

HTML: Fundamentos y estructura básica

¿Qué es HTML?

HTML (HyperText Markup Language) es el lenguaje estándar para estructurar el contenido en la web.

- **HyperText:** Permite enlazar documentos entre sí mediante hipervínculos.
- **Markup Language:** Utiliza etiquetas para describir la estructura de la información.

HTML no es un lenguaje de programación, sino un lenguaje de marcado que organiza elementos en una página.

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Ejemplo de Documento</title>
</head>
<body>
  <h1>Bienvenidos al curso</h1>
  <p>Este es un ejemplo básico de HTML.</p>
</body>
</html>
```

Explicación de las secciones:

- **<!DOCTYPE html>:** Informa al navegador que se usará la versión HTML5.
- **<html>:** Contenedor raíz de todo el contenido.
 - **Atributo lang:** Indica el idioma del documento (importante para SEO y accesibilidad).
- **<head>:** Contiene metadatos, configuraciones y recursos externos como CSS o JavaScript.
 - **<meta charset="UTF-8">:** Define la codificación de caracteres (UTF-8 permite caracteres especiales).
 - **<title>:** Título que aparece en la pestaña del navegador.
- **<body>:** Contiene el contenido visible de la página web.

Etiquetas principales en HTML

1. Estructura del contenido:

- `<div>`: Contenedor genérico, útil para agrupar elementos.
- `<p>`: Representa un párrafo.
- `<h1>`-`<h6>`: Encabezados jerárquicos; `<h1>` es el más importante.

`<h1>`Título Principal`</h1>`

`<h2>`Subtítulo`</h2>`

Hipervínculos:

- `<a>`: Define enlaces a otras páginas o recursos.

`Ir a Google`

- **Atributo `target="_blank"`**: Abre el enlace en una nueva pestaña.

``: Muestra imágenes en la página.

``

- **`src`**: Ruta de la imagen.
- **`alt`**: Texto alternativo para accesibilidad y SEO.

Atributos en HTML

- `class`**: Define una categoría reutilizable para aplicar estilos CSS.

html

`<p class="destacado">Texto destacado</p>`

Copiar Editar
- `id`**: Identificador único para un elemento específico.

html

`<div id="principal">Contenido principal</div>`

Copiar Editar
- Otros atributos comunes:**
 - **`src`**: Fuente de un recurso (imágenes, scripts).
 - **`href`**: Dirección de un enlace.



2. ¿Qué es un elemento en HTML?

Un **elemento** es el conjunto completo que combina:

1. La **etiqueta de apertura**.
2. El **contenido interno** (opcional).
3. La **etiqueta de cierre** (en elementos que lo requieren).

Ejemplo completo de un elemento:

html

 Copiar  Editar

```
<h1>¡Hola, mundo!</h1>
```

En este caso:

- El **elemento** es todo el conjunto: `<h1>¡Hola, mundo!</h1>`.
- El **contenido interno** es `¡Hola, mundo!`.

Tipos de elementos:

1. Elementos en bloque (block-level):

- Ocupan todo el ancho disponible y comienzan en una nueva línea.
- Ejemplos: `<div>`, `<p>`, `<h1>` - `<h6>`, ``, `<table>`.

2. Elementos en línea (inline):

- Ocupan solo el espacio necesario y no interrumpen el flujo del texto.
- Ejemplos: `<a>`, ``, ``, ``.

Elementos semánticos:



- Proporcionan significado adicional al contenido, mejorando la accesibilidad y el SEO.
- Ejemplos: `<header>`, `<footer>`, `<article>`, `<section>`.

Métodos de integración de CSS

1. En línea:

- Definir estilos directamente en la etiqueta HTML.
- Útil para pruebas rápidas, pero no recomendado para proyectos grandes.

html



 Copiar  Editar

```
<p style="color: red; font-size: 16px;">Texto rojo</p>
```

2. Interno:

- Usar un bloque `<style>` dentro de `<head>`.
- Adecuado para estilos que afectan solo una página.

html



 Copiar  Editar

```
<style>
  p { color: blue; font-size: 18px; }
</style>
```

3. Externo:

- Enlazar un archivo `.css` separado con `<link>`.
- Método preferido para mantener el código limpio y modular.

html

 Copiar  Editar

```
<link rel="stylesheet" href="estilos.css">
```

Selectores básicos en CSS

1. Selector de tipo:

Aplica estilos a todas las etiquetas de un tipo específico.

CSS

Copiar Editar

```
p { color: green; }
```

2. Selector de clase:

Aplica estilos a elementos con una clase específica.

CSS

Copiar Editar

```
.destacado { font-weight: bold; }
```

3. Selector de id:

Aplica estilos a un elemento único con un id.

CSS

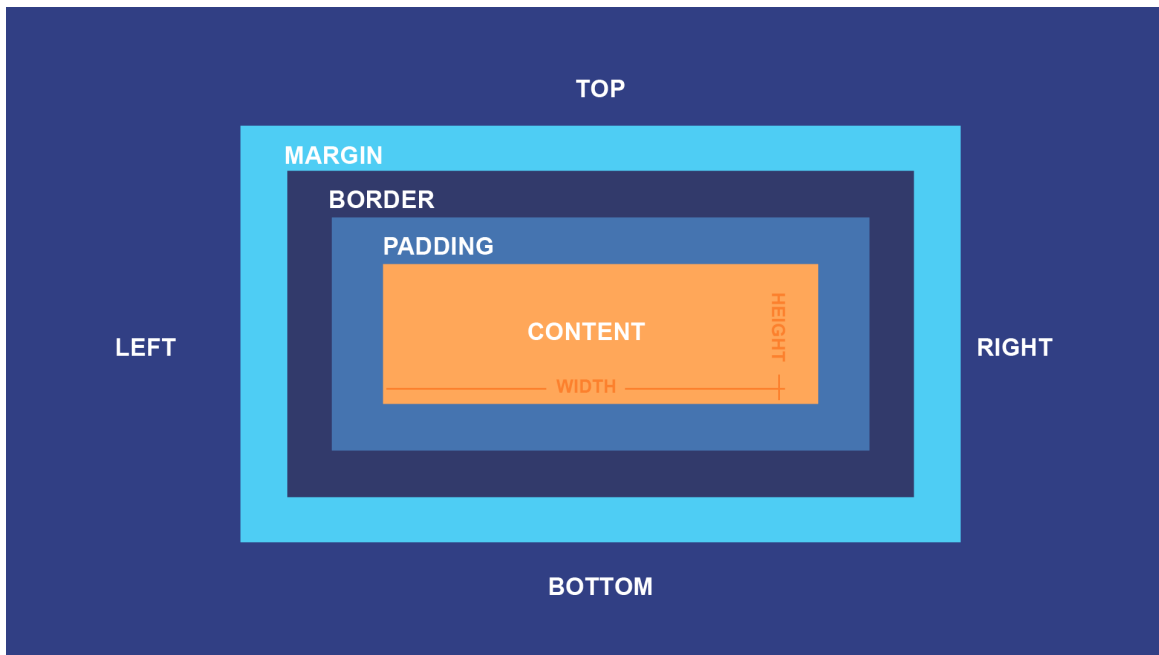
Copiar Editar

```
#principal { background-color: lightgray; }
```

Modelo de caja en CSS (Box Model)

Cada elemento en CSS se representa como una "caja" que incluye:



1. **Contenido:** Texto o elementos internos.
2. **Padding:** Espacio entre el contenido y el borde.
3. **Border:** Línea que rodea el contenido y el relleno.
4. **Margin:** Espacio fuera del borde.



1. `block` (Elementos de nivel bloque)

- **Descripción:**
Los elementos con `display: block` ocupan todo el ancho disponible de la línea. Comienzan en una nueva línea automáticamente y empujan los elementos siguientes hacia abajo.
- **Características principales:**
 - Son ideales para dividir secciones del contenido.
 - Respetan márgenes y padding.
- **Ejemplos de etiquetas con comportamiento `block` por defecto:**
 - `<div>`, `<p>`, `<h1>` – `<h6>`, `<section>`.
- **Ejemplo visual:**

html

 Copiar  Editar

```
<div style="background-color: lightblue;">Elemento bloque 1</div>
<div style="background-color: lightcoral;">Elemento bloque 2</div>
```

Resultado: Cada `div` ocupará toda la línea.

3. flex (Modelo Flexible)

- **Descripción:**

`display: flex` permite al contenedor (elemento padre) organizar sus hijos de manera flexible, distribuyéndolos en filas o columnas. Es ideal para layouts dinámicos y responsivos.

- **Características principales:**

- Permite alinear, distribuir y ajustar elementos hijos de forma sencilla.
- Admite propiedades como `justify-content` y `align-items`.

- **Ejemplo visual básico:**

html

Copiar Editar

```
<div style="display: flex; gap: 10px; background-color: lightgray;">
  <div style="background-color: lightblue;">Elemento 1</div>
  <div style="background-color: lightcoral;">Elemento 2</div>
  <div style="background-color: lightgreen;">Elemento 3</div>
</div>
```

Resultado: Los elementos hijos (Elemento 1 , Elemento 2 , Elemento 3) estarán en la misma línea, con un espacio uniforme entre ellos.

4. grid (Modelo de Rejilla)

- **Descripción:**

`display: grid` organiza los elementos hijos en una cuadrícula formada por filas y columnas. Es ideal para layouts más complejos, como tableros o paneles de control.

- **Características principales:**

- Se define un contenedor padre con filas y columnas explícitas.
- Permite colocar elementos en áreas específicas de la cuadrícula.

- **Ejemplo visual básico:**

html

Copiar Editar

```
<div style="display: grid; grid-template-columns: repeat(3, 1fr); gap: 10px;">
  <div style="background-color: lightblue;">Elemento 1</div>
  <div style="background-color: lightcoral;">Elemento 2</div>
  <div style="background-color: lightgreen;">Elemento 3</div>
</div>
```

Resultado: Los elementos hijos se distribuyen en 3 columnas iguales.

