U1 — Bases y Elementos de Texto en HTML

Portada

- **Título:** Bases y Elementos de Texto en HTML
- **Subtítulo:** Introducción, estructura y semántica básica
- **Autor / Curso / Fecha**
- **Imagen sugerida:** icono de HTML o captura de un editor

Tabla de contenidos

---

1. Introducción

Explicación breve sobre la importancia de HTML como lenguaje de marcado para estructurar contenido en la web. Propósito del PDF: ofrecer una guía clara sobre las bases y los elementos de texto más utilizados y su semántica.

---

2. ¿Qué es HTML y cuál es su rol?

- HTML (HyperText Markup Language) es el lenguaje estándar para crear y estructurar páginas y aplicaciones web.
- Define la **estructura** y el **significado** del contenido (títulos, párrafos, listas, enlaces, etc.).
- No es un lenguaje de programación: describe *qué* es cada parte del contenido, no *cómo* se comporta (eso lo hace CSS/JS).

---

3. El proceso de renderizado en el navegador (resumen técnico)

1. **Solicitud y respuesta HTTP/HTTPS:** el navegador solicita un recurso y recibe HTML (y luego CSS/JS/medios).
2. **Parsing:** el navegador analiza (parsea) el HTML y construye el **DOM** (Document Object Model).
3. **CSSOM:** al cargar CSS el navegador construye la CSSOM.
4. **Render tree:** se combina DOM + CSSOM para crear el árbol de renderizado.
5. **Layout (reflow):** cálculo del tamaño y posición de cada elemento.
6. **Paint:** se rasterizan los píxeles en la pantalla.

> Nota: entender este flujo ayuda a escribir HTML eficiente y depurar problemas de renderizado y rendimiento.

---

4. Estructura mínima de un documento HTML

Explicación de cada parte y su sintaxis mínima.

Ejemplo mínimo

```html
<!DOCTYPE html>
<html lang="es">
  <head>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1" />
    <title>Documento de ejemplo</title>
  </head>
  <body>
    <h1>Hola mundo</h1>
    <p>Este es un documento HTML mínimo.</p>
  </body>
</html>
```

`<!DOCTYPE html>`

- Indica al navegador que el documento es HTML5.
- No es una etiqueta HTML, sino una declaración de tipo.

`<html>`

- Elemento raíz que envuelve todo el contenido HTML.
- Se recomienda incluir el atributo `lang` para accesibilidad y SEO.

`<head>`

- Contiene metadatos, enlaces a estilos, scripts que deben cargarse en head, y el `<title>`.
- **No** es contenido visible por defecto (salvo el `<title>` que aparece en la pestaña del navegador).

`<body>`

- Contiene el contenido visible: texto, imágenes, formularios, etc.

---

5. Metadatos y elementos comunes dentro de `<head>`

- `<meta charset="utf-8">` — define la codificación de caracteres.
- `<meta name="viewport" content="width=device-width, initial-scale=1">` — hace responsivo el layout en móviles.
- `<title>` — texto que aparece en la pestaña del navegador y resultados de búsqueda.
- `<meta name="description" content="...">` — descripción para motores de búsqueda.
- `<link rel="stylesheet" href="styles.css">` — enlaza CSS externo.
- `<script src="script.js" defer></script>` — enlaza JavaScript; `defer` recomienda retrasar ejecución hasta parseo del HTML.

---

6. Elementos de texto

Encabezados: `<h1>` a `<h6>`

- `<h1>` representa el título principal; deben usarse en orden lógico (evitar saltos de nivel sin razón).
- Importancia semántica y para SEO: los motores usan encabezados para entender la jerarquía.

Ejemplo:

```
<h1>Título principal</h1>
<h2>Subtítulo</h2>
<h3>Sección</h3>
```

Párrafos: `<p>`

- Agrupan bloques de texto.
- No deben contener elementos de nivel de bloque como otro `<p>`.

Ejemplo:

```
<p>Este es un párrafo. Aquí va texto explicativo.</p>
```

Saltos de línea: `<br />`

- Inserta un salto de línea simple dentro de texto (uso puntual — **no** para separar párrafos completos).

Ejemplo:

```
<p>Línea 1<br />Línea 2</p>
```

Elementos inline relacionados (breve mención)

- `<strong>`: indica importancia (semántica; se renderiza en negrita por defecto).
- `<em>`: enfatiza (cursiva por defecto).
- `<code>`: fragmento de código en línea.
- `<small>`, `<mark>`, `<abbr>`: usos específicos.

---

7. Semántica estructural básica

Explicación de cada etiqueta y ejemplo de uso.

`<header>`

- Representa la cabecera introductoria del documento o de una sección.
- Suele contener logo, títulos, navegación principal.

`<footer>`

- Pie de página: información de autor, copyright, enlaces legales.

`<main>`

- Contenido principal del documento. Debe ser único por página.

`<section>`

- Agrupa contenido temáticamente relacionado. Usar cuando el contenido tiene un encabezado propio.

`<article>`

- Contenido independiente y autocontenido (ej: artículo de blog, entrada de noticias).

`<nav>`

- Contiene enlaces de navegación principales.

`<aside>`

- Contenido tangencial: barras laterales, widgets, info relacionada que no es parte central.

Ejemplo estructural combinado:

```
<body>
  <header>
    <h1>Mi sitio</h1>
    <nav> ... </nav>
  </header>

  <main>
    <article>
      <h2>Artículo 1</h2>
      <p>Contenido del artículo...</p>
    </article>

    <section>
      <h2>Sección temática</h2>
      <p>Texto de la sección...</p>
    </section>

    <aside>
      <p>Contenido relacionado</p>
    </aside>
  </main>

  <footer>
    <p>© 2025 — Autor</p>
  </footer>
</body>
```

---

8. Ejemplos completos y comentados

Se incluyen ejemplos prácticos con comentarios inline que expliquen por qué se usa cada etiqueta y alternativas.

(En el PDF se pueden añadir 2-3 ejemplos de páginas sencillas: blog, página de producto, landing page.)

---

9. Buenas prácticas y accesibilidad

- Usar encabezados de forma jerárquica (`<h1>` -> `<h2>` -> `<h3>`) sin saltos innecesarios.
- Incluir `lang` en `<html>` y `meta charset="utf-8"`.
- Textos alternativos (`alt`) en imágenes para accesibilidad.
- Evitar usar `<br />` para espaciados que deben ser controlados por CSS.
- Usar etiquetas semánticas (`<main>` , `<nav>` , etc.) para mejorar SEO y accesibilidad.

- Formularios con `label` correctamente asociados.
- Control de foco y orden lógico para navegación por teclado.

---

10. Errores comunes y cómo evitarlos

- **No cerrar etiquetas** (aunque HTML5 puede corregirlo, es mala práctica).
- **Mala jerarquía de encabezados**: dificulta la navegación y accesibilidad.
- **Usar elementos** `div` **en lugar de semánticos** sin necesidad.
- **Usar** `<br />` **para separar secciones enteras** en lugar de `<p>` o CSS.

---

11. Ejercicios propuestos

1. Crear una página con `header`, `nav`, `main` (con 2 `article`), `aside`, y `footer`.
2. Reestructurar una página sin semántica usando etiquetas semánticas apropiadas.
3. Escribir ejemplos de texto usando `<h1>`..`<h3>`, `<p>`, `<strong>`, `<em>`, y `<code>`.

---

12. Resumen y recursos para profundizar

- Resumen de conceptos clave.
- Enlaces sugeridos (MDN Web Docs, W3C, especificaciones HTML Living Standard).

---

13. Glosario

- **DOM**: Document Object Model.
- **CSSOM**: CSS Object Model.
- **Viewport**: área visible de la página en el dispositivo.
- **Semántica**: significado de las etiquetas para máquinas y humanos.

Anexo: Plantillas y snippets listos para copiar

- Plantilla base (ya incluida en la sección 4).
- Plantilla con layout semántico (ya incluida en la sección 7).

Fin del documento