

# HTML: El Arquitecto de la Web

## 1. ¿Qué es HTML? El Esqueleto de la Página

HTML (HyperText Markup Language - Lenguaje de Marcado de Hipertexto) no es un lenguaje de programación como PHP o JavaScript; es un **lenguaje de marcado**. Piensa en él como el **esqueleto** o la **estructura de metal y ladrillo** de cualquier edificio web.

HTML define:

- **La Estructura:** ¿Esto es un título? ¿Esto es un párrafo? ¿Esto es una lista de elementos?
- **El Contenido:** El texto, las imágenes, los enlaces y los videos que ves.

Es la primera capa que se carga en cualquier navegador. Sin HTML, solo tendrías datos sin organizar; con HTML, tienes una jerarquía legible y navegable.

### Usos Comunes de HTML:

- **Creación de la Estructura de Documentos:** Definir el cuerpo, encabezados, y pies de página de la web.
- **Gestión de Contenido Multimedia:** Insertar imágenes, videos y audio.
- **Creación de Enlaces (Hipertexto):** Conectar una página con otra, creando la "web" de documentos interconectados.
- **Recolección de Datos con Formularios:** Crear campos para que el usuario pueda introducir información (aunque el procesamiento lo haga PHP).

---

## 2. Sintaxis Básica: Las Etiquetas (Tags) y Elementos

La sintaxis de HTML se basa en **etiquetas** (Tags), las cuales envuelven el contenido para darle significado.

### 2.1. Etiquetas de Apertura y Cierre

La mayoría de las etiquetas funcionan en pares, como un sándwich que envuelve el contenido:

1. **Etiqueta de Apertura:** `<p>`
2. **Contenido:** Esto es un párrafo.
3. **Etiqueta de Cierre:** `</p>` (Siempre lleva una barra diagonal /).

El conjunto de la etiqueta de apertura, el contenido y la etiqueta de cierre se llama **Elemento**.

### 2.2. Elementos Vacíos (Self-Closing)

Algunas etiquetas no envuelven contenido, sino que insertan algo. Estas son **etiquetas de autocierre** o vacías.

Etiqueta	Propósito	Analogía
<code>&lt;br&gt;</code>	Salto de línea.	El "Enter" de la web.

Etiqueta	Propósito	Analogía
<code>&lt;hr&gt;</code>	Línea horizontal (separador temático).	Un separador de capítulo.
<code>&lt;img&gt;</code>	Inserta una imagen.	Colgar un cuadro en la pared.
<code>&lt;input&gt;</code>	Inserta un campo de formulario.	Un buzón o una casilla.

Ejemplo de Elementos:

```
<h1>Título Principal de la Página</h1>
```

```

<hr>
```

### 3. Atributos: Los Detalles y Personalizaciones

Los **atributos** son modificadores que se añaden a la etiqueta de apertura para proporcionar información adicional sobre el elemento, dándole detalles y personalizaciones.

**Sintaxis:** Se declaran dentro de la etiqueta de apertura, usando el formato `nombre="valor"`.

Atributo	Uso Común	Propósito	Analogía
<code>id</code>	Para identificar un elemento de forma <b>única</b> .	La <b>cédula de identidad</b> del elemento.	
<code>class</code>	Para agrupar varios elementos bajo un estilo común.	La <b>categoría o grupo</b> del elemento (ej. todos los botones).	
<code>href</code>	En la etiqueta <code>&lt;a&gt;</code> , especifica la URL de destino.	El <b>destino</b> del enlace.	
<code>src</code>	En etiquetas como <code>&lt;img&gt;</code> , especifica la fuente (ruta) del archivo.	La <b>dirección</b> donde encontrar el recurso.	

Ejemplo de Atributos:

```
<header id="cabecera-principal">

  <button class="btn-accion">Comprar</button>
  <button class="btn-accion">Añadir al carrito</button>

  <a href="https://google.com" target="_blank">Ir a Google (en nueva pestaña)</a>
</header>
```

### 4. Estructura del Documento: El Plan de Construcción

Todo documento HTML moderno sigue una estructura básica obligatoria para que el navegador sepa cómo interpretarlo.

#### 4.1. Declaración `<!DOCTYPE html>`

Es la primera línea. Le dice al navegador que el documento es de tipo HTML5 (la versión moderna).

## 4.2. El Elemento Raíz <html>

El contenedor principal de todo el documento. Contiene dos hijos obligatorios: <head> y <body>. Se recomienda añadir el atributo lang="es" para indicar el idioma.

## 4.3. <head>: La Cabeza (Metadatos)

Contiene la información **no visible** para el usuario, pero crítica para el navegador y los motores de búsqueda (Google). Aquí se carga el título, el CSS, el JavaScript y los metadatos.

## 4.4. <body>: El Cuerpo (Contenido Visible)

Contiene **todo el contenido visible** de la página web: títulos, párrafos, imágenes, enlaces, etc.

### Ejemplo de Estructura Completa:

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Mi Enciclopedia Web</title>
  <link rel="stylesheet" href="style.css">
</head>
<body>
  <h1>Bienvenido a la Web</h1>
  <p>La estructura está lista.</p>
</body>
</html>
```

## 5. Tipos de Elementos: En Bloque vs. En Línea

Los elementos HTML se clasifican en cómo se comportan y cómo interactúan con el espacio a su alrededor.

### 5.1. Elementos de Bloque (Block)

- **Comportamiento:** Ocupan **todo el ancho** disponible, creando un "bloque" propio, como si tuvieran un salto de línea antes y después.
- **Usos Comunes:** Estructura (títulos, párrafos, divisiones, listas).
- **Ejemplos:** <h1>, <p>, <div>, <ul>.

### 5.2. Elementos en Línea (Inline)

- **Comportamiento:** Solo ocupan el espacio que necesitan para su contenido y se mantienen en la misma línea, uno al lado del otro, como las palabras en una frase.
- **Usos Comunes:** Texto (enlaces, negritas, énfasis).
- **Ejemplos:** <a>, <span>, <strong>, <img>.

### Ejemplo de Diferencias:

<p>Soy un elemento de Bloque 1.</p>

<p>Soy un elemento de Bloque 2.</p>

<span>Soy un elemento en Línea 1.</span>

<span>Soy un elemento en Línea 2.</span>

## 6. Semántica: El Significado de las Etiquetas 🧠

Desde HTML5, la **Semántica** es fundamental. Consiste en elegir la etiqueta correcta que mejor describa el **significado** del contenido, no solo su apariencia.

- **Mal uso (No Semántico):** Usar un <div> (división genérica) para el menú de navegación.
- **Buen uso (Semántico):** Usar un <nav> para el menú de navegación.

### 6.1. Etiquetas Semánticas Clave

Etiqueta	Significado Semántico	Analogía
<header>	Contenido introductorio o de navegación.	El <b>encabezado</b> del edificio.
<nav>	Contenedor de enlaces de navegación.	El <b>mapa o índice</b> del sitio.
<main>	El contenido principal y <b>único</b> de la página.	El <b>corazón</b> de la casa.
<article>	Contenido independiente (como una noticia o <i>post</i> ).	Una <b>nota de prensa</b> completa.
<section>	Una sección temática dentro de un documento.	Un <b>capítulo</b> del libro.
<footer>	Información de pie de página (derechos de autor, contacto).	La <b>base</b> del edificio.

Ejemplo Semántico:

```
<body>
  <header>
    <h1>Título del Proyecto</h1>
    <nav>
      <ul>
        <li><a href="#inicio">Inicio</a></li>
        <li><a href="#acerca">Acerca de</a></li>
      </ul>
    </nav>
  </header>

  <main>
    <section>
      <h2>Capítulo Uno: Fundamentos</h2>
      <p>Aquí va el texto explicativo...</p>
    </section>
  </main>

  <footer>
    <p>&copy; 2025 Mi Enciclopedia Web</p>
  </footer>
</body>
```

## 7. Enlaces e Imágenes: La Conexión y la Decoración

### 7.1. Hipervínculos (<a>): El Conector

El elemento <a> (de *anchor*) es el motor de la web. Requiere el atributo `href` (Hipertexto Referencia) para indicar a dónde ir.

```
<a href="https://google.com">Buscar en Google</a>
```

```
<a href="contacto.html">Ir al formulario de contacto</a>
```

```
<a href="#seccion-5">Saltar a la Sección 5</a>
```

### 7.2. Imágenes (<img>): El Decorador

El elemento <img> es vacío (autocierre) y requiere dos atributos críticos:

- **src** (Source): La ruta del archivo de la imagen.
- **alt** (Texto Alternativo): Una descripción de la imagen. **Es crucial para la accesibilidad** (para lectores de pantalla) y para el SEO.

```

```

## 8. Formularios: La Interacción con PHP y MySQL

Los formularios son la forma principal en que los usuarios envían datos a la cocina (PHP). El elemento <form> es el contenedor de todos los campos de entrada.

### 8.1. El Elemento <form>: La Bandeja de Pedidos

El formulario necesita dos atributos principales:

1. **action**: Especifica a qué archivo de PHP se enviarán los datos (nuestro Chef).
2. **method**: Define cómo se envían los datos:
  - **GET**: Los datos se añaden a la URL (útil para búsquedas).
  - **POST**: Los datos se envían "por detrás" del servidor (invisible en la URL), ideal para datos sensibles (contraseñas, emails).

### 8.2. Campos de Entrada (<input>) y Etiquetado

Los datos se recogen con el elemento <input>. El atributo `type` define el tipo de campo (texto, email, contraseña, etc.), y el atributo **name** es crucial, ya que es la "llave" que PHP usará para recibir el dato (por ejemplo, `$_POST['nombre_usuario']`).

```
<form action="procesar.php" method="POST">
  <label for="campo-nombre">Tu Nombre:</label>

  <input type="text" id="campo-nombre" name="nombre_usuario" required>

  <br>

  <label for="campo-email">Tu Email:</label>
  <input type="email" id="campo-email" name="correo_usuario" required>

  <br>

  <input type="submit" value="Enviar Datos">
</form>
```