

# Curso HTML5, CSS y JavaScript

---

## Módulo 2: Introducción a HTML 5

- [emilianoagustingallo@gmail.com](mailto:emilianoagustingallo@gmail.com)
- <https://www.linkedin.com/in/emiliano-gallo/>



# HTML5

- Es la más reciente versión de HTML
- Incorpora una serie de elementos útiles para las web actuales
- Incorpora etiquetas que dan soporte a nuevas funcionalidades: es decir, aquellas que nos sirven para extender el HTML. Por ej: vídeo, sonido, lienzos donde diseñar dibujos
- Incorpora etiquetas que componen la web semántica: algunas etiquetas que realmente no proponen nuevas funcionalidades, sino que sirven para componer sitios indicando qué es cada bloque de código.

# Funcionalidades multimedia

- Audio: inserta un audio dentro de una página `<audio></audio>`
- Video: inserta un video dentro de una página `<video></video>`
- Embed: inserta contenido externo dentro de una página (ej: plugins) `<embed></embed>`
- Track: especifica varias pistas de sonido o video `<track></track>`
- Source: especifica varias fuentes para un mismo audio o video `<source></source>`

# Funcionalidades de formulario

- Meter: permite trabajar con medidas y escalas `<meter></meter>`
- Progress: implementa barra de progreso `<progress></progress>`
- Output: realiza y muestra cálculos matemáticos `<output></output>`
- Datalist: crea campos con funcionalidad de autocompletar `<datalist></datalist>`

# Funcionalidades de dibujo

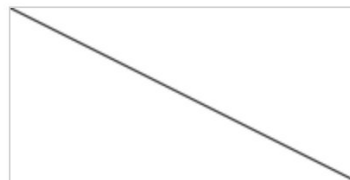
- Canvas: genera un lienzo en la página donde realiza cualquier tipo de diseño, soporta dibujo de todo tipo de formas, degradados, imágenes `<canvas></canvas>`

```
<!DOCTYPE html>
<html>
<body>

<canvas id="myCanvas" width="200" height="100" style="border:1px solid #d3d3d3;">
Tu navegador no soporta la etiqueta HTML Canvas</canvas>

<script>
var c = document.getElementById("myCanvas");
var ctx = c.getContext("2d");
ctx.moveTo(0,0);
ctx.lineTo(200,100);
ctx.stroke();
</script>

</body>
</html>
```



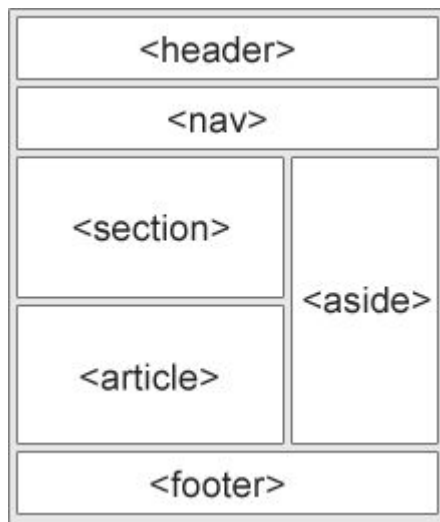
# Web semántica

Describe su significado tanto para el navegador como para el desarrollador

- Artículo: especifica un artículo, es decir, una unidad de contenido `<article></article>`
- Section: es una sección dentro de un documento `<section></section>`
- Header: la cabecera de una página `<header></header>`
- Footer: el pie de página o informaciones que formen el pie de una sección `<footer></footer>`
- Aside: es una parte de la web que muestra contenido accesorio, generalmente colocado en un panel lateral `<aside></aside>`
- Nav: navegador principal de una página web `<nav></nav>`

# Elementos

HTML tiene varios elementos semánticos que nos ayudan a definir las partes de una web



- `<header>` - Define la cabecera del documento
- `<nav>` - Define los links de navegación
- `<section>` - Define la sección del documento
- `<article>` - Define un contenido independiente
- `<aside>` - Define un contenido aparte como una barra lateral
- `<footer>` - Define el pie de un documento
- `<details>` - Define información adicional que el usuario puede mostrar u ocultar a demanda
- `<summary>` - Define un encabezado para `<details>`

# Formularios HTML

Se utiliza un formulario HTML para recopilar información ingresada por un usuario. La entrada del usuario se puede enviar a un servidor para su procesamiento

```
<!DOCTYPE html>
<html>
<body>

<h2>Formulario HTML</h2>

<form action="/action_page.php">
  <label for="fname">Nombre:</label><br>
  <input type="text" id="fname" name="fname" value="John"><br>
  <label for="lname">Apellido:</label><br>
  <input type="text" id="lname" name="lname" value="Doe"><br><br>
  <input type="submit" value="Enviar">
</form>

</body>
</html>
```

## Formulario HTML

Nombre:

Apellido:



# Diseños layouts

Los sitios webs usualmente muestran contenido en columnas



# Técnicas de diseños

Hay 4 diferentes técnicas para crear diseños multicolumna. Cada una de ella tiene pros y contras

- Flotante `float`
- Flexible `flexbox`
- Grilla `grid`
- Framework (ej. Bootstrap)

# Diseño flotante float

La propiedad `float` es usada para posicionar y formatear contenido. Por ej: para dejar que una imagen flote a la izquierda de un texto. Los elementos flotantes están vinculados al flujo de documentos, lo que puede dañar la flexibilidad

- La propiedad `float` especifica como el elemento debería float
  - Left
  - Right
  - None
  - inherits
- La propiedad `clear` especifica qué elementos pueden flotar junto al elemento y de qué lado

# Diseño flexible flex

El uso de **flexbox** asegura que los elementos se comporten de manera predecible cuando el diseño de la página debe acomodarse a diferentes tamaños de pantalla y diferentes dispositivos de visualización.

Facilita el diseño de una estructura de diseño flexible y responsive sin utilizar **float** o **position**

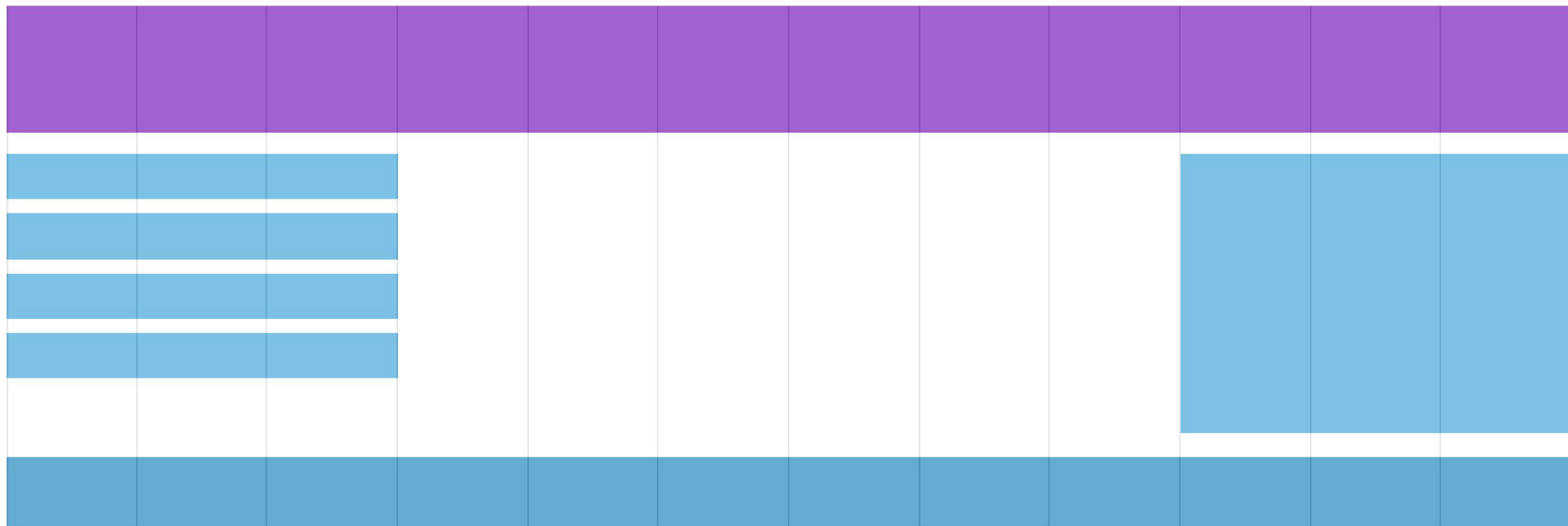
- flex-direction
- flex-wrap
- flex-flow
- justify-content
- align-items
- align-content

[https://www.w3schools.com/css/css3\\_flexbox.asp](https://www.w3schools.com/css/css3_flexbox.asp)

<https://flexboxfroggy.com/#es>

# Diseño de grilla **grid**

Ofrece un sistema de diseño basado en cuadrícula, con filas y columnas, lo que facilita el diseño de páginas web sin tener que usar **float** y **position**.



# Tipos de rutas

Contamos con 2 formas de acceder a los archivos dentro de un archivo HTML

- Ruta relativa: siempre se parte desde donde se encuentra el archivo HTML
  - En la misma ubicación: “NombreArchivo.extensión”
  - Una carpeta mas adentro: “NombreCarpeta/NombreArchivo.extensión”
  - Una carpeta mas arriba: “../NombreArchivo.extensión”
- Ruta absoluta: siempre se parte desde el raiz del disco rígido: “C: NombreUsuario/Mis documentos/NombreArchivo.extensión”

# A trabajar!!



Desarrollar un sitio utilizando **flex** que luzca similar a la de la imagen

