

# Unidad 2 - HTML

## Estructura del documento en HTML5

### ¿Qué es HTML?

- HTML (HyperText Markup Language) es un lenguaje de marcas que usa etiquetas para estructurar y referenciar distintos tipos de información en la web.

### Etiquetas HTML

- HTML ofrece etiquetas predefinidas para estructurar y dar contenido a una página web.
- Estructura básica → `<etiqueta>contenido</etiqueta>`

```
<p>Este texto contiene una  
<strong>palabra importante</strong>.</p>
```

### Semántica en HTML5

- La semántica implica proporcionar significado a los elementos HTML, separando:
  - **Estructura y contenido** (HTML5).
  - **Estilo visual** (CSS).
- HTML4 (no semántico):

```
<p>Resalto esta <b>palabra</b>.</p>
```

- HTML5 (semántico):

```
<p>Resalto esta <strong>palabra</strong>.</p>
```

- La etiqueta `<strong>` denota importancia, no estilo.

### Estructura y atributos de etiquetas

- **Apertura y cierre** → `<etiqueta>contenido</etiqueta>`

- **Atributos:**

- Pueden ser opcionales u obligatorios.
- Tipos:
  - **Conjunto de valores** → `<html lang="es">`
  - **Valores libres** → `<p id="valor_unico">Texto</p>`
  - **Booleanos** → `<input required>`
- Reglas comunes:
  - Nombres en **kebab case** (minúsculas y separados por guiones).
  - Evitar caracteres no estándar, como acentos o emojis.

- Atributos comunes:

- **id**: Identificador único, usar solo cuando sea necesario. No puede repetirse en un documento.
- **class**: Más flexible, permite aplicar estilos a varios elementos que comparten la misma clase.

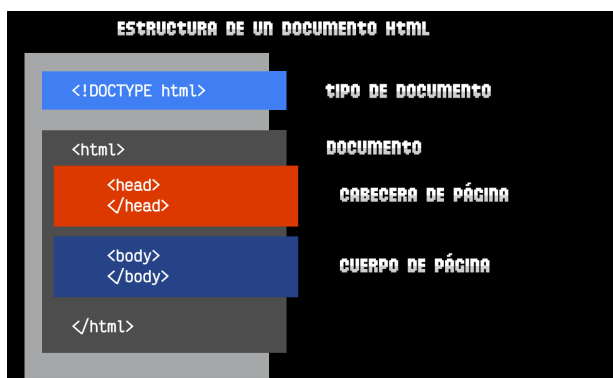
```
<div id = "pagina">
  <div class ="anuncio">Aquí irá un anuncio</div>
  <div id = "articulo">Aquí irá el contenido de texto del artículo</div>
  <div class = "anuncio">Aquí irá un anuncio</div>
</div>
```

## Comentarios en HTML

- Se utilizan para documentar el código. El navegador los ignora.

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

## Estructura del documento HTML



- `<!DOCTYPE html>` → Define el tipo de documento.
- `<html>` → Raíz del documento. Puede incluir atributos como `lang`.
- `<head>` → Contiene metadatos (información no visible). Incluye:
  - `<title>` → Título visible en la pestaña del navegador.
  - **Meta etiquetas:**
    - `<meta charset ="UTF-8">` → Indica al navegador la codificación a utilizar.
    - `<meta name = "description" content = "Free Web tutorials">` → Descripción breve del contenido
    - `<meta name = "keywords" content = "HTML, CSS, JavaScript">` → Palabras clave de la web
    - `<meta name = "author" content = "Autor de la web">`
    - `<meta name = "robots" content = "index , follow"/>`
      - `index` → Sugiere que se quiere indexar el contenido de la página en el buscador.
      - `noindex` → Indica que no debe indexar el contenido de la página en el buscador.

- `follow` → Sugiere que siga los enlaces que encuentre en la página.
- `nofollow` → Indica que no debe seguir los enlaces que encuentre en la página.
- **Recursos externos** → `<link rel="stylesheet" href="styles.css">`
- **Otras etiquetas**
  - `<base>` → Define una ruta base para enlaces relativos.
  - **Atributos:**
    - `href` : Ruta base.
    - `target` :
      - `self` (por defecto): Misma pestaña.
      - `blank` : Nueva pestaña.

```
<base href="https://misitio.root/" target="_blank">
```

- `<meta name="viewport">` → Optimiza la vista en dispositivos móviles.

- **Uso recomendado:**

```
<meta name="viewport" content="width=device-width, initial-scale=1">
```

- `width=device-width` : Ajusta al ancho del dispositivo.
- `initial-scale=1` : Sin zoom inicial.
- Para diseños fijos → `content="width=1024px"` .
- `user-scalable` → Controla el zoom (por defecto activado).
- `<body>` → Contiene el contenido visible.

```
<!DOCTYPE html>
<html lang="es">
  <head>
    <title>Mi Página Web</title>
    <meta charset="UTF-8">
  </head>
  <body>
    <p>Contenido de la página</p>
  </body>
</html>
```

## Validación del código HTML

- Es importante validar el código para garantizar que cumple los estándares y funciona correctamente. Utiliza herramientas como W3C Validator.

## Elementos semánticos

## Elementos Semánticos en HTML

- Los elementos semánticos facilitan la comprensión del contenido tanto para humanos como para máquinas, mejorando accesibilidad y SEO.

- `<article>` → Contenido independiente, como artículos o publicaciones.

```
<article>
  <h2>Título del Artículo</h2>
  <p>Contenido del artículo.</p>
</article>
```

- `<nav>` → Enlaces de navegación.

```
<nav>
  <ul>
    <li><a href="#">Inicio</a></li>
    <li><a href="#">Contacto</a></li>
  </ul>
</nav>
```

- `<header>` → Cabecera de un documento o sección.

```
<header>
  <h1>Bienvenidos</h1>
</header>
```

- `<footer>` → Pie de página con información adicional.

```
<footer>
  <p>&copy; 2024 Mi Sitio Web</p>
</footer>
```

- `<section>` → Agrupación temática de contenido.

```
<section>
  <h2>Sección de Noticias</h2>
  <p>Texto relevante.</p>
</section>
```

- `<main>` → Contenido principal de la página.

```
<main>
  <h2>Contenido Principal</h2>
</main>
```

- `<aside>` → Información complementaria.

```
<aside>
  <p>Barra lateral.</p>
</aside>
```

- `<figure>` → Imágenes o gráficos con leyendas.

```
<figure>
  
  <figcaption>Pie de imagen.</figcaption>
</figure>
```

## Elementos de texto

`<a>` → Fragmento de texto que enlaza a otra página diferente.

- `href` → Especifica la URL del recurso enlazado.
  - **Tipos de URL:**
    - Absoluta → Enlace completo (ej. `https://example.com`).
    - Relativa → Enlace dentro del mismo sitio.
    - Anclas → Fragmentos de texto identificados con `#id` para navegar dentro de la misma página.
    - **Scroll to text fragments** → Navega a un texto específico usando el formato `#::~text=<inicio>,<fin>`.

```
<a href="https:\\misitio.com">link normal</a>
<a href="#ancla">ancla</a>
```

- `download` → Descarga el recurso especificado en lugar de abrirlo.
  - Puedes asignar un nombre para el archivo descargado.
  - Solo funciona con recursos del mismo dominio.

```
<a href="/archivo.pdf" download="documento.pdf">Descargar</a>
```

- `target` → Especifica dónde abrir el enlace:
  - `_blank` → Nueva pestaña o ventana.
  - `_self` → En la pestaña o `<iframe>` actual.
  - `_parent` → En el marco padre inmediato.
  - `_top` → En el marco raíz.
- `hreflang` → Indica el idioma del recurso enlazado (código ISO 639-1).
- `ping` → Notifica a un servidor externo al seguir un enlace.
- `rel` → Define la relación del enlace con la página actual (ej. `nofollow`, `noopener`).

- `type` → Sugerencia del tipo de contenido enlazado (no obligatorio).

## Elementos de agrupación

### Agrupación de elementos

- `<div>` → Utilizado para agrupar y organizar elementos. Puede incluir atributos como `id` o `class`. Se recomienda evitar `<div>` con un solo elemento.

```
<div id="subpárrafo" class="párrafo">
  <h3>Título del párrafo</h3>
  <p>Mi primer párrafo</p>
  <div>
    <p>Información adicional <a href="#">link</a></p>
    <hr>
    <p>Otra información</p>
  </div>
</div>
```

- `<p>` → Define párrafos y puede contener enlaces.

### Listas

- **Listas ordenadas** `<ol>` → Incluyen numeración personalizable (`type`, `start`, `reversed`) y valores específicos con `value`.

```
<ol start="5" type="i" reversed>
  <li>Primer elemento</li>
  <li value="25">Cuarto ítem.</li>
</ol>
```

- **Listas no ordenadas** `<ul>` → Utilizan viñetas para los elementos.

```
<ul>
  <li>Elemento 1</li>
  <li>Elemento 2</li>
</ul>
```

- **Listas descriptivas** `<dl>` → Combina `<dt>` (término) y `<dd>` (descripción).

```
<dl>
  <dt>Gato</dt>
  <dd>Animal que dominó Internet.</dd>
</dl>
```

### Citas y referencias

- `<blockquote>` → Bloques de citas, con atributo opcional `cite`.
- `<q>` → Citas breves.

- `<cite>` → Referencias a títulos de obras.

## Formato de texto

### • Énfasis y relevancia

- `<strong>` → Resalta texto importante.
- `<em>` → Texto enfatizado.
- `<mark>` → Texto resaltado (simula rotulador).

```
<p><strong>Importante</strong> y <mark>marcado</mark>.</p>
```

### • Otras etiquetas

- `<s>` → Indica texto obsoleto.

```
<s>Precio antiguo: <u>3.99 €</u> Precio nuevo: 2€</s>
```

- `<span>` → Fragmento de texto genérico para aplicar estilos o scripts.

```
<p>Este <strong>párrafo</strong> de texto tiene <span>algunos fragmento  
texto</span>que mantienen su <span>disposición original</span>.</p>
```

### • Espaciado y saltos:

- `<br>` → Salto de línea.
- `<wbr>` → Permite dividir palabras largas para saltos.

## Elementos de Código

- `<pre>` y `<code>` → Formateo de texto/código respetando saltos y espacios.

```
<pre>
  <code>
    function saludo() {
      console.log("Hola Mundo");
    }
  </code>
</pre>
```

## Formularios

- `<form>` → Define la estructura de formularios. Requiere atributos:
  - `method` → GET o POST.
  - `action` → URL de envío.
  - `name` → Nombre del formulario.

```
<form action="procesar.php" method="post">
  <input type="text" name="usuario" placeholder="Nombre">
  <input type="submit" value="Enviar">
</form>
```

- **Tipos de campos:**

- **Atributos generales:**

- `value` → Indica un valor por defecto.
- `autocomplete="off"` → Autocompleta el input.
- `type` → Tipo de input.
- `placeholder` → Texto descriptivo.
- `name` → Nombre del input.
- `hidden` → Campo oculto al usuario.
- `spellcheck="false"` → Deshabilita la comprobación ortográfica.

```
<input type="text" value="David" placeholder="Nombre de usuario" autoco
```

- **Texto:**

- Texto alfanumérico libre (texto corto) → `<input type="text">`
- Texto para búsquedas → `<input type="search">`
- Número de teléfono → `<input type="tel">`
- Dirección URL → `<input type="url">`
- Dirección de correo email → `<input type="email">`
- Clave o contraseña → `<input type="password">`
- Campo oculto al usuario → `<input type="hidden">`

- **Números:**

- Números y caracteres permitidos → `<input type="number">`
- Números dentro de un rango → `<input type="range">`
  - `min` → Número mínimo.
  - `max` → Número máximo.
  - `step` → Valor que aumenta o disminuye del rango por cada interacción.

```
<input type="number" value="18" name="edad" />
<input type="range" name="donacion" value="20" min="5" max="200" step="
<output>25</output>
```

- **Fechas y horas:**

- Fecha → `<input type="date">`



- `min`, `max`, `step`
- Hora → `<input type="time">`
  - `min`, `max`, `step`
- Fecha y hora local del sistema del usuario → `<input type="datetime-local">`
- **Otros campos:**
  - Casilla de chequeo (booleano) → `<input type="checkbox">`
    - `checked` → Marcado por defecto
  - Casillas de chequeo con selección única → `<input type="radio">`
    - Todos los elementos tiene que tener el mismo atributo `name`.
    - `checked` → Marcado por defecto

```
<input type="checkbox" name="username" checked> Enviar publicidad al
<input type="radio" name="categoria" value="user" checked> Particula
<input type="radio" name="categoria" value="business"> Empresa
<input type="radio" name="categoria" value="business"> Otro
```

- Paleta de colores → `<input type="color">`
  - `value` → Color en hexadecimal
- Selector de archivos → `<input type="file">`
  - El atributo `enctype` de la etiqueta `<form>` toma el valor `multipart/form-data`.
  - `accept` → Indica el tipo de archivo que acepta.
    - `image/jpeg` → Imágenes JPEG.
    - `image/png` → Imágenes PNG.
    - `application/pdf` → Documentos PDF.
    - `application/zip` → Archivos comprimidos ZIP.
  - `multiple` → Carga múltiples archivos.

```
<form action="" method="get" enctype="multipart/form-data">
  <input type="file" name="foto_usr" id="" accept="image/*">
</form>
```

- **Selectores y listas:**
  - `<select>` → Desplegables con `<option>`. Soporta grupos con `<optgroup>` y selecciones múltiples.
  - `<datalist>` → Combina selección con entrada personalizada.

```
<input list="lenguajes" name="lenguaje">
<datalist id="lenguajes">
  <option value="HTML">
```

```
<option value="CSS">
</datalist>
```

- **Agrupación:**

- `<fieldset>` y `<legend>` → Agrupan campos relacionados.
- `<label>` → Asocia texto a un campo (permite activar el campo al hacer clic).

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

- **Botones:**

- `<input type="submit">` → Enviar formulario.
- `<input type="reset">` → Reiniciar formulario.
- `<input type="image">` → Envío de formulario con imagen
- Botón sin funcionalidad → `<input type="button">` o `<button>`

```
<input type="submit" value="Enviar formulario"/>
<input type="reset" value="Reset"/>
<input type="image" src="boton_enviar.jpg"/>
<button>Boton</button>
```

- Al utilizar varios botones de envío en un formulario, hay que utilizar los siguientes atributos:
  - `formmethod` → Modifica el atributo `method` del `<form>` para los envíos del botón.
  - `formaction` → Modifica el atributo `action` del `<form>` para los envíos del botón.
  - `formenctype` → Modifica el atributo `enctype` del `<form>` para los envíos del botón.
  - `formnovalidate` → Modifica el atributo `novalidate` del `<form>` para los envíos del botón.
  - `formtarget` → Modifica el atributo `target` del `<form>` para los envíos del botón.

```
<input type="submit" value="Enviar formulario"/>
<input type="submit" formaction="formulario_admin.php" value="Enviar fo
<input type="submit" formnovalidate value="Enviar formulario sin valida
```

## Validación de formularios

- **¿Qué es?**

Es un proceso automático que verifica que los datos ingresados sean correctos y tengan el formato adecuado antes de enviarse. Ayuda a:

- Evitar errores comunes y datos malintencionados.
- Garantizar la seguridad y calidad de la información recibida.

- **¿Por qué es importante?**
  - Corrige errores del usuario.
  - Protege contra manipulaciones.
  - Asegura que el servidor gestione datos válidos y seguros.
- **Escenarios de validación**
  - **Sin validación:** Los datos enviados no se verifican en el cliente ni en el servidor.
    - **Riesgos:** Posibilidad de recibir datos incorrectos, inconsistentes o inseguros, comprometiendo el sistema.
  - **Validación en el frontend:** Se realiza directamente en el navegador antes de enviar los datos al servidor.
    - **Ventajas:** Respuesta inmediata para el usuario.
    - **Desventajas:** Los usuarios malintencionados pueden eludir la validación del navegador.
  - **Validación en el backend:** Se verifica en el servidor después de recibir los datos.
    - **Ventajas:** Proporciona una capa de seguridad robusta contra manipulaciones.
    - **Desventajas:** Latencia, ya que el usuario podría tener que corregir errores y reenviar el formulario.
  - **Doble validación (frontend y backend):** Combina la rapidez del frontend con la seguridad del backend.
    - **Ventajas:** Detecta errores temprano y asegura que los datos sean válidos y confiables.
- **Validación en HTML5:** HTML5 incluye herramientas y atributos integrados que facilitan la validación de formularios sin necesidad de JavaScript adicional.
  - **Tipos de datos con validación automática:**
    - `email` → Formatos válidos para correos electrónicos.
    - `url` → URLs correctamente estructuradas.
    - `number` → Sólo números válidos.
    - `date` → Fechas en formatos específicos.
  - **Atributos útiles para validación:**
    - `maxlength` y `minlength` → Restringen la longitud mínima o máxima del texto.
    - `min` y `max` → Definen rangos de valores numéricos.
    - `required` → Hace obligatorio un campo.
    - `pattern` → Permite patrones específicos mediante expresiones regulares.
    - `disabled` y `readonly` → Controlan si el campo puede ser editado o enviado.
  - **Uso del Atributo `pattern`:** El atributo `pattern` utiliza **expresiones regulares** para validar entradas más complejas o específicas.

Expresión regular	Carácter especial	Significado	Descripción
.	Punto	Comodín	Cualquier carácter único.
A B	Pipe	Opciones lógicas	Opciones alternativas (o A o B).
C(A B)	Paréntesis	Agrupaciones	Agrupaciones alternativas (o CA o CB).
[0-9]	Corchetes	Rango	Un número del 0 al 9.
[A-Z]		Letras	Una letra mayúscula.
[^A-Z]	^ en corchetes	Exclusión	Excluye letras mayúsculas.
[0-9]*	Asterisco	Cierre	Cero o más dígitos.
[0-9]+	Signo más	Cierre+	Uno o más dígitos.
[0-9]{3}	Llaves	Precisión	Exactamente tres dígitos.
[0-9]{2,4}		Coincidencia (rango)	De 2 a 4 dígitos.
b?	Interrogación	Carácter opcional	El carácter <b>b</b> puede aparecer o puede que no.
\.	Barra invertida	Escape	El carácter . (no como comodín).

■ Avanzado con lookaheads ( `(?=...)` ):

- **Lookahead positivo:** Valida un patrón sin consumirlo.
- **Atributo `title`:** Personaliza el mensaje de error mostrado al usuario.

```
<label for="textField">Introduce un texto con al menos una letra may
<input type="text" id="textField" name="textField" pattern="(?=.*[A-
```