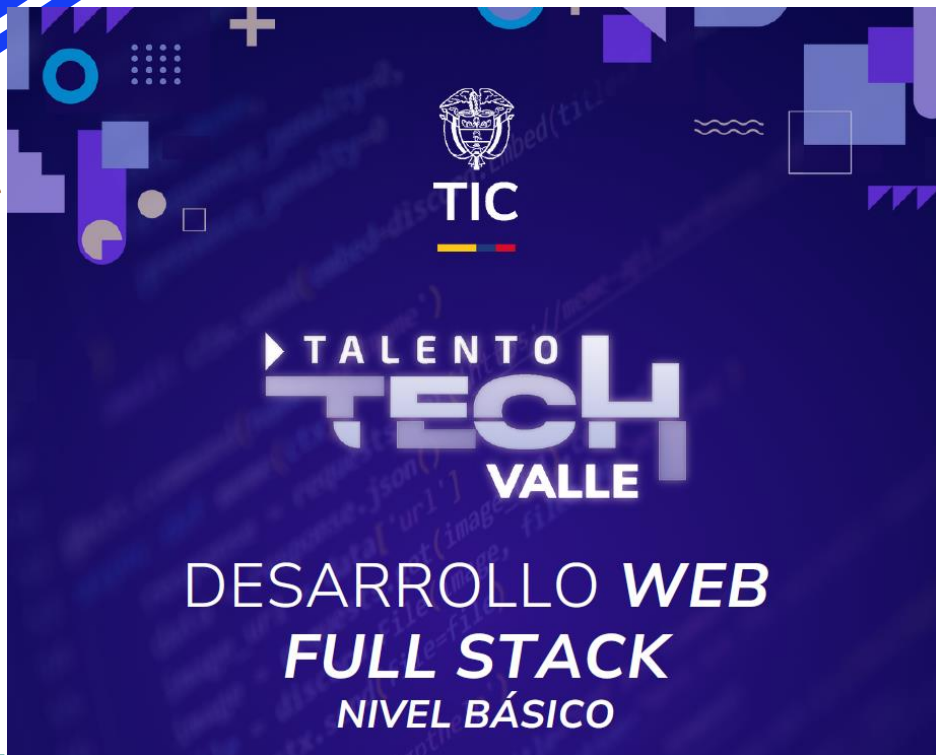


*Alexander Ocoro*



*Mentor*



*Warner Fdo. Valencia*



*Ejecutor*

# HTML y CSS Básico II

Objetivo: Refinar el uso de selectores y aprender a aplicar estilos más avanzados para mejorar la presentación y la organización visual de una página web.

# Refinamiento de Selectores

Objetivo: Comprender y aplicar selectores avanzados para estilizar secciones específicas de una página web.

# Refinamiento de Selectores

## Teoría: Selectores Avanzados en CSS

### 1. Selector de descendiente (Elemento Elemento)

- Aplica estilos a todos los elementos secundarios específicos dentro de un contenedor.
- Ejemplo: `div p` selecciona todos los `<p>` dentro de cualquier `<div>`.

### 2. Selector de hijo directo (Elemento > Elemento)

- Aplica estilos a un elemento específico que es hijo directo de un contenedor.
- Ejemplo: `div > p` selecciona solo los `<p>` que son hijos directos de cualquier `<div>`.

### 3. Selector de hermano adyacente (Elemento + Elemento)

- Aplica estilos a un elemento que es inmediatamente posterior (hermano) a otro elemento específico.
- Ejemplo: `h2 + p` selecciona el primer `<p>` que aparece después de un `<h2>`.

### 4. Selector de atributo ([atributo])

- Selecciona elementos que tienen un atributo específico.
- Ejemplo: `input[type="text"]` selecciona todos los `<input>` de tipo texto.

## Ejemplo Práctico: Aplicar Selectores Avanzados

1. **Paso 1:** Agrega un `<section>` con diferentes elementos en `index.html`.

```
<section>
  <div class="container">
    <h2>Título de Sección</h2>
    <p>Este es un párrafo en la sección.</p>
    <p>Otro párrafo en la sección.</p>
    <p>Este párrafo aparece después de un h2.</p>
  </div>
</section>
```

2. **Paso 2:** En `estilos.css`, aplica los siguientes selectores avanzados:

```
/* Selecciona todos los párrafos dentro de div.container */
.container p {
  color: #555;
}

/* Selecciona solo los párrafos hijos directos de div.container */
.container > p {
  font-weight: bold;
}

/* Selecciona el primer párrafo que sigue a h2 */
h2 + p {
  color: #007bff;
}

/* Selecciona todos los inputs de tipo texto */
input[type="text"] {
  border: 2px solid #333;
}
```

# Propiedades de Estilo en CSS

Objetivo: Aprender a aplicar colores de fondo, bordes y a manejar el espacio alrededor de los elementos con padding y margin.

## Teoría: Propiedades de Estilo en CSS

### 1. Colores de Fondo (`background-color`)

- Permite aplicar un color de fondo a los elementos, ya sea en nombre de color, valor hexadecimal o RGB.
- Ejemplo:

```
.container {  
  background-color: #f5f5f5;  
}
```

### 2. Bordes (`border`)

- Propiedad que define el grosor, estilo y color de un borde.
- Ejemplo:

```
.container {  
  border: 1px solid #ccc;  
}
```

### 3. Espacio Interior (`padding`) y Exterior (`margin`)

- `padding`: Define el espacio entre el borde y el contenido.
- `margin`: Define el espacio alrededor de un elemento.
- Ejemplo:

```
.container {  
  padding: 20px;  
  margin: 10px auto;  
}
```

### Ejemplo Práctico: Agregar Fondo, Bordes y Espacio a Elementos

1. **Paso 1:** En el archivo `estilos.css`, aplica fondo, bordes y espacios a un contenedor específico:

```
/* Fondo y bordes */  
.container {  
  background-color: #f9f9f9;  
  border: 2px solid #e1e1e1;  
  border-radius: 8px;  
  padding: 15px;  
  margin: 20px auto;  
  width: 80%;  
}
```

2. **Paso 2:** Aplica `margin` y `padding` a otros elementos, como `<h2>` o `<p>`.



# Estilos para Formularios y Multimedia

Objetivo: Personalizar la apariencia de formularios y elementos multimedia.

## Teoría: Estilos para Formularios y Multimedia

### 1. Personalización de Formularios

- **Inputs y etiquetas:** Se pueden personalizar los bordes, fondos y colores de los elementos de entrada.
- **Botones:** Usando `background-color`, `border-radius` y `padding` para mejorar la apariencia.

### 2. Estilos de Multimedia

- **Videos y audio:** Cambiar el tamaño y añadir bordes o espacios.
- Ejemplo para un botón y un input:

```
input[type="text"],
input[type="email"] {
  width: 100%;
  padding: 8px;
  margin-bottom: 10px;
  border: 1px solid #ccc;
  border-radius: 5px;
}

button {
  background-color: #007bff;
  color: #fff;
  padding: 10px 15px;
  border: none;
  border-radius: 5px;
  cursor: pointer;
}

button:hover {
  background-color: #0056b3;
}
```

## Ejemplo Práctico: Estilizar el Formulario y el Video

1. **Paso 1:** En `estilos.css`, estiliza el formulario agregando espacio y colores:

```
form {  
  width: 300px;  
  margin: 0 auto;  
  padding: 20px;  
  border: 1px solid #ddd;  
  background-color: #fafafa;  
}  
  
input,  
button {  
  margin-top: 10px;  
}
```

2. **Paso 2:** Aplica bordes y tamaños al elemento `<video>`:

```
video {  
  display: block;  
  margin: 0 auto;  
  border: 5px solid #eee;  
  width: 100%;  
  max-width: 600px;  
}
```

# Estructura de Contenedores

Objetivo: Personalizar la apariencia de formularios y elementos multimedia.

## Teoría: Estructura de Contenedores en CSS

### 1. Uso de `<div>` como contenedor

- `<div>` es un contenedor genérico que ayuda a organizar y aplicar estilos a grupos de elementos.

### 2. Alineación de Elementos con CSS

- Propiedades de alineación en CSS para centrar o distribuir elementos:
  - `text-align: center` para centrar texto.
  - `margin: auto` para centrar contenedores.
  - Ejemplo de alinear un contenedor:

```
.contenedor {  
  width: 80%;  
  margin: 0 auto;  
  background-color: #f1f1f1;  
  padding: 20px;  
}
```

## Ejemplo Práctico: Crear un Contenedor y Organizar la Página

1. **Paso 1:** Agrupa secciones en contenedores `<div>` en el archivo `index.html`.

```
<div class="contenedor">
  <header>
    <h1>Título de la Página</h1>
  </header>
  <section>
    <h2>Sección 1</h2>
    <p>Contenido de la primera sección.</p>
  </section>
  <section>
    <h2>Sección 2</h2>
    <p>Contenido de la segunda sección.</p>
  </section>
  <footer>
    <p>Pie de página.</p>
  </footer>
</div>
```

### 2. Paso 2

: Aplica estilos al contenedor en `estilos.css`:

```
.contenedor {
  width: 80%;
  margin: 0 auto;
  background-color: #f9f9f9;
  padding: 20px;
  border: 1px solid #ddd;
}
```