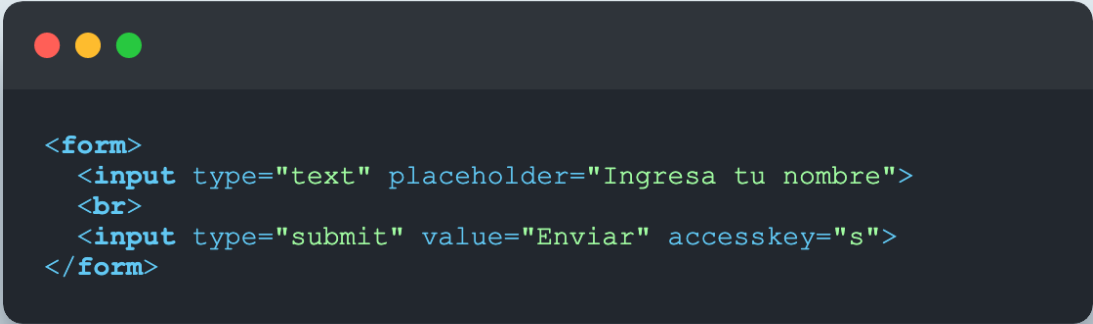
- **accesskey**: Este atributo se utiliza para asignar una tecla de acceso rápido a un elemento, permitiendo al usuario acceder rápidamente al elemento al presionar la tecla de acceso. Por ejemplo: `<button accesskey="s">Guardar</button>`

Es importante notar que algunos atributos pueden ser utilizados solo en algunas etiquetas específicas, y no todos los atributos son compatibles con todas las versiones de HTML.

5.1.1 accesskey

El atributo “accesskey” en HTML permite asignar una tecla de acceso rápido a un elemento de la página, de modo que el usuario puede activarlo presionando esa tecla mientras tiene presionada la tecla “Alt” (en Windows) o “Control” (en Mac).

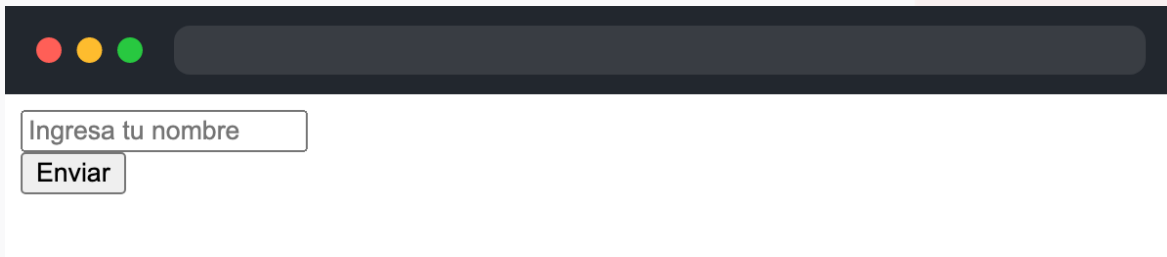
Por ejemplo, si queremos asignar la tecla “s” como acceso rápido a un botón de envío de un formulario, podemos utilizar el siguiente código:



```
<form>
  <input type="text" placeholder="Ingresa tu nombre">
  <br>
  <input type="submit" value="Enviar" accesskey="s">
</form>
```

El código se renderiza de esta manera al abrirlo en un navegador.

5 ATRIBUTOS



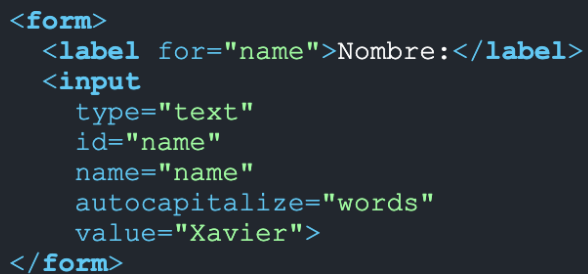
A screenshot of a web browser window. The browser has a dark title bar with three colored window control buttons (red, yellow, green) on the left. Below the title bar is a white page. The page contains a form with a text input field and a button. The input field has a light gray border and contains the placeholder text 'Ingresa tu nombre'. Below the input field is a button with a light gray border and the text 'Enviar'.

En este caso, cuando el usuario presiona “Alt+s” (en Windows) o “Control+s” (en Mac), el botón de envío del formulario será activado.

5.1.2 autocapitalize

El atributo “autocapitalize” en HTML es específico para los dispositivos móviles y se utiliza para especificar cómo se deben capitalizar las letras automáticamente en un elemento de entrada de texto.

Por ejemplo, si queremos que un campo de formulario para ingresar el nombre de una persona se capitalice automáticamente la primera letra de cada palabra, podemos utilizar el siguiente código:

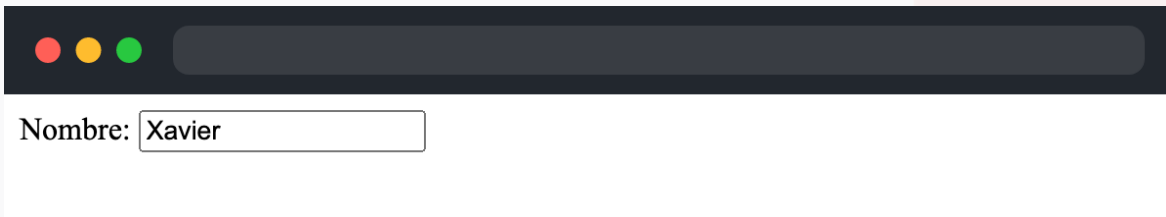


```
<form>
  <label for="name">Nombre:</label>
  <input
    type="text"
    id="name"
    name="name"
    autocapitalize="words"
    value="Xavier">
</form>
```

A screenshot of a code editor window with a dark background and a light orange border. The code editor shows HTML code for a form. The code is as follows:

El código se renderiza de esta manera al abrirlo en un navegador.

5 ATRIBUTOS



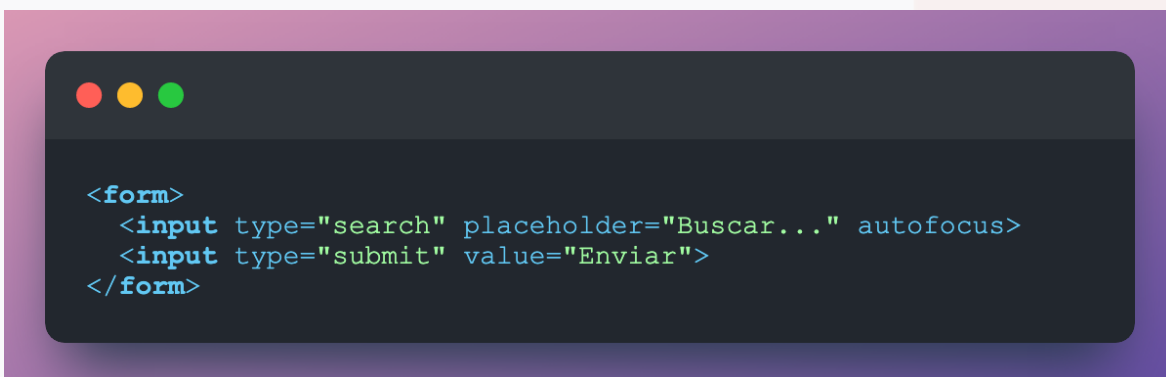
En este caso, cuando el usuario ingresa su nombre en el campo, la primera letra de cada palabra se capitalizará automáticamente.

Existen diferentes valores que se pueden utilizar en el atributo `autocapitalize`, como “off” (para deshabilitar la capitalización automática), “sentences” (para capitalizar automáticamente la primera letra de cada oración) y “characters” (para capitalizar automáticamente todas las letras).

5.1.3 autofocus

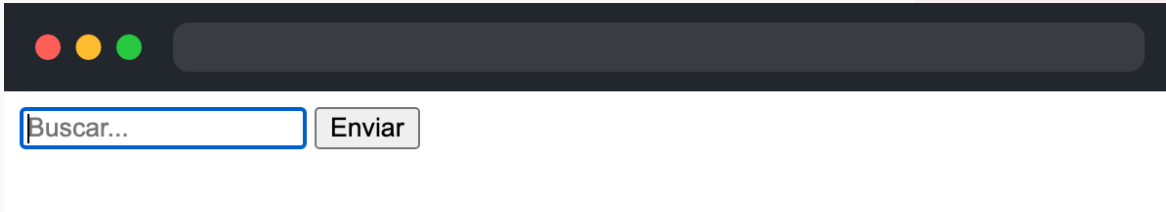
El atributo “autofocus” en HTML se utiliza para indicar que un elemento de entrada debe recibir el foco automáticamente cuando se carga la página. Esto significa que el cursor se posicionará automáticamente en ese elemento y el usuario podrá comenzar a escribir sin tener que hacer clic en él.

Por ejemplo, si queremos que un campo de búsqueda tenga el foco automáticamente cuando se cargue la página, podemos utilizar el siguiente código:



5 ATRIBUTOS

El código se renderiza de esta manera al abrirlo en un navegador.



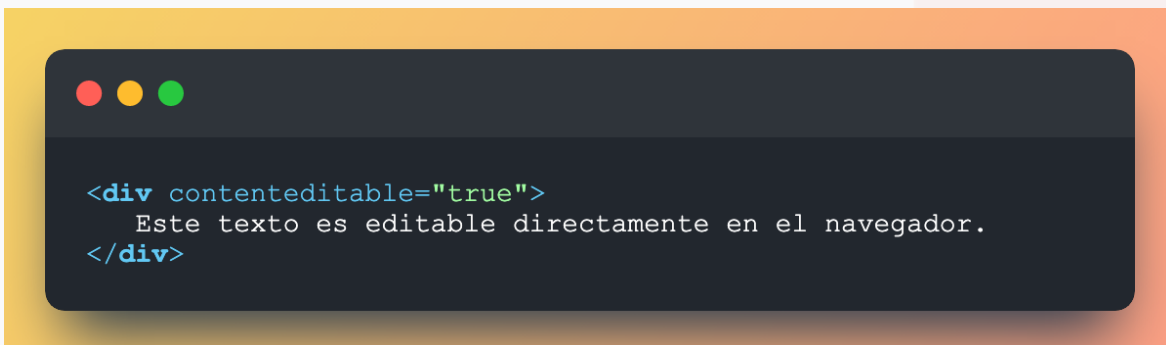
En este caso, cuando se carga la página, el cursor se posicionará automáticamente en el campo de búsqueda y el usuario podrá comenzar a escribir sin tener que hacer clic en él.

Es importante tener en cuenta que solo un elemento de una página puede tener el atributo autofocus, si en una pagina tiene varios elementos con este atributo solo uno sera el que tendra el foco.

5.1.4 contenteditable

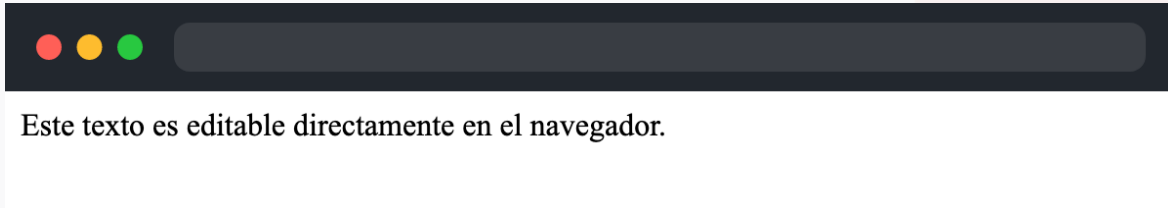
El atributo “contenteditable” en HTML se utiliza para indicar que el contenido de un elemento puede ser editado directamente en el navegador. Esto permite al usuario modificar el contenido de un elemento sin tener que utilizar un formulario o un campo de entrada específico.

Por ejemplo, si queremos que una sección de una página sea editable directamente por el usuario, podemos utilizar el siguiente código:



5 ATRIBUTOS

El código se renderiza de esta manera al abrirlo en un navegador.



En este caso, cuando el usuario hace clic en la sección, el contenido será editable directamente en el navegador, permitiendo al usuario modificar el texto.

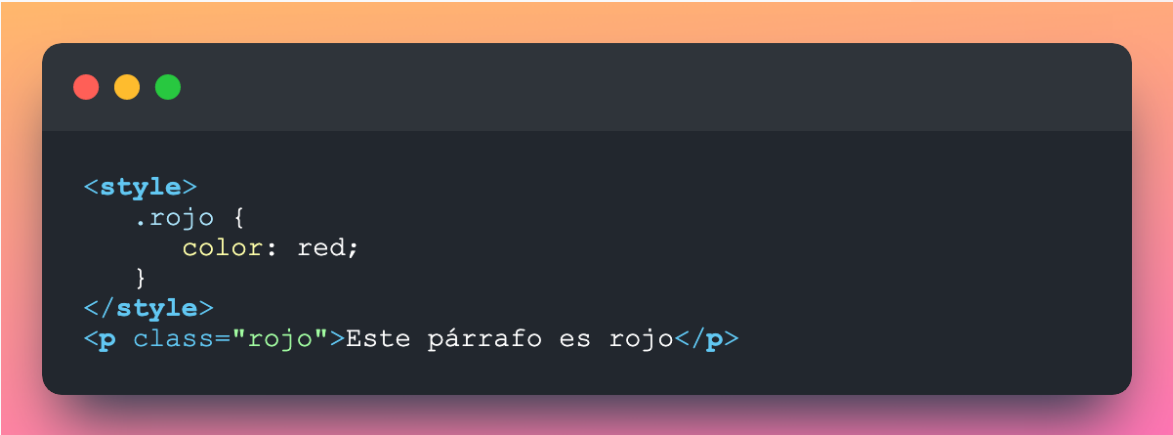
Es recomendable usarlo solo en navegadores modernos y no es recomendable usarlo en campos de entrada de información confidencial o sensibles.

5.1.5 class

El atributo “class” en HTML se utiliza para asignar uno o varios nombres de clase a un elemento. Los nombres de clase son utilizados en conjunto con hojas de estilo en cascada (CSS) para especificar el estilo de los elementos en una página web. Los nombres de clase pueden ser reutilizados en varios elementos de la página, lo que permite aplicar estilos consistentes a varios elementos con un solo conjunto de reglas CSS.

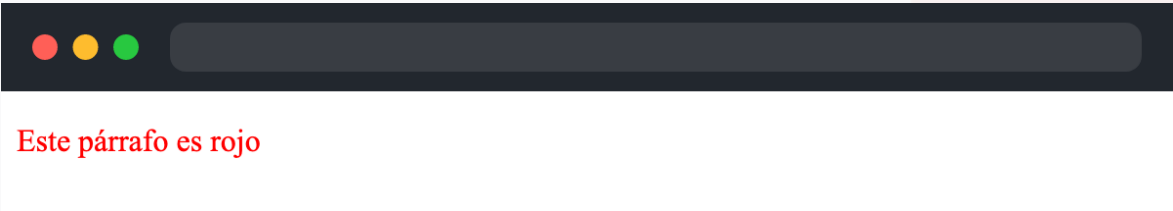
Por ejemplo, si queremos asignar la clase “rojo” a un elemento “p”, podemos utilizar el siguiente código:

5 ATRIBUTOS

A code editor window with a dark background and three colored window control buttons (red, yellow, green) in the top-left corner. The code is as follows:

```
<style>
  .rojo {
    color: red;
  }
</style>
<p class="rojo">Este párrafo es rojo</p>
```

El código se renderiza de esta manera al abrirlo en un navegador.

A browser window with a dark header bar containing three colored window control buttons (red, yellow, green) and a search bar. The main content area is white and displays the text "Este párrafo es rojo" in red color.

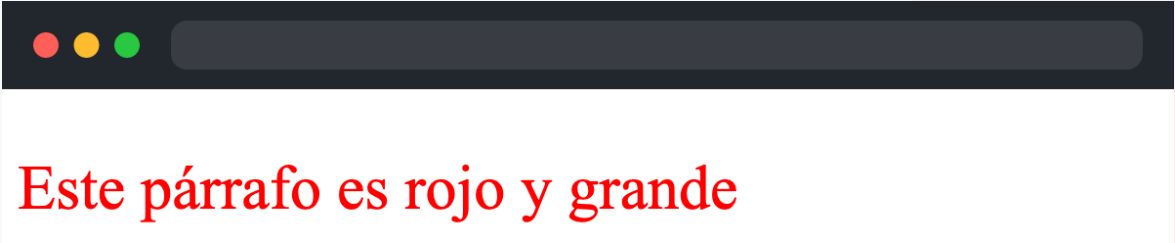
Este párrafo es rojo

En este caso, el atributo “class” está asignando la clase “rojo” al elemento “p” , y con una regla CSS se puede especificar el color rojo.

Es importante tener en cuenta que un elemento puede tener varias clases asignadas, separadas por un espacio, y cada clase tiene su propia regla CSS.

```
<style>
  .rojo {
    color: red;
    font-size: 2rem;
  }
</style>
<p class="rojo grande">Este párrafo es rojo y grande</p>
```

El código se renderiza de esta manera al abrirlo en un navegador.



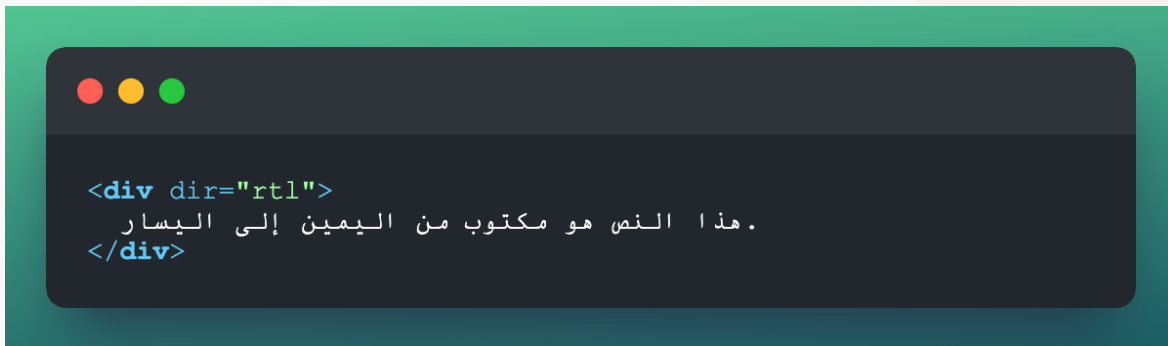
Este párrafo es rojo y grande

Además, es importante mencionar que el atributo `class` es uno de los más utilizados en HTML para aplicar estilos a elementos, es altamente recomendable utilizarlo en conjunto con un archivo CSS para separar el contenido y estilo de la página.

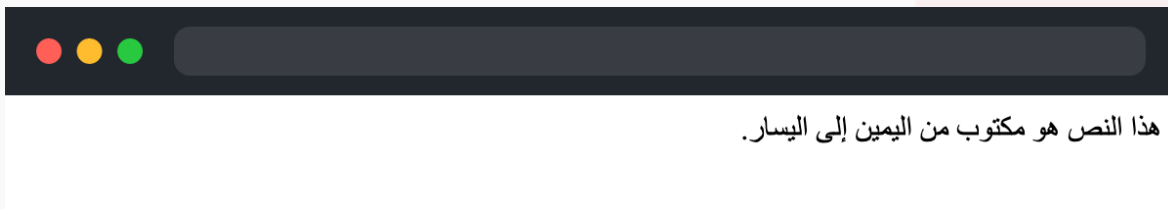
5.1.6 `dir`

El atributo `“dir”` en HTML se utiliza para especificar la dirección del texto en un elemento. Puede tener dos valores: `“ltr”` (de izquierda a derecha) y `“rtl”` (de derecha a izquierda).

Por ejemplo, si queremos que una sección de una página tenga el texto de derecha a izquierda, podemos utilizar el siguiente código:



El código se renderiza de esta manera al abrirlo en un navegador.



En este caso, el texto dentro de la sección estará escrito de derecha a izquierda.

Es importante tener en cuenta que este atributo es especialmente útil para idiomas que se escriben de derecha a izquierda, como el árabe o el hebreo. También puede ser útil para elementos que contienen texto mixto, como una dirección postal que contiene tanto números como letras.

5.1.7 draggable

El atributo “draggable” en HTML se utiliza para indicar que un elemento puede ser arrastrado y soltado. Esto permite al usuario mover elementos dentro de una página o entre páginas o aplicaciones mediante una acción de arrastrar y soltar.

Por ejemplo, si queremos que una imagen sea arrastrable, podemos utilizar el siguiente código:

5 ATRIBUTOS

```

```

El código se renderiza de esta manera al abrirlo en un navegador.



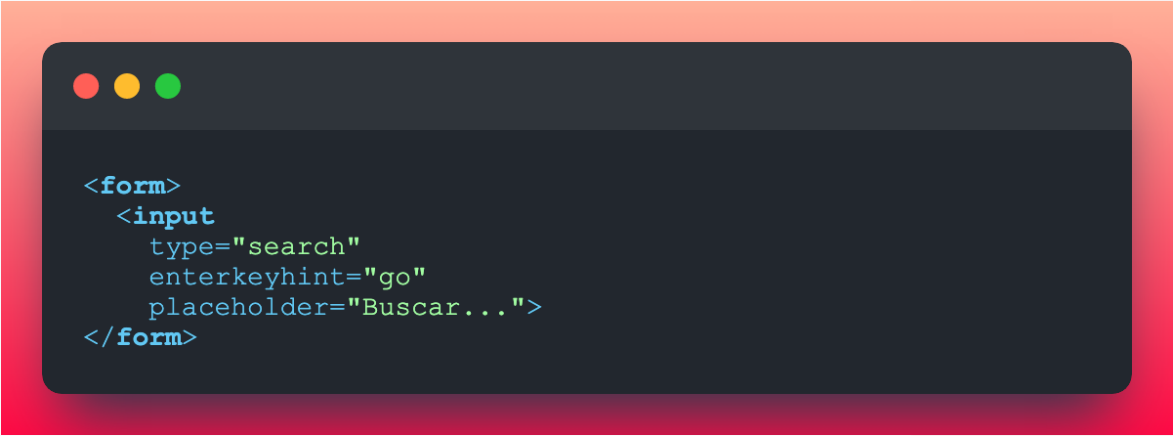
En este caso, cuando el usuario hace clic en la imagen y la arrastra, podrá moverla dentro de la página o incluso a otra aplicación.

Es importante tener en cuenta que para que el elemento sea arrastrable se debe de establecer el atributo `draggable` en `true`, y también se debe de establecer eventos de `"dragstart"`, `"drag"` y `"dragend"` para poder darle una funcionalidad al elemento.

5.1.8 enterkeyhint

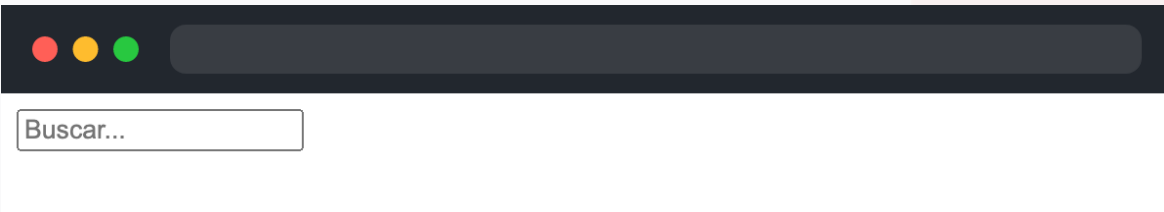
El atributo “enterkeyhint” en HTML es un atributo que se utiliza para sugerir al navegador cómo manejar la tecla “Enter” cuando el usuario la presiona en un elemento de entrada. Puede tener varios valores como “enter” , “done” , “go” , “next” y “previous” .

Por ejemplo, si queremos que cuando el usuario presione “Enter” en un campo de búsqueda, el navegador realice la búsqueda, podemos utilizar el siguiente código:

A screenshot of a code editor window with a dark background and a red-to-orange gradient border. The code is written in a light blue font and defines an HTML form with a search input field. The input field has a type of "search", an enterkeyhint of "go", and a placeholder text "Buscar...".

```
<form>
  <input
    type="search"
    enterkeyhint="go"
    placeholder="Buscar...">
</form>
```

El código se renderiza de esta manera al abrirlo en un navegador.



En este caso, cuando el usuario presiona “Enter” en el campo de búsqueda, el navegador entenderá que debe de realizar la búsqueda.

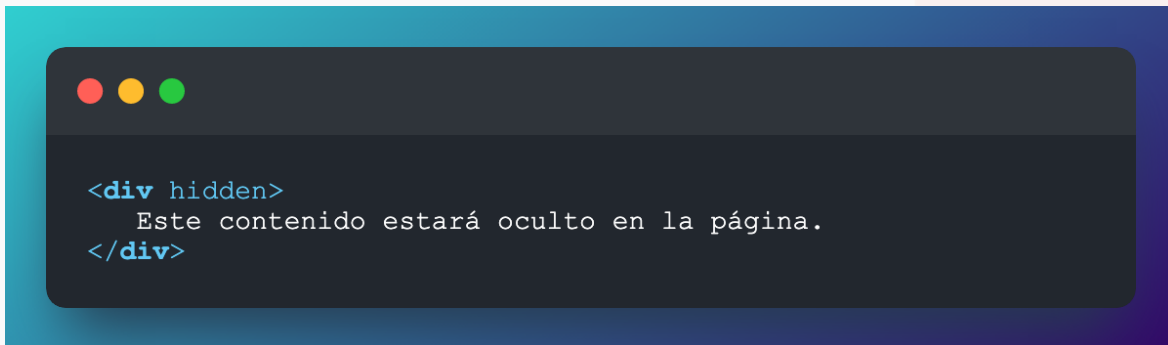
Es recomendable usarlo solo en navegadores modernos y en conjunto con JavaScript para asegurar su funcionamiento en todos los navegadores.

5 ATRIBUTOS

5.1.9 hidden

El atributo “hidden” en HTML se utiliza para indicar que un elemento debe estar oculto en la página. El elemento estará presente en el documento HTML, pero no será visible en la página.

Por ejemplo, si queremos ocultar una sección de una página, podemos utilizar el siguiente código:



El código se renderiza de esta manera al abrirlo en un navegador.



En este caso, el contenido dentro de la sección estará oculto en la página.

Es importante tener en cuenta que el elemento oculto sigue ocupando espacio en la página, solo no es visible, también se puede utilizar la propiedad CSS `display: none` para lograr el mismo resultado, pero esta propiedad no solo oculta el elemento sino que también lo elimina del flujo del documento.

5 ATRIBUTOS

5.1.10 id

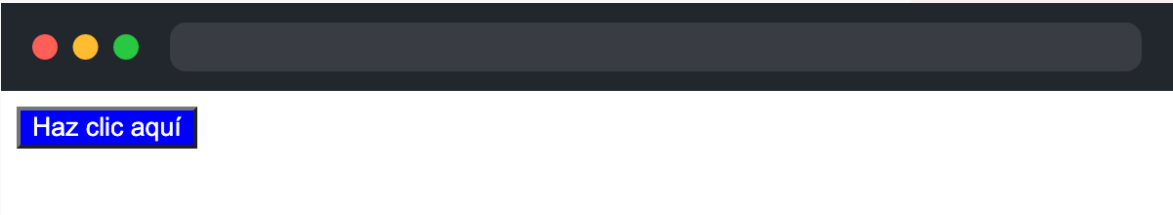
El atributo “id” en HTML se utiliza para asignar un identificador único a un elemento. Los identificadores son utilizados en conjunto con hojas de estilo en cascada (CSS) y JavaScript para especificar el estilo y la funcionalidad de los elementos en una página web. Un identificador solo puede ser utilizado una vez en una página, y es útil para seleccionar elementos específicos para aplicar estilos o agregar funcionalidad.

Por ejemplo, si queremos asignar el identificador “boton1” a un elemento “button”, podemos utilizar el siguiente código:



```
<style>
  #boton1 {
    background-color: blue;
    color: white;
  }
</style>
<button id="boton1">Haz clic aquí</button>
```

El código se renderiza de esta manera al abrirlo en un navegador.



Haz clic aquí

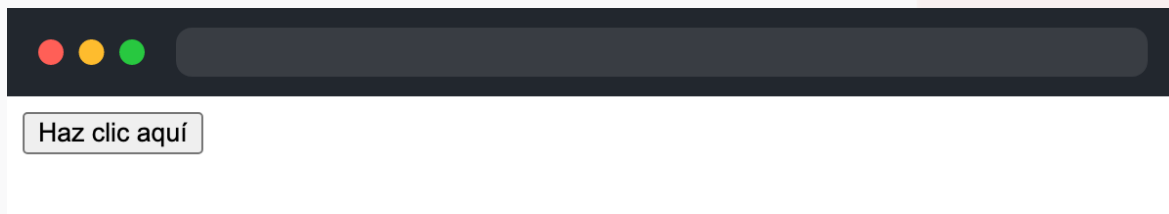
En este caso, el atributo “id” está asignando el identificador “boton1” al elemento “button”, y con una regla CSS se puede especificar el estilo de ese botón específico.

5 ATRIBUTOS

Además, el atributo id también es utilizado para seleccionar elementos específicos en JavaScript, lo que permite agregar funcionalidad a ese elemento específico.



El código se renderiza de esta manera al abrirlo en un navegador.



En este caso se está seleccionando al elemento con id “boton1” y se está agregando un evento click al mismo.

Es importante mencionar que el atributo id es único en una página y es utilizado para seleccionar elementos específicos, es recomendable utilizarlo con cuidado y evitar su uso en elementos que se repiten varias veces en una página.

5.1.11 inert

El atributo “inert” en HTML es un atributo que se utiliza para indicar que un elemento no debe ser interactivo. El elemento estará presente en el documento HTML, pero no

5 ATRIBUTOS

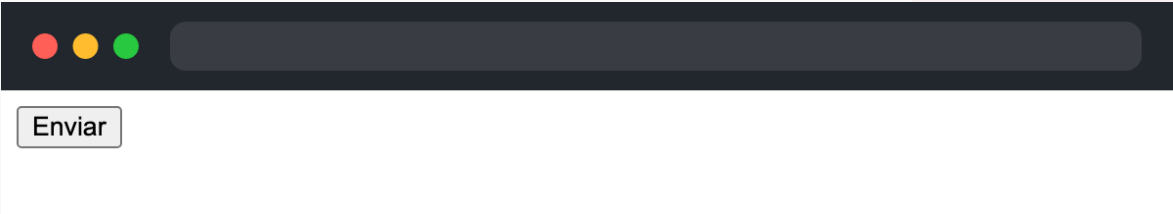
podrá ser seleccionado ni interactuado con el cursor, ni recibirá eventos de entrada como clic o toque.

Por ejemplo, si queremos desactivar un botón en un formulario, podemos utilizar el siguiente código:



```
<button type="submit" inert>Enviar</button>
```

El código se renderiza de esta manera al abrirlo en un navegador.



Enviar

En este caso, el botón estará presente en el documento HTML, pero no podrá ser interactuado, no podrá ser presionado, ni recibirá eventos de entrada.

Es recomendable usarlo solo en navegadores modernos y en conjunto con JavaScript para asegurar su funcionamiento en todos los navegadores y para asegurar que el botón no es interactuado.

5.1.12 inputmode

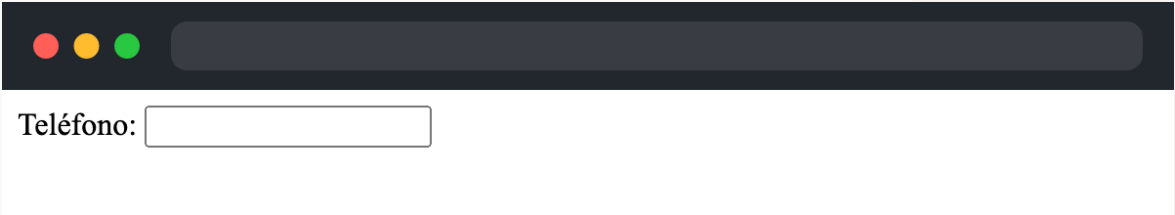
El atributo “inputmode” en HTML es un atributo que se utiliza para indicar el tipo de entrada esperado en un elemento de entrada de texto. Puede tener varios valores como “verbatim”, “latin”, “latin-name”, “latin-prose”, “full-width-latin”, “kana”, “katakana”, “numeric”, “tel”, “email”, “url” .

5 ATRIBUTOS

Por ejemplo, si queremos que un campo de formulario solo acepte números, podemos utilizar el siguiente código:

```
<form>
  <label for="teléfono">Teléfono:</label>
  <input
    type="text"
    id="teléfono"
    name="teléfono"
    inputmode="numeric">
</form>
```

El código se renderiza de esta manera al abrirlo en un navegador.



Teléfono:

En este caso, el campo de teléfono solo aceptará números.

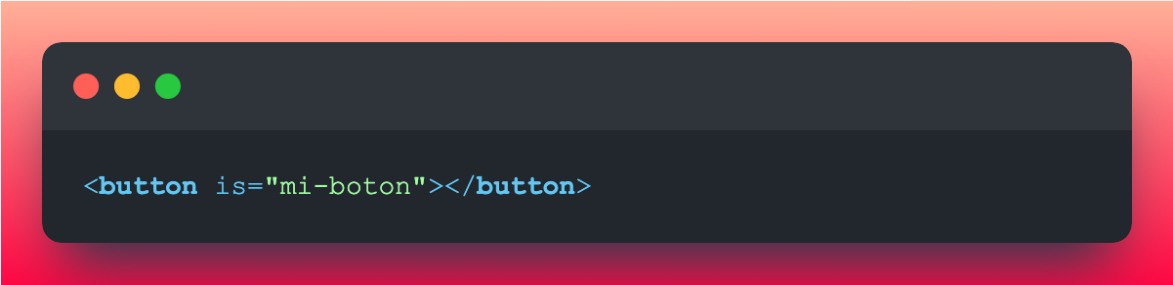
Es recomendable usarlo solo en navegadores modernos y en conjunto con JavaScript para asegurar su funcionamiento en todos los navegadores y para asegurar que el input se está restringiendo de acuerdo al tipo de entrada esperado.

5.1.13 is

El atributo “is” en HTML es un atributo para el uso en Custom Elements, es utilizado para especificar el nombre de la clase de un elemento personalizado.

5 ATRIBUTOS

Por ejemplo, si queremos crear un elemento personalizado “mi-boton” , podemos utilizar el siguiente código:



```
<button is="mi-boton"></button>
```

En este caso, el elemento “button” se comportaría como si fuera un “mi-boton” debido a que esta siendo modificado por la clase “mi-boton” .

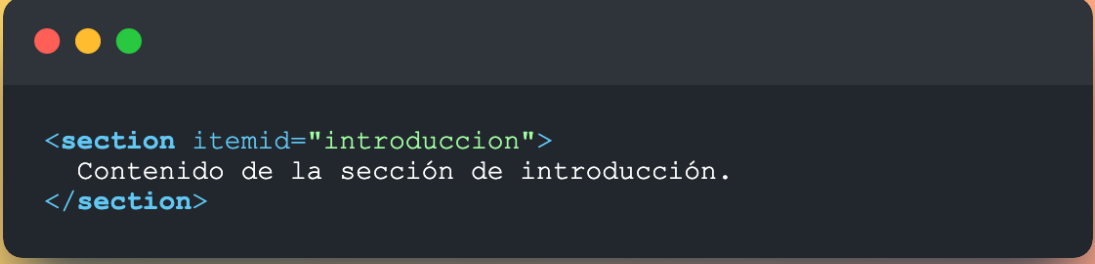
Es recomendable usarlo solo en navegadores modernos y en conjunto con JavaScript para asegurar su funcionamiento en todos los navegadores y para asegurar que el elemento personalizado esta siendo aplicado correctamente.

5.1.14 itemid

El atributo “itemid” en HTML es un atributo que se utiliza para especificar un identificador único para un elemento en una página. Este identificador se puede utilizar para enlazar a ese elemento específico desde otras partes de la página o desde otras páginas web.

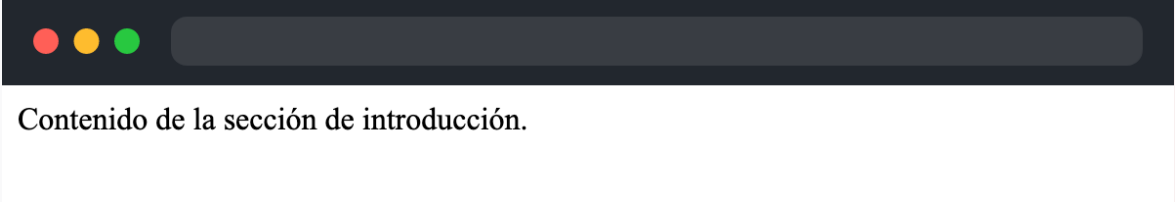
Por ejemplo, si queremos enlazar a una sección específica de una página, podemos utilizar el siguiente código:

5 ATRIBUTOS



```
<section itemid="introduccion">
  Contenido de la sección de introducción.
</section>
```

El código se renderiza de esta manera al abrirlo en un navegador.



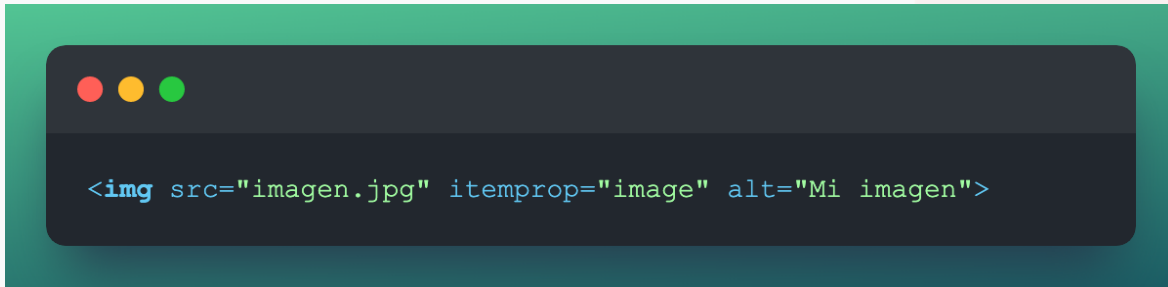
Contenido de la sección de introducción.

En este caso, el identificador “introduccion” se puede utilizar para enlazar a esta sección específica desde otras partes de la página o desde otras páginas web.

5.1.15 itemprop

El atributo “itemprop” en HTML es utilizado en conjunto con el vocabulario de metadatos de Microdata, para especificar las propiedades de un elemento en una página. Estas propiedades pueden ser utilizadas para proporcionar información adicional sobre el elemento a los motores de búsqueda y a otros agentes de recuperación de información.

Por ejemplo, si queremos proporcionar información adicional sobre una imagen en una página, podemos utilizar el siguiente código:



En este caso, se está especificando que la imagen es una propiedad de item.

Es importante tener en cuenta que el atributo `itemprop` es utilizado en conjunto con la estructura de Microdata, para dar significado a los elementos de una página y proporcionar información adicional a los motores de búsqueda y otros agentes de recuperación de información.

5.1.16 itemref

El atributo “`itemref`” en HTML es utilizado en conjunto con el vocabulario de metadatos de Microdata, para especificar los elementos relacionados con un elemento en una página. Estos elementos relacionados pueden proporcionar información adicional sobre el elemento, pero están localizados fuera del elemento en sí.

Por ejemplo, si queremos proporcionar información adicional sobre un artículo en una página, pero esta información se encuentra en una sección diferente de la página, podemos utilizar el siguiente código:

```
<article itemscope>
  <h1 itemprop="name">Mi artículo</h1>
  <div itemprop="author" itemscope
    itemtype="http://schema.org/Person">
    <span itemprop="name">Juan Pérez</span>
  </div>
  <div itemref="notes"></div>
</article>
...
<div id="notes">
  Notas adicionales sobre el artículo.
</div>
```

El código se renderiza de esta manera al abrirlo en un navegador.

Mi artículo

Juan Pérez

...

Notas adicionales sobre el artículo.

5.1.17 itemscope

El atributo “itemscope” en HTML es utilizado en conjunto con el vocabulario de metadatos de Microdata, para indicar que un elemento es una entidad independiente con propiedades. Este atributo permite especificar las propiedades de un elemento utilizando los atributos “itemprop” y “itemtype” .

5 ATRIBUTOS

Por ejemplo, si queremos proporcionar información adicional sobre un artículo en una página, podemos utilizar el siguiente código:

```
<article itemscope>
  <h1 itemprop="name">Mi artículo</h1>
  <div
    itemprop="author"
    itemscope
    itemtype="http://schema.org/Person">
    <span itemprop="name">Juan Pérez</span>
  </div>
  <p itemprop="text">Contenido del artículo</p>
</article>
```

El código se renderiza de esta manera al abrirlo en un navegador.

Mi artículo

Juan Pérez

Contenido del artículo

En este caso, el atributo “itemscope” está indicando que el elemento “article” es una entidad independiente con propiedades como “name”, “author” y “text” .

Es importante tener en cuenta que el atributo itemscope es utilizado en conjunto con la estructura de Microdata, para dar significado a los elementos de una página y proporcionar información adicional a los motores de búsqueda y otros agentes de recuperación de información.

5.1.18 itemtype

El atributo “itemtype” en HTML es utilizado en conjunto con el vocabulario de metadatos de Microdata, para especificar el tipo de entidad representado por un elemento con el atributo “itemscope”. Puede especificar una URL que identifica un vocabulario específico que se utilizará para describir las propiedades del elemento.

Por ejemplo, si queremos proporcionar información adicional sobre una persona en una página, podemos utilizar el siguiente código:

```
<div itemscope itemtype="http://schema.org/Person">
  <span itemprop="name">Juan Pérez</span>
  <span itemprop="jobTitle">Desarrollador web</span>
  <a itemprop="url"
    href="https://www.ejemplo.com">Sitio web</a>
</div>
```

El código se renderiza de esta manera al abrirlo en un navegador.

Juan Pérez Desarrollador web [Sitio web](https://www.ejemplo.com)

En este caso, el atributo “itemtype” está especificando que el elemento “div” representa una entidad del tipo “<http://schema.org/Person>”, lo cual indica que se esperan propiedades como “name”, “jobTitle” y “url” para describir a esa persona. Esto permite a los motores de búsqueda y otros agentes de recuperación de información entender y utilizar esta información de manera más eficiente.

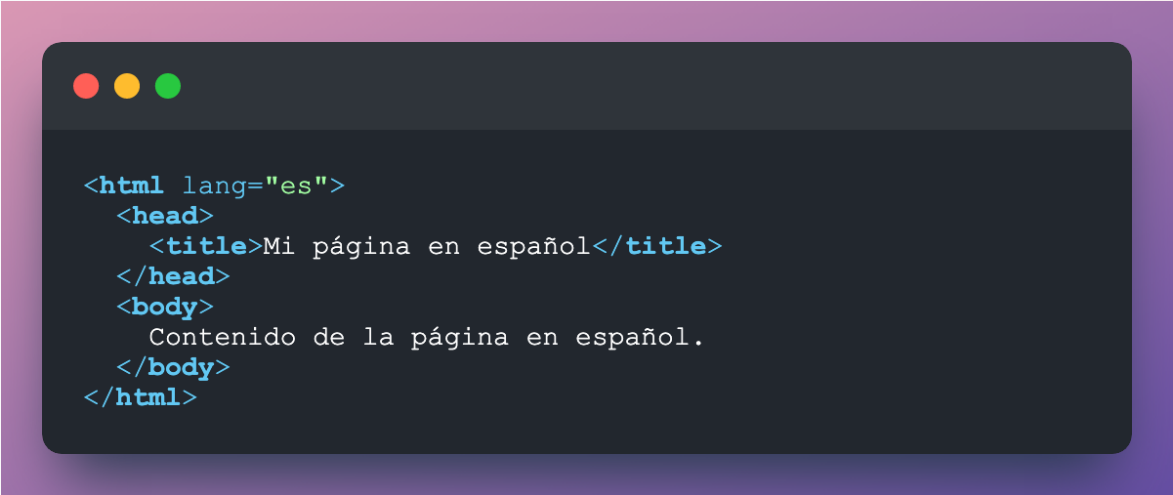
5 ATRIBUTOS

Es importante tener en cuenta que el atributo `itemtype` es utilizado en conjunto con la estructura de Microdata y es utilizado para indicar el tipo de entidad representada por un elemento con `itemscope`, permitiendo una mejor interpretación de la información en la página y facilitando su procesamiento.

5.1.19 lang

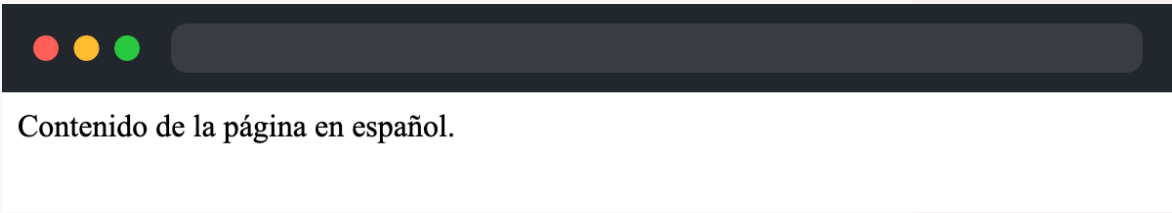
El atributo “`lang`” en HTML se utiliza para especificar el idioma del contenido de un elemento. Este atributo es útil para los motores de búsqueda y otros agentes de recuperación de información, ya que les permite entender mejor el contenido de una página y mejorar la accesibilidad para personas con discapacidades auditivas.

Por ejemplo, si queremos especificar que el contenido de una página está en español, podemos utilizar el siguiente código:



```
<html lang="es">
  <head>
    <title>Mi página en español</title>
  </head>
  <body>
    Contenido de la página en español.
  </body>
</html>
```

El código se renderiza de esta manera al abrirlo en un navegador.



Contenido de la página en español.

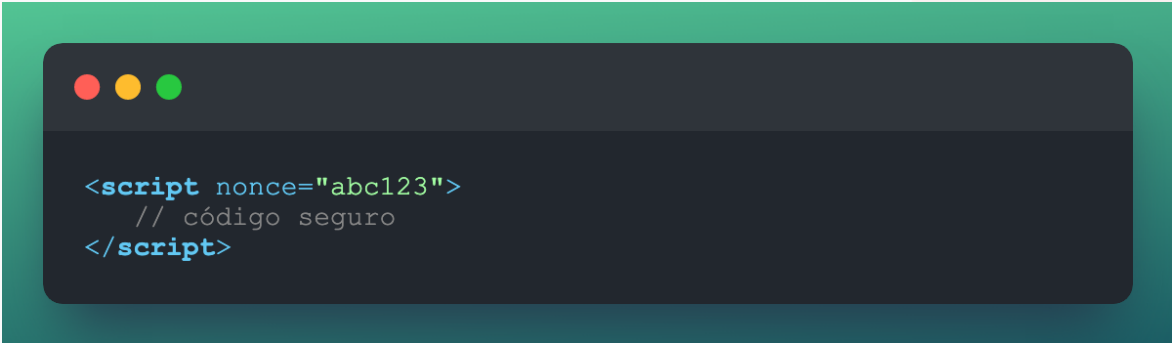
En este caso, el atributo “lang” debe ser utilizado en el elemento raíz de la página (en este caso, el elemento “html”) y en los elementos donde se cambie el idioma del contenido. Esto ayudará a los motores de búsqueda y otros agentes de recuperación de información a entender mejor el contenido de la página y mejorar la accesibilidad para personas con discapacidades auditivas.

Además, es importante especificar el idioma correcto ya que esto puede mejorar la experiencia de los usuarios y mejorar la accesibilidad para personas con discapacidades.

5.1.20 nonce

El atributo “nonce” en HTML es utilizado para proporcionar un valor único que se utilizará para verificar la seguridad de los scripts ejecutados en una página web. Este valor se genera de forma aleatoria y se utiliza para garantizar que solo los scripts que lo incluyen son ejecutados, evitando posibles ataques de inyección de script.

Por ejemplo, si queremos asegurarnos de que solo los scripts que incluyen el nonce “abc123” son ejecutados en nuestra página, podemos utilizar el siguiente código:



```
<script nonce="abc123">  
  // código seguro  
</script>
```

5 ATRIBUTOS

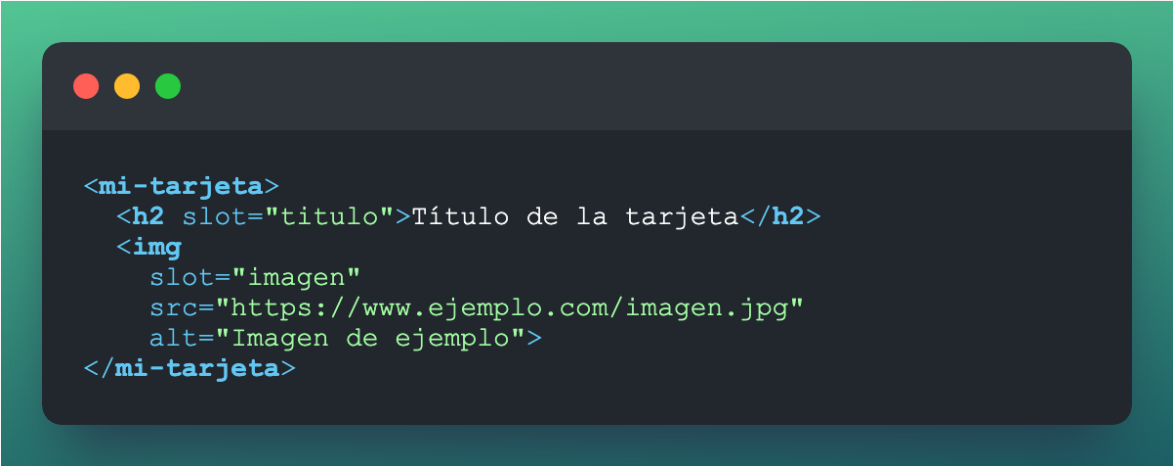
En este caso, el atributo “nonce” está proporcionando un valor único que se utilizará para verificar la seguridad del script ejecutado.

Es importante tener en cuenta que el atributo nonce se utiliza como medida de seguridad adicional, no es una solución completa para evitar ataques de inyección de script, sino que se debe combinar con otras medidas de seguridad para asegurar la seguridad de una página web.

5.1.21 Atributo slot

El atributo “slot” en HTML se utiliza en conjunto con la tecnología de “Web Components” para especificar un lugar donde los elementos hijos de un componente personalizado pueden ser insertados. Un componente personalizado puede tener varios “slots” donde los elementos hijos pueden ser insertados, y cada slot puede tener un nombre específico.

Por ejemplo, si queremos crear un componente personalizado que contenga un título y un botón, podemos utilizar el siguiente código:



```
<mi-tarjeta>
  <h2 slot="titulo">Título de la tarjeta</h2>
  
</mi-tarjeta>
```

El código se renderiza de esta manera al abrirlo en un navegador.



Título de la tarjeta



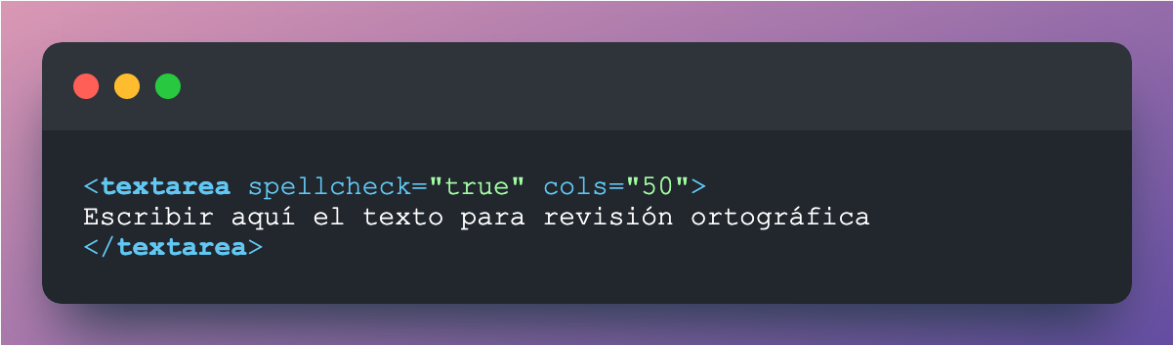
En este ejemplo, el componente “mi-tarjeta” tiene dos “slots” con nombres “título” y “imagen”, y los elementos “h2” y “img” son insertados en estos “slots” respectivamente.

De esta manera, el componente “mi-tarjeta” puede ser reutilizado en diferentes partes de la aplicación web y se pueden cambiar el título y la imagen sin tener que modificar el código del componente.

5.1.22 spellcheck

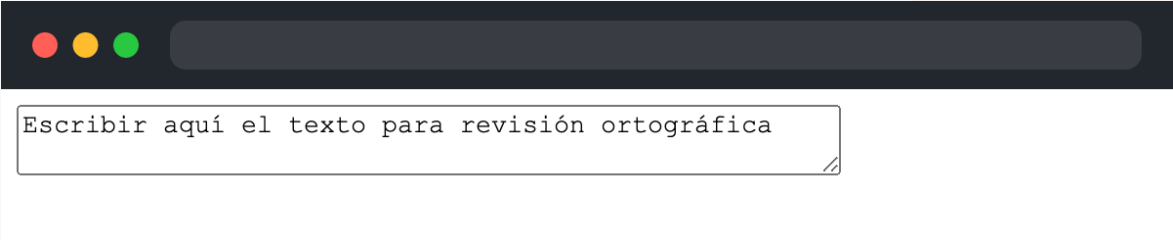
El atributo “spellcheck” en HTML se utiliza para habilitar o deshabilitar la revisión ortográfica automática en un elemento. Cuando se establece en “true” , la revisión ortográfica automática está habilitada y se marcan las palabras mal escritas con un subrayado rojo. Cuando se establece en “false” , la revisión ortográfica automática está deshabilitada.

Por ejemplo, si queremos habilitar la revisión ortográfica automática en un elemento “textarea” , podemos utilizar el siguiente código:

A screenshot of a code editor window with a dark background and a purple gradient border. The code editor shows the following HTML code:

```
<textarea spellcheck="true" cols="50">
Escribir aquí el texto para revisión ortográfica
</textarea>
```

El código se renderiza de esta manera al abrirlo en un navegador.

A screenshot of a web browser window. The browser's address bar is empty. Below the address bar is a text input field (textarea) with the placeholder text "Escribir aquí el texto para revisión ortográfica". The input field has a light gray border and a small cursor icon at the bottom right.

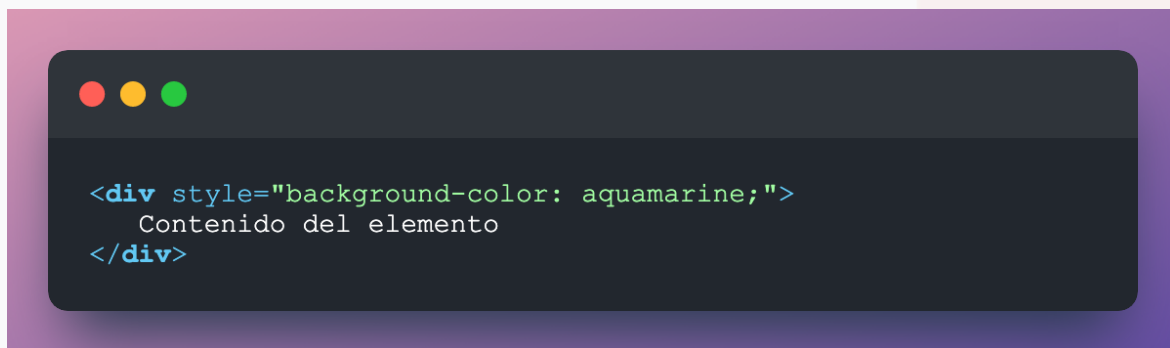
En este caso, el atributo “spellcheck” está establecido en “true” y se habilitará la revisión ortográfica automática en el elemento “textarea” .

Es importante tener en cuenta que el soporte para el atributo spellcheck puede variar entre los navegadores, es recomendable verificar la compatibilidad antes de usarlo en una aplicación web.

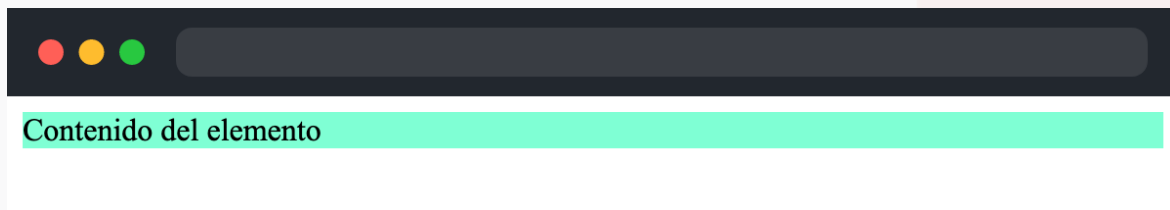
5.1.23 Atributo style

El atributo “style” en HTML se utiliza para especificar el estilo CSS de un elemento. Puede contener una o varias reglas de estilo CSS, y se aplican al elemento en el que se encuentra el atributo.

Por ejemplo, si queremos cambiar el color de fondo de un elemento “div” a azul, podemos utilizar el siguiente código:



El código se renderiza de esta manera al abrirlo en un navegador.



En este caso, el atributo “style” está especificando que el color de fondo del elemento “div” es azul.

Es importante tener en cuenta que el uso del atributo “style” no es recomendable para proyectos grandes o con un gran número de elementos, ya que hace difícil el mantenimiento y la reutilización de estilos. Es mejor utilizar un archivo CSS externo y enlazarlo en la página para tener una mayor separación entre contenido y estilo.

5 ATRIBUTOS

5.1.24 tabindex

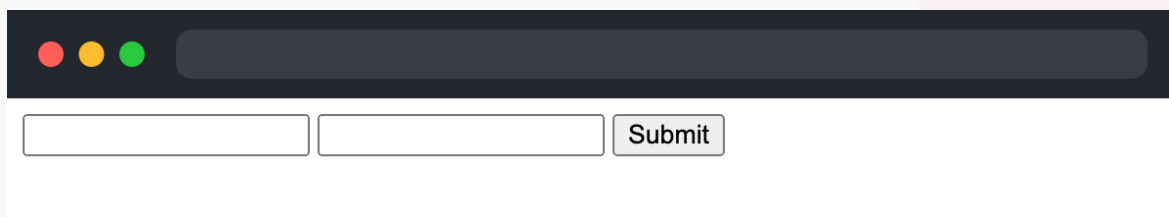
El atributo “tabindex” en HTML se utiliza para especificar el orden en el que los elementos son seleccionados cuando el usuario presiona la tecla “Tab” en el teclado. Los elementos con un valor de tabindex mayor serán seleccionados antes que los elementos con un valor menor. Los elementos con un valor de tabindex negativo no podrán ser seleccionados mediante la tecla “Tab” .

Por ejemplo, si queremos especificar el orden de selección de los elementos “input” en una página, podemos utilizar el siguiente código:



```
<input type="text" tabindex="1">
<input type="password" tabindex="2">
<input type="submit" tabindex="3">
```

El código se renderiza de esta manera al abrirlo en un navegador.



En este caso, el primer elemento “input” tiene un valor de tabindex de 1, el segundo elemento “input” tiene un valor de tabindex de 2 y el tercer elemento “input” tiene un valor de tabindex de 3. Esto significa que cuando el usuario presiona la tecla “Tab” , los elementos serán seleccionados en el orden especificado.

Es importante tener en cuenta que el valor de tabindex debe ser utilizado con precaución, ya que puede afectar negativamente la experiencia del usuario si no se

5 ATRIBUTOS

utiliza de manera adecuada.

5.1.25 Atributo title

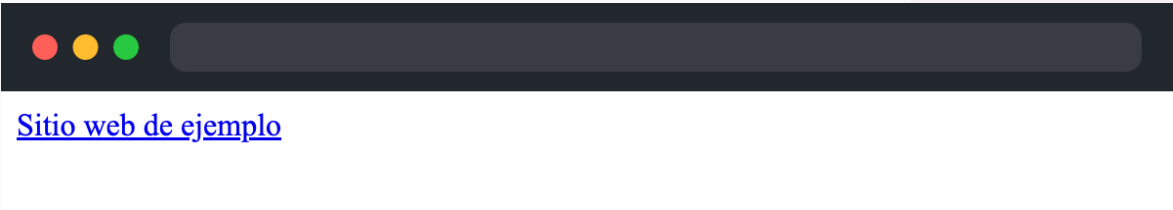
El atributo “title” en HTML se utiliza para proporcionar una descripción adicional de un elemento. El valor del atributo es mostrado como un “tooltip” cuando el usuario coloca el cursor sobre el elemento o lo selecciona. Este atributo es comúnmente utilizado en enlaces, pero también puede ser utilizado en otros elementos como imágenes, botones, etc.

Por ejemplo, si queremos proporcionar una descripción adicional para un enlace, podemos utilizar el siguiente código:



```
<a  
  href="https://www.ejemplo.com"  
  title="Visitar sitio web de ejemplo">  
  Sitio web de ejemplo</a>
```

El código se renderiza de esta manera al abrirlo en un navegador.



[Sitio web de ejemplo](https://www.ejemplo.com)

En este caso, el atributo “title” está proporcionando una descripción adicional del enlace que aparecerá como un “tooltip” cuando el usuario coloca el cursor sobre el enlace o lo selecciona.

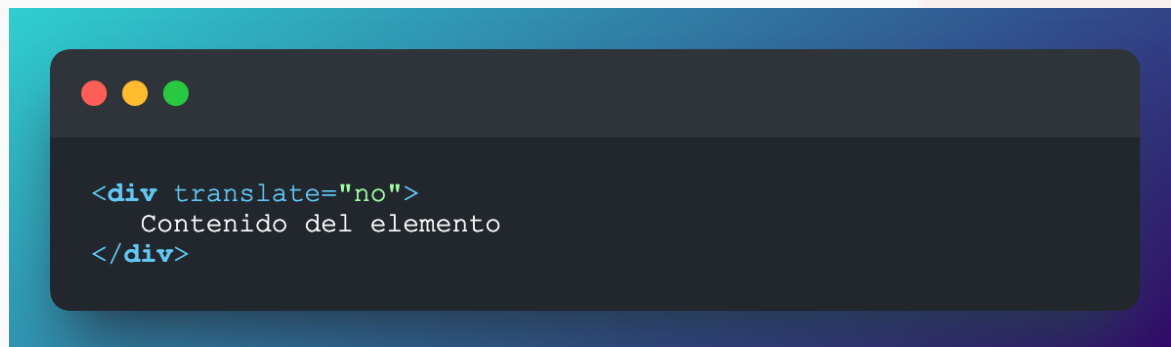
5 ATRIBUTOS

Es importante tener en cuenta que el atributo “title” puede ser utilizado en diferentes elementos y su contenido puede variar dependiendo del contexto y objetivo de la página, puede ser utilizado para brindar información adicional sobre un elemento o para brindar ayuda al usuario.

5.1.26 translate

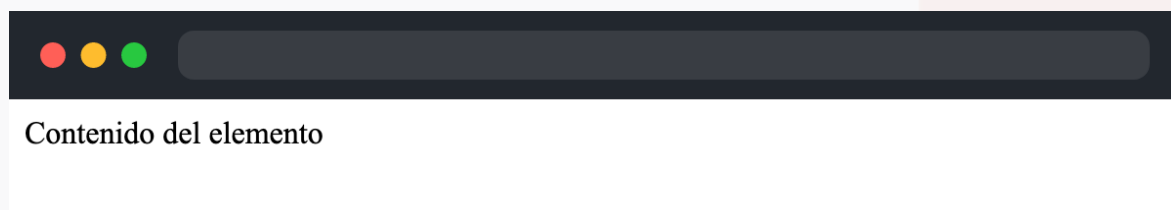
El atributo “translate” en HTML se utiliza para especificar si el contenido de un elemento debe ser traducido automáticamente o no. El valor puede ser “yes”, “no” o “auto”. Si se establece en “yes”, el contenido será traducido automáticamente, si se establece en “no” el contenido no será traducido automáticamente y si se establece en “auto” el navegador determinará si el contenido debe ser traducido automáticamente o no.

Por ejemplo, si queremos evitar que el contenido de un elemento “div” sea traducido automáticamente, podemos utilizar el siguiente código:



```
<div translate="no">
  Contenido del elemento
</div>
```

El código se renderiza de esta manera al abrirlo en un navegador.



5 ATRIBUTOS

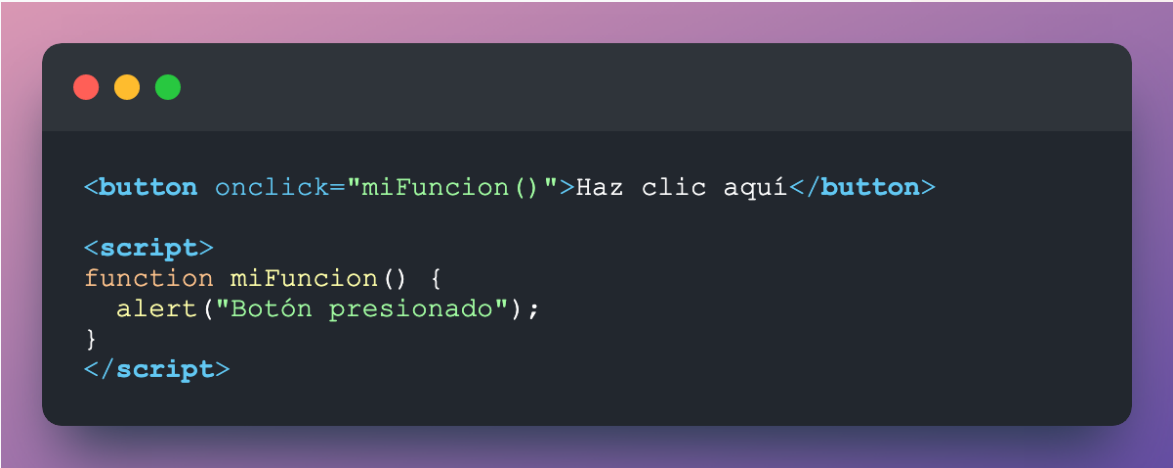
En este caso, el atributo “translate” está establecido en “no” y el contenido del elemento “div” no será traducido automáticamente.

Es recomendable verificar la compatibilidad antes de usarlo en una aplicación web.

5.2 Eventos

Los eventos en HTML son acciones que ocurren en una página web, como hacer clic en un botón, pasar el mouse sobre un elemento, cargar una página, etc. Pueden ser controlados mediante atributos de eventos en los elementos HTML.

Un ejemplo práctico sería agregar un evento de “click” a un botón. El siguiente código mostraría un botón en una página web y ejecutaría una función JavaScript cuando el botón es presionado:

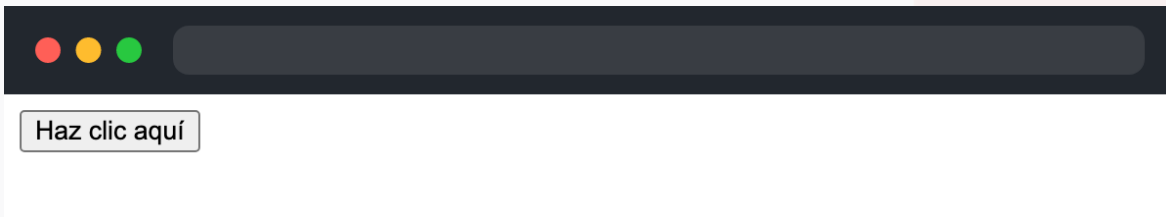


```
<button onclick="miFuncion()">Haz clic aquí</button>

<script>
function miFuncion() {
  alert("Botón presionado");
}
</script>
```

El código se renderiza de esta manera al abrirlo en un navegador.

5 ATRIBUTOS



En este ejemplo, cuando el usuario hace clic en el botón “Haz clic aquí” , se ejecutará la función JavaScript “miFuncion” y se mostrará un mensaje de alerta en la pantalla.

5.2.1 onauxclick

El atributo “onauxclick” en HTML es un evento que se activa cuando se hace clic con el botón auxiliar del mouse (normalmente el botón derecho del mouse) en un elemento. Este evento puede ser utilizado para ejecutar una función JavaScript específica cuando se activa.

Por ejemplo, si queremos mostrar un mensaje de alerta cuando se haga clic con el botón derecho del mouse en un elemento “p” , podemos utilizar el siguiente código:



En este caso, cuando el usuario haga clic con el botón derecho del mouse en el elemento “p” , se mostrará un mensaje de alerta “No permitido hacer clic derecho” .

Es importante mencionar que el atributo onauxclick es un evento de mouse y su

soporte puede variar entre los navegadores, es recomendable verificar la compatibilidad antes de usarlo en una aplicación web. Además, es recomendable utilizarlo junto con eventos como `oncontextmenu` para evitar que el menú contextual del navegador aparezca al hacer clic derecho.

5.2.2 onbeforeinput

El atributo “`onbeforeinput`” en HTML es un evento que se activa antes de que el usuario ingrese o modifique el contenido de un elemento de entrada (por ejemplo, un elemento “`input`” o “`textarea`”). Este evento puede ser utilizado para ejecutar una función JavaScript específica justo antes de que el usuario realice cambios en el contenido del elemento.

Por ejemplo, si queremos limitar la longitud de un elemento “`input`” a 10 caracteres, podemos utilizar el siguiente código:

A screenshot of a code editor with a dark background and light green accents. The code is written in a light blue font. It shows an HTML `<input>` tag with attributes `type="text"`, `onbeforeinput="limitar(event)"`, and `maxlength="10">`. Below the tag is a JavaScript `<script>` block containing a function `limitar(event)` that checks if the input value length is greater than or equal to 10. If so, it calls `event.preventDefault()` to prevent the default behavior.

```
<input
  type="text"
  onbeforeinput="limitar(event)"
  maxlength="10">
<script>
  function limitar(event) {
    if (event.target.value.length >= 10) {
      event.preventDefault();
    }
  }
</script>
```

En este caso, antes de que el usuario ingrese o modifique el contenido del elemento “`input`”, se ejecutará la función “`limitar`” y se verificará si el nuevo contenido tiene

5 ATRIBUTOS

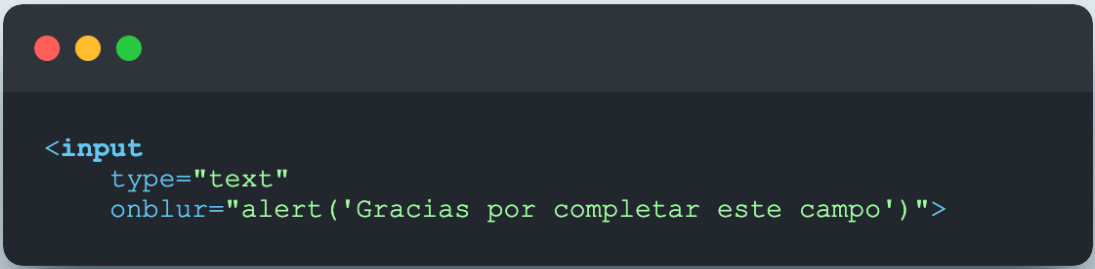
una longitud mayor o igual a 10 caracteres. Si es así, se cancela el evento y no se permite que se realice el cambio.

Es importante mencionar que el atributo `onbeforeinput` es un evento de entrada de datos y su soporte puede variar entre los navegadores, es recomendable verificar la compatibilidad antes de usarlo en una aplicación web.

5.2.3 onblur

El atributo `onblur` en HTML es un evento que se activa cuando un elemento pierde el foco (es decir, cuando el usuario deja de seleccionar o hacer clic en un elemento). Este evento puede ser utilizado para ejecutar una función JavaScript específica cuando el usuario abandona un elemento.

Por ejemplo, si queremos mostrar un mensaje de alerta cuando el usuario abandona un elemento `input`, podemos utilizar el siguiente código:



```
<input
  type="text"
  onblur="alert('Gracias por completar este campo')">
```

En este caso, cuando el usuario abandona el elemento `input`, se mostrará un mensaje de alerta `Gracias por completar este campo`.

Además de mostrar un mensaje de alerta, el evento `onblur` también puede ser utilizado para verificar la validez de los datos ingresados en un formulario, o para ejecutar cualquier otra acción necesaria cuando el usuario abandona un elemento.

Es importante mencionar que el atributo `onblur` es un evento de foco y su soporte puede variar entre los navegadores, es recomendable verificar la compatibilidad

antes de usarlo en una aplicación web.

5.2.4 oncancel

Es importante mencionar que el atributo `oncancel` es un evento de cancelación y su soporte puede variar entre los navegadores. Aunque es un evento soportado por algunos navegadores, su uso no es muy común en aplicaciones web. Es recomendable verificar la compatibilidad antes de usarlo en una aplicación web.

En este caso, cuando se cancela una acción en el elemento `<input>`, se mostrará un mensaje de alerta `"Acción cancelada"`.



Por ejemplo, si queremos mostrar un mensaje de alerta cuando se cancela una acción en un elemento `<input>`, podemos utilizar el siguiente código:

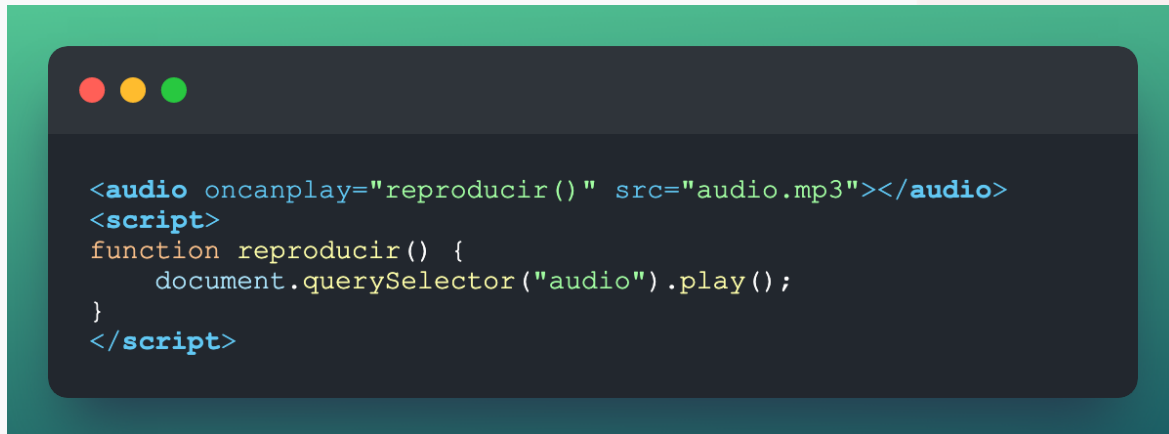
El atributo `oncancel` en HTML es un evento que se activa cuando se cancela una acción que se estaba realizando en un elemento. Este evento puede ser utilizado para ejecutar una función JavaScript específica cuando se cancela una acción.

5.2.5 oncanplay

El atributo `oncanplay` en HTML es un evento el cual se activa cuando un elemento multimedia (como un video o audio) está listo para ser reproducido. Este evento pue-

5 ATRIBUTOS

de ser utilizado para ejecutar una función JavaScript específica cuando el elemento multimedia está listo para ser reproducido.



En este ejemplo, cuando el archivo de audio está listo para ser reproducido, se activa el evento “canplay” y se llama a la función “reproducir()” que reproducirá el audio.

Es importante mencionar que el evento canplay está soportado por la mayoría de los navegadores modernos, pero es recomendable verificar la compatibilidad antes de usarlo en una aplicación web.

5.2.6 oncanplaythrough

El atributo “oncanplaythrough” en HTML es un evento que se activa cuando un elemento multimedia (como un video o audio) ha cargado suficiente contenido para permitir la reproducción sin interrupciones. Este evento puede ser utilizado para ejecutar una función JavaScript específica cuando el elemento multimedia está listo para ser reproducido sin interrupciones.

Por ejemplo, si queremos mostrar un botón “Play” cuando un video esté listo para ser reproducido sin interrupciones, podemos utilizar el siguiente código:

A code editor window with a dark background and three colored window control buttons (red, yellow, green) in the top-left corner. The code is written in a light blue/cyan monospace font. It consists of an HTML `<video>` tag with an `oncanplaythrough` attribute set to `"mostrarBoton()"` and a `src` attribute set to `"video.mp4"`. Below the tag is a `<script>` block containing a JavaScript function `mostrarBoton()` that uses `document.querySelector("#play-button")` to find an element and sets its `style.display` to `"block"`.

```
<video
  oncanplaythrough="mostrarBoton() "
  src="video.mp4"></video>
<script>
function mostrarBoton() {
  let boton = document.querySelector("#play-button");
  boton.style.display = "block";
}
</script>
```

En este ejemplo, cuando el video ha cargado suficiente contenido para permitir la reproducción sin interrupciones, se activa el evento “oncanplaythrough” y se llama a la función “mostrarBoton()” que mostrará el botón “Play” para reproducir el video.

Es importante mencionar que el evento oncanplaythrough está soportado por la mayoría de los navegadores modernos, pero es recomendable verificar la compatibilidad antes de usarlo en una aplicación web.

5.2.7 onchange

El atributo “onchange” en HTML es un evento que se activa cuando el valor de un elemento cambia (por ejemplo, cuando se selecciona una opción diferente en un “select”, cuando se escribe en un “input” o cuando se activa o desactiva un “checkbox” o “radio button”). Este evento puede ser utilizado para ejecutar una función JavaScript específica cuando el valor de un elemento cambia.

Por ejemplo, si queremos mostrar un mensaje de alerta cuando el usuario selecciona una opción diferente en un “select”, podemos utilizar el siguiente código:

```
<select onchange="alert('Selección: ' + this.value)">
  <option value="opcion1">Opción 1</option>
  <option value="opcion2">Opción 2</option>
  <option value="opcion3">Opción 3</option>
</select>
```

En este caso, cuando el usuario selecciona una opción diferente en el “select”, se mostrará un mensaje de alerta con el valor de la opción seleccionada.

Además de mostrar un mensaje de alerta, el evento onchange también puede ser utilizado para verificar la validez de los datos ingresados en un formulario, o para ejecutar cualquier otra acción necesaria cuando el valor de un elemento.

5.2.8 onclick

El atributo “onclick” en HTML es un evento que se activa cuando se hace clic en un elemento. Este evento puede ser utilizado para ejecutar una función JavaScript específica cuando el usuario hace clic en un elemento.

Por ejemplo, si queremos mostrar un mensaje de alerta cuando el usuario hace clic en un botón, podemos utilizar el siguiente código:

```
<button onclick="alert('Clic') ">Presionar</button>
```

5 ATRIBUTOS

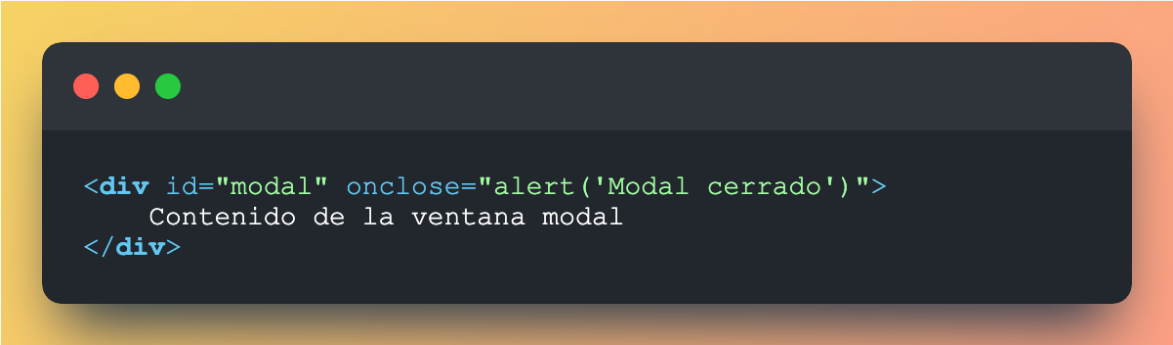
En este caso, cuando el usuario hace clic en el botón “Presionar” , se mostrará un mensaje de alerta “Clic” .

Además de mostrar un mensaje de alerta, el evento onclick también puede ser utilizado para cambiar el contenido de una página, redirigir a otra página, o para ejecutar cualquier otra acción necesaria cuando el usuario hace clic en un elemento. Es uno de los eventos más comunes en HTML y es muy útil para interactuar con el usuario en una página web.

5.2.9 onclose

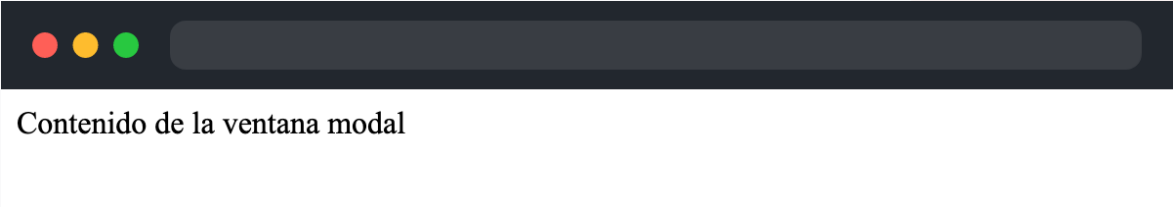
El atributo “onclose” en HTML es un evento que se activa cuando una ventana o un elemento es cerrado. Este evento puede ser utilizado para ejecutar una función JavaScript específica cuando una ventana o un elemento es cerrado.

Por ejemplo, si queremos mostrar un mensaje de alerta cuando un usuario cierra una ventana modal, podemos utilizar el siguiente código:



```
<div id="modal" onclose="alert('Modal cerrado')">
  Contenido de la ventana modal
</div>
```

El código se renderiza de esta manera al abrirlo en un navegador.



Contenido de la ventana modal

5 ATRIBUTOS

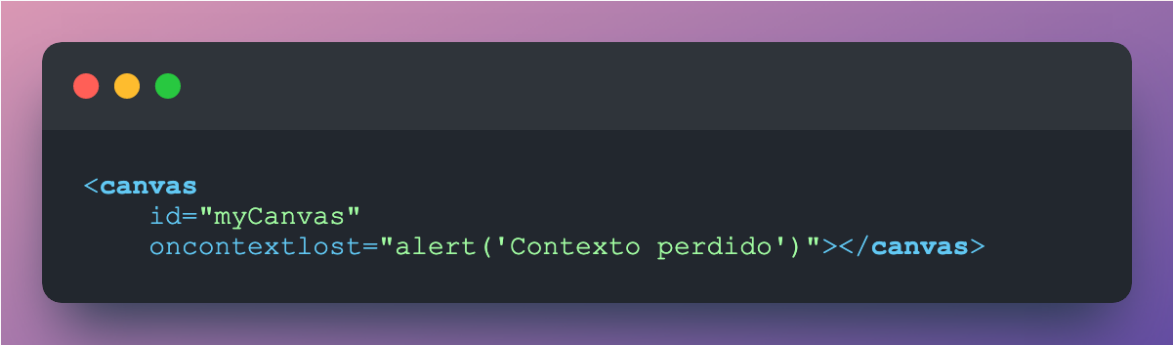
En este caso, cuando el usuario cierra la ventana modal, se mostrará un mensaje de alerta “Modal cerrado” .

Es importante mencionar que el atributo `onclose` está soportado solo en algunos navegadores y su soporte puede variar entre ellos. Aunque es un evento soportado por algunos navegadores, su uso no es muy común en aplicaciones web. Es recomendable verificar la compatibilidad antes de usarlo en una aplicación web y existen otras alternativas de javascript para detectar cuando una ventana o un elemento es cerrado.

5.2.10 oncontextlost

El atributo “`oncontextlost`” en HTML es un evento que se activa cuando el contexto de un canvas es perdido. Este evento puede ser utilizado para ejecutar una función JavaScript específica cuando el contexto del canvas es perdido. El contexto del canvas es el estado en el que se encuentra el canvas y todos los datos necesarios para dibujar en él, como las transformaciones y las propiedades de estilo.

Por ejemplo, si queremos mostrar un mensaje de alerta cuando el contexto de un canvas es perdido, podemos utilizar el siguiente código:



```
<canvas
  id="myCanvas"
  oncontextlost="alert('Contexto perdido')"></canvas>
```

En este caso, cuando el contexto del canvas es perdido, se mostrará un mensaje de alerta “Contexto perdido” .

Es importante mencionar que el evento `oncontextlost` es soportado por la mayoría de

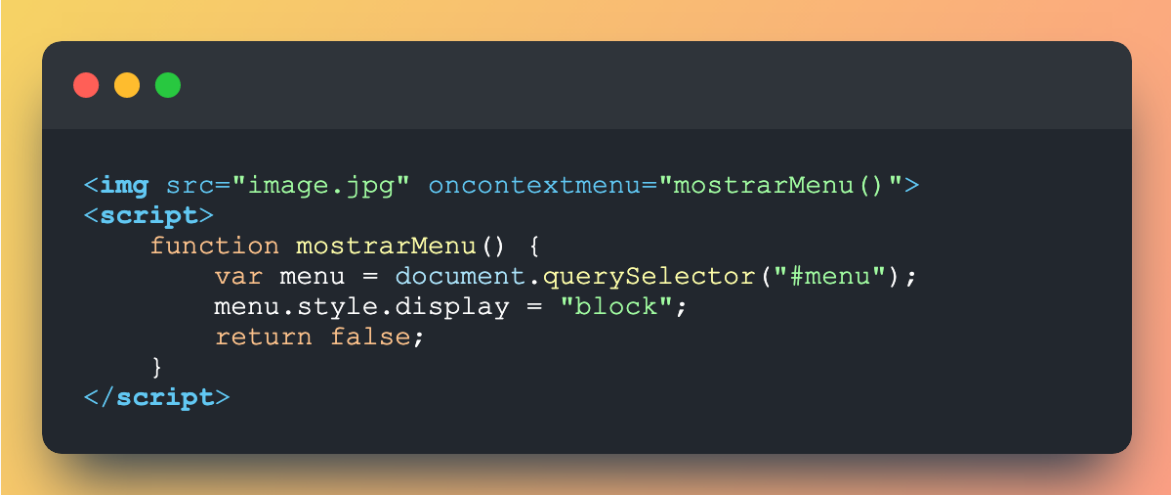
5 ATRIBUTOS

los navegadores modernos, pero su uso es menos común en aplicaciones web y es recomendable verificar la compatibilidad antes de usarlo en una aplicación web. Es importante mencionar que este evento se activa solo cuando el contexto del canvas es perdido de forma permanente, no temporal.

5.2.11 oncontextmenu

El atributo “oncontextmenu” en HTML es un evento que se activa cuando se hace clic derecho en un elemento. Este evento puede ser utilizado para ejecutar una función JavaScript específica cuando el usuario hace clic derecho en un elemento.

Por ejemplo, si queremos mostrar un menú personalizado cuando el usuario hace clic derecho en una imagen, podemos utilizar el siguiente código:



```

<script>
  function mostrarMenu() {
    var menu = document.querySelector("#menu");
    menu.style.display = "block";
    return false;
  }
</script>
```

En este caso, cuando el usuario hace clic derecho en la imagen, se mostrará un menú personalizado.

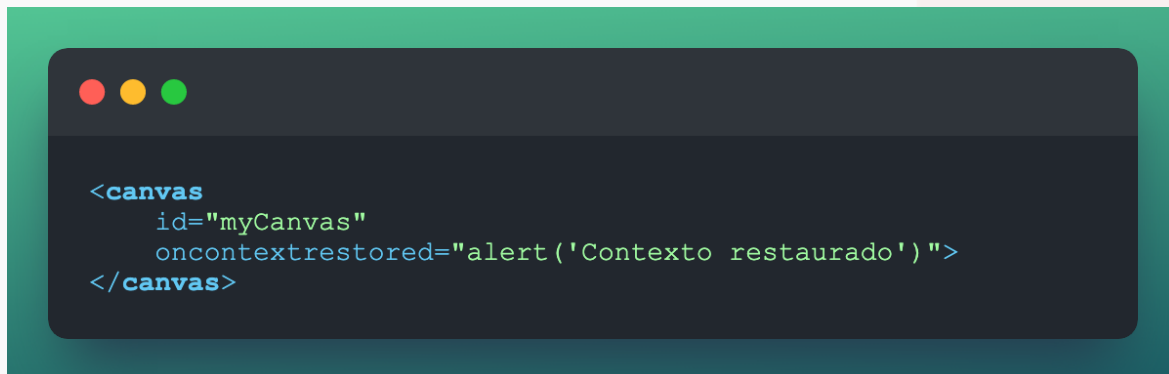
Es importante mencionar que el evento oncontextmenu es soportado por la mayoría de los navegadores modernos, pero su uso es menos común en aplicaciones web. Además, algunos navegadores pueden bloquear el menú contextual por defecto cuando se utiliza este evento. Es recomendable verificar la compatibilidad antes de

usarlo en una aplicación web y utilizarlo de forma responsable para evitar una mala experiencia de usuario.

5.2.12 oncontextrestored

El atributo “oncontextrestored” en HTML es un evento que se activa cuando el contexto de un canvas es restaurado. Este evento puede ser utilizado para ejecutar una función JavaScript específica cuando el contexto del canvas es restaurado. El contexto del canvas es el estado en el que se encuentra el canvas y todos los datos necesarios para dibujar en él, como las transformaciones y las propiedades de estilo.

Por ejemplo, si queremos mostrar un mensaje de alerta cuando el contexto de un canvas es restaurado, podemos utilizar el siguiente código:



En este caso, cuando el contexto del canvas es restaurado, se mostrará un mensaje de alerta “Contexto restaurado” .

Es importante mencionar que el evento oncontextrestored es soportado por la mayoría de los navegadores modernos, pero su uso es menos común en aplicaciones web y es recomendable verificar la compatibilidad antes de usarlo en una aplicación web. Es importante mencionar que este evento se activa solo cuando el contexto del canvas es restaurado después de haber sido perdido temporalmente.

5.2.13 oncopy

El atributo “oncopy” en HTML es un evento que se activa cuando el usuario copia el contenido de un elemento. Este evento puede ser utilizado para ejecutar una función JavaScript específica cuando el usuario copia el contenido de un elemento.

Por ejemplo, si queremos mostrar un mensaje de alerta cuando el usuario copia el contenido de una caja de texto, podemos utilizar el siguiente código:

```
<input type="text" oncopy="alert('Contenido copiado')">
```

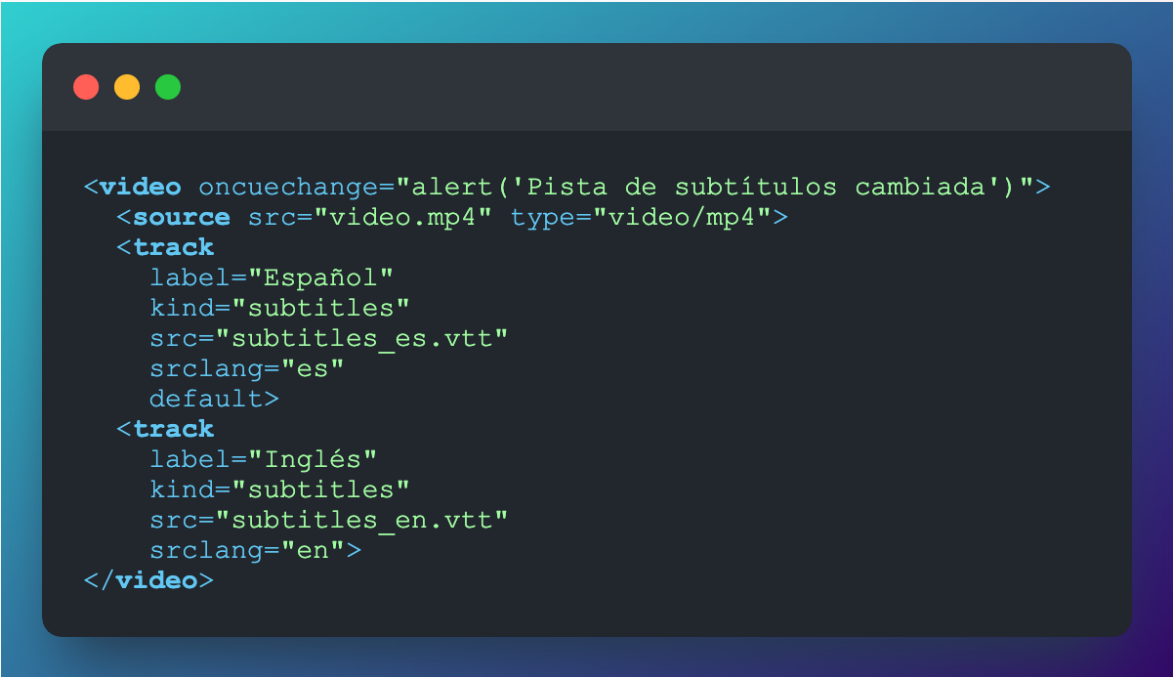
En este caso, cuando el usuario copia el contenido de la caja de texto, se mostrará un mensaje de alerta “Contenido copiado” .

Es importante mencionar que el evento oncopy es soportado por la mayoría de los navegadores modernos, pero su uso es menos común en aplicaciones web. Es recomendable verificar la compatibilidad antes de usarlo en una aplicación web y existen otras alternativas de javascript para detectar cuando se copia contenido.

5.2.14 oncuechange

El atributo “oncuechange” en HTML es un evento que se activa cuando cambia la pista de subtítulos o la descripción de audio en un elemento de audio o video. Este evento puede ser utilizado para ejecutar una función JavaScript específica cuando se cambia la pista de subtítulos o la descripción de audio.

Por ejemplo, si queremos mostrar un mensaje de alerta cuando se cambia la pista de subtítulos en un video, podemos utilizar el siguiente código:



```
<video oncuechange="alert('Pista de subtítulos cambiada')">
  <source src="video.mp4" type="video/mp4">
  <track
    label="Español"
    kind="subtitles"
    src="subtitles_es.vtt"
    srclang="es"
    default>
  <track
    label="Inglés"
    kind="subtitles"
    src="subtitles_en.vtt"
    srclang="en">
</video>
```

En este caso, cuando se cambia la pista de subtítulos en el video, se mostrará un mensaje de alerta “Pista de subtítulos cambiada” .

Es importante mencionar que el evento `oncuechange` es soportado por la mayoría de los navegadores modernos, pero su uso es menos común en aplicaciones web. Es recomendable verificar la compatibilidad antes de usarlo en una aplicación web y existen otras alternativas de javascript para detectar cuando se cambia la pista de subtítulos o la descripción de audio.

5.2.15 oncut

El atributo “oncut” en HTML es un evento que se activa cuando el usuario corta (corta y pega) el contenido de un elemento. Este evento puede ser utilizado para ejecutar una función JavaScript específica cuando el usuario corta el contenido de un elemento.

5 ATRIBUTOS

Por ejemplo, si queremos mostrar un mensaje de alerta cuando el usuario corta el contenido de una caja de texto, podemos utilizar el siguiente código:

```
<input type="text" oncut="alert('Contenido cortado')">
```

En este caso, cuando el usuario corta el contenido de la caja de texto, se mostrará un mensaje de alerta “Contenido cortado” .

Es importante mencionar que el evento `oncut` es soportado por la mayoría de los navegadores modernos, pero su uso es menos común en aplicaciones web. Es recomendable verificar la compatibilidad antes de usarlo en una aplicación web y existen otras alternativas de javascript para detectar cuando se corta contenido.

5.2.16 ondblclick

El atributo “`ondblclick`” en HTML es un evento que se activa cuando se hace doble clic en un elemento. Este evento puede ser utilizado para ejecutar una función JavaScript específica cuando el usuario hace doble clic en un elemento.

Por ejemplo, si queremos mostrar un mensaje de alerta cuando el usuario hace doble clic en un botón, podemos utilizar el siguiente código:



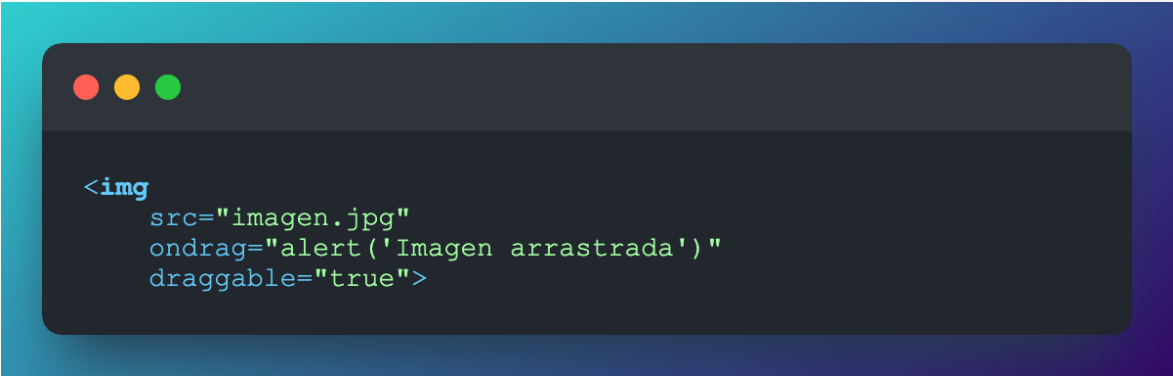
En este caso, cuando el usuario hace doble clic en el botón, se mostrará un mensaje de alerta “Doble clic realizado” .

Es importante mencionar que el evento `ondblclick` es soportado por la mayoría de los navegadores modernos, pero su uso es menos común en aplicaciones web, debido a que es más común utilizar eventos de clic simple. Es recomendable verificar la compatibilidad antes de usarlo en una aplicación web.

5.2.17 ondrag

El atributo “`ondrag`” en HTML es un evento que se activa cuando un elemento es arrastrado. Este evento puede ser utilizado para ejecutar una función JavaScript específica cuando un elemento es arrastrado. El evento `ondrag` se activa en el elemento objetivo, es decir, el elemento que es arrastrado, mientras que el evento `ondragstart` se activa en el elemento origen, es decir, el elemento que inicia el arrastre.

Por ejemplo, si queremos mostrar un mensaje de alerta cuando el usuario arrastra una imagen, podemos utilizar el siguiente código:



```

```

En este caso, cuando el usuario arrastra la imagen, se mostrará un mensaje de alerta “Imagen arrastrada” .

Es importante mencionar que el evento `ondrag` es soportado por la mayoría de los navegadores modernos, pero su uso es menos común en aplicaciones web. Es recomendable verificar la compatibilidad antes de usarlo en una aplicación web y existen otras alternativas de javascript para detectar cuando se arrastra un elemento.

5.2.18 ondragend

El atributo “`ondragend`” en HTML es un evento que se activa al finalizar una operación de arrastre. Este evento puede ser utilizado para ejecutar una función JavaScript específica cuando se termina de arrastrar un elemento. El evento `ondragend` se activa en el elemento objetivo, es decir, el elemento que es arrastrado, y se activa cuando el usuario suelta el elemento.

Por ejemplo, si queremos mostrar un mensaje de alerta cuando el usuario suelta una imagen después de arrastrarla, podemos utilizar el siguiente código:



```

```

En este caso, cuando el usuario suelta la imagen después de arrastrarla, se mostrará un mensaje de alerta “Imagen soltada” .

Es importante mencionar que el evento `ondragend` es soportado por la mayoría de los navegadores modernos, pero su uso es menos común en aplicaciones web. Es recomendable verificar la compatibilidad antes de usarlo en una aplicación web y existen otras alternativas de javascript para detectar cuando se suelta un elemento después de arrastrarlo.

5.2.19 ondragenter

El atributo “`ondragenter`” en HTML es un evento que se activa cuando un elemento es arrastrado y entra en el área de otro elemento. Este evento puede ser utilizado para ejecutar una función JavaScript específica cuando un elemento es arrastrado y entra en el área de otro elemento. El evento `ondragenter` se activa en el elemento destino, es decir, el elemento que es el receptor del arrastre.

Por ejemplo, si queremos mostrar un mensaje de alerta cuando el usuario arrastra una imagen hacia una caja de texto, podemos utilizar el siguiente código:

```

<input
  type="text"
  ondragenter="alert('Arrastrada hacia la caja')">
```

En este caso, cuando el usuario arrastra la imagen hacia la caja de texto, se mostrará un mensaje de alerta “Imagen arrastrada hacia la caja de texto” .

Es importante mencionar que el evento `ondragenter` es soportado por la mayoría de los navegadores modernos, pero su uso es menos común en aplicaciones web. Es recomendable verificar la compatibilidad antes de usarlo en una aplicación web y existen otras alternativas de javascript para detectar cuando un elemento es arrastrado hacia otro elemento.

5.2.20 `ondragleave`

El atributo “`ondragleave`” en HTML es un evento que se activa cuando un elemento es arrastrado y sale del área de otro elemento. Este evento puede ser utilizado para ejecutar una función JavaScript específica cuando un elemento es arrastrado y sale del área de otro elemento. El evento `ondragleave` se activa en el elemento destino, es decir, el elemento que es el receptor del arrastre.

Por ejemplo, si queremos mostrar un mensaje de alerta cuando el usuario arrastra una imagen fuera de una caja de texto, podemos utilizar el siguiente código:


```

<input
  type="text"
  ondragleave="alert('Arrastrada fuera de la caja')">
```

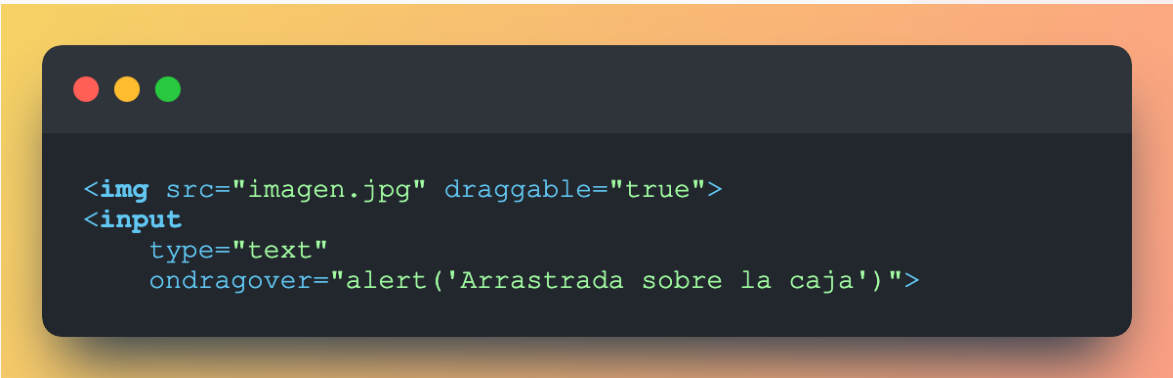
En este caso, cuando el usuario arrastra la imagen fuera de la caja de texto, se mostrará un mensaje de alerta “Imagen arrastrada fuera de la caja de texto” .

Es importante mencionar que el evento `ondragleave` es soportado por la mayoría de los navegadores modernos, pero su uso es menos común en aplicaciones web. Es recomendable verificar la compatibilidad antes de usarlo en una aplicación web y existen otras alternativas de javascript para detectar cuando un elemento es arrastrado fuera de otro elemento.

5.2.21 ondragover

El atributo “`ondragover`” en HTML es un evento que se activa cuando un elemento es arrastrado sobre otro elemento. Este evento puede ser utilizado para ejecutar una función JavaScript específica cuando un elemento es arrastrado sobre otro elemento. El evento `ondragover` se activa en el elemento destino, es decir, el elemento que es el receptor del arrastre. Es importante mencionar que por defecto el navegador no permite soltar un elemento en otro elemento si no se utiliza este evento, es necesario incluirlo para poder soltar el elemento arrastrado en otro elemento.

Por ejemplo, si queremos mostrar un mensaje de alerta cuando el usuario arrastra una imagen sobre una caja de texto, podemos utilizar el siguiente código:



```

<input
  type="text"
  ondragover="alert('Arrastrada sobre la caja')">
```

En este caso, cuando el usuario arrastra la imagen sobre la caja de texto, se mostrará un mensaje de alerta “Imagen arrastrada sobre la caja de texto” .

Es importante mencionar que el evento `ondragover` es soportado por la mayoría de los navegadores modernos, pero su uso es menos común en aplicaciones web. Es recomendable verificar la compatibilidad antes de usarlo en una aplicación web y existen otras alternativas de javascript para detectar cuando un elemento es arrastrado sobre otro elemento.

5.2.22 ondragstart

El atributo “`ondragstart`” en HTML es un evento que se activa al comenzar una operación de arrastre. Este evento puede ser utilizado para ejecutar una función JavaScript específica cuando se comienza a arrastrar un elemento. El evento `ondragstart` se activa en el elemento origen, es decir, el elemento que es arrastrado.

Por ejemplo, si queremos mostrar un mensaje de alerta cuando el usuario comienza a arrastrar una imagen, podemos utilizar el siguiente código:

```

```

En este caso, cuando el usuario comienza a arrastrar la imagen, se mostrará un mensaje de alerta “Comienza a arrastrar la imagen” .

5.2.23 ondrop

El atributo “ondrop” en HTML es un evento que se activa cuando un elemento es soltado en otro elemento después de haber sido arrastrado. Este evento puede ser utilizado para ejecutar una función JavaScript específica cuando se suelta un elemento en otro elemento. El evento ondrop se activa en el elemento destino, es decir, el elemento en el cual se suelta el elemento arrastrado.

Por ejemplo, si queremos mostrar un mensaje de alerta cuando el usuario suelta una imagen en una caja de texto, podemos utilizar el siguiente código:

```

<input
  type="text"
  ondrop="alert('Imagen soltada en la caja de texto')">
```

5 ATRIBUTOS

En este caso, cuando el usuario suelta la imagen en la caja de texto, se mostrará un mensaje de alerta “Imagen soltada en la caja de texto” .

Es importante mencionar que el evento `ondrop` es soportado por la mayoría de los navegadores modernos, pero su uso es menos común en aplicaciones web. Es recomendable verificar la compatibilidad antes de usarlo en una aplicación web y existen otras alternativas de javascript para detectar cuando un elemento es soltado en otro elemento. Además es necesario incluir el evento `ondragover` para poder soltar un elemento en otro elemento.

5.2.24 `ondurationchange`

El atributo “`ondurationchange`” en HTML es un evento que se activa cuando la duración de un medio (como un video o audio) cambia. Este evento puede ser utilizado para ejecutar una función JavaScript específica cuando la duración de un medio cambia. El evento `ondurationchange` se activa en el elemento del medio, es decir, el elemento que representa el video o audio.

Por ejemplo, si queremos mostrar un mensaje de alerta cuando la duración de un video cambia, podemos utilizar el siguiente código:



```
<video
  ondurationchange="alert('La duración ha cambiado')">
  <source src="video.mp4" type="video/mp4">
</video>
```

En este caso, cuando la duración del video cambia, se mostrará un mensaje de alerta “La duración del video ha cambiado” .

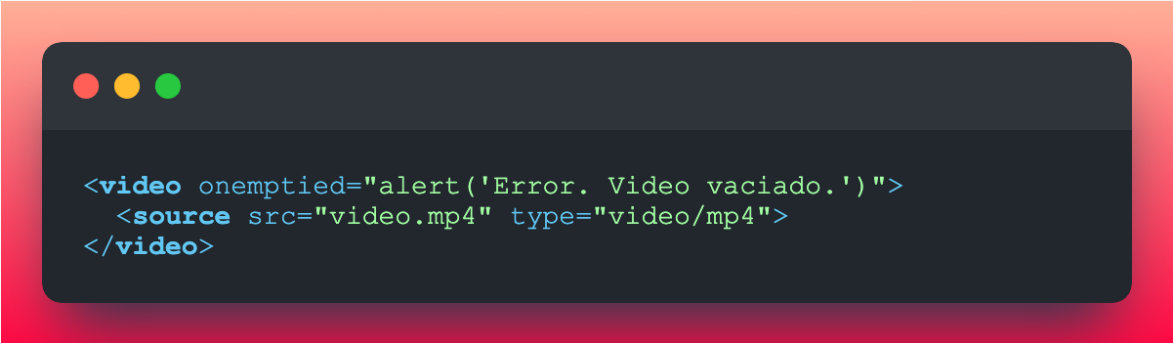
5 ATRIBUTOS

Es importante mencionar que el evento `ondurationchange` es soportado por la mayoría de los navegadores modernos, pero su uso es menos común en aplicaciones web. Es recomendable verificar la compatibilidad antes de usarlo en una aplicación web.

5.2.25 `onemptied`

El atributo `onemptied` en HTML es un evento que se activa cuando un medio (como un video o audio) es vaciado debido a un error. Este evento puede ser utilizado para ejecutar una función JavaScript específica cuando un medio es vaciado debido a un error. El evento `onemptied` se activa en el elemento del medio, es decir, el elemento que representa el video o audio.

Por ejemplo, si queremos mostrar un mensaje de alerta cuando un video es vaciado debido a un error, podemos utilizar el siguiente código:



```
<video onemptied="alert('Error. Video vaciado.')">
  <source src="video.mp4" type="video/mp4">
</video>
```

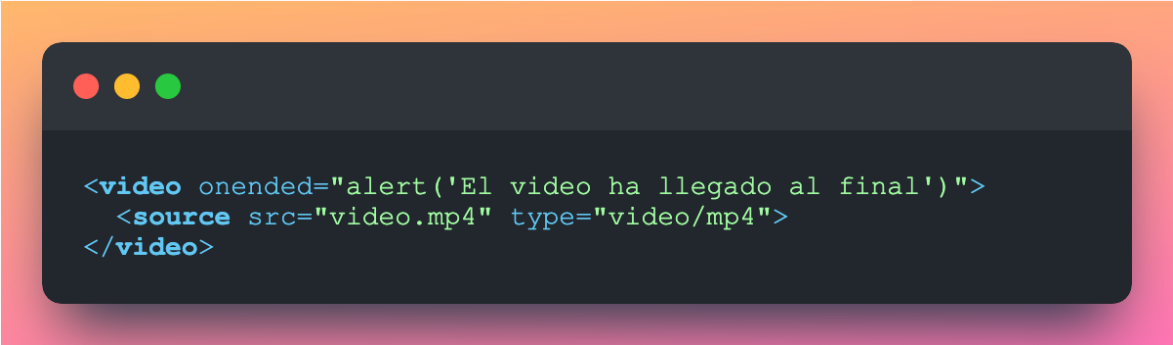
En este caso, cuando el video es vaciado debido a un error, se mostrará un mensaje de alerta “El video ha sido vaciado debido a un error” .

Es importante mencionar que el evento `onemptied` es soportado por la mayoría de los navegadores modernos, pero su uso es menos común en aplicaciones web. Es recomendable verificar la compatibilidad antes de usarlo en una aplicación web.

5.2.26 onended

El atributo “onended” en HTML es un evento que se activa cuando un medio (como un video o audio) llega al final de su reproducción. Este evento puede ser utilizado para ejecutar una función JavaScript específica cuando un medio llega al final de su reproducción. El evento onended se activa en el elemento del medio, es decir, el elemento que representa el video o audio.

Por ejemplo, si queremos mostrar un mensaje de alerta cuando un video llega al final de su reproducción, podemos utilizar el siguiente código:



```
<video onended="alert('El video ha llegado al final')">
  <source src="video.mp4" type="video/mp4">
</video>
```

En este caso, cuando el video llega al final de su reproducción, se mostrará un mensaje de alerta “El video ha llegado al final” .

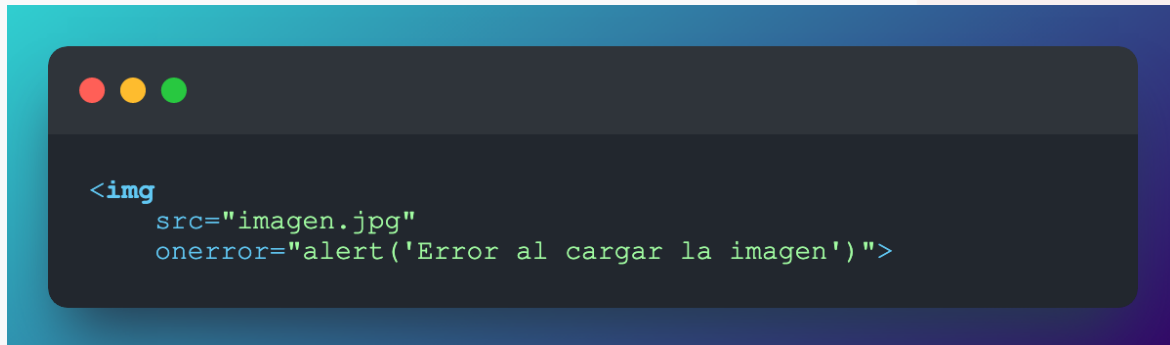
Es importante mencionar que el evento onended es soportado por la mayoría de los navegadores modernos, pero su uso es menos común en aplicaciones web. Es recomendable verificar la compatibilidad antes de usarlo en una aplicación web.

5.2.27 onerror

El atributo “onerror” en HTML es un evento que se activa cuando ocurre un error en un elemento. Este evento puede ser utilizado para ejecutar una función JavaScript específica cuando ocurre un error en un elemento. El evento onerror se activa en el elemento que tiene el error.

5 ATRIBUTOS

Por ejemplo, si queremos mostrar un mensaje de alerta cuando ocurre un error al cargar una imagen, podemos utilizar el siguiente código:



En este caso, cuando ocurre un error al cargar la imagen, se mostrará un mensaje de alerta “Error al cargar la imagen” .

Es importante mencionar que el evento onerror es soportado por la mayoría de los navegadores modernos, y su uso es común en aplicaciones web para manejar errores. Es recomendable verificar la compatibilidad antes de usarlo en una aplicación web.

5.2.28 onfocus

El atributo “onfocus” en HTML es un evento que se activa cuando un elemento gana el foco. El foco es la capacidad de un elemento para recibir entrada del usuario, como escribir en un campo de texto o seleccionar un botón. Este evento puede ser utilizado para ejecutar una función JavaScript específica cuando un elemento gana el foco. El evento onfocus se activa en el elemento que gana el foco.

Por ejemplo, si queremos mostrar un mensaje de alerta cuando un campo de texto gana el foco, podemos utilizar el siguiente código:

```
<input
  type="text"
  onfocus="alert('Campo de texto enfocado')">
```

En este caso, cuando el campo de texto gana el foco, se mostrará un mensaje de alerta “El campo de texto ha ganado el foco” .

5.2.29 onformdata

El atributo “onformdata” en HTML es un evento que se activa cuando un objeto FormData es creado. El objeto FormData es una interfaz JavaScript que permite construir una colección de pares clave-valor para enviar un formulario mediante una petición HTTP. Este evento puede ser utilizado para ejecutar una función JavaScript específica cuando se construye un objeto FormData. El evento onformdata se activa en el elemento que tiene el formulario.

Por ejemplo, si queremos mostrar un mensaje de alerta cuando se construye un objeto FormData, podemos utilizar el siguiente código:

```
<form onformdata="alert('Objeto FormData creado')">
  <input type="text" name="nombre">
  <input type="submit" value="Enviar">
</form>
```


5 ATRIBUTOS

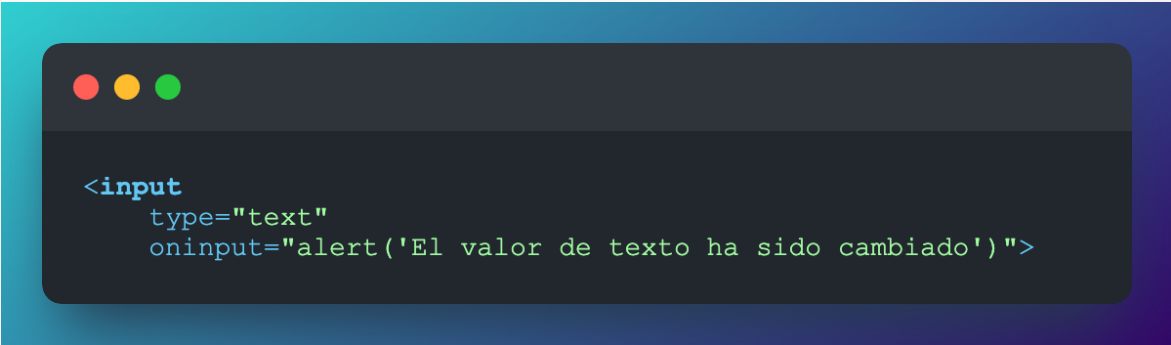
En este caso, cuando se construye un objeto FormData al hacer submit del formulario, se mostrará un mensaje de alerta “El objeto FormData ha sido creado” .

Es importante mencionar que el evento onformdata es soportado por algunos navegadores modernos, pero su uso es menos común en aplicaciones web. Es recomendable verificar la compatibilidad antes de usarlo en una aplicación web.

5.2.30 oninput

El atributo “oninput” en HTML es un evento que se activa cuando el valor de un elemento es cambiado. Este evento puede ser utilizado para ejecutar una función JavaScript específica cuando el valor de un elemento es cambiado. El evento oninput se activa en el elemento cuyo valor es cambiado.

Por ejemplo, si queremos mostrar un mensaje de alerta cuando el valor de un campo de texto es cambiado, podemos utilizar el siguiente código:



```
<input
  type="text"
  oninput="alert('El valor de texto ha sido cambiado')">
```

En este caso, cuando el valor del campo de texto es cambiado, se mostrará un mensaje de alerta “El valor del campo de texto ha sido cambiado” .

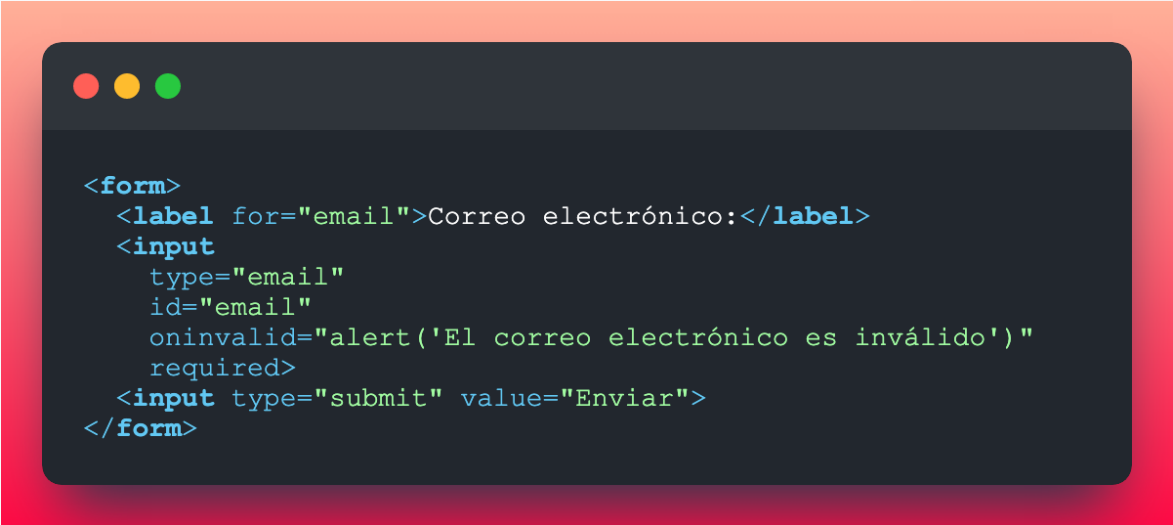
5.2.31 oninvalid

El atributo “oninvalid” en HTML es un evento que se activa cuando un elemento no cumple con las reglas de validación establecidas. La validación es el proceso de

5 ATRIBUTOS

comprobar que los datos introducidos en un formulario son válidos antes de enviarlo. Este evento puede ser utilizado para ejecutar una función JavaScript específica cuando un elemento no cumple con las reglas de validación. El evento `oninvalid` se activa en el elemento que no cumple con las reglas de validación.

Por ejemplo, si queremos mostrar un mensaje de alerta cuando un campo de correo electrónico no cumple con las reglas de validación, podemos utilizar el siguiente código:



```
<form>
  <label for="email">Correo electrónico:</label>
  <input
    type="email"
    id="email"
    oninvalid="alert('El correo electrónico es inválido')"
    required>
  <input type="submit" value="Enviar">
</form>
```

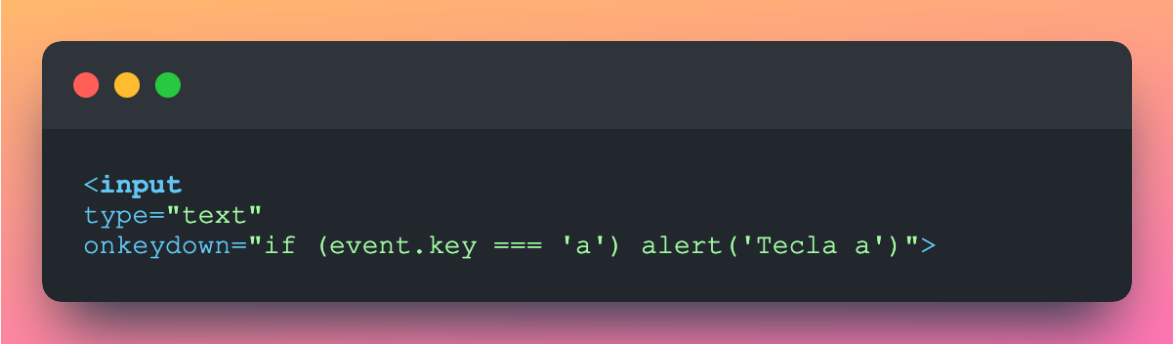
En este caso, cuando se intenta enviar el formulario y el campo de correo electrónico no cumple con las reglas de validación (no es un correo electrónico válido), se mostrará un mensaje de alerta “El correo electrónico no es válido” .

5.2.32 onkeydown

El atributo “onkeydown” en HTML es un evento que se activa cuando se presiona una tecla en el teclado. Este evento puede ser utilizado para ejecutar una función JavaScript específica cuando se presiona una tecla en el teclado. El evento `onkeydown` se activa en el elemento en el que se está presionando la tecla.

5 ATRIBUTOS

Por ejemplo, si queremos mostrar un mensaje de alerta cuando se presiona la tecla “a” en un campo de texto, podemos utilizar el siguiente código:

A code editor window with a dark background and a light blue border. It contains the following HTML code:

```
<input  
type="text"  
onkeydown="if (event.key === 'a') alert('Tecla a')">
```

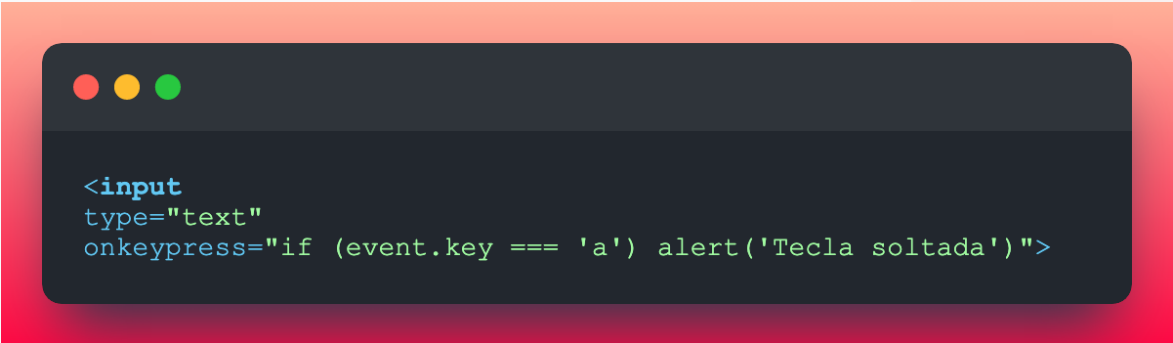
```
<input  
type="text"  
onkeydown="if (event.key === 'a') alert('Tecla a')">
```

En este caso, cuando se presiona la tecla “a” en el campo de texto, se mostrará un mensaje de alerta “Tecla a” .

5.2.33 onkeypress

El atributo “onkeypress” en HTML es un evento que se activa cuando se presiona una tecla en el teclado y se suelta. Este evento puede ser utilizado para ejecutar una función JavaScript específica cuando se presiona una tecla en el teclado y se suelta. El evento onkeypress se activa en el elemento en el que se está presionando la tecla.

Por ejemplo, si queremos mostrar un mensaje de alerta cuando se presiona y suelta la tecla “a” en un campo de texto, podemos utilizar el siguiente código:

A code editor window with a dark background and a light blue border. It contains the following HTML code:

```
<input  
type="text"  
onkeypress="if (event.key === 'a') alert('Tecla soltada')">
```

```
<input  
type="text"  
onkeypress="if (event.key === 'a') alert('Tecla soltada')">
```

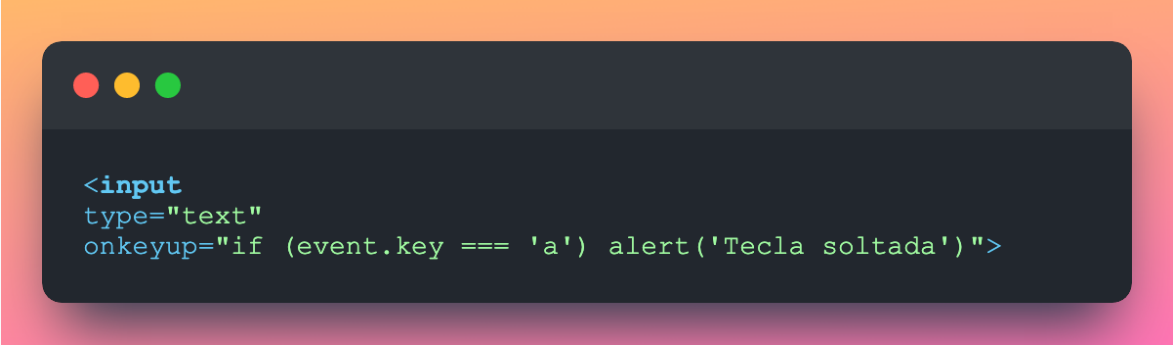
5 ATRIBUTOS

En este caso, cuando se presiona y suelta la tecla “a” en el campo de texto, se mostrará un mensaje de alerta “Tecla soltada” .

5.2.34 onkeyup

El atributo “onkeyup” en HTML es un evento que se activa cuando se suelta una tecla en el teclado. Este evento puede ser utilizado para ejecutar una función JavaScript específica cuando se suelta una tecla en el teclado. El evento onkeyup se activa en el elemento en el que se está soltando la tecla.

Por ejemplo, si queremos mostrar un mensaje de alerta cuando se suelta la tecla “a” en un campo de texto, podemos utilizar el siguiente código:



```
<input  
type="text"  
onkeyup="if (event.key === 'a') alert('Tecla soltada')">
```

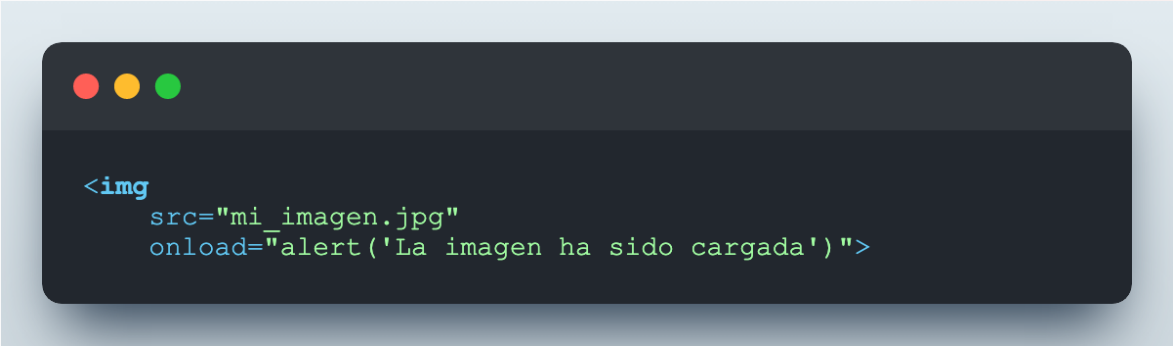
En este caso, cuando se suelta la tecla “a” en el campo de texto, se mostrará un mensaje de alerta “La tecla a ha sido soltada” .

5.2.35 onload

El atributo “onload” en HTML es un evento que se activa cuando se ha terminado de cargar un elemento, como una imagen, un iframe, un script, etc. Este evento puede ser utilizado para ejecutar una función JavaScript específica una vez que un elemento ha sido cargado. El evento onload se activa en el elemento que ha sido cargado.

5 ATRIBUTOS

Por ejemplo, si queremos mostrar un mensaje de alerta cuando se ha terminado de cargar una imagen, podemos utilizar el siguiente código:



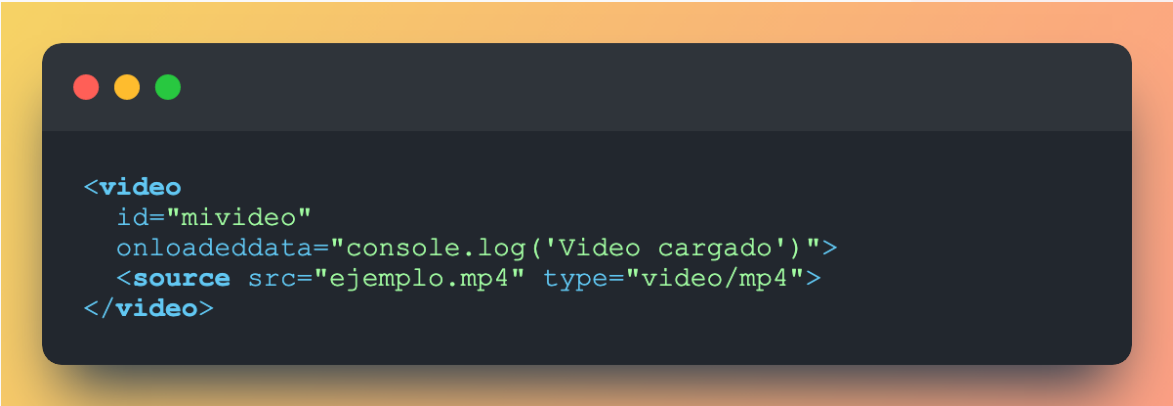
```

```

En este caso, cuando la imagen ha sido cargada, se mostrará un mensaje de alerta “La imagen ha sido cargada” .

5.2.36 onloadeddata

El atributo **onloadeddata** en HTML se utiliza para especificar una función JavaScript que se ejecutará cuando se complete la carga de los datos de un elemento multimedia (como un video o audio). Por ejemplo, si queremos mostrar un mensaje en la consola cuando se cargue un video en una página web, podríamos utilizar el siguiente código:



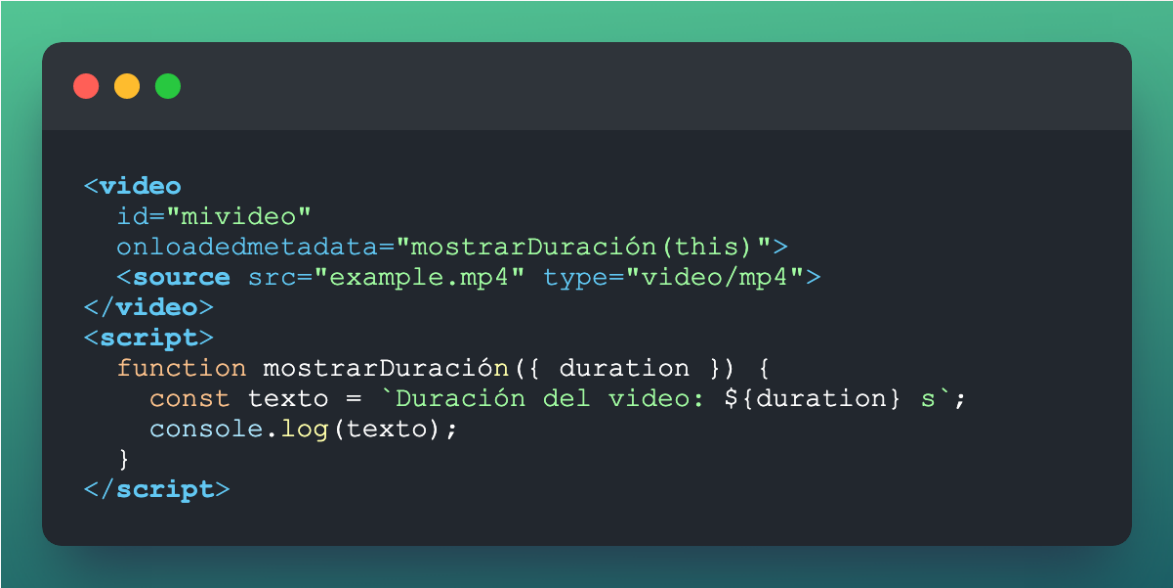
```
<video  
  id="mivideo"  
  onloadeddata="console.log('Video cargado')">  
  <source src="ejemplo.mp4" type="video/mp4">  
</video>
```

5 ATRIBUTOS

En este ejemplo, cuando el video con id “myvideo” termine de cargar los datos, se ejecutará la función **console.log('Video cargado')**, lo que mostrará el mensaje “Video cargado” en la consola del navegador.

5.2.37 onloadedmetadata

El atributo **onloadedmetadata** en HTML se utiliza para especificar una función JavaScript que se ejecutará cuando se hayan cargado los metadatos de un elemento multimedia (como un video o audio). Los metadatos incluyen información como la duración, las dimensiones y el formato del archivo multimedia. Por ejemplo, si queremos mostrar la duración del video en una página web, podríamos utilizar el siguiente código:

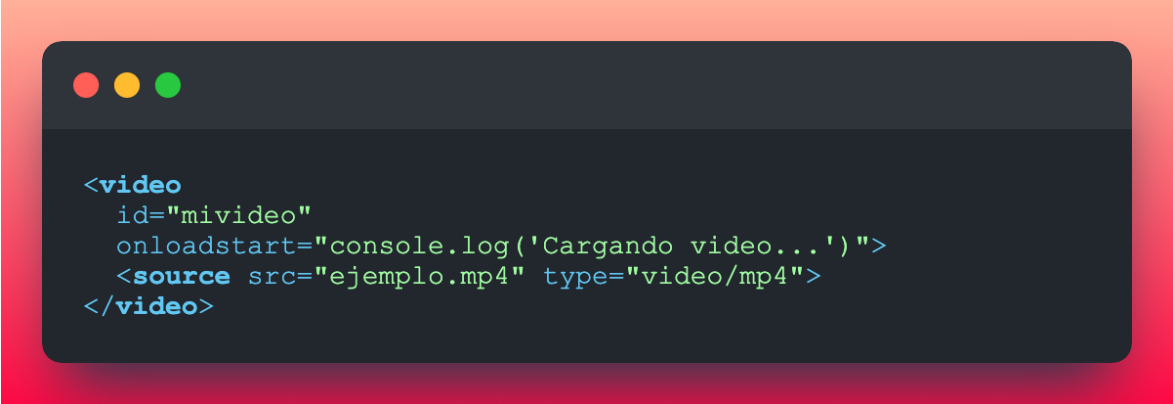


```
<video
  id="mivideo"
  onloadedmetadata="mostrarDuración(this)">
  <source src="example.mp4" type="video/mp4">
</video>
<script>
  function mostrarDuración({ duration }) {
    const texto = `Duración del video: ${duration} s`;
    console.log(texto);
  }
</script>
```

En este ejemplo, cuando los metadatos del video con id “myvideo” se hayan cargado, se ejecutará la función **mostrarDuración(this)**, que mostrará en la consola del navegador “Duración del video: x s” donde x es la duración en segundos del video.

5.2.38 onloadstart

El atributo **onloadstart** en HTML se utiliza para especificar una función JavaScript que se ejecutará al iniciar la carga de un elemento multimedia (como un video o audio). Esto sucede justo después de que se haya establecido la fuente del archivo multimedia y antes de que se cargue ningún dato. Por ejemplo, si queremos mostrar un mensaje en la consola cuando comience a cargar un video en una página web, podríamos utilizar el siguiente código:



```
<video
  id="mivideo"
  onloadstart="console.log('Cargando video...')">
  <source src="ejemplo.mp4" type="video/mp4">
</video>
```

En este ejemplo, cuando comience a cargar el video con id “myvideo”, se ejecutará la función **console.log('Cargando video...')**, lo que mostrará el mensaje “Cargando video...” en la consola del navegador.

También puede ser utilizado para mostrar una animación de carga o un spinner mientras el video se está cargando.



```
<video id="mivideo" onloadstart="animar()">
  <source src="ejemplo.mp4" type="video/mp4">
</video>
<script>
function animar() {
  var rueda = document.getElementById("spinner");
  rueda.style.display = "block";
}
</script>
```

En este ejemplo, se ejecutará la función **showSpinner()** cuando comience a cargar el video, y esta función mostrará el elemento con id “rueda” en la página (ejemplo una rueda o una animación de carga) para indicar que el video se está cargando.

5.2.39 onmousedown

El atributo **onmousedown** en HTML se utiliza para especificar una función JavaScript que se ejecutará cuando se presione (o “se haga clic”) un botón del mouse en un elemento específico de la página. Este evento se activa justo antes de que el evento **onmouseup** (que se activa cuando se suelta el botón del mouse) y **onclick** (que se activa cuando se hace clic en el elemento).

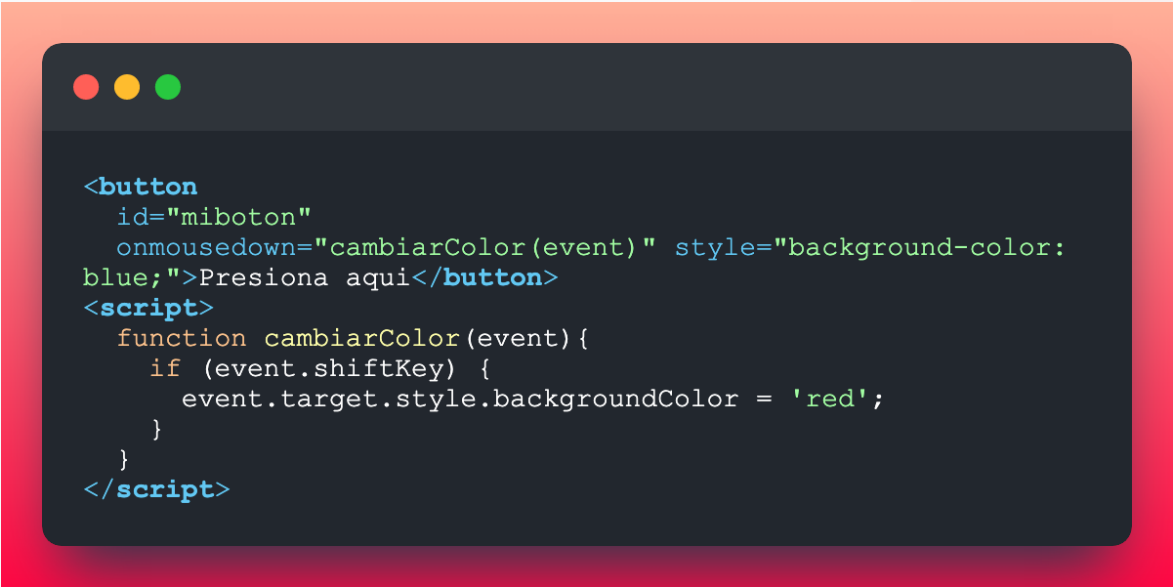
Por ejemplo, si queremos cambiar el color de fondo de un botón cuando se presione, podríamos utilizar el siguiente código:



```
<button
  id="miboton"
  onmousedown="cambiarColor(this)"
  style="background-color: blue;">Presiona aqui</button>
<script>
  function cambiarColor(elemento){
    elemento.style.backgroundColor = 'red';
  }
</script>
```

En este ejemplo, cuando se presione el botón con id “miboton” , se ejecutará la función **cambiarColor(this)**, lo que cambiará el color de fondo del botón a rojo.

También se pueden usar eventos del mouse combinado con eventos del teclado para crear una interacción más compleja, por ejemplo si queremos cambiar el color de fondo de un botón cuando se presione con el mouse y al mismo tiempo se presione la tecla “shift” se podría utilizar el siguiente código:



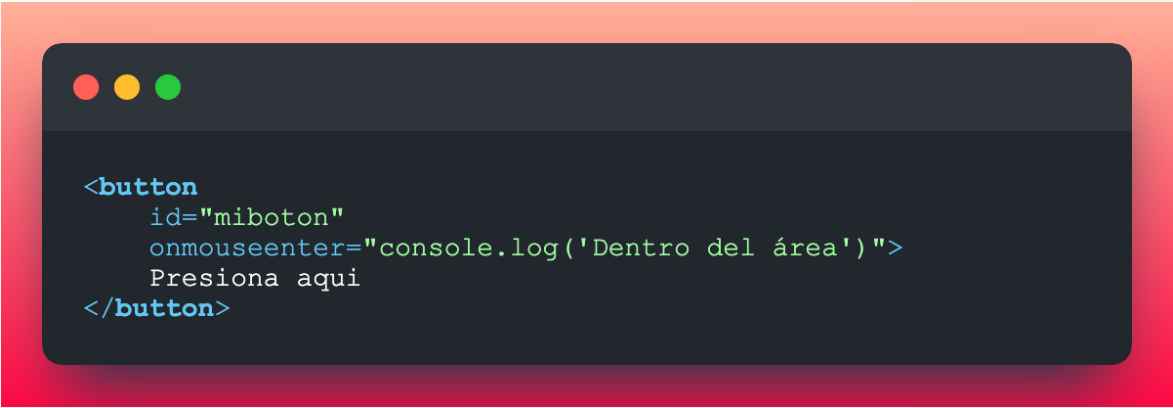
```
<button
  id="miboton"
  onmousedown="cambiarColor(event)" style="background-color:
blue;">Presiona aqui</button>
<script>
  function cambiarColor(event) {
    if (event.shiftKey) {
      event.target.style.backgroundColor = 'red';
    }
  }
</script>
```

En este ejemplo, cuando se presione el botón con id “miboton” y se presione la tecla “shift” al mismo tiempo, se ejecutará la función **cambiarColor(event)**, lo que cambiará el color de fondo del botón a rojo.

5.2.40 onmouseenter

El atributo **onmouseenter** en HTML se utiliza para especificar una función JavaScript que se ejecutará cuando el cursor del mouse entre en el área de un elemento específico de la página. Este evento es similar al evento **onmouseover**, pero solo se activa cuando el cursor entra en el área del elemento, mientras que **onmouseover** se activa cada vez que el cursor pasa por encima del elemento, incluso si el cursor ya está dentro del elemento.

Por ejemplo, si queremos mostrar un mensaje en la consola cuando el cursor del mouse entre en el área de un botón, podríamos utilizar el siguiente código:



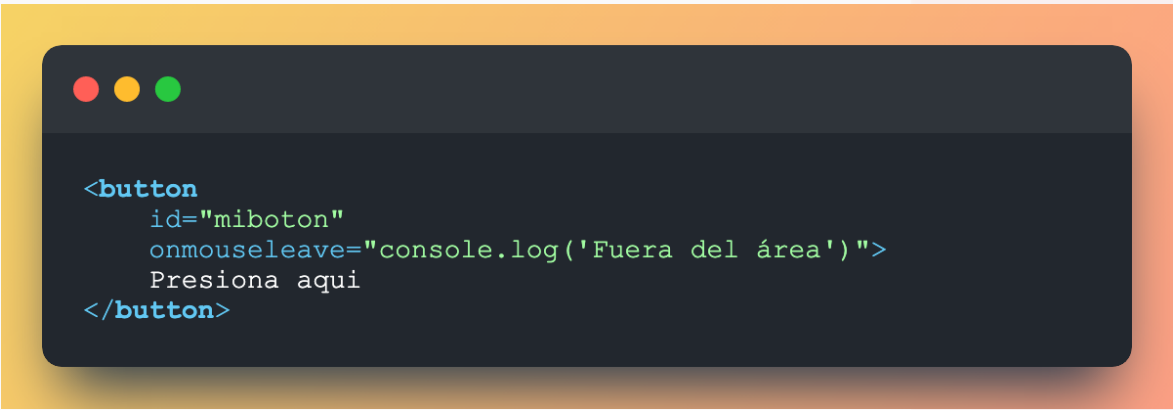
```
<button
  id="miboton"
  onmouseenter="console.log('Dentro del área')">
  Presiona aqui
</button>
```

En este ejemplo, cuando el cursor entre en el área del botón con id “miboton” , se ejecutará la función **console.log('Dentro del área')**, lo que mostrará el mensaje “Dentro del área” en la consola del navegador.

5.2.41 onmouseleave

El atributo **onmouseleave** en HTML se utiliza para especificar una función JavaScript que se ejecutará cuando el cursor del mouse salga del área de un elemento específico de la página. Este evento es similar al evento **onmouseout**, pero solo se activa cuando el cursor sale del área del elemento, mientras que **onmouseout** se activa cada vez que el cursor sale de cualquier elemento hijo del elemento, incluso si el cursor todavía está dentro del área del elemento padre.

Por ejemplo, si queremos mostrar un mensaje en la consola cuando el cursor del mouse salga del área de un botón, podríamos utilizar el siguiente código:



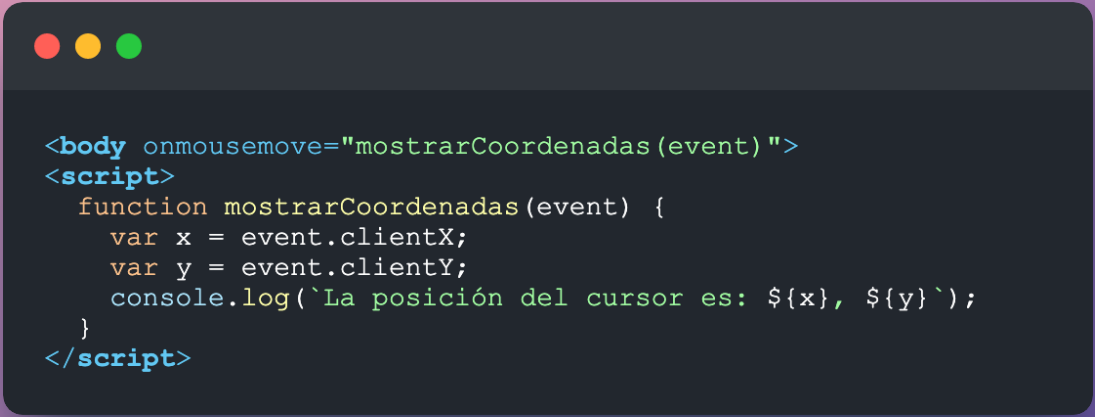
```
<button
  id="miboton"
  onmouseleave="console.log('Fuera del área')">
  Presiona aqui
</button>
```

En este ejemplo, cuando el cursor salga del área del botón con id “miboton”, se ejecutará la función **console.log('Fuera del área')**, lo que mostrará el mensaje “Fuera del área” en la consola del navegador.

5.2.42 onmousemove

El atributo **onmousemove** en HTML se utiliza para especificar una función JavaScript que se ejecutará cuando el cursor del mouse se mueva dentro del área de un elemento específico de la página. Este evento se activa cada vez que el cursor se mueve dentro del área del elemento, lo que permite realizar acciones en tiempo real en función de la posición del cursor.

Por ejemplo, si queremos mostrar la posición actual del cursor en una página web, podríamos utilizar el siguiente código:



```
<body onmousemove="mostrarCoordenadas(event)">
<script>
  function mostrarCoordenadas(event) {
    var x = event.clientX;
    var y = event.clientY;
    console.log(`La posición del cursor es: ${x}, ${y}`);
  }
</script>
```

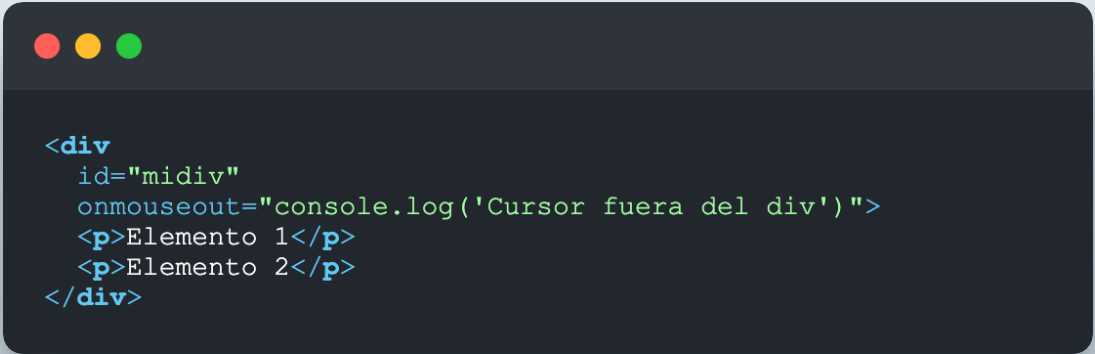
En este ejemplo, cada vez que el cursor se mueva dentro del área del body, se ejecutará la función **mostrarCoordenadas(event)**, lo que mostrará en la consola del navegador “La posición del cursor es: x, y” donde x e y son las coordenadas actuales del cursor en la página.

Este atributo puede ser utilizado para hacer un rastreo del movimiento del mouse, para crear efectos de hover, para crear dibujos o para crear mapas interactivos.

5.2.43 onmouseout

El atributo **onmouseout** en HTML se utiliza para especificar una función JavaScript que se ejecutará cuando el cursor del mouse salga de cualquier elemento hijo de un elemento específico de la página. Este evento se activa cada vez que el cursor sale de cualquier elemento hijo del elemento, incluso si el cursor todavía está dentro del área del elemento padre. Es similar al evento **onmouseleave**, pero este solo se activa cuando el cursor sale del área del elemento.

Por ejemplo, si queremos mostrar un mensaje en la consola cuando el cursor del mouse salga de un div que contiene varios elementos, podríamos utilizar el siguiente código:



```
<div
  id="midiv"
  onmouseout="console.log('Cursor fuera del div')">
  <p>Elemento 1</p>
  <p>Elemento 2</p>
</div>
```

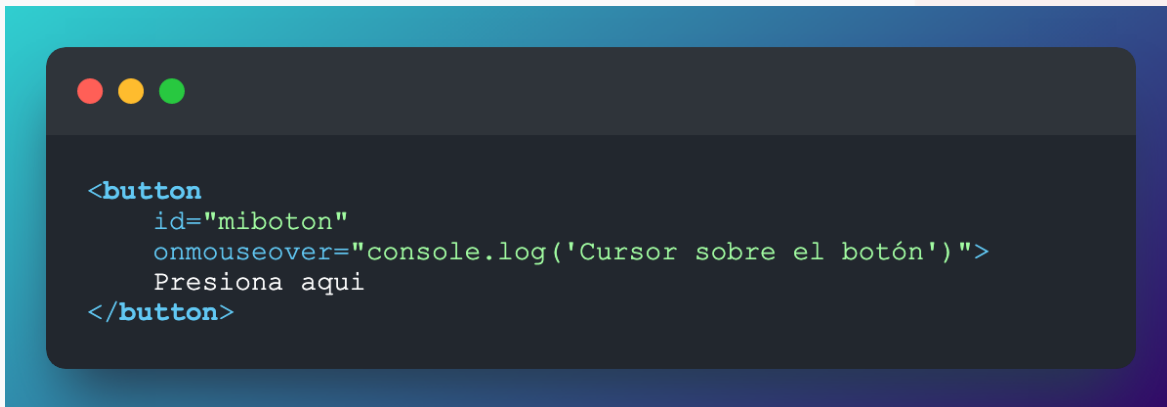
En este ejemplo, cada vez que el cursor salga de cualquier elemento dentro del div con id “midiv”, se ejecutará la función **console.log('Cursor fuera del div')**, lo que mostrará el mensaje “Cursor fuera del div” en la consola del navegador.

Este atributo es útil para crear efectos de hover para elementos que tienen varios elementos hijos o para crear un comportamiento diferente cuando el cursor sale de un elemento en particular.

5.2.44 onmouseover

El atributo **onmouseover** en HTML se utiliza para especificar una función JavaScript que se ejecutará cuando el cursor del mouse pase por encima de un elemento específico de la página. Este evento se activa cada vez que el cursor pasa por encima del elemento, incluso si el cursor ya está dentro del área del elemento. Es similar al evento **onmouseenter**, pero este solo se activa cuando el cursor entra en el área del elemento.

Por ejemplo, si queremos mostrar un mensaje en la consola cuando el cursor del mouse pase por encima de un botón, podríamos utilizar el siguiente código:



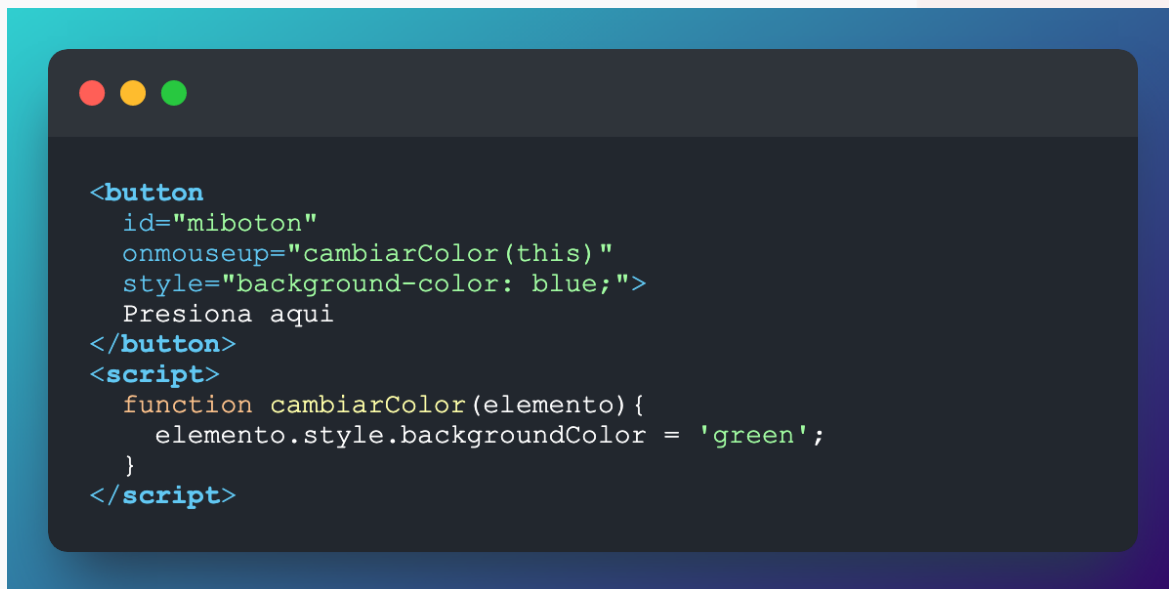
En este ejemplo, cada vez que el cursor pase por encima del botón con id “miboton”, se ejecutará la función **console.log('Cursor sobre el botón')**, lo que mostrará el mensaje “Cursor sobre el botón” en la consola del navegador.

Este atributo es utilizado para crear efectos de hover, para mostrar información adicional o para crear interacciones en una página web.

5.2.45 onmouseup

El atributo **onmouseup** en HTML se utiliza para especificar una función JavaScript que se ejecutará cuando se suelte el botón del mouse sobre un elemento específico de la página. Este evento se activa justo después del evento **onmousedown** (que se activa cuando se presiona el botón del mouse sobre el elemento) y antes del evento **onclick** (que se activa cuando se hace clic en el elemento).

Por ejemplo, si queremos cambiar el color de fondo de un botón cuando se suelte el botón del mouse sobre él, podríamos utilizar el siguiente código:



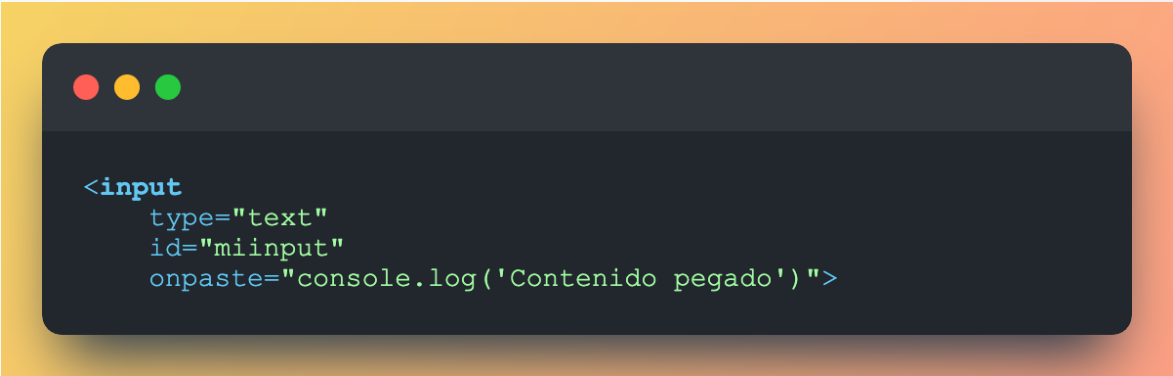
En este ejemplo, cuando se suelte el botón del mouse sobre el botón con id “miboton”, se ejecutará la función **cambiarColor(this)**, lo que cambiará el color de fondo del botón a verde.

Este atributo es utilizado para crear interacciones y comportamientos específicos cuando el usuario suelta el botón del mouse sobre un elemento.

5.2.46 onpaste

El atributo **onpaste** en HTML se utiliza para especificar una función JavaScript que se ejecutará cuando se pegue (o “se pegue”) contenido en un elemento específico de la página. Este evento se activa justo después de que el usuario haya pegado el contenido en el elemento y antes de que el contenido sea insertado.

Por ejemplo, si queremos mostrar un mensaje en la consola cuando se pegue algo en un input de texto, podríamos utilizar el siguiente código:



```
<input
  type="text"
  id="miinput"
  onpaste="console.log('Contenido pegado') ">
```

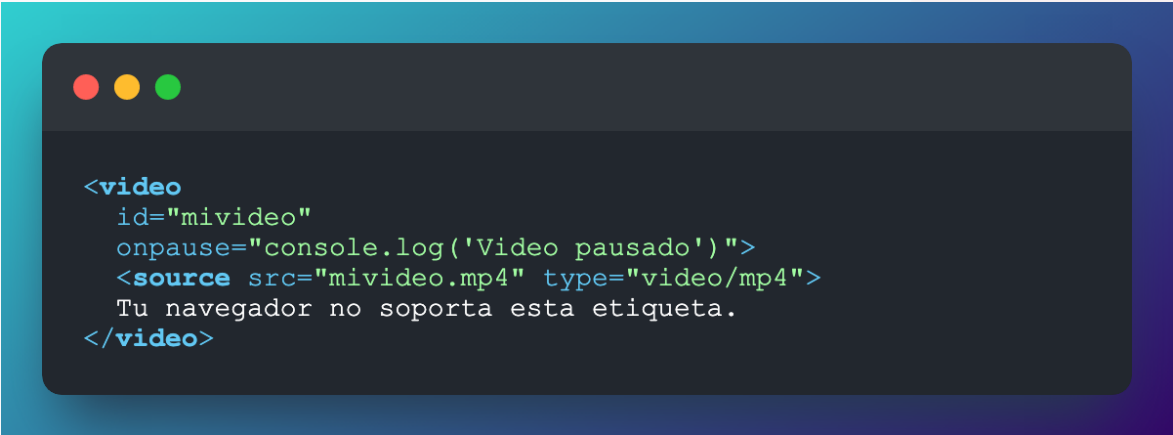
En este ejemplo, cuando se pegue algo en el input de texto con id “miinput” , se ejecutará la función **console.log('Contenido pegado')**, lo que mostrará el mensaje “Contenido pegado” en la consola del navegador.

Este atributo puede ser utilizado para validar el contenido pegado, para modificarlo antes de insertarlo o para controlar el número de caracteres que se pueden pegar en un elemento específico.

5.2.47 onpause

El atributo **onpause** en HTML se utiliza para especificar una función JavaScript que se ejecutará cuando se pausa un elemento multimedia (como un video o un audio) en una página web. Este evento se activa cuando el usuario hace clic en el botón de pausa o cuando se llama al método **pause()** en el elemento multimedia desde JavaScript.

Por ejemplo, si queremos mostrar un mensaje en la consola cuando se pausa un video, podríamos utilizar el siguiente código:



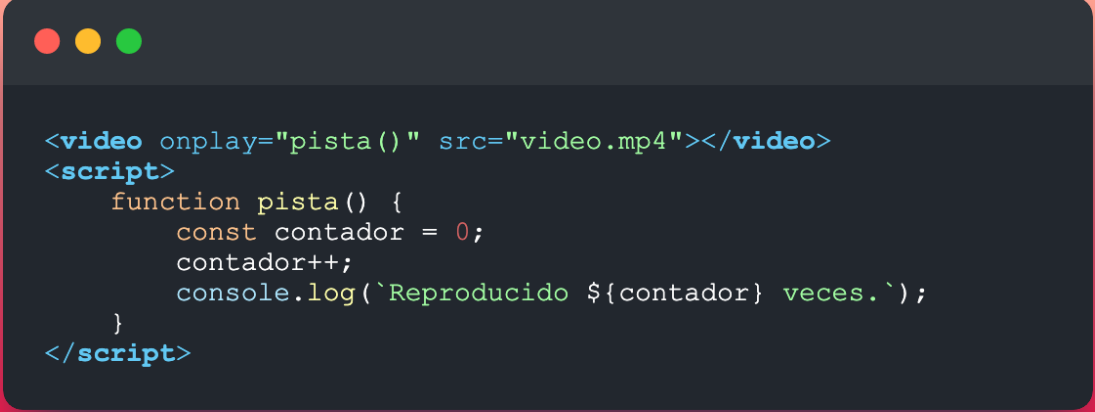
```
<video
  id="mivideo"
  onpause="console.log('Video pausado')">
  <source src="mivideo.mp4" type="video/mp4">
  Tu navegador no soporta esta etiqueta.
</video>
```

En este ejemplo, cuando se haga clic en el botón de pausa del video con id “mivideo” o cuando se llame al método **pause()** en el video, se ejecutará la función **console.log('Video pausado')**, lo que mostrará el mensaje “Video pausado” en la consola del navegador.

5.2.48 onplay

El atributo “onplay” en HTML es un evento que se activa cuando un medio, como un video o un audio, comienza a reproducirse. Un ejemplo práctico podría ser usar el atributo “onplay” para activar una función en JavaScript que registra el número de veces que un video ha sido reproducido en un sitio web.

Ejemplo:

A code editor window with a dark background and three colored window control buttons (red, yellow, green) in the top-left corner. The code is written in a light blue/cyan monospace font. It consists of an HTML `<video>` tag with an `onplay` event listener and a `src` attribute, followed by a `<script>` block containing a JavaScript function `pista()` that increments a counter and logs it to the console.

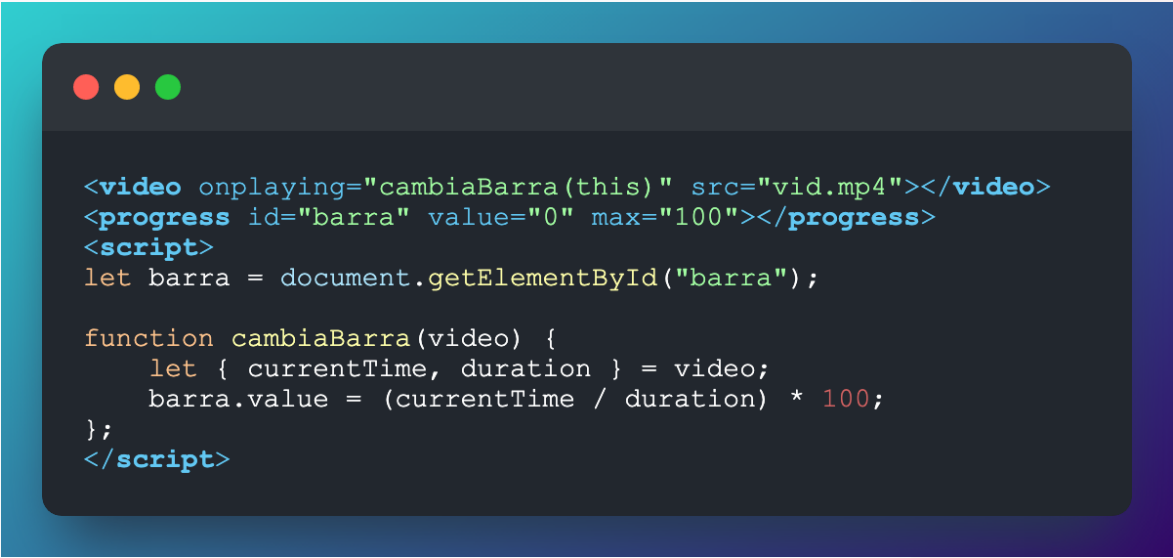
```
<video onplay="pista()" src="video.mp4"></video>
<script>
  function pista() {
    const contador = 0;
    contador++;
    console.log(`Reproducido ${contador} veces.`);
  }
</script>
```

En este ejemplo, cada vez que el video es reproducido, la función “pista” se activa y registra la cantidad de veces que ha sido reproducido en la consola del navegador.

5.2.49 onplaying

El atributo “onplaying” en HTML es un evento que se activa mientras un medio, como un video o un audio, se está reproduciendo. Un ejemplo práctico podría ser usar el atributo “onplaying” para actualizar una barra de progreso en tiempo real mientras se reproduce un video en un sitio web.

Ejemplo:



```
<video onplaying="cambiaBarra(this)" src="vid.mp4"></video>
<progress id="barra" value="0" max="100"></progress>
<script>
let barra = document.getElementById("barra");

function cambiaBarra(video) {
  let { currentTime, duration } = video;
  barra.value = (currentTime / duration) * 100;
};
</script>
```

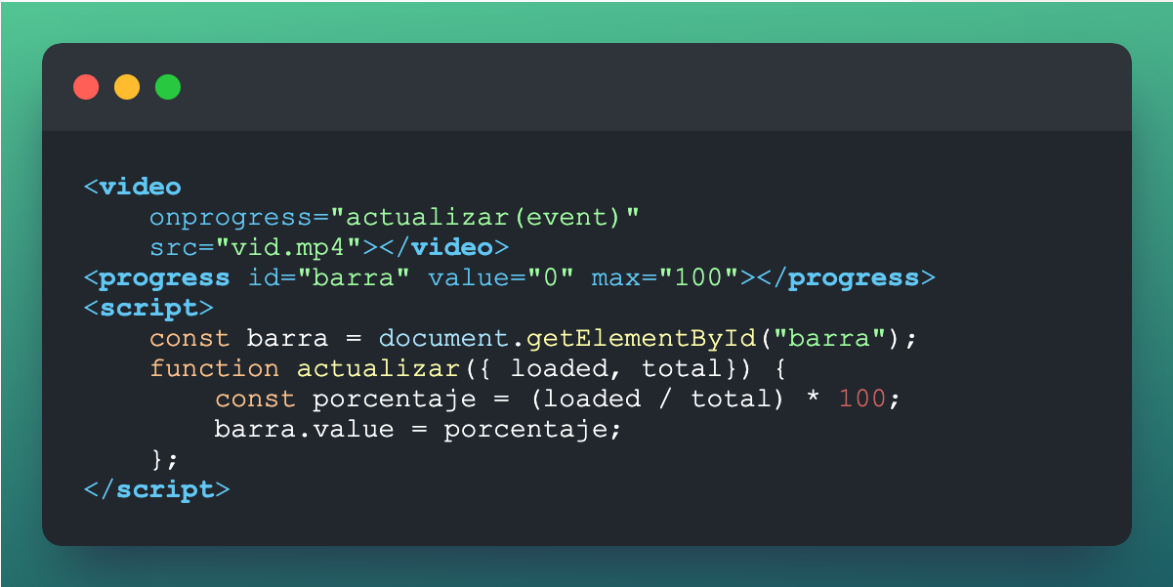
En este ejemplo, mientras el video se está reproduciendo, el evento “onplaying” se activa y actualiza el valor de la barra de progreso en función del tiempo de reproducción actual del video.

5.2.50 onprogress

El atributo “onprogress” en HTML es un evento que se activa cuando se está descargando o cargando un medio, como un video o un audio. Puede ser utilizado para mostrar una barra de progreso de la descarga o para notificar al usuario cuando el archivo está listo para ser reproducido.

Un ejemplo práctico podría ser usar el atributo “onprogress” para mostrar una barra de progreso mientras se descarga un video en un sitio web.

Ejemplo:



```
<video
  onprogress="actualizar(event) "
  src="vid.mp4"></video>
<progress id="barra" value="0" max="100"></progress>
<script>
  const barra = document.getElementById("barra");
  function actualizar({ loaded, total}) {
    const porcentaje = (loaded / total) * 100;
    barra.value = porcentaje;
  };
</script>
```

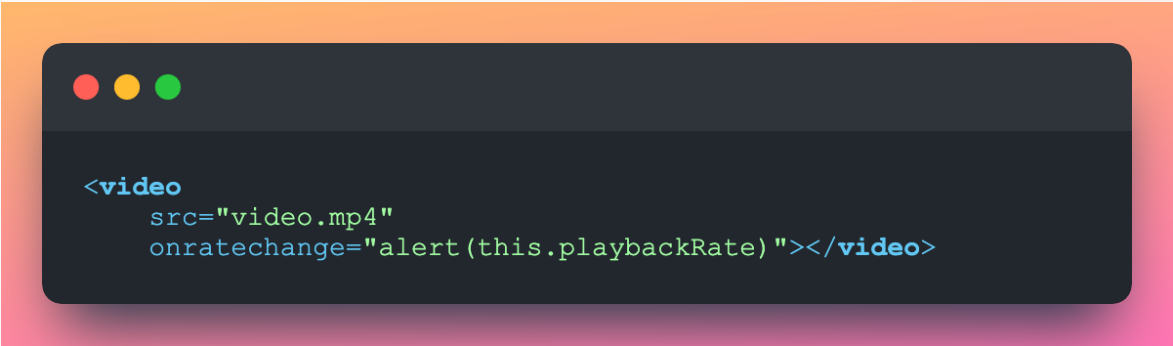
En este ejemplo, mientras el video se está descargando, el evento “onprogress” se activa y actualiza el valor de la barra de progreso en función de la cantidad de datos descargados.

5.2.51 onratechange

El atributo “onratechange” en HTML es un evento que se activa cuando el usuario cambia la velocidad de reproducción de un medio, como un video o un audio.

Un ejemplo práctico podría ser usar el atributo “onratechange” para mostrar una notificación al usuario cuando cambia la velocidad de reproducción de un video en un sitio web.

Ejemplo:



```
<video  
  src="video.mp4"  
  onratechange="alert (this.playbackRate) "></video>
```

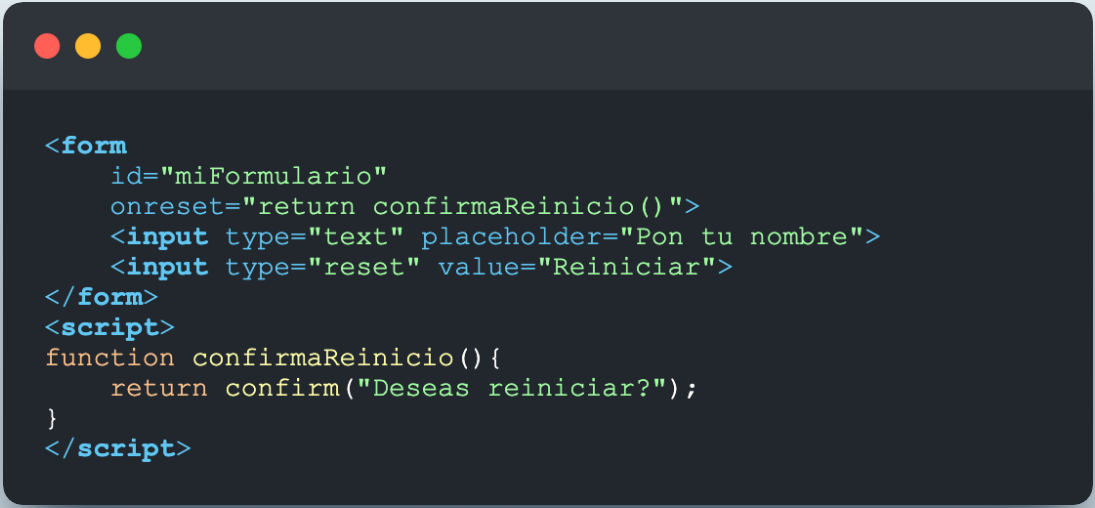
En este ejemplo, cada vez que el usuario cambia la velocidad de reproducción del video, el evento “onratechange” se activa y muestra una notificación en la consola del navegador con la velocidad de reproducción.

5.2.52 onreset

El atributo “onreset” en HTML es un evento que se activa cuando un usuario intenta restablecer un formulario en una página web. Es utilizado para ejecutar una función de JavaScript antes de que el formulario sea restablecido.

Un ejemplo práctico podría ser usar el atributo “onreset” para mostrar una alerta al usuario antes de que se restablezca un formulario.

Ejemplo:



```
<form
  id="miFormulario"
  onreset="return confirmaReinicio()">
  <input type="text" placeholder="Pon tu nombre">
  <input type="reset" value="Reiniciar">
</form>
<script>
function confirmaReinicio() {
  return confirm("Deseas reiniciar?");
}
</script>
```

En este ejemplo, al presionar el botón “Reset” en el formulario, se activará la función personalizada de JavaScript “confirmaReinicio” y se mostrará una alerta al usuario preguntando si está seguro de que quiere restablecer el formulario. Si el usuario presiona “Cancel” en la alerta, el formulario no será restablecido. Si el usuario presiona “OK” en la alerta, el formulario será restablecido.

Es importante mencionar que, si la función devuelve falso, el formulario no se restablecerá.

5.2.53 onresize

El atributo “onresize” en HTML es un evento que se activa cuando un elemento en una página web cambia de tamaño. Es comúnmente utilizado para ejecutar una función de JavaScript que se ajuste al cambio de tamaño de la ventana del navegador.

Un ejemplo práctico podría ser usar el atributo “onresize” para ajustar el tamaño de una imagen en una página web de acuerdo con el tamaño de la ventana del

5 ATRIBUTOS

navegador.

Ejemplo:



En este ejemplo, cada vez que el tamaño de la ventana del navegador cambia, se activará la función personalizada de JavaScript “ajustarTamaño” y la imagen se ajustará al 50% del ancho de la ventana.

Es importante mencionar que el evento `onresize` solo puede ser aplicado a la ventana del navegador o al `body` del documento, no a los elementos individuales.

5.2.54 onscroll

El atributo “`onscroll`” en HTML es un evento que se activa cuando un usuario hace scroll en una página web, ya sea mediante el uso del mouse, el touchpad o las teclas de flecha. Es comúnmente utilizado para ejecutar una función de JavaScript que realice una acción específica al hacer scroll en una página.

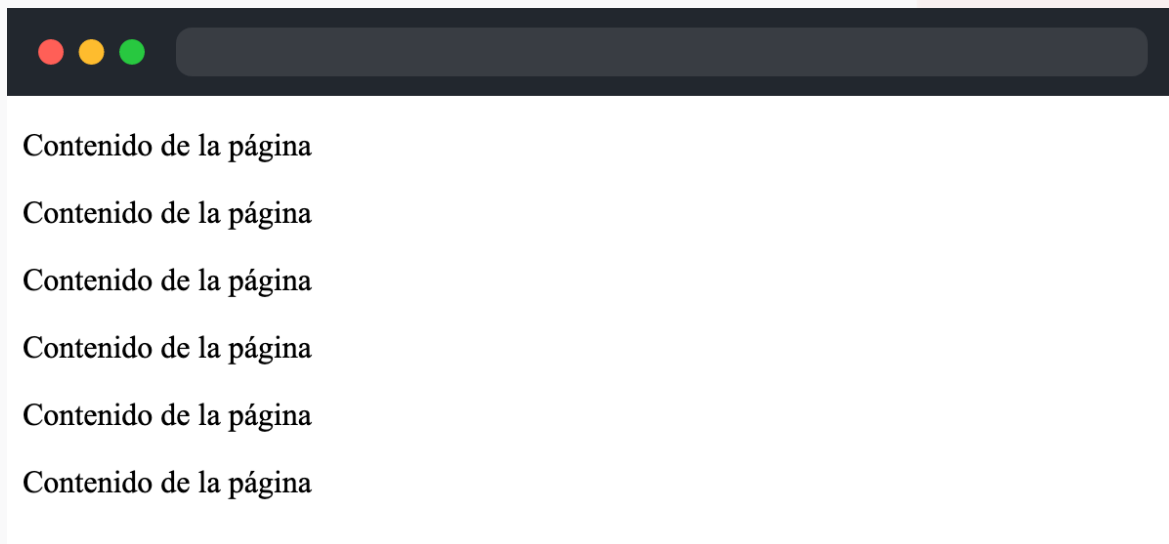
Un ejemplo práctico podría ser usar el atributo “`onscroll`” para mostrar un botón “Ir arriba” en la parte inferior derecha de una página web mientras el usuario hace scroll.

Ejemplo:


```
<body onscroll="mostrarBoton()">
  <button
    id="botonSuperior"
    style="display:none">Ir arriba</button>
  <p>Contenido de la página</p>
  <p>Contenido de la página</p>
  <p>Contenido de la página</p>
  <p>Contenido de la página</p>
  <p>Contenido de la página</p>
  <p>Contenido de la página</p>
</body>
<script>
function mostrarBoton() {
  const btn = document.getElementById("botonSuperior");
  if (document.body.scrollTop > 20
    || document.documentElement.scrollTop > 20) {
    btn.style.display = "block";
  } else {
    btn.style.display = "none";
  }
}
</script>
```

El código se renderiza de esta manera al abrirlo en un navegador.

5 ATRIBUTOS



En este ejemplo, al hacer scroll en la página, se activará la función personalizada de JavaScript “mostrarBoton” y mostrará el botón “Ir arriba” en la parte inferior derecha de la página. Al presionar el botón, el usuario será llevado al inicio de la página.

Es importante mencionar que para llevar al usuario al inicio de la página, debes agregar más código javascript para manejar el evento del click del botón y usar el metodo **scrollTo(x,y)** o **scrollIntoView()**

5.2.55 onscrollend

El atributo “onscrollend” en HTML es un evento que se activa cuando un usuario ha dejado de hacer scroll en una página web. Es un evento menos comúnmente utilizado en comparación con “onscroll” y es soportado solo en algunos navegadores modernos.

Un ejemplo práctico podría ser usar el atributo “onscrollend” para ejecutar una función de JavaScript que cambie el color de fondo de una sección de la página una vez que el usuario ha dejado de hacer scroll.

5 ATRIBUTOS

Ejemplo:



El código se renderiza de esta manera al abrirlo en un navegador.



En este ejemplo, una vez que el usuario ha dejado de hacer scroll en la sección “miSeccion” , se activará la función personalizada de JavaScript “cambiarColor” y cambiará el color de fondo de la sección a rojo.

Es importante mencionar que la compatibilidad del evento onscrollend varia dependiendo del navegador, y no está soportado en todos ellos, así que es posible que no funcione en todos los casos.

5.2.56 onsecuritypolicyviolation

El atributo “onsecuritypolicyviolation” en HTML es un evento que se activa cuando una violación de política de seguridad ha sido detectada en una página web. Es comúnmente utilizado para ejecutar una función de JavaScript que maneje esta violación.

5 ATRIBUTOS

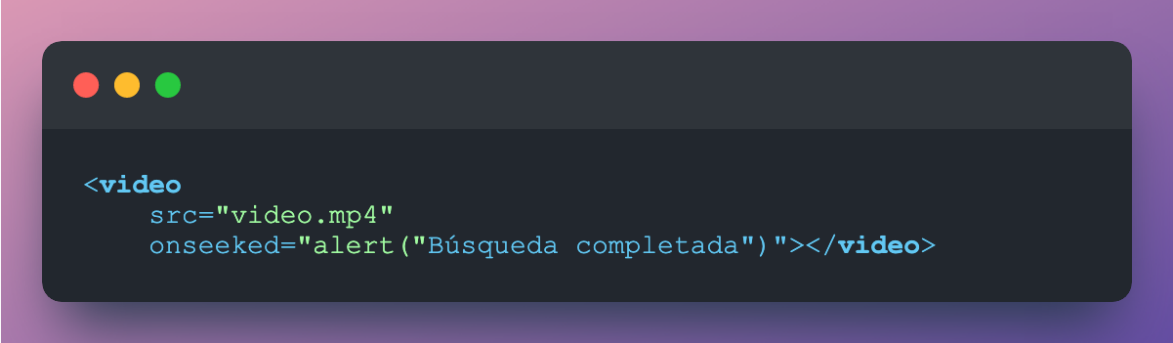
Es importante mencionar que el evento “onsecuritypolicyviolation” es soportado solo en algunos navegadores modernos y no en todos.

5.2.57 onseeked

El atributo “onseeked” en HTML es un evento que se activa cuando un usuario ha terminado de buscar en un medio, como un video o un audio. Es utilizado para ejecutar una función de JavaScript después de que el usuario ha terminado de buscar en el medio.

Un ejemplo práctico podría ser usar el atributo “onseeked” para mostrar una notificación al usuario después de que ha terminado de buscar en un video en un sitio web.

Ejemplo:



```
<video  
  src="video.mp4"  
  onseeked="alert('Búsqueda completada') "></video>
```

En este ejemplo, al finalizar la búsqueda en el video, se activará la función personalizada de JavaScript “showSeekComplete” y se mostrará una notificación en la consola del navegador indicando que la búsqueda ha sido completada.

5.2.58 onseeking

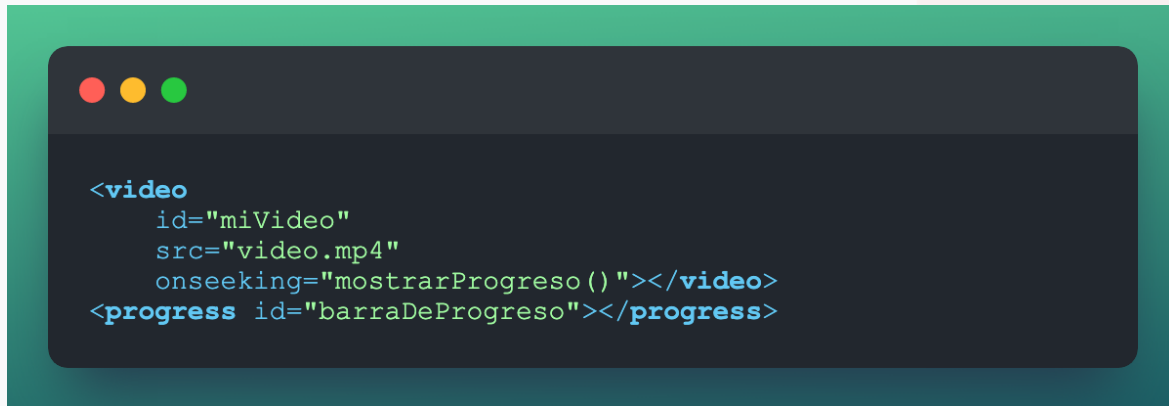
El atributo “onseeking” en HTML es un evento que se activa cuando un usuario está buscando en un medio, como un video o un audio. Es utilizado para ejecutar

5 ATRIBUTOS

una función de JavaScript mientras el usuario está buscando en el medio.

Un ejemplo práctico podría ser usar el atributo “onseeking” para mostrar una barra de progreso mientras el usuario busca en un video en un sitio web.

Ejemplo:



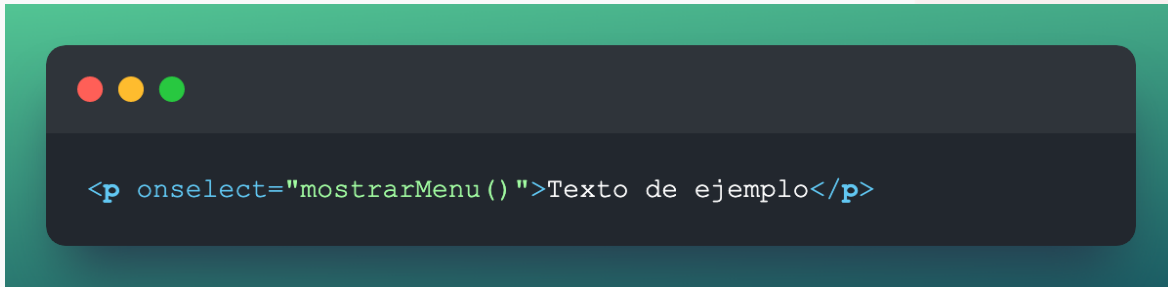
En este ejemplo, mientras el usuario busca en el video, se activará la función personalizada de JavaScript “mostrarProgreso” y se actualizará una barra de progreso en la página web que indica el tiempo actual en relación con la duración total del video.

5.2.59 onselect

El atributo “onselect” en HTML es un evento que se activa cuando un usuario selecciona texto en una página web. Es utilizado para ejecutar una función de JavaScript cuando el usuario selecciona texto en una página web.

Un ejemplo práctico podría ser usar el atributo “onselect” para mostrar un menú contextual con opciones adicionales, después de que el usuario ha seleccionado texto en una página web.

Ejemplo:



En este ejemplo, al seleccionar texto en el párrafo con id “miTexto” , se activará la función personalizada de JavaScript personalizada de JavaScript “mostrarMenu” .

5.2.60 onslotchange

El atributo “onslotchange” en HTML es un evento que se activa cuando el contenido de un slot en una página web cambia. Es utilizado para ejecutar una función de JavaScript cuando el contenido de un slot ha cambiado.

Un ejemplo práctico podría ser usar el atributo “onslotchange” para actualizar el contenido de una sección de una página web cada vez que el contenido de un slot cambia.

Ejemplo:

```
<template id="plantilla">
  <slot name="contenido"></slot>
</template>

<div id="miSeccion" onslotchange="actualizarContenido()">
  <template id="plantilla">
    <p>Texto de ejemplo</p>
  </template>
</div>

<script>
  function actualizarContenido() {
    const contenido = document.getElementById("plantilla")
      .contenido;
    document.getElementById("miSeccion").innerHTML = "";

    document.getElementById("miSeccion").appendChild(contenido);
  }
</script>
```

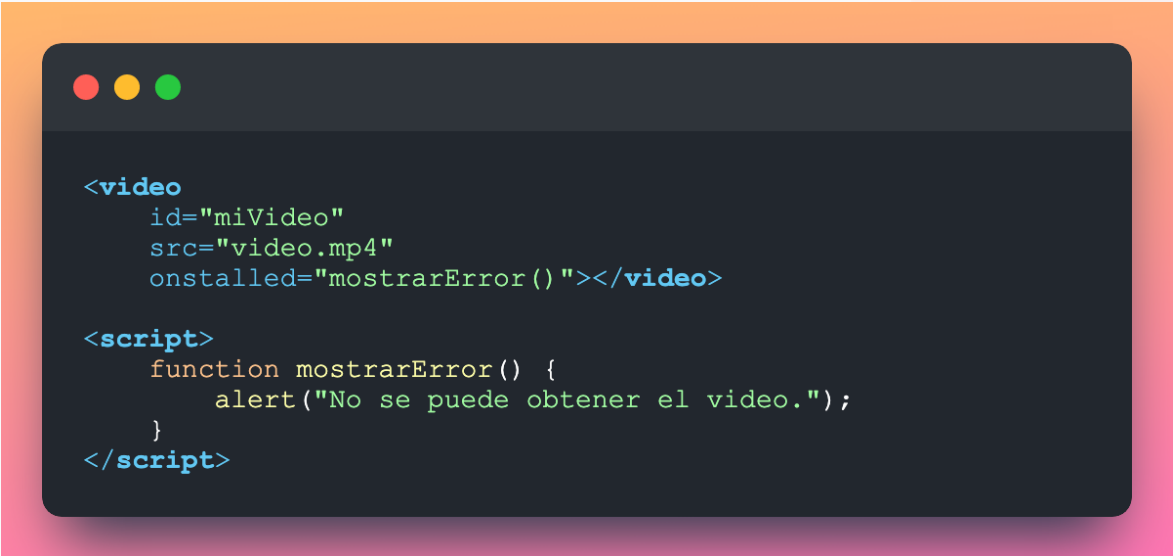
En este ejemplo, cada vez que el contenido del slot “contenido” en el template “plantilla” cambia, se activa la función “actualizarContenido” y actualiza el contenido de la sección “miSeccion”. De esta manera, cualquier cambio en el contenido del slot se reflejará automáticamente en la sección “miSeccion” de la página. Es importante mencionar que las secciones **<template>** son utilizadas para almacenar contenido que se va a utilizar en algún momento en el futuro, pero que no se desea mostrar inmediatamente, el atributo slot es utilizado para definir un lugar donde se desea que el contenido sea insertado, y cuando el contenido es insertado en ese lugar es cuando se activa el evento onslotchange.

5.2.61 onstalled

El atributo “onstalled” en HTML es un evento que se activa cuando el navegador no puede obtener un recurso multimedia, como un video o un audio debido a un problema de red. Es utilizado para ejecutar una función de JavaScript cuando el navegador no puede obtener un recurso multimedia.

Un ejemplo práctico podría ser usar el atributo “onstalled” para mostrar un mensaje de error al usuario cuando el navegador no puede obtener un video debido a un problema de red.

Ejemplo:



```
<video
  id="miVideo"
  src="video.mp4"
  onstalled="mostrarError()"></video>

<script>
  function mostrarError() {
    alert("No se puede obtener el video.");
  }
</script>
```

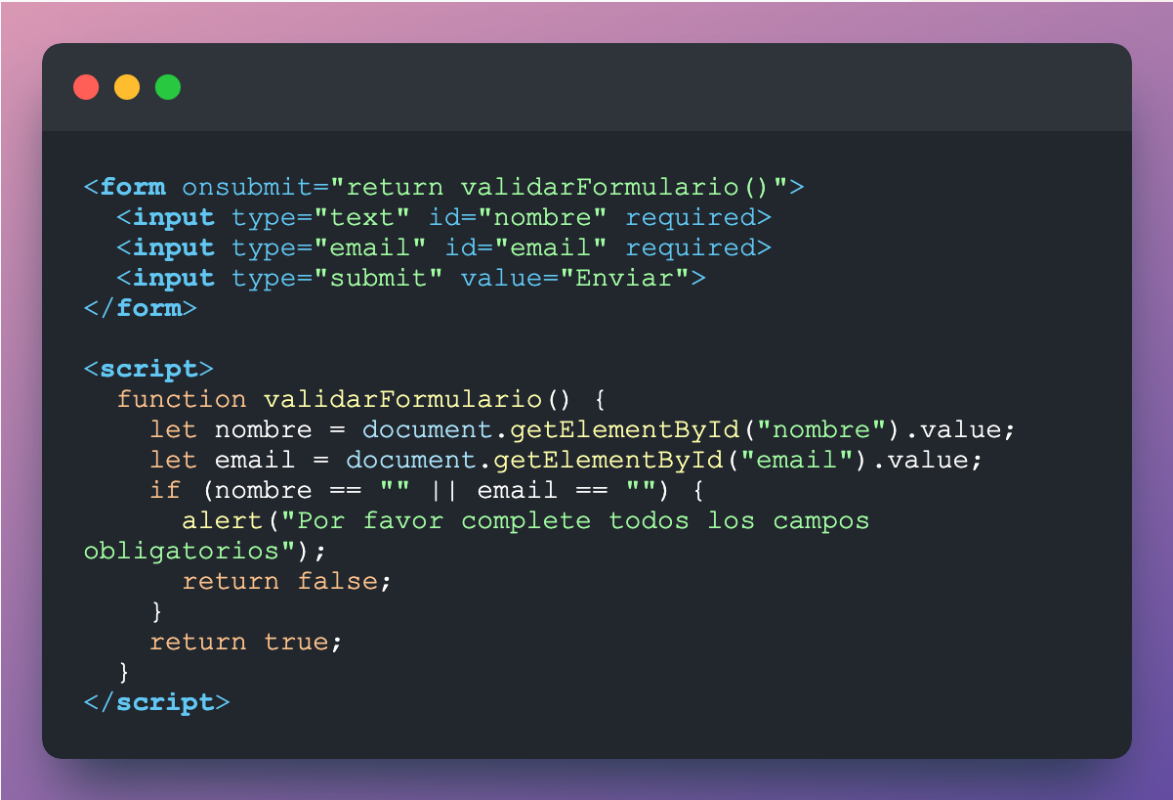
En este ejemplo, si el navegador no puede obtener el video debido a un problema de red, se activará la función personalizada de JavaScript personalizada de JavaScript “mostrarError” .

5.2.62 onsubmit

El atributo “onsubmit” en HTML es un evento que se activa cuando un usuario envía un formulario en una página web. Es utilizado para ejecutar una función de JavaScript cuando el formulario es enviado, permitiendo al desarrollador validar o procesar la información del formulario antes de enviarlo al servidor.

Un ejemplo práctico podría ser usar el atributo “onsubmit” para validar que los campos obligatorios del formulario estén completos antes de enviar el formulario.

Ejemplo:



```
<form onsubmit="return validarFormulario()">
  <input type="text" id="nombre" required>
  <input type="email" id="email" required>
  <input type="submit" value="Enviar">
</form>

<script>
  function validarFormulario() {
    let nombre = document.getElementById("nombre").value;
    let email = document.getElementById("email").value;
    if (nombre == "" || email == "") {
      alert("Por favor complete todos los campos obligatorios");
      return false;
    }
    return true;
  }
</script>
```

En este ejemplo, cuando el usuario presiona el botón “Enviar” , se activa la función personalizada de JavaScript “validarFormulario” . La función podría validar los campos “nombre” y “email” usando JavaScript (que no se muestra como parte del

ejemplo) y podría tener código que envíe el formulario. Es importante mencionar que en este ejemplo se está utilizando el atributo **required** en los input del formulario, si el valor de esos input es vacío, el formulario no se enviará automáticamente, pero con el uso de `onsubmit` se puede personalizar la acción a tomar desde presionar el botón hasta enviar el formulario.

6 Siguientes pasos

6.1 Herramientas

1. **CSS:** CSS (Cascading Style Sheets) es un lenguaje de hojas de estilo utilizado para definir la presentación de documentos HTML. Esto incluye el diseño, el color, la fuente, la alineación, el tamaño, etc.
2. **JavaScript:** JavaScript es un lenguaje de programación interpretado que se utiliza para crear contenido dinámico en páginas web. Esto incluye animaciones, juegos, aplicaciones web, etc.
3. **jQuery:** jQuery es una biblioteca de JavaScript que facilita la manipulación de elementos HTML, así como la creación de animaciones y aplicaciones web.
4. **Bootstrap:** Bootstrap es un marco de trabajo de código abierto para el desarrollo de sitios web y aplicaciones web. Proporciona un conjunto de herramientas para crear diseños responsivos y aplicaciones web.
5. **PHP:** PHP es un lenguaje de programación de servidor que se utiliza para crear contenido dinámico en páginas web. Esto incluye la creación de formularios, bases de datos, etc.

6.2 Recursos

1. [W3Schools](#): W3Schools es una de las mejores fuentes de información para

6 SIGUIENTES PASOS

aprender HTML. Ofrece tutoriales, ejemplos y referencias para ayudar a los usuarios a aprender HTML.

2. [MDN Web Docs](#): MDN Web Docs es una excelente fuente de información para aprender HTML. Ofrece tutoriales, ejemplos y referencias para ayudar a los usuarios a aprender HTML.
3. [Codecademy](#): Codecademy es una plataforma de aprendizaje en línea que ofrece cursos interactivos para aprender HTML. Ofrece tutoriales, ejemplos y referencias para ayudar a los usuarios a aprender HTML.
4. [HTML Dog](#): HTML Dog es una excelente fuente de información para aprender HTML. Ofrece tutoriales, ejemplos y referencias para ayudar a los usuarios a aprender HTML.
5. [Tutorialspoint](#): Tutorialspoint es una excelente fuente de información para aprender HTML. Ofrece tutoriales, ejemplos y referencias para ayudar a los usuarios a aprender HTML.

6.3 Despedida

Estos son los siguientes pasos recomendados:

1. Aprender CSS: CSS es un lenguaje de hojas de estilo que se utiliza para dar estilo y formato a los documentos HTML.
2. Aprender JavaScript: JavaScript es un lenguaje de programación que se utiliza para crear contenido interactivo en páginas web.
3. Aprender sobre el diseño web: El diseño web se refiere a la planificación, creación y mantenimiento de sitios web.
4. Aprender sobre el desarrollo web: El desarrollo web se refiere al proceso de crear sitios web, aplicaciones web y servicios web.
5. Aprender sobre el alojamiento web: El alojamiento web se refiere al proceso de almacenar los archivos de un sitio web en un servidor web.

6.4 Preguntas de entrevista

Estas son 99 preguntas comunes en una entrevista sobre HTML para practicar y probar tus conocimientos:

1. ¿Qué es HTML?

HTML (Hypertext Markup Language) es un lenguaje de marcas utilizado para estructurar el contenido de una página web.

2. ¿Cómo se utiliza HTML para estructurar el contenido de una página web?

HTML utiliza etiquetas para definir diferentes tipos de contenido, como encabezados, párrafos, imágenes, enlaces y más. Estas etiquetas se utilizan para estructurar el contenido de una página web en una jerarquía lógica, lo que permite a los navegadores web mostrar el contenido de una manera consistente y accesible.

3. ¿Cuál es la estructura básica de un documento HTML?

La estructura básica de un documento HTML incluye una etiqueta de **<doctype>**, una etiqueta de **<html>** (que engloba el contenido de la página), una etiqueta de **<head>** (que contiene metadatos sobre la página) y una etiqueta de **<body>** (que contiene el contenido de la página visible).

4. ¿Qué es una etiqueta de ancla en HTML?

Una etiqueta de ancla en HTML es una etiqueta utilizada para crear un enlace dentro de una página web. Utiliza la etiqueta **<a>** y requiere el atributo `href` para especificar el destino del enlace.

5. ¿Cómo se puede agregar una imagen a una página web utilizando HTML?

Se puede agregar una imagen a una página web utilizando la etiqueta **** de HTML y especificando la ubicación de la imagen mediante el atributo `src`.

6. ¿Qué es una lista ordenada y una lista no ordenada en HTML?

Una lista ordenada en HTML es una lista en la que cada elemento está numerado y se representa con la etiqueta ****. Una lista no ordenada en HTML es una lista en la que cada elemento no está numerado y se representa con la etiqueta ****

7. ¿Qué es el atributo “alt” en la etiqueta **** de HTML?

El atributo “alt” en la etiqueta **** de HTML es utilizado para proporcionar un texto alternativo que se mostrará en caso de que la imagen no se pueda mostrar. Es importante proporcionar texto alternativo para mejorar la accesibilidad del sitio web

8. ¿Qué son los metadatos en HTML?

Los metadatos en HTML son información adicional sobre la página que no se muestra en la página en sí, pero que proporciona información sobre la página a los navegadores web, motores de búsqueda y otros sistemas. Los metadatos se incluyen en la sección “head” del documento HTML.

9. ¿Cómo se pueden crear tablas en HTML?

Las tablas en HTML se crean utilizando la etiqueta **<table>**, con etiquetas **<tr>** para definir las filas, etiquetas **<th>** o **<td>** para definir las celdas y, opcionalmente, etiquetas **<thead>**, **<tbody>** y **<tfoot>** para definir los encabezados, el cuerpo y el pie de tabla.

10. ¿Cómo se puede especificar el color de fondo de un elemento en HTML?

El color de fondo de un elemento en HTML se puede especificar utilizando el atributo “style” y especificando el valor “background-color” .

11. ¿Qué es el atributo “class” en HTML?

El atributo “class” en HTML se utiliza para asignar una o varias clases a un elemento. Esto permite a los desarrolladores aplicar estilos CSS específicos a un conjunto de elementos con el mismo nombre de clase.

12. ¿Cómo se puede crear un formulario en HTML?

Los formularios en HTML se crean utilizando la etiqueta **<form>**, con etiquetas dentro del formulario para definir campos de entrada, como **<input>**, **<select>** y **<textarea>**, y etiquetas para los botones de envío y resetear.

13. ¿Qué es el atributo “action” en la etiqueta **<form>** de HTML?

El atributo “action” en la etiqueta **<form>** de HTML se utiliza para especificar el destino del formulario, es decir, a qué URL se enviará la información del formulario cuando se envíe.

14. ¿Qué es el atributo “method” en la etiqueta **<form>** de HTML?

El atributo “method” en la etiqueta **<form>** de HTML se utiliza para especificar el método de envío de información del formulario, ya sea “get” o “post” .

15. ¿Qué es la semántica en HTML?

La semántica en HTML se refiere a la correcta utilización de etiquetas y atributos para describir el significado del contenido dentro de una página web. Esto mejora la accesibilidad del sitio y facilita la comprensión por parte de los motores de búsqueda.

16. ¿Qué es el atributo “data-” en HTML?

El atributo “data-” en HTML es un atributo genérico que se puede utilizar para almacenar cualquier tipo de información en un elemento HTML, esto es útil para almacenar información adicional sobre un elemento y utilizarlo en javascript.

17. ¿Cómo se pueden agregar estilos a una página web utilizando HTML?

Los estilos pueden ser aplicados a una página web utilizando HTML de varias maneras. Una forma es utilizar la etiqueta **<style>** en la sección “head” del documento HTML y escribir reglas CSS dentro de ella. También se pueden aplicar estilos utilizando el atributo “style” en un elemento HTML específico, y especificando los estilos directamente.

18. ¿Qué son las entidades de caracteres en HTML?

Las entidades de caracteres en HTML son códigos especiales que se utilizan para representar caracteres especiales, como símbolos, que de otra manera no podrían ser incluidos en el código HTML. Ejemplos de entidades de caracteres incluyen “<” para < y “&” para &.

19. ¿Qué es la **<div>** en HTML?

La etiqueta **<div>** en HTML es una etiqueta de contenedor genérica que se utiliza para agrupar elementos HTML con un propósito específico. Por ejemplo, un “div” puede ser utilizado para agrupar varios elementos en una sección de una página web.

20. ¿Cómo se pueden crear enlaces a otras páginas y a recursos externos en HTML?

Los enlaces a otras páginas y a recursos externos se pueden crear en HTML utilizando la etiqueta **<a>** y especificando la dirección del destino del enlace en el atributo “href” .

21. ¿Cómo se pueden crear secciones en una página web utilizando HTML?

Secciones en una página web se pueden crear utilizando las etiquetas **<header>**, **<nav>**, **<main>**, **<article>** , **<aside>** y **<footer>**, cada una de las cuales tiene un significado semántico específico.

22. ¿Qué es el atributo “placeholder” en HTML?

El atributo “placeholder” en HTML es un atributo que se utiliza para proporcionar una descripción de ejemplo de lo que debe ser ingresado en un campo de formulario. Es una sugerencia al usuario y desaparece cuando el usuario comienza a escribir en el campo

23. ¿Qué es el atributo “required” en HTML?

El atributo “required” en HTML es un atributo que se utiliza para hacer que un campo de formulario sea obligatorio. Esto significa que el usuario no puede enviar el formulario si el campo no está completo.

24. ¿Qué es el atributo “disabled” en HTML?

El atributo “disabled” en HTML es un atributo que se utiliza para deshabilitar un elemento. Un elemento deshabilitado no puede ser utilizado o modificado por el usuario.

25. ¿Qué es la etiqueta **<figure>** en HTML?

La etiqueta **<figure>** en HTML es una etiqueta utilizada para contener elementos autocontenidos, como imágenes o videos, y su caption (descripción). Es útil para dar contexto o información sobre ese elemento.

26. ¿Qué es la etiqueta **<mark>** en HTML?

La etiqueta **<mark>** en HTML es una etiqueta utilizada para resaltar texto dentro de un párrafo o contenido, como un texto marcado o subrayado, para indicar que es de importancia especial.

27. ¿Qué es la **<time>** en HTML?

La **<time>** en HTML es una etiqueta utilizada para representar fechas y horas, y se puede utilizar en conjunto con atributos “datetime” para especificar el momento exacto que puede ser útiles para buscadores.

28. ¿Cómo se pueden crear títulos y subtítulos en una página web utilizando HTML?

Los títulos y subtítulos en una página web se pueden crear utilizando las etiquetas **<h1>** a **<h6>** para los encabezados, donde “h1” es el título principal y “h6” es el subtítulo más pequeño.

29. ¿Qué es el atributo “lang” en HTML?

El atributo “lang” en HTML es utilizado para especificar el idioma del contenido en un elemento. Esto puede ser utilizado por los navegadores para habilitar funciones específicas del idioma, como la ortografía automática o la lectura de pantalla.

30. ¿Cómo se pueden crear botones en una página web utilizando HTML?

Los botones en una página web se pueden crear utilizando la etiqueta **<button>** y, opcionalmente, utilizando atributos como “type” , “name” y “value” para especificar la función del botón.

31. ¿Qué es el atributo “for” en HTML?

El atributo “for” en HTML es utilizado para asociar una etiqueta de **<label>** con un elemento de formulario. El valor del atributo “for” debe ser igual al atributo “id” del elemento de formulario asociado.

32. ¿Qué es el atributo “form” en HTML?

El atributo “form” en HTML es utilizado para asociar un elemento de formulario con un formulario específico. El valor del atributo “form” debe ser igual al atributo “id” del formulario al que el elemento debe ser enviado.

33. ¿Qué es el atributo “value” en HTML?

El atributo “value” en HTML es utilizado para especificar el valor predeterminado de un elemento de formulario, como un campo de entrada o un botón de opción.

34. ¿Qué es el atributo “name” en HTML?

El atributo “name” en HTML es utilizado para dar un nombre a un elemento de formulario, lo que permite que los datos enviados por el formulario se asocien con el elemento correcto en el lado del servidor.

35. ¿Cómo se pueden insertar imágenes en una página web utilizando HTML?

Las imágenes se pueden insertar en una página web utilizando la etiqueta **** y especificando la dirección del archivo de imagen en el atributo “src” .

36. ¿Qué es el atributo “alt” en HTML?

6 SIGUIENTES PASOS

El atributo “alt” en HTML es utilizado para proporcionar una descripción alternativa de una imagen, esto es especialmente útil para personas con discapacidad visual.

37. ¿Cómo se pueden crear listas en HTML?

Las listas se pueden crear en HTML utilizando las etiquetas **** para listas desordenadas y “ol” para listas ordenadas, y etiquetas **** para cada elemento de la lista.

38. ¿Qué es el atributo “srcset” en HTML?

El atributo “srcset” en HTML se utiliza para proporcionar varias opciones de imagen para una sola imagen “img”, con distintas resoluciones y tamaños, de esta manera los navegadores pueden elegir la mejor imagen según la resolución de la pantalla del dispositivo.

39. ¿Qué es el atributo “sizes” en HTML?

El atributo “sizes” en HTML se utiliza en conjunto con el atributo “srcset” para indicarle al navegador el tamaño de la imagen que se espera que se muestre en pantalla dependiendo del tamaño de la ventana del navegador.

40. ¿Cómo se pueden crear menús de navegación en una página web utilizando HTML?

Los menús de navegación se pueden crear en una página web utilizando la etiqueta **<nav>** y etiquetas **<a>** para cada elemento del menú, dentro de una lista desordenada **** o ordenada **** para mejorar el accesibilidad.

41. ¿Cómo se pueden crear áreas de interacción en una página web utilizando HTML?

Las áreas de interacción en una página web se pueden crear utilizando diferentes elementos de formularios como “input”, “select”, “textarea”, “button”, entre otros. Estos elementos pueden ser utilizados con atributos como “type”, “name”, “value” y “required” para especificar la función de cada área de

interacción.

42. ¿Cómo se pueden crear tablas en HTML?

Las tablas se pueden crear en HTML utilizando la etiqueta **<table>** y elementos como **<thead>**, **<tbody>**, **<tfoot>**, **<tr>**, **<th>** y **<td>** para construir las filas y las celdas de la tabla.

43. ¿Qué es el atributo “scope” en HTML?

El atributo “scope” en HTML se utiliza en las etiquetas **<th>** para especificar el alcance de un encabezado de tabla, ya sea un encabezado de columna, de fila o de grupo.

44. ¿Cómo se pueden crear formularios en HTML?

Los formularios se pueden crear en HTML utilizando la etiqueta **<form>** y elementos como **<input>**, **<select>**, **<textarea>**, **<button>**, entre otros, para crear los campos y botones del formulario.

45. ¿Qué es el atributo “action” en HTML?

El atributo “action” en HTML se utiliza en la etiqueta **<form>** para especificar la URL a la cual los datos del formulario deben ser enviados una vez que el usuario ha presionado el botón “submit” .

46. ¿Qué es el atributo “method” en HTML?

El atributo “method” en HTML se utiliza en la etiqueta **<form>** para especificar cómo se deben enviar los datos del formulario al servidor, ya sea utilizando el método HTTP “get” o “post” .

47. ¿Qué es la etiqueta **<iframe>** en HTML?

La etiqueta **<iframe>** en HTML es utilizada para insertar un documento HTML dentro de otro documento HTML, permitiendo mostrar contenido de otra página web dentro de la página actual.

48. ¿Qué es el atributo “src” en HTML?

6 SIGUIENTES PASOS

El atributo “src” en HTML se utiliza en varias etiquetas, como **** y **<iframe>**, para especificar la URL del recurso que se debe cargar en el elemento.

49. ¿Qué es la etiqueta **<meta>** en HTML?

La etiqueta **<meta>** en HTML es utilizada para proporcionar información adicional sobre una página web, como la descripción, las palabras clave, los caracteres de codificación, entre otros.

50. ¿Cómo se pueden crear enlaces a una ubicación específica dentro de una página utilizando HTML?

Los enlaces a una ubicación específica dentro de una página se pueden crear utilizando la etiqueta **<a>** y especificando el ID del destino en el atributo “href”, utilizando el símbolo #.

51. ¿Cómo se pueden crear cajas de texto para ingresar datos en una página web utilizando HTML?

Las cajas de texto para ingresar datos en una página web se pueden crear utilizando la etiqueta **<input>** y especificando el “type” como “text”.

52. ¿Qué es el atributo “maxlength” en HTML?

El atributo “maxlength” en HTML es utilizado para establecer un límite en la cantidad de caracteres que pueden ser ingresados en un campo de formulario, como una caja de texto.

53. ¿Qué es el atributo “pattern” en HTML?

El atributo “pattern” en HTML se utiliza para especificar un patrón de validación para el campo de entrada. El valor debe ser una expresión regular.

54. ¿Qué es el atributo “autofocus” en HTML?

El atributo “autofocus” en HTML es utilizado para indicar al navegador que un campo de formulario debe recibir automáticamente el focus cuando la página se carga.

55. ¿Qué es la etiqueta **<audio>** en HTML?

La etiqueta **<audio>** en HTML se utiliza para insertar un archivo de audio en una página web y proporciona controles de reproducción para el usuario.

56. ¿Qué es la etiqueta **<video>** en HTML?

La etiqueta **<video>** en HTML se utiliza para insertar un archivo de video en una página web y proporciona controles de reproducción para el usuario.

57. ¿Qué es la etiqueta **<source>** en HTML?

La etiqueta **<source>** en HTML se utiliza para proporcionar varias fuentes para un elemento multimedia, como **<audio>** o **<video>**, permitiendo que el navegador elija la mejor opción según el tipo de archivo y la compatibilidad del navegador.

58. ¿Qué es la etiqueta **<track>** en HTML?

La etiqueta **<track>** en HTML se utiliza para añadir subtítulos o descripciones de audio a elementos multimedia, como **<audio>**

59. ¿Qué es la etiqueta **<template>** en HTML?

La etiqueta **<template>** en HTML se utiliza para contener contenido que debe ser utilizado como plantilla y no se debe mostrar en la página hasta que sea activada mediante JavaScript.

60. ¿Qué es la etiqueta **<canvas>** en HTML?

La etiqueta **<canvas>** en HTML se utiliza para crear un área en la que se pueden dibujar gráficos mediante scripts, como JavaScript.

61. ¿Qué es la etiqueta **<svg>** en HTML?

La etiqueta **<svg>** en HTML se utiliza para insertar gráficos vectoriales en una página web. El contenido de esta etiqueta está escrito en un lenguaje específico, SVG (Scalable Vector Graphics), en lugar de una imagen raster.

62. ¿Qué es la etiqueta **<data>** en HTML?

La etiqueta **<data>** en HTML se utiliza para especificar un contenido adicional para un elemento, el cual estará disponible mediante JavaScript pero no se mostrará en el navegador.

63. ¿Cómo se pueden crear formularios de búsqueda en una página web utilizando HTML?

Los formularios de búsqueda se pueden crear en una página web utilizando la etiqueta **<form>** y etiqueta **<input>** con el atributo “type” establecido en “search” .

64. ¿Cómo se pueden crear campos de contraseña en una página web utilizando HTML?

Los campos de contraseña se pueden crear en una página web utilizando la etiqueta **<input>** con el atributo “type” establecido en “password” .

65. ¿Qué es el atributo “placeholder” en HTML?

El atributo “placeholder” en HTML se utiliza para proporcionar una sugerencia de texto a un campo de formulario antes de que el usuario haya ingresado algún valor.

66. ¿Qué es el atributo “required” en HTML?

El atributo “required” en HTML se utiliza para indicar que un campo de formulario es obligatorio y debe ser completado antes de enviar el formulario.

67. ¿Qué es el atributo “disabled” en HTML?

El atributo “disabled” en HTML se utiliza para deshabilitar temporalmente un elemento de formulario, impidiendo que el usuario interactúe con él.

68. ¿Qué es el atributo “readonly” en HTML?

El atributo “readonly” en HTML se utiliza para hacer un campo de formulario solo de lectura, impidiendo que el usuario lo modifique pero permitiendo que su valor sea enviado al servidor cuando se envía el formulario.

69. ¿Qué es el atributo “multiple” en HTML?

El atributo “multiple” en HTML se utiliza para permitir que un usuario seleccione varias opciones en una lista desplegable o campo de archivos.

70. ¿Cómo se pueden crear botones en una página web utilizando HTML?

Los botones se pueden crear en una página web utilizando la etiqueta **<button>** o la etiqueta **<input>** con el atributo “type” establecido en “button” o “submit” .

71. ¿Qué es el atributo “form” en HTML?

El atributo “form” en HTML se utiliza para especificar a qué formulario pertenece un elemento, cuando este se encuentra fuera de la estructura del formulario pero aún así debe ser enviado con él.

72. ¿Qué es el atributo “autocomplete” en HTML?

El atributo “autocomplete” en HTML se utiliza para indicar al navegador si debe o no autocompletar automáticamente el valor de un campo de formulario con un valor previamente ingresado.

73. ¿Qué es el atributo “novalidate” en HTML?

El atributo “novalidate” en HTML se utiliza para indicar al navegador que no debe validar automáticamente el formulario antes de enviarlo.

74. ¿Qué es el atributo “formaction” en HTML?

El atributo “formaction” en HTML se utiliza para especificar la URL a la cual se debe enviar el formulario cuando se presiona un botón específico.

75. ¿Qué es el atributo “formenctype” en HTML?

El atributo “formenctype” en HTML se utiliza para especificar el tipo de codificación utilizado para enviar los datos del formulario al servidor.

76. ¿Qué es el atributo “formmethod” en HTML?

El atributo “formmethod” en HTML se utiliza para especificar el método HTTP utilizado para enviar los datos del formulario al servidor.

77. ¿Qué es el atributo “formnovalidate” en HTML?

El atributo “formnovalidate” en HTML se utiliza para indicar al navegador que no debe validar automáticamente el formulario cuando se presiona un botón específico.

78. ¿Qué es el atributo “formtarget” en HTML?

El atributo “formtarget” en HTML se utiliza para especificar en qué ventana o marco se debe mostrar el resultado de enviar el formulario.

79. ¿Cómo se pueden crear menús desplegables en una página web utilizando HTML?

Los menús desplegables se pueden crear en una página web utilizando la etiqueta **<select>** junto con las etiquetas **<option>** para cada opción en el menú.

80. ¿Qué es el atributo “size” en HTML?

El atributo “size” en HTML se utiliza para especificar el número de filas que se deben mostrar en una lista desplegable o en un área de texto.

81. ¿Qué es el atributo “optgroup” en HTML?

El atributo “optgroup” en HTML se utiliza para agrupar las opciones en un menú desplegable, permitiendo organizar las opciones de manera lógica y fácil de navegar para el usuario.

82. ¿Qué es el atributo “datalist” en HTML?

El atributo “datalist” en HTML se utiliza para especificar una lista de opciones predefinidas para un campo de formulario, que el navegador puede utilizar para sugerir posibles valores al usuario.

83. ¿Qué es el atributo “challenge” en HTML?

El atributo “challenge” en HTML se utiliza para proporcionar un desafío para un campo de contraseña para prevenir ataques de fuerza bruta automatizada.

84. ¿Qué es el atributo “keytype” en HTML?

El atributo “keytype” en HTML se utiliza para especificar el tipo de teclado virtual que se debe mostrar en un campo de formulario móvil.

85. ¿Qué es el atributo “incremental” en HTML?

El atributo “incremental” en HTML se utiliza para indicar al navegador que debe ir guardando automáticamente un campo de formulario a medida que el usuario ingresa valores.

86. ¿Cómo se pueden crear tablas en una página web utilizando HTML?

Las tablas se pueden crear en una página web utilizando la etiqueta **<table>**, la etiqueta **<tr>** para cada fila, la etiqueta **<th>** para encabezados de columna y la etiqueta **<td>** para datos en cada celda.

87. ¿Qué es el atributo “scope” en HTML?

El atributo “scope” en HTML se utiliza para especificar el alcance de una celda de encabezado en una tabla, ya sea para toda la columna, fila o grupo de celdas relacionadas.

88. ¿Qué es el atributo “colspan” en HTML?

El atributo “colspan” en HTML se utiliza para especificar cuántas columnas una celda debe extenderse en una tabla.

89. ¿Qué es el atributo “rowspan” en HTML?

El atributo “rowspan” en HTML se utiliza para especificar cuántas filas una celda debe extenderse en una tabla.

90. ¿Cómo se pueden crear imágenes en una página web utilizando HTML?

Las imágenes se pueden crear en una página web utilizando la etiqueta **** y especificando la ruta del archivo de imagen en el atributo “src” .

91. ¿Qué es el atributo “alt” en HTML?

El atributo “alt” en HTML se utiliza para proporcionar una descripción alternativa de una imagen, que se mostrará si la imagen no se puede cargar o si el usuario utiliza un lector de pantalla.

92. ¿Qué es el atributo “usemap” en HTML?

El atributo “usemap” en HTML se utiliza para especificar un mapa de imagen asociado a una imagen, permitiendo que el usuario haga clic en diferentes áreas de la imagen para activar enlaces.

93. ¿Qué es el atributo “ismap” en HTML?

El atributo “ismap” en HTML se utiliza para indicar que una imagen es un mapa de imagen del lado del cliente, en lugar de un mapa de imagen del lado del servidor.

94. ¿Cómo se pueden crear listas en una página web utilizando HTML?

Las listas se pueden crear en una página web utilizando las etiquetas **** (lista no ordenada) o **** (lista ordenada) junto con la etiqueta **** para cada elemento en la lista.

95. ¿Qué es el atributo “type” en HTML?

El atributo “type” en HTML se utiliza para especificar el tipo de elemento, como “checkbox” o “text” para un campo de formulario, “bullet” o “number” para una lista, o “image” para un botón de envío.

96. ¿Qué es el atributo “value” en HTML?

El atributo “value” en HTML se utiliza para especificar el valor predeterminado de un elemento de formulario, como un campo de texto o una casilla de verificación.

97. ¿Qué es el atributo “label” en HTML?

El atributo “label” en HTML se utiliza para asociar una etiqueta de texto con un elemento de formulario, proporcionando una descripción para el elemento.

98. ¿Cómo se pueden crear enlaces en una página web utilizando HTML?

Los enlaces se pueden crear en una página web utilizando la etiqueta `<a>` y especificando la URL del enlace en el atributo `“href”`.

99. ¿Qué es el atributo `“target”` en HTML?

El atributo `“target”` en HTML se utiliza para especificar en qué ventana o marco se debe abrir un enlace. El valor predeterminado es `“_self”`, que abre el enlace en la misma ventana o marco. Otros valores comunes incluyen `“_blank”` para abrir el enlace en una nueva ventana o marco.