**🌟 Introducción sencilla a HTML: ¿Qué es?**

Imagina que estás construyendo una casa 🏠:

* HTML es la estructura básica de la casa (paredes, puertas, ventanas).
* CSS (que aprenderás más adelante) sería la decoración y pintura.
* JavaScript o TypeScript (que ya conoces) sería la electricidad, las cañerías, y todo lo que hace que la casa funcione o "haga cosas".

En resumen, HTML le dice al navegador cómo es la estructura y el contenido de tu página web.

----------------------------------------------------------------------------------------------------------------------------------------------

**📌 ¿Cómo funciona HTML?**

HTML significa HyperText Markup Language o Lenguaje de Marcas de Hipertexto. (llamado lenguaje de etiquetas)

Es decir:

* HyperText porque es texto con enlaces que llevan a otros textos o páginas.
* Markup porque usamos etiquetas (<etiqueta>) para decirle al navegador qué tipo de contenido estamos mostrando.

----------------------------------------------------------------------------------------------------------------------------------------------

**🧱 Las etiquetas HTML: El "bloque básico"**

Las etiquetas son como cajas que contienen información.

Ejemplo simple:



• <p>: es la etiqueta que significa párrafo (como un bloque de texto).

• El texto "Hola, mundo!" es lo que ve el usuario.

• </p> indica el cierre de esa etiqueta o caja.

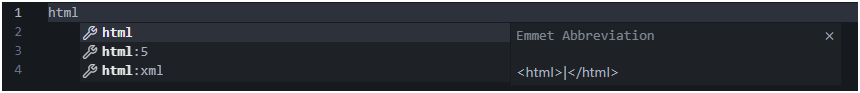
Analogía:

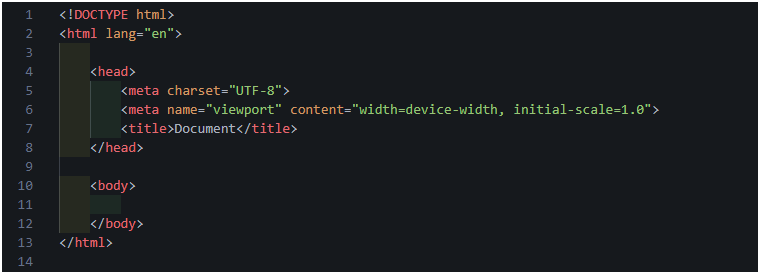
* Imagina que etiquetas HTML son como cajones con etiquetas que indican qué guardas dentro (ropa, zapatos, libros).
* El navegador abre cada etiqueta y presenta el contenido según corresponda.

----------------------------------------------------------------------------------------------------------------------------------------------

**⚙️ Estructura básica de un documento HTML**

Todo documento HTML tiene esta estructura mínima:

* al escribir “html” en visual studio code aparece el acceso a autocompletar esta estructura, elegir **html:5**
* Se genera la siguiente estructura: estructura inicial vacía.



Piensa en head como la parte interna del libro (índice, datos técnicos) y en body como el contenido real (capítulos e imágenes). Si TypeScript tuviera un programa, el <body> sería como el console.log () que muestra algo al usuario, mientras que <head> sería como las configuraciones iniciales que no se ven directamente.

*Aquí va una explicación sencilla, línea por línea, del código:*

Indica al navegador qué versión de HTML estás usando. En este caso, HTML5 (la más reciente y usada).

Es la etiqueta raíz que indica el inicio del documento HTML completo. El atributo lang="en" indica que el idioma del contenido es inglés (en). Si fuese español pondríamos lang="es".



Define el encabezado del documento. Es donde se colocan metadatos (información sobre la página que no es visible directamente al usuario).

Define el conjunto de caracteres que la página va a utilizar.  
UTF-8 es el más utilizado porque permite casi cualquier carácter, incluyendo letras acentuadas, emojis, caracteres internacionales, etc.



Esto hace que tu página sea responsive (que se vea bien en celulares, tablets y computadoras).

* width=device-width hace que el ancho de la página coincida con el ancho del dispositivo.
* initial-scale=1.0 indica que la página se cargará con zoom estándar (sin acercar o alejar).

Define el título de la pestaña en el navegador. Por ejemplo, aquí dice "Document", pero podrías colocar algo más descriptivo como "Mi primera página web".



Cierra la sección <head>.



Aquí comienza la parte visible de tu página web. Todo lo que coloques dentro de <body> será visible para el usuario.



Cierra la sección <body>.



Cierra la etiqueta raíz <html>. Marca el fin del documento HTML.

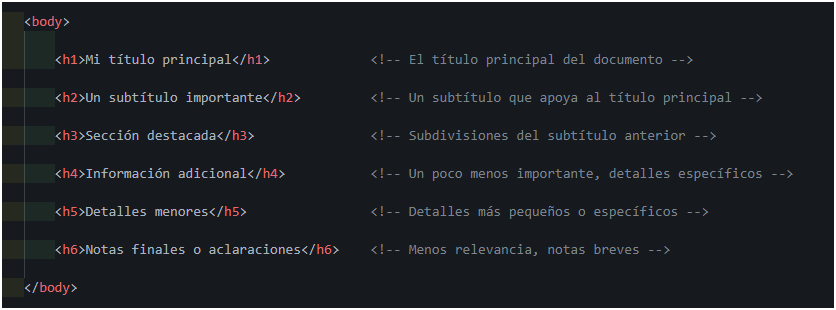
**🎯 Ejemplos de etiquetas básicas**

Vamos a presentarle algunas etiquetas básicas con ejemplos que pueda probar, recuerda lo que coloques dentro de **<body>** será visible para el usuario.

**Encabezados (Títulos) 🡺 (<h1> a <h6>):**

Son como encabezados en un documento de Word, de más grande (h1) a más pequeño (h6).

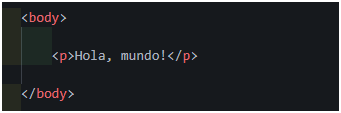
*Ejemplo:*



**Párrafos (<p>):**

Para texto normal, como el cuerpo de un correo.

Ejemplo:



**Enlaces (<a>):**

Como un “texto azul” que lleva a otro lugar.

Ejemplo:



* Href: es lo que dicta a dónde ir.

**Imágenes (<img>):**

No necesita cerrar la etiqueta, es como un return rápido en una función.

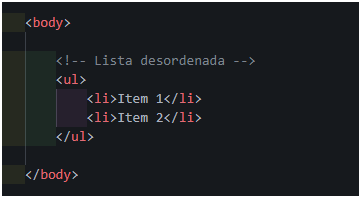
Ejemplo:



* **src** : es la URL de la imagen.
* **alt :** descripción breve si la imagen no se carga.

**Listas desordenadas (<ul>) y listas ordenadas (<ol>):**

***Lista desordenada (<ul>)***

******

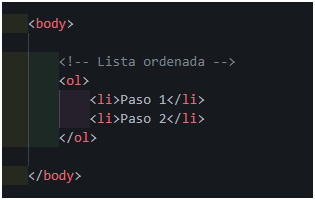
**<ul>**: Significa "unordered list" (lista desordenada). Esto crea una lista donde los elementos no tienen un orden numérico, y por defecto, el navegador les pone un puntito (bullet point) al inicio de cada elemento.

**<li>**: Significa "list ítem" (elemento de la lista). Cada <li> es un ítem dentro de la lista, como un elemento en un array.

**Qué hace**: Aquí estamos creando una lista con dos elementos: "Ítem 1" e "Ítem 2". En el navegador, se verá algo así:

* Ítem 1
* Ítem 2

**Ejemplo mental**: Imagina que estás haciendo una lista de compras en un cuaderno. La etiqueta <ul> es como decir "esta es mi lista de compras", y cada <li> es un producto que apuntas, como "leche" o "pan". El navegador automáticamente pone un puntito al lado de cada ítem.

***Lista ordenada (<ol>)***

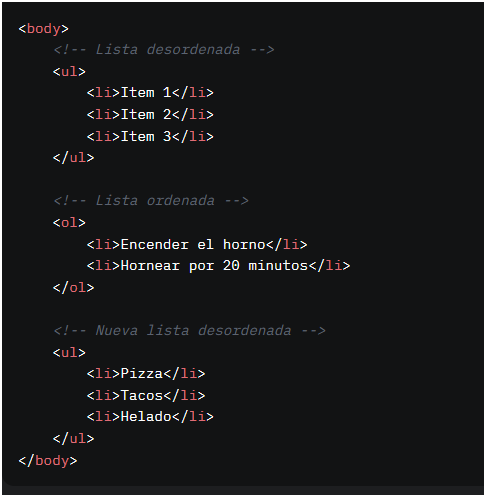
**<ol>:** Significa "ordered list" (lista ordenada). Esto crea una lista donde los elementos tienen un orden numérico, y el navegador automáticamente les pone números (1, 2, 3...) al inicio de cada elemento.

**<li>:** Igual que antes, cada <li> es un elemento de la lista.

**Qué hace**: Aquí estamos creando una lista con dos elementos: "Paso 1" y "Paso 2". En el navegador, se verá algo así:

1. Paso 1
2. Paso 2

**Ejemplo mental**: Imagina que estás escribiendo una receta con pasos numerados: "1. Mezclar los ingredientes, 2. Hornear". La etiqueta <ol> es como decir "esta es mi lista de pasos", y cada <li> es un paso numerado.



**Divisor (<div>):**

La etiqueta **<div>** (que viene de "división") es como una caja o contenedor invisible que usas para agrupar y organizar otros elementos en tu página web, como texto, imágenes, listas, etc. No tiene un significado especial por sí misma (como **<h1>** que indica un título o **<p>** que indica un párrafo), pero es súper útil para estructurar tu contenido.

Analogía con TypeScript: Piensa en un <div> como un objeto o una variable en TypeScript que puede contener otras cosas dentro. Por ejemplo, si tienes un objeto como

**Let**  **mensaje**:**string** = "Hola";

**Let** **edad**:**number** = 25;

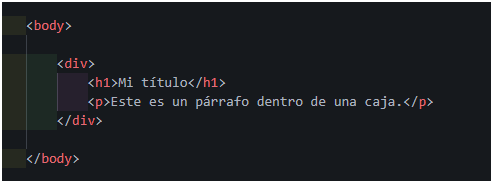
El **<div>** sería como esa caja que agrupa el texto y el número. En HTML, usas **<div>** para agrupar elementos y luego darles estilo (con CSS) o interactuar con ellos (con TypeScript/JavaScript).

**Ejemplo mental:** Imagina que estás organizando cosas en una mudanza. Un **<div>** es como una caja de cartón donde puedes meter diferentes cosas (juguetes, libros, ropa) para mantenerlas juntas y organizadas.

**¿Cómo se usa un <div>? Ejemplo básico**

La etiqueta **<div>** es muy simple: se abre con **<div>**, pones cosas dentro, y se cierra con **</div>**. Por sí sola, no cambia cómo se ve el contenido, pero es un "contenedor" que te permite trabajar con ese grupo de elementos más adelante.

**Ejemplo básico**:



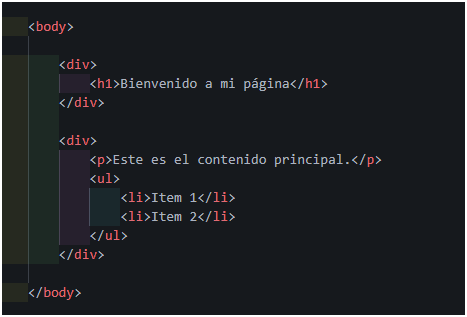
* Aquí, el **<div>** es como una caja que contiene un título (**<h1>**) y un párrafo (**<p>**).
* En el navegador, esto se verá igual que si no tuvieras el **<div>**, pero ahora tienes esos elementos agrupados para poder trabajar con ellos como un conjunto.

**¿Para qué sirve un <div>?**

El **<div>** es útil por dos razones principales:

1. **Organización**: Te ayuda a estructurar tu página dividiéndola en secciones lógicas.
2. **Estilización e interacción**: Más adelante, con CSS puedes darle estilo a un **<div>** (como cambiar su color, tamaño o posición), y con TypeScript/JavaScript puedes interactuar con él (como hacer que se oculte o muestre al hacer clic).

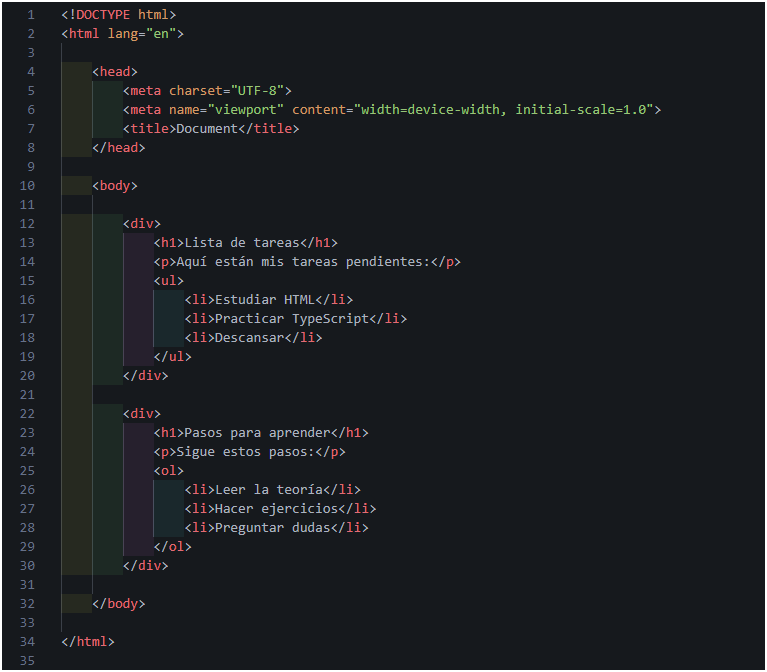
**Ejemplo práctico con organización**: Imagina que quieres dividir tu página en dos secciones: una para un encabezado y otra para el contenido principal.



* Aquí usamos dos **<div>** para separar el encabezado (**<h1>**) del contenido principal (**<p>** y **<ul>**).
* En el navegador, se verá igual que antes, pero ahora está organizado en "cajas" que puedes manipular por separado.

**Ejemplo práctico con <div>**

Vamos a crear un ejemplo más completo que combine <div> con las listas que ya vimos antes, para que tu amigo pueda probarlo.

**Ejemplo completo**:

**Qué hace este código**:

* Usamos dos **<div>** para separar dos secciones: una para las tareas pendientes y otra para los pasos de aprendizaje.
* Cada **<div>** contiene un título (**<h1>**), un párrafo (**<p>**), y una lista (**<ul>** o **<ol>**).
* En el navegador, se verá como dos bloques de contenido organizados, uno debajo del otro.

*En el navegador se debería ver de la siguiente forma:*

