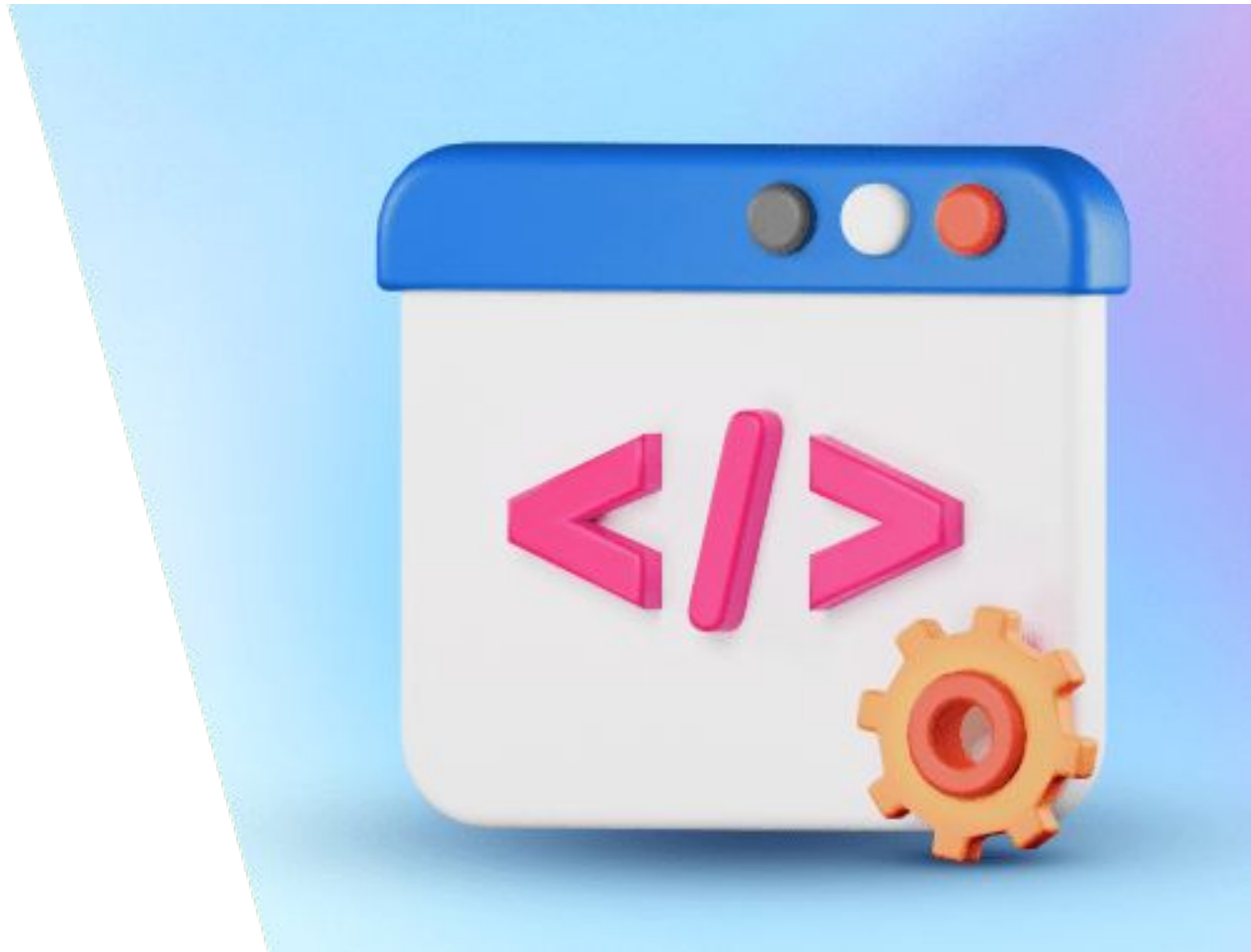


# Estructura y Etiquetas en HTML

HyperText Markup Language



# HTML

HTML significa HyperText Markup Language (Lenguaje de Marcado de Hipertexto) y es el lenguaje estándar para crear páginas web. No es un lenguaje de programación (como Python o JavaScript), sino un lenguaje de marcado, es decir, se utiliza para dar estructura y organizar el contenido en la web.

<https://developer.mozilla.org/en-US/docs/Web/HTML>

## ¿Qué es HTML?

- HTML significa HyperText Markup Language (Lenguaje de Marcado de Hipertexto) y es el lenguaje estándar para crear páginas web.
- HTML **no es un lenguaje de programación** (como Python o JavaScript) no tiene estructuras de control (selectivas, de iteración...) y no te permite utilizar estructuras de datos y/o realizar cálculos.
- Es un lenguaje de maquetado, es decir, se utiliza para dar estructura y organizar el contenido en la web.

## ¿Qué hace HTML?

- Estructura el contenido: textos, imágenes, tablas, enlaces, formularios, etc.
- Le dice al navegador qué debe mostrar y cómo están organizados los elementos.
- Funciona junto con CSS (estilo visual) y JavaScript (interactividad) para crear páginas web completas.
- Funciona en base a etiquetas encerradas entre <>

# **<HTML> TAGS**



# etiquetas en HTML(Tags)

1. El método principal para marcar contenido usando HTML son las etiquetas. Las etiquetas usan la sintaxis

`<tagname>contenido</tagname>.`

# etiquetas en HTML(Tags)

Por ejemplo, los párrafos usan la etiqueta p, así que podríamos crear el párrafo "Hello World" con:

```
<p>Hello World</p>.
```

La etiqueta del lado izquierdo, como <p>, se conoce como etiqueta de apertura.

La etiqueta del lado derecho, como </p>, se conoce como etiqueta de cierre.



# etiquetas en HTML(Tags)

Todo lo que está dentro de las etiquetas es el contenido, que es lo que realmente se mostrará en la página.

La combinación de la etiqueta de apertura, el contenido y la etiqueta de cierre se conoce como un **elemento completo**.

# tipos de etiquetas en HTML

1. Etiquetas que se abren y cierran (etiquetas en pares) Estas etiquetas tienen una etiqueta de apertura y una de cierre, y entre ellas se coloca el contenido que afectarán.

```
<etiqueta>Contenido</etiqueta>
```

# Ejemplos:

Etiqueta	Descripción	Ejemplo
<p>	Párrafo	<p>Hola mundo</p>
<h1> a <h6>	Encabezados	<h1>Título</h1>
<div>	Contenedor en bloque	<div>Contenido</div>
<span>	Contenedor en línea	<span>Texto</span>
<a>	Enlace	<a href="url">Ir</a>
<strong>	Texto en negrita	<strong>Importante</strong>
<em>	Texto en cursiva	<em>Énfasis</em>

Algunas etiquetas no tienen contenido, por lo que las llamamos etiquetas vacías o etiquetas autocontenidas.

Estas pueden usar solo la sintaxis de apertura o bien:


`<tagname />`.

Por ejemplo, las líneas horizontales usan la etiqueta `hr`, así que podemos crear una línea horizontal con:

`<hr>` o `<hr />`

2. Etiquetas auto-cerradas (autocontenidas o vacías) estas etiquetas no necesitan una etiqueta de cierre. Se utilizan solas, y normalmente insertan contenido o generan una acción visual directa.

```
<etiqueta/>
```

 En HTML5, el / al final no es obligatorio, pero puede usarse por compatibilidad con XHTML.

# Ejemplos:

Etiqueta	Descripción	Ejemplo
<img>	Inserta una imagen	
 	Salto de línea	Hola Mundo
<hr>	Línea horizontal	<hr>
<input>	Campo de entrada (formulario)	<input type="text" placeholder="Nombre">
<meta>	Metadatos (en <head>)	<meta charset="UTF-8">
<link>	Vincula hojas de estilo	<link rel="stylesheet" href="style.css">

## INDENTACIÓN

Profundidad del nivel de texto para facilitar lectura



## COMENTARIOS

Anotaciones dirigidas al programador para aclarar conceptos

# Indentación en HTML

No es obligatorio indentar el código HTML, pero sí es altamente recomendable.

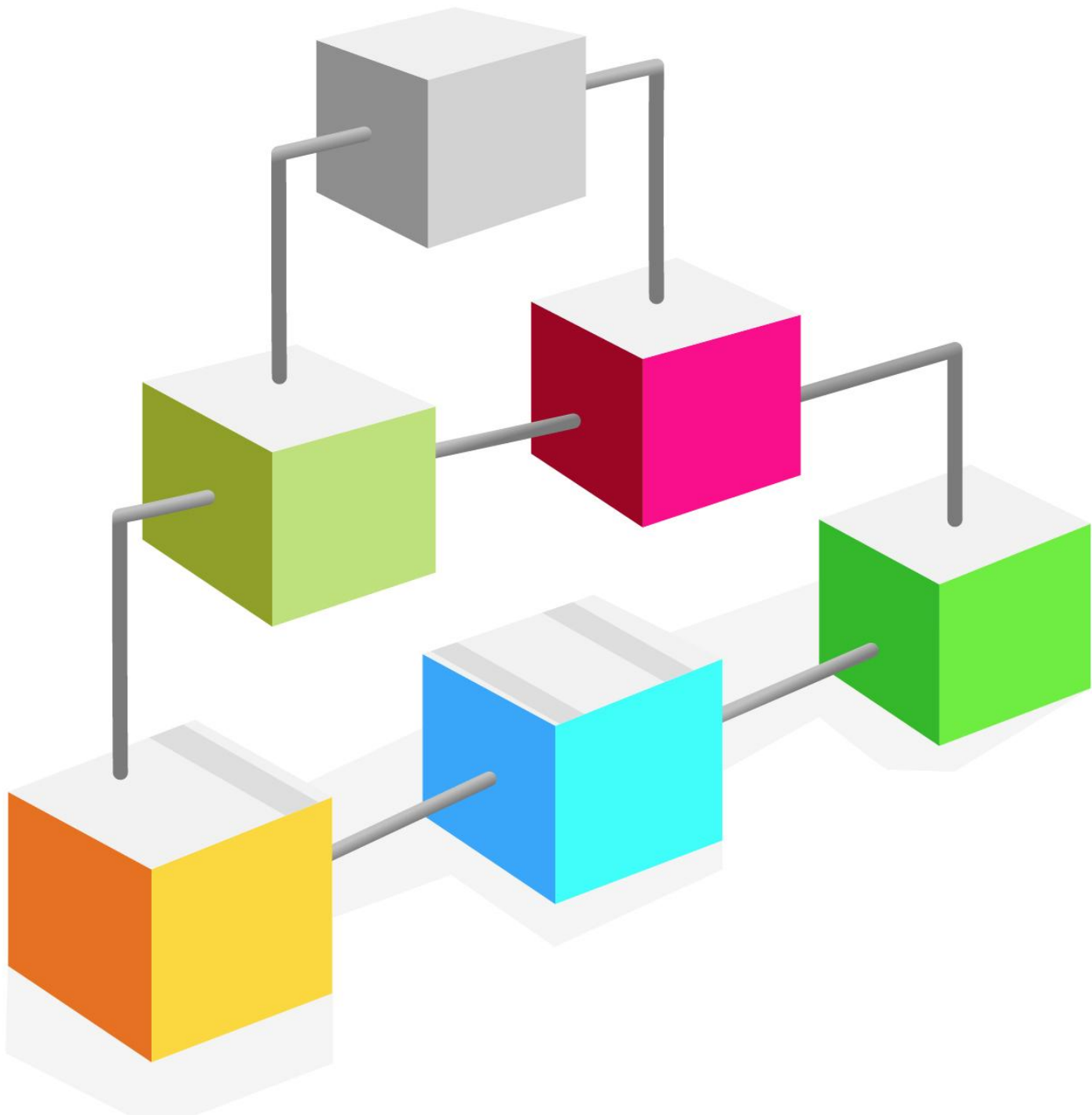
```
<html>  
  <head>  
    <title>Mi Página</title>  
  </head>  
  <body>  
    <h1>Hola Mundo</h1>  
    <p>Bienvenido a HTML</p>  
  </body>  
</html>
```



## ¿Qué pasa si no lo haces?

El navegador interpretará igual el HTML, aunque no esté indentado, pero será poco legible para las personas que puedan continuar o mantener este código (aun para el autor):

```
<html><head><title>Mi Página</title></head>  
  <body> <h1>Hola Mundo</h1><p>Bienvenido a  
HTML</p>  
  </body></html>
```



# Estructura Básica código HTML

```
<!DOCTYPE html>  
<html>  
  <head>  
    <title>Título</title>  
  </head>  
  <body>  
    <!-- Contenido visible -->  
  </body>  
</html>
```

**!DOCTYPE**  
**HTML**  
HYPERTEXT MARKUP LANGUAGE

# <!DOCTYPE html>




La etiqueta <!DOCTYPE html> es una declaración de tipo de documento, y se usa al inicio de un archivo HTML.

Es una instrucción para el navegador que le indica qué versión de HTML se está utilizando. En este caso:

**<!DOCTYPE html>**

Declara que el documento está escrito en HTML5, la versión actual y más usada.

## ¿Para qué sirve?

-  Activa el “modo estándar” en los navegadores, para que interpreten el HTML correctamente y no en “modo quirks” (modo de compatibilidad con versiones antiguas).
-  Evita problemas de diseño y renderizado.
-  No es una etiqueta HTML, sino una declaración especial reconocida por el navegador.

## Declaración en versiones anteriores a HTML5

La etiqueta `<!DOCTYPE html>` es una declaración de tipo de documento, y se usa al inicio de un archivo HTML.

### HTML 4.01 Transitional:

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

### HTML 1.0:

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

### HTML 4.01 Strict:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

## ¿Se puede omitir?

No se recomienda omitir, incluso en versiones antiguas:


- Si no se declara `<!DOCTYPE>`, el navegador puede entrar en modo “quirks” (modo antiguo), y el comportamiento visual del sitio puede ser impredecible o incorrecto.
- La declaración es necesaria para que el navegador sepa cómo interpretar correctamente el HTML.





# <html>

La etiqueta <html> define el inicio del documento HTML. Es la raíz de todo el contenido que estará estructurado dentro de la página web.

 Función principal. Indica al navegador que todo el contenido entre <html> y </html> forma parte del documento HTML. Dentro de ella deben estar exactamente dos secciones:

**<head>** Información sobre la página (metadatos, enlaces a CSS, título, etc.)

**<body>** Contenido visible por el usuario (texto, imágenes, botones, etc.)



## Características:

- Obligatoria en HTML: todo documento debe incluirla.
- Puede incluir un atributo como lang para indicar el idioma:

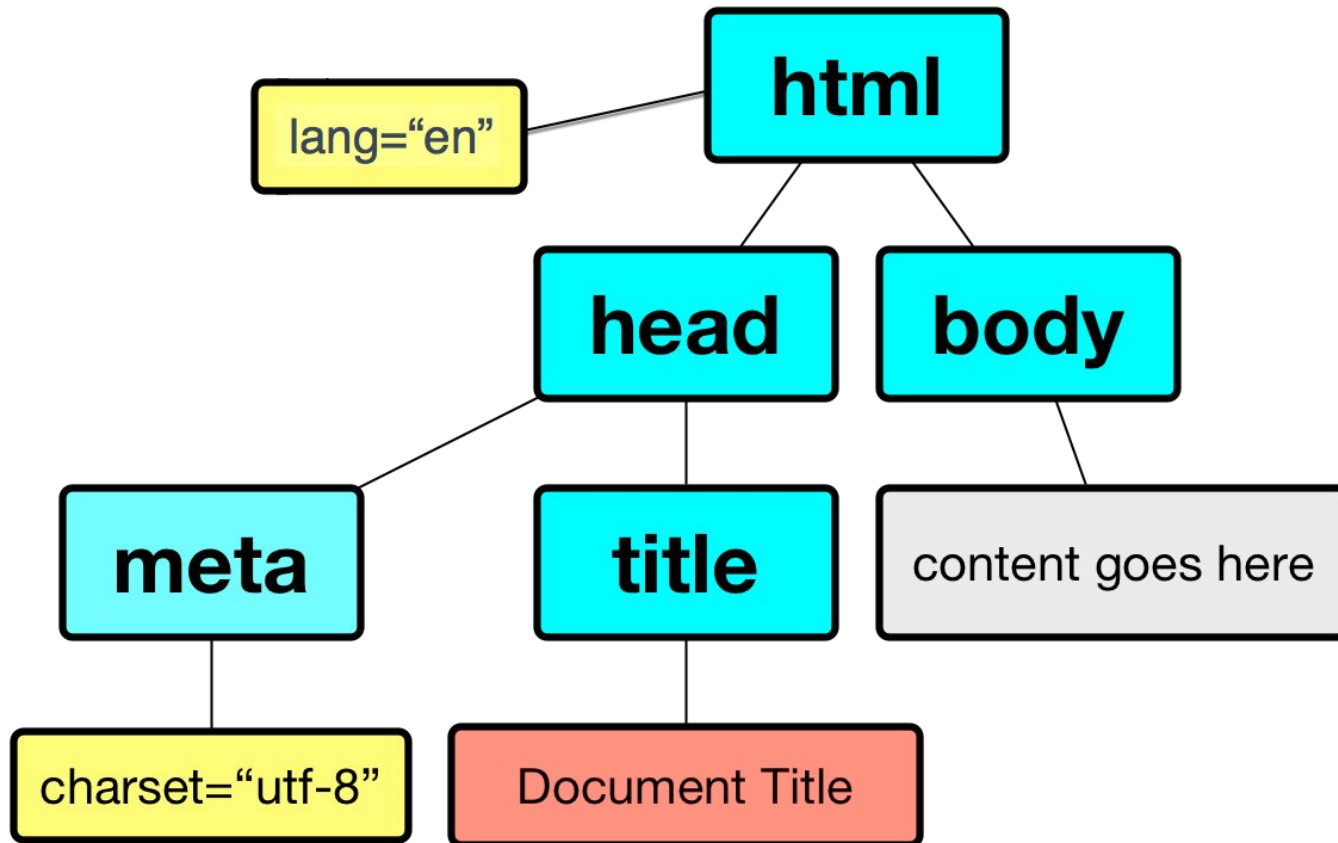
```
<html lang="es">
```

Esto es útil para accesibilidad y para que los motores de búsqueda y lectores de pantalla interpreten correctamente el idioma del contenido.



## Conclusión:

- `<html>` es la etiqueta contenedora principal de un documento HTML.
- Todo el código HTML debe estar dentro de esta etiqueta.
- Es un requisito esencial para que el navegador interprete correctamente la estructura del sitio web.



# <head>

La etiqueta <head> se utiliza para contener información "invisible" que describe y configura la página web. Esta información no se muestra directamente en el navegador, pero es crucial para el funcionamiento, estilo, posicionamiento y comportamiento de la página.

La etiqueta contiene metadatos sobre el documento HTML (es decir, cualquier cosa que no se muestra en la página).

El <head> del archivo HTML debe ser la primera etiqueta dentro de la etiqueta <html>.



# ¿Qué contiene `<head>`?

Etiqueta	Función
<title>	Define el título que aparece en la pestaña del navegador
<meta>	Metadatos como codificación, descripción, autor, etc.
<link>	Enlace a archivos externos, como hojas de estilo CSS
<style>	Estilos CSS internos
<script>	Código JavaScript o enlaces a archivos JS externos
<base>	Establece una URL base para enlaces relativos

## ¿Por qué es importante?

- ✓ Proporciona información al navegador, motores de búsqueda y dispositivos.
- ✓ Mejora la accesibilidad y el posicionamiento SEO.
- ✓ Permite vincular archivos externos como hojas de estilo o scripts.

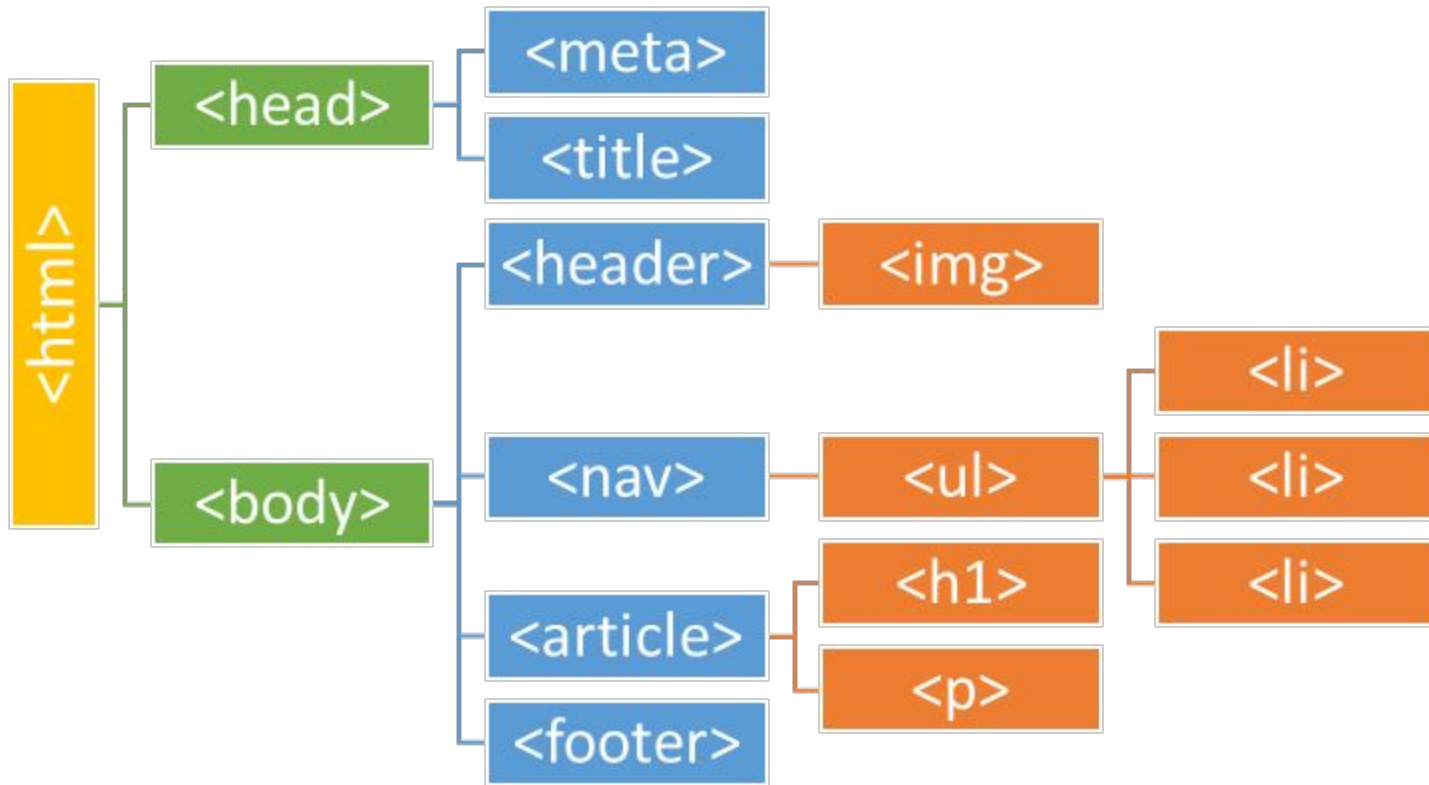
La etiqueta `<title>` es la única requerida dentro de `<head>`.

Esta etiqueta le da un título a la página, que se utiliza como el nombre de la pestaña en la mayoría de los navegadores y también se usa en los resultados de búsqueda de muchos motores de búsqueda.



## Conclusión

- ✓ La etiqueta <head> sirve como cabecera informativa del documento HTML.
- ✓ Aunque no contiene contenido visible, controla muchas funciones clave de la página web, por lo que es indispensable.



# ¿Qué es la etiqueta `<body>`?

La etiqueta `<body>` representa el cuerpo del documento HTML. Todo lo que se quiere mostrar al usuario en el navegador —como texto, imágenes, botones, enlaces, formularios, videos, etc.— debe estar dentro de esta etiqueta.

La etiqueta que contiene el contenido de la página web.

El `<body>` del archivo HTML debe ser la segunda etiqueta dentro de `<html>`, justo debajo de `<head>`.



## ¿Qué puede contener `<body>`?



Etiqueta	Propósito
<code>&lt;h1&gt;</code> a <code>&lt;h6&gt;</code>	Encabezados de distintos niveles
<code>&lt;p&gt;</code>	Párrafos de texto
<code>&lt;img&gt;</code>	Imágenes
<code>&lt;a&gt;</code>	Enlaces
<code>&lt;ul&gt;</code> , <code>&lt;ol&gt;</code>	Listas no ordenadas y ordenadas
<code>&lt;table&gt;</code>	Tablas
<code>&lt;div&gt;</code> , <code>&lt;span&gt;</code>	Contenedores para estructurar contenido
<code>&lt;form&gt;</code>	Formularios interactivos
<code>&lt;button&gt;</code>	Botones








# ¿Qué puede pasar si el HTML tiene errores?

Cuando un documento HTML tiene errores de sintaxis o de estructura, el navegador intentará interpretar el contenido de todas formas, pero pueden ocurrir varios problemas dependiendo de la gravedad del error.

1.  El navegador "adivina" lo que quisiste hacer. Los navegadores modernos son tolerantes a errores. Intentan "reparar" el HTML mal escrito aplicando su propio algoritmo de corrección. A veces esto funciona bien, pero otras veces causa comportamientos inesperados.
2.  El diseño de la página se rompe. Si etiquetas están mal cerradas o anidadas incorrectamente, los estilos CSS pueden aplicarse mal o no aplicarse.

3.  Problemas de accesibilidad y SEO. Herramientas de accesibilidad (como lectores de pantalla) podrían interpretar mal el contenido. Motores de búsqueda (como Google) podrían ignorar o malinterpretar parte del contenido.
4.  Errores al integrar con JavaScript o CSS. Si el HTML no está bien formado, JavaScript puede no encontrar elementos esperados con `getElementById()` o `querySelector()`. Estilos definidos en CSS podrían no aplicarse si hay errores en el DOM.
5.  Errores de validación. Si usas validadores como el W3C Markup Validator, se mostrará una lista de errores y advertencias que deben corregirse para cumplir los estándares web.



# BUENAS PRÁCTICAS





# Buenas prácticas para evitar errores

- ✓ Indentación clara y anidado correcto.
- ✓ Cierre de todas las etiquetas (incluso si no es obligatorio en HTML5).
- ✓ Validar el código usando herramientas como: W3C Validator
- ✓ Usar editores modernos (Visual Studio Code, Sublime, etc.) con resaltado de errores

## Actividad 3

Investiga y realiza un listado con ejemplos sobre las diferentes formas de darle formato a un documento HTML sin usar CSS.

Puntos a considerar:

Cómo poner negritas, cursivas o texto subrayado.

Cómo insertar saltos de línea o espacios.

Cómo mostrar texto en código o con distinto tamaño.

Otras etiquetas de formato que encuentres.