

1 HTML (Estructura y Contenido)

Elementos utilizados para definir la estructura y el contenido de la página web.

1.1 Estructura Principal

- `<!DOCTYPE html>`: Declara que el documento es de tipo HTML5.
- `<html>`: Elemento raíz que engloba todo el contenido.
- `<head>`: Contiene metadatos, estilos y el título de la página.
- `<body>`: Contiene el contenido visible por el usuario.
- `<header>`, `<main>`, `<footer>`: Secciones estructurales.
- `<nav>`: Sección con enlaces de navegación.
- `<section>`, `<article>`: Agrupan contenido con sentido temático.

Ejemplo de Estructura Principal

```
<!DOCTYPE html>
<html lang="es">
<head>
    <meta charset="UTF-8">
    <title>Ejemplo</title>
</head>
<body>
    <header>
        <nav>...</nav>
    </header>
    <main>
        <section>
            <article>...</article>
        </section>
        <aside>...</aside>
    </main>
    <footer>
        <p>Pie de página</p>
    </footer>
</body>
</html>
```

1.2 Formularios

Los formularios permiten la interacción del usuario con la página para enviar información.

- `<form>`: Contenedor del formulario. Atributos:

- `method`: Define el método de envío. `POST` envía datos de forma segura; `GET` los muestra en la URL.
- `action`: URL a la que se envían los datos (si se omite, se envían a la misma página).

- `<input>`: Campo de entrada. Atributos:

- `type="text"`: Campo de texto.
- `type=radio"`: Opción de selección única.
- `required`: Campo obligatorio.
- `name`: Nombre del campo, clave para el envío de datos.
- `value`: Valor que se enviará.

- `<select>`: Lista desplegable. Contiene `<option>`.

- `<textarea>`: Área de texto multilínea.

- `<label>`: Etiqueta asociada a un campo mediante `for`.

- `<button type="submit">`: Envía el formulario.

- `<button type=reset">`: Limpia los campos.

- `<fieldset>` / `<legend>`: Agrupan y nombran conjuntos de campos.

Ejemplo de Formulario

```
<form method="POST" action="/procesar">
    <fieldset>
        <legend>Datos Personales</legend>
        <div>
            <label for="nombre_id">Nombre:</label>
            <input type="text" id="nombre_id" name="nombre_usuario" required>
        </div>
        <div>
            <input type="radio" id="hombre" name="sexo" value="h" checked>
            <label for="hombre">Hombre</label>
            <input type="radio" id="mujer" name="sexo" value="m">
            <label for="mujer">Mujer</label>
        </div>
        <div>
            <label for="pais_id">País:</label>
            <select id="pais_id" name="pais">
                <option value="ES" selected>España</option>
                <option value="FR">Francia</option>
            </select>
        </div>
        <div>
            <label for="com_id">Comentarios:</label>
            <textarea id="com_id" name="comentarios" rows="4"></textarea>
        </div>
        <button type="submit">Enviar</button>
        <button type="reset">Limpiar</button>
    </fieldset>
</form>
```

2 CSS (Estilos y Maquetación)

Define la apariencia visual, formato y diseño de los elementos HTML.

2.1 Selectores

Los selectores permiten aplicar estilos a elementos específicos:

- `p`: Selecciona todos los párrafos.
- `.clase`: Selecciona elementos con esa clase.
- `#id`: Selecciona un elemento con un identificador único.
- `elemento elemento`: Selecciona descendientes.
- `elemento:hover`: Aplica estilos cuando el cursor pasa por encima.

Ejemplo de Selectores

```
<style>
    p { color: blue; }
    .noticia { font-weight: bold; }
    #articulo-1 { background-color: #f0f0f0; }
    a:hover { color: red; }
</style>
```

2.2 Modelo de Caja (Box Model)

Cada elemento en CSS es una caja con contenido, relleno, borde y margen.

- `width`, `height`: Dimensiones del contenido.
- `padding`: Espacio interno.
- `margin`: Espacio externo.
- `border`: Línea alrededor del elemento.
- `border-radius`: Bordes redondeados.
- `box-shadow`: Sombra de la caja.
- `box-sizing: border-box`: Incluye padding y borde dentro del ancho total.

Ejemplo de Modelo de Caja

```
<style>
    .caja {
        box-sizing: border-box;
        width: 300px;
        min-height: 100px;
        padding: 20px;
        margin: 15px;
        border: 2px solid #333;
        border-radius: 8px;
        box-shadow: 5px 5px 10px rgba(0,0,0,0.2);
    }
</style>
```

2.3 Tipografía y Texto

Propiedades que afectan la apariencia del texto:

- **font-family**: Define la fuente.
- **font-size**: Tamaño del texto.
- **font-weight**: Grosor (normal, bold).
- **color**: Color del texto.
- **text-align**: Alineación del texto (left, center, right).

Ejemplo de Tipografía

```
<style>
    .titulo-principal {
        font-family: "Segoe UI", Roboto, sans-serif;
        font-size: 2.5rem;
        font-weight: 700;
        color: #3d5a6c;
        text-align: center;
    }
</style>
```

2.4 Fondos

Permiten definir imágenes o colores de fondo.

- **background-color**: Color de fondo.

- **background-image**: Imagen de fondo.
- **background-size**: Escala la imagen (cover, contain).
- **background-position**: Posición de la imagen.
- **background-attachment: fixed**: Hace que el fondo quede fijo.

Ejemplo de Fondos

```
<style>
  .hero-section {
    height: 100vh;
    background-image: url('imagenes/fondo.jpg');
    background-size: cover;
    background-position: center;
    background-attachment: fixed;
  }
</style>
```

2.5 Posicionamiento

Controla la ubicación de los elementos en la página.

- **position**: Define el tipo de posicionamiento (static, relative, absolute, fixed, sticky).
- **top**, **bottom**, **left**, **right**: Posición respecto al contenedor.
- **z-index**: Controla la superposición de elementos.

Ejemplo de Posicionamiento

```
<style>
  .header-principal {
    position: sticky;
    top: 0;
    z-index: 100;
    background-color: white;
  }
</style>
```

2.6 Flexbox

Sistema de diseño flexible que distribuye elementos en fila o columna.

- **display: flex**: Activa el modelo Flexbox.

- **justify-content**: Alinea horizontalmente (center, space-between...).
- **align-items**: Alinea verticalmente (center, flex-start...).
- **flex-direction**: Dirección del eje (row o column).

Ejemplo de Flexbox

```
<style>
    .flex-container {
        display: flex;
        justify-content: space-between;
        align-items: center;
    }
    .columna-izq, .columna-der {
        width: 45%;
        background: #e3f2fd;
        padding: 10px;
    }
</style>
```

2.7 Grid

Sistema de diseño basado en cuadrículas.

- **display: grid**: Activa el modelo Grid.
- **grid-template-columns**: Define columnas.
- **gap**: Espacio entre celdas.

Ejemplo de Grid

```
<style>
    .grid-container {
        display: grid;
        grid-template-columns: repeat(3, 1fr);
        gap: 10px;
    }
    .card {
        background-color: #dceefc;
        padding: 15px;
        border-radius: 6px;
        text-align: center;
    }
</style>
```

2.8 Transiciones y Efectos Hover

Añaden animaciones suaves al interactuar con los elementos.

- **transition**: Controla la duración y tipo de animación.
- **transform: scale()**: Cambia el tamaño del elemento.
- **:hover**: Activa un estilo al pasar el cursor.

Ejemplo de Efectos y Responsive

```
<style>
    .boton-interactivo {
        background-color: blue;
        color: white;
        padding: 10px 20px;
        border-radius: 5px;
        cursor: pointer;
        transition: background-color 0.3s ease, transform 0.3s ease;
    }
    .boton-interactivo:hover {
        background-color: darkblue;
        transform: scale(1.05);
    }

    /* Diseño responsive */
    @media (max-width: 600px) {
        .boton-interactivo {
            width: 100%;
        }
    }
</style>
```