

INTRODUCCIÓN AL DESARROLLO FRONTEND CON REACT 2023



SECRETARÍA DE
EXTENSIÓN
UNIVERSITARIA
UTN - FRC



*UTN
Facultad Regional Córdoba

Agencia
CÓRDOBA
JOVEN

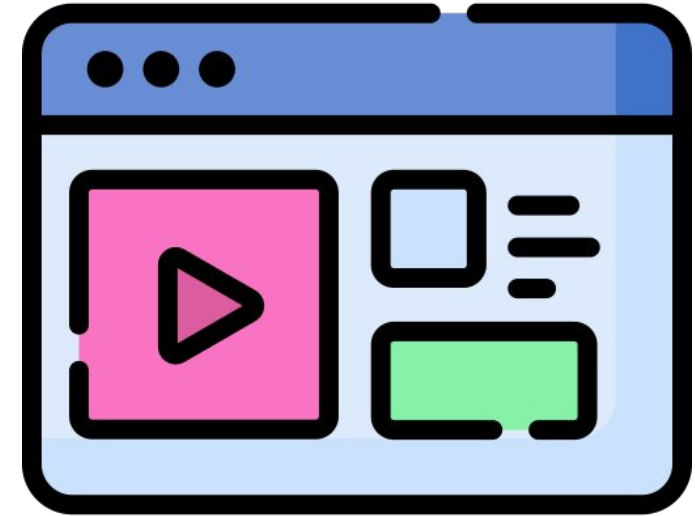


CÓRDOBA
entre todos

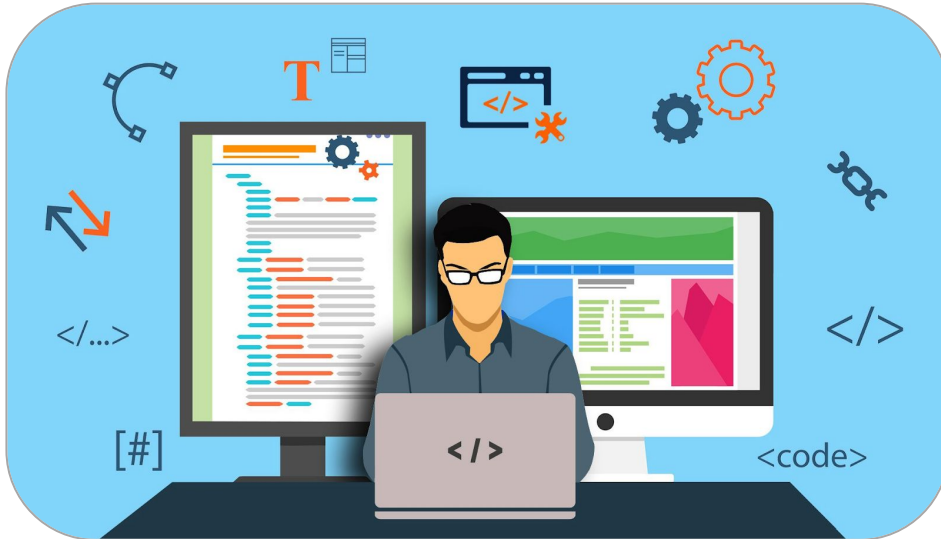
Clase 02 - Contenido

Fundamentos de HTML y CSS

- Introducción a HTML
- Estructura básica de un archivo HTML.
- Etiquetas HTML y sus atributos más comunes.
- Introducción a CSS y su sintaxis.
- Selectores y propiedades CSS básicas.



Objetivos de la Clase



- **Entender el Rol de HTML y CSS**
 - Diferenciar entre estructura (HTML) y estilo (CSS).
- **Crear una Página Web Básica**
 - Utilizar etiquetas HTML para estructurar contenido.
- **Aplicar Estilos con CSS**
 - Utilizar selectores, propiedades y valores en CSS para diseñar una página.
- **Realizar Diseño Responsivo**
 - Usar media queries para adaptar el diseño a diferentes dispositivos.
- **Utilizar Herramientas de Desarrollo Web**
 - Familiarizarse con las herramientas de desarrollo de navegadores como Chrome DevTools.
- **Introducción a Frameworks y Librerías**
 - Tener una noción básica de herramientas como Bootstrap, Tailwind y SASS.

¿Qué son HTML y CSS?

HTML



HTML (HyperText Markup Language):

- Lenguaje de marcado estándar para crear páginas web.
- Estructura el contenido en la web (texto, imágenes, listas, tablas, etc.)

CSS (Cascading Style Sheets):

- Lenguaje de diseño que se utiliza para describir la apariencia de documentos HTML.
- Controla el diseño, colores, fuentes y más.

CSS



¿Qué son HTML y CSS?

HTML (HyperText Markup Language)

```
<!DOCTYPE html>
<html>
<head>
  <title>Mi Página</title>
</head>
<body>
  <h1>Hola, mundo!</h1>
</body>
</html>
```

CSS (Cascading Style Sheets)

```
body {
  background-color: #f2f2f2;
}
h1 {
  color: blue;
  font-size: 36px;
}
```

HTML: Importancia en Desarrollo Web

- **Estructura:** HTML proporciona la estructura básica de las páginas web, permitiendo organizar el contenido de manera efectiva.
- **Universalidad:** Es la base de casi todos los sitios web y es compatible con todos los navegadores modernos.
- **Facilidad de Uso:** Es uno de los lenguajes más fáciles de aprender, lo que lo convierte en un punto de partida ideal para quienes deseen entrar al mundo del desarrollo web.
- **Integración:** HTML se integra sin problemas con otros lenguajes de programación y tecnologías web, como CSS y JavaScript, permitiendo la creación de aplicaciones web más complejas.
- **Accesibilidad:** Con HTML se pueden crear páginas web que son accesibles a personas con discapacidades, mejorando la usabilidad y la inclusividad.



Estructura Básica de un Archivo HTML

```
<!DOCTYPE html>
<html>
  <head>
    <title>Título de la Página</title>
  </head>
  <body>
    <h1>Encabezado Principal</h1>
    <p>Texto del párrafo</p>
  </body>
</html>
```



Descripción de Componentes:

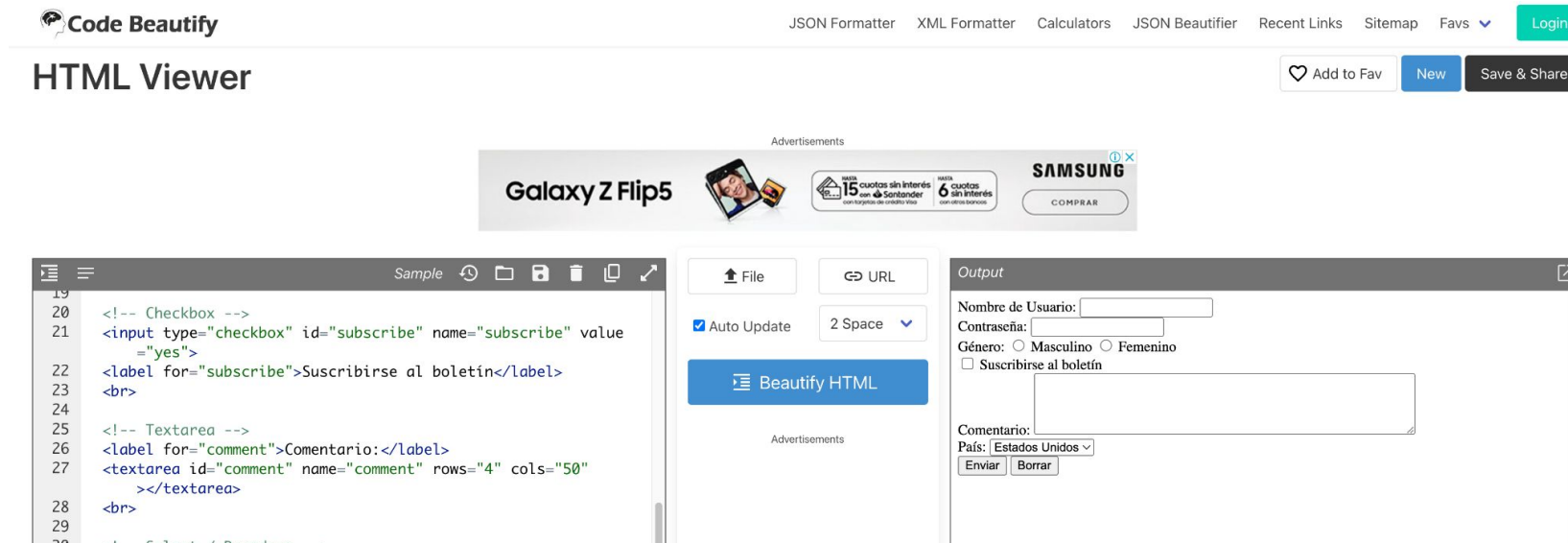
- **<!DOCTYPE html>:** Define el estándar del documento; en este caso, HTML5.
- **<html>:** Contenedor principal que encapsula todo el contenido de la página web.
- **<head>:** Contiene meta-información, enlaces a estilos CSS, scripts y el título del documento.
 - **<title>:** Título que aparece en la pestaña del navegador.
- **<body>:** Contiene todo el contenido visible de la página, como texto, imágenes y enlaces.

Notas adicionales

- Es crucial seguir esta estructura básica para asegurar que la página web se muestre correctamente en diferentes navegadores.
- Omitir alguno de estos elementos básicos puede resultar en problemas de visualización o de funcionalidad.

HTML: Cómo lo podemos probar?

- **VSCode + Navegador:** Podemos utilizar un editor de texto y guardar el archivo con extensión .html para ver el formulario en un navegador web.
- **Con Code Beautify:**
 - <https://codebeautify.org/htmlviewer>



Elementos y Etiquetas HTML

¿Qué son los Elementos y Etiquetas HTML?

- Un **elemento HTML** está formado por una etiqueta de apertura, el contenido y una etiqueta de cierre.
- Las **etiquetas HTML** son el código que rodea al contenido y le da significado.

Componentes de una Etiqueta HTML:

- **Etiqueta de Apertura:** Se coloca al principio del elemento.
 - Ejemplo: `<p>`
- **Contenido:** Este es el contenido real que se mostrará en la página web.
 - Ejemplo: Este es un párrafo.
- **Etiqueta de Cierre:** Se coloca al final del elemento.
 - Ejemplo: `</p>`

```
<p>Este es un párrafo.</p>
```

En este caso, `<p>` es la etiqueta de apertura, Este es un párrafo. es el contenido, y `</p>` es la etiqueta de cierre.

```

```

En este caso, el elemento `` es "autocerrado", es decir, no necesita una etiqueta de cierre. Algunas etiquetas, como `
` y `<hr>`, también son autocerradas.

Atributos HTML

¿Qué son los Atributos HTML?

- Los atributos proporcionan información adicional sobre un elemento HTML y se colocan dentro de la etiqueta de apertura.
- Se usan para definir propiedades como enlaces, fuentes de imágenes, y más.

Ejemplo de Atributos en la Etiqueta <a>:

```
<a href="https://www.ejemplo.com" target="_blank">Visitar sitio</a>
```

- **href:** Especifica la URL a la que enlaza el elemento.
- **target:** Define cómo se abrirá el nuevo recurso (e.g., en una nueva ventana/tab con `_blank`).

Ejemplo de Atributos en la Etiqueta :

```

```

- **src:** Indica la ruta de la imagen.
- **alt:** Proporciona una descripción textual de la imagen para accesibilidad.
- **width:** Establece el ancho de la imagen en píxeles.

Atributos HTML Comunes en Diversas Etiquetas

```
<a href="https://www.ejemplo.com" class="mi-clase" id="mi-id">Enlace</a>
```

- **class:** Permite asignar una o más clases al elemento para su estilización con CSS.
- **id:** Proporciona una identificación única para el elemento dentro de la página.
- **style:** Aplica estilos CSS directamente dentro del elemento HTML.

Listas HTML

- HTML ofrece dos tipos principales de listas:
 - ordenadas ()
 - desordenadas ().
- Los elementos de la lista se definen con la etiqueta .

```
<!-- Lista Desordenada -->
<ul>
  <li>Elemento 1</li>
  <li>Elemento 2</li>
</ul>

<!-- Lista Ordenada -->
<ol>
  <li>Primer elemento</li>
  <li>Segundo elemento</li>
</ol>
```

- Elemento 1
- Elemento 2

1. Primer elemento
2. Segundo elemento

Tablas HTML

- Las tablas se crean con la etiqueta **<table>**,
- y se componen de
 - filas (**<tr>**)
 - celdas (**<td>**)
 - encabezados (**<th>**)

```
<table border="1">
  <tr>
    <th>Encabezado 1</th>
    <th>Encabezado 2</th>
  </tr>
  <tr>
    <td>Celda 1</td>
    <td>Celda 2</td>
  </tr>
</table>
```

Encabezado 1	Encabezado 2
Celda 1	Celda 2

Formularios en HTML

- Permite al usuario ingresar y enviar información al servidor o manipularla con JavaScript.
- Los formularios son muy versátiles y pueden ser usados para diversas aplicaciones, desde simples búsquedas hasta autenticación de usuario o encuestas en línea.
- Los formularios se definen con la etiqueta **<form>**.
- y pueden contener diferentes tipos de
 - elementos de entrada (**<input>**, **<textarea>**, **<select>**, etc.).
 - y etiquetas (**<label>**) (son importantes para la accesibilidad en los formularios).

```
<form action="/submit">
  <label for="nombre">Nombre:</label>
  <input type="text" id="nombre" name="nombre">
  <input type="submit" value="Enviar">
</form>
```

Nombre:

Formularios en HTML

```
<form action="/submit" method="post">
  <!-- Texto -->
  <label for="username">Nombre de Usuario:</label>
  <input id="username" type="text" name="username" required>
  <br>

  <!-- Contraseña -->
  <label for="password">Contraseña:</label>
  <input id="password" type="password" name="password" required minlength="8">
  <br>

  <!-- Radio Buttons -->
  <label>Género:</label>
  <input type="radio" id="male" name="gender" value="male">
  <label for="male">Masculino</label>
  <input type="radio" id="female" name="gender" value="female">
  <label for="female">Femenino</label>
  <br>

  <!-- Checkbox -->
  <input type="checkbox" id="subscribe" name="subscribe" value="yes">
  <label for="subscribe">Suscribirse al boletín</label>
  <br>

  <!-- Textarea -->
  <label for="comment">Comentario:</label>
  <textarea id="comment" name="comment" rows="4" cols="50"></textarea>
  <br>

  <!-- Select / Dropdown -->
  <label for="country">País:</label>
  <select id="country" name="country">
    <option value="usa">Estados Unidos</option>
    <option value="uk">Reino Unido</option>
    <option value="other">Otro</option>
  </select>
  <br>

  <!-- Botones -->
  <input type="submit" value="Enviar">
  <input type="reset" value="Borrar">
</form>
```

Nombre de Usuario:

Contraseña:

Género: ☐ Masculino ☐ Femenino

☐ Suscribirse al boletín

Comentario:

País:

Descripción de Elementos:

- Campo de texto y de contraseña con validación (required, minlength).
- Radio buttons para seleccionar el género.
- Una casilla de verificación para suscribirse al boletín.
- Un área de texto para comentarios.
- Una lista desplegable para seleccionar el país.
- Botones para enviar y borrar el formulario.

La Importancia del HTML Semántico

¿Qué es el HTML Semántico?

El HTML Semántico se refiere al uso de etiquetas HTML que describen de manera precisa su contenido, no solo su apariencia o función. Estas etiquetas proporcionan contexto tanto a los navegadores como a los desarrolladores.

Importancia del HTML Semántico

- **Accesibilidad:** Mejora la accesibilidad al facilitar la interpretación del contenido por lectores de pantalla.
- **SEO:** Mejora el rendimiento en motores de búsqueda al proporcionar una mejor estructura.
- **Mantenibilidad:** Facilita la lectura y el mantenimiento del código al hacerlo más autoexplicativo.

Etiquetas Semánticas Comunes

- **<header>:** Se utiliza para contener elementos de encabezado como títulos y navegación.
- **<footer>:** Se utiliza para contener información que se sitúa en la parte inferior de la página o de una sección.
- **<article>:** Define un contenido independiente y autocontenido que puede ser distribuido y sindicado.
- **<section>:** Representa una sección de contenido con un tema relacionado, típicamente con un encabezado.

HTML Semántico: Ejemplo

```
<!DOCTYPE html>
<html>
<head>
  <title>HTML Semántico</title>
</head>
<body>
  <header>
    <h1>Encabezado de la Página</h1>
  </header>

  <section>
    <h2>Primera Sección</h2>
    <p>Contenido de la primera sección.</p>
  </section>

  <article>
    <h2>Un Artículo</h2>
    <p>Contenido del artículo.</p>
  </article>

  <footer>
    <p>Derechos Reservados, 2023.</p>
  </footer>
</body>
</html>
```

CSS: Importancia en Desarrollo Web



- **Separación de Contenido y Diseño:** Permite separar el contenido (HTML) del diseño y estilo (CSS), facilitando el mantenimiento y la accesibilidad.
- **Consistencia:** Un solo archivo CSS puede controlar múltiples páginas, asegurando que el diseño sea coherente en todo el sitio web.
- **Flexibilidad y Control:** Ofrece un gran nivel de control sobre la apariencia, incluyendo layout, colores, tipografías y más.
- **Interacción y Experiencia del Usuario:** Con CSS se pueden crear efectos visuales atractivos y experiencias de usuario más interactivas.
- **Optimización y Rendimiento:** Menor cantidad de código y archivos más pequeños se traducen en tiempos de carga más rápidos para el usuario.
- **Responsive Design:** CSS hace posible crear diseños que se adaptan a diferentes tamaños de pantalla, desde móviles hasta pantallas de escritorio.

HTML sin CSS

```
<!DOCTYPE html>
<html>
<head>
  <title>Página sin CSS</title>
</head>
<body>

<h1>Bienvenido a mi página</h1>

<p>Este es un párrafo de texto simple.</p>

<ul>
  <li>Elemento 1</li>
  <li>Elemento 2</li>
</ul>

</body>
</html>
```

Bienvenido a mi página

Este es un párrafo de texto simple.

- Elemento 1
- Elemento 2

HTML Con CSS

```
<!DOCTYPE html>
<html>
<head>
  <title>Página con CSS</title>
  <style>
    body {
      font-family: Arial, sans-serif;
      background-color: #f4f4f4;
      margin: 20px;
      padding: 20px;
      border: 2px solid #ccc;
    }

    h1 {
      color: #333366;
      border-bottom: 1px solid #ccc;
    }

    p {
      color: #666;
      font-size: 1.1em;
    }

    ul {
      background-color: #eee;
      padding: 10px;
    }

    ul li {
      color: #333;
    }
  </style>
</head>
<body>
  <h1>Bienvenido a mi página</h1>
  <p>Este es un párrafo de texto con estilo.</p>
  <ul>
    <li>Elemento 1</li>
    <li>Elemento 2</li>
  </ul>
</body>
</html>
```

Bienvenido a mi página

Este es un párrafo de texto con estilo.

- Elemento 1
- Elemento 2

Sintaxis de CSS: Selector, Propiedad y Valor

- Selector: Identifica el elemento HTML que se quiere estilizar.

```
/* El selector es "p", para los párrafos */  
p { /*...*/ }
```

- Propiedad: Especifica qué característica del elemento se va a cambiar.

```
/* "color" y "font-size" son propiedades */  
color: red;  
font-size: 16px;
```

- Valor: Indica el valor que tomará la propiedad especificada.

```
/* "red" y "16px" son valores */  
color: red;  
font-size: 16px;
```



Formas de Incluir CSS en HTML

Existen tres maneras principales de incluir estilos CSS en una página HTML:

- **Inline CSS:** Se aplica directamente en el elemento HTML usando el atributo style.

```
<p style="color: red;">Texto en rojo</p>
```

Internal CSS: Se coloca dentro del elemento `<head>` del HTML, utilizando la etiqueta `<style>`.

- **Internal CSS:** Se define el estilo para toda la página dentro de la etiqueta `<style>`.

```
<head>
  <style>
    p {
      color: red;
    }
  </style>
</head>
```

- **External CSS:** Se almacena en un archivo separado .css y se vincula al HTML mediante la etiqueta `<link>`.

```
p {
  color: red;
}
```

```
<head>
  <link rel="stylesheet" type="text/css"
  href="styles.css">
</head>
```

Formas de Incluir CSS: Pros y Contras

- **Inline CSS:**
 - Pros: Rápido y específico.
 - Contras: No reutilizable, dificulta el mantenimiento.
- **Internal CSS:**
 - Pros: Fácil de manejar en páginas pequeñas.
 - Contras: No es reutilizable en múltiples páginas.
- **External CSS:**
 - Pros: Reutilizable, fácil de mantener, separa el contenido del diseño.
 - Contras: Requiere un archivo externo.

Selectores CSS Básicos

- **Selector de Tipo:** Selecciona elementos HTML por su nombre de etiqueta.

Ejemplo:

```
/* Esto afectará a todos los párrafos en la página */  
p {  
  color: blue;  
}
```

```
<p id="miParrafo">Este es un texto en negrita.</p>
```

- **Selector de ID:** Selecciona un elemento específico basado en su atributo id.

Ejemplo:

```
/* Esto afectará sólo al elemento con id="miParrafo" */  
#miParrafo {  
  font-weight: bold;  
}
```

- **Selector de Clase:** Selecciona uno o más elementos que tienen un atributo class específico.

Ejemplo:

```
/* Esto afectará a todos los elementos con class="textoDestacado" */  
.textoDestacado {  
  font-style: italic;  
}
```

```
<p class="textoDestacado">Este es un texto en  
cursiva.</p>
```


Selectores CSS Básicos: Cuándo usar cada tipo?

- **Selectores de Tipo:**
 - Útil para dar un estilo general a un tipo específico de elemento en toda la página.
- **Selectores de ID:**
 - Útil para dar un estilo específico a un único elemento; los ID deben ser únicos en una página.
- **Selectores de Clase:**
 - Útil para dar un estilo que se pueda aplicar a varios elementos en la página; las clases pueden reutilizarse.

Propiedades CSS Básicas

Propiedades CSS más Comunes y su Uso

- **Color (color):** Controla el color del texto dentro de un elemento.

Ejemplo:

```
p {  
  color: red; /* Texto en rojo */  
}
```

- **Tamaño de Fuente (font-size):** Controla el tamaño de la fuente del texto.

Ejemplo:

```
p {  
  font-size: 16px; /* Tamaño de fuente de 16 píxeles */  
}
```

- **Margen (margin):** Controla el espacio fuera del elemento, entre el borde del elemento y los elementos circundantes.

Ejemplo:

```
p {  
  margin: 10px; /* Margen de 10 píxeles alrededor del párrafo */  
}
```

Propiedades CSS Básicas

- **Relleno (padding):** Controla el espacio entre el contenido del elemento y su borde.

Ejemplo:

```
p {  
  padding: 10px;  /* Relleno de 10 píxeles dentro del párrafo */  
}
```

- **Borde (border):** Define el borde alrededor del elemento.

Ejemplo:

```
p {  
  border: 1px solid black;  /* Borde sólido, 1 píxel de grosor, color negro */  
}
```

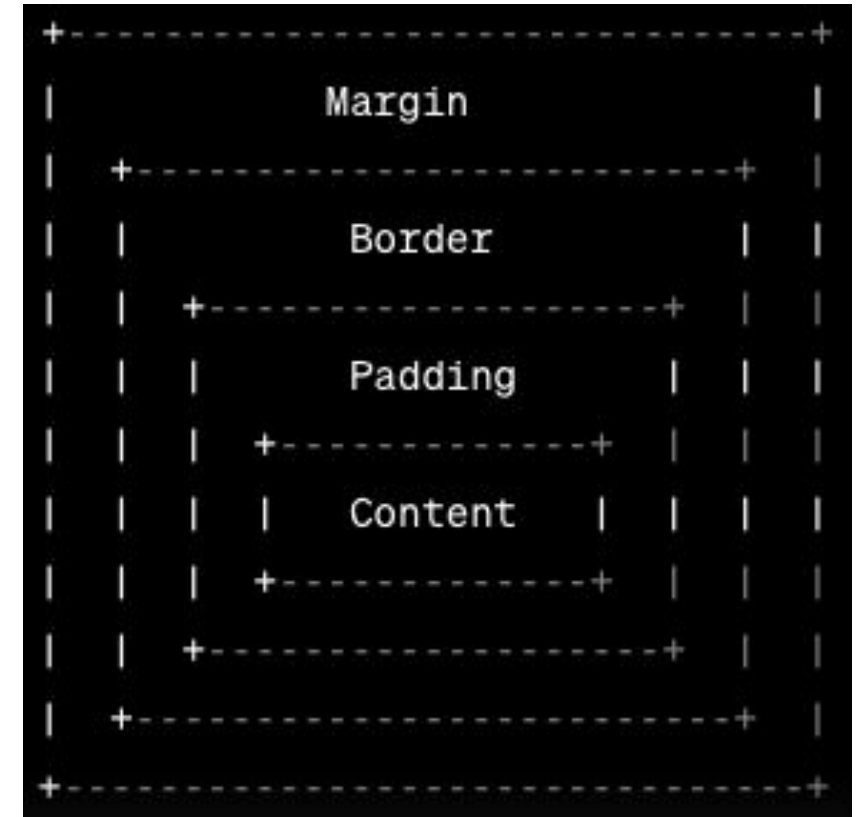
Ejemplo completo con todas las propiedades

```
p {  
  color: red;  
  font-size: 16px;  
  margin: 10px;  
  padding: 10px;  
  border: 1px solid black;  
}
```

Modelo de Caja (Box Model)

El Modelo de Caja es un concepto crucial en CSS que describe cómo se estructuran los elementos en la página web. Consiste en:

- **Margen (margin):** Es el espacio que rodea un elemento, separándolo de otros elementos.
- **Borde (border):** Es la línea que rodea el padding y el contenido.
- **Relleno (padding):** Es el espacio entre el borde y el contenido real del elemento.
- **Contenido (content):** Es el área donde se muestra el contenido real, como texto, imágenes, etc.



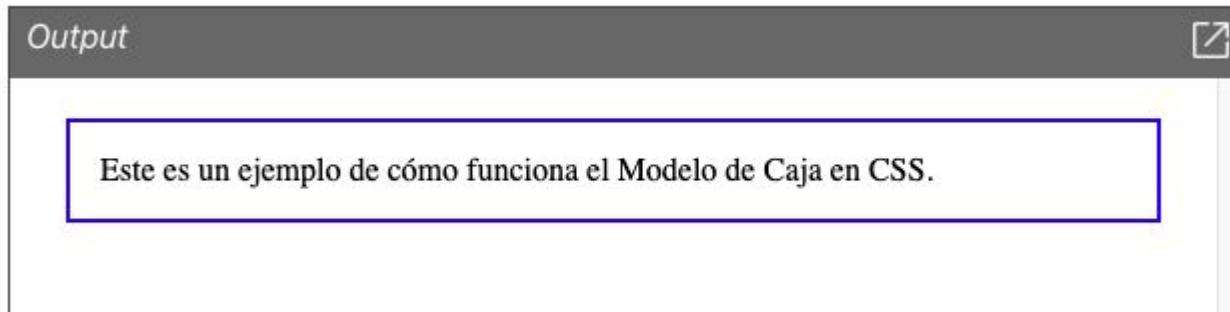
Ejemplo de Código CSS



```
div {  
  margin: 20px;           /* Espacio externo */  
  border: 2px solid blue; /* Línea que rodea el padding y el contenido */  
  padding: 15px;          /* Espacio interno */  
  /* El contenido se define en el HTML */  
}
```



```
<div>Este es un ejemplo de cómo funciona el Modelo de Caja en CSS.</div>
```



Introducción al Posicionamiento y Layout en CSS

- **Position (position):** Define cómo se posicionará un elemento en el layout. Las opciones más comunes son static, relative, absolute, y fixed. Ejemplo:

```
/* Posiciona un elemento de manera absoluta dentro de su contenedor */  
div {  
  position: absolute;  
  top: 20px;  
  left: 30px;  
}
```

- **Float (float):** Utilizado para flotar un elemento a la izquierda o a la derecha de su contenedor.

Ejemplo:

```
/* Flota una imagen a la derecha del texto */  
img {  
  float: right;  
}
```

- **Flexbox:** Un modelo de diseño que permite diseñar estructuras complejas con un diseño más predecible. Ejemplo:

```
/* Crea un contenedor flexbox */  
.container {  
  display: flex;  
}
```

- **Grid:** Un sistema de layout bidimensional que transforma el contenedor en un plano de cuadrícula.

Ejemplo:

```
/* Crea un contenedor de cuadrícula */  
.container {  
  display: grid;  
}
```

Posicionamiento y Layout en CSS:



- **Position y Float:** Útiles para diseños más simples y casos específicos.
- **Flexbox:** Excelente para diseños unidimensionales (una fila o una columna).
- **Grid:** Ideal para diseños bidimensionales (filas y columnas).

Unidades de Medida en CSS

- **Píxeles (px):** Una unidad de medida fija que no escala con otros elementos.

Ejemplo:

```
/* Fija el tamaño del texto en 16 píxeles */  
p {  
  font-size: 16px;  
}
```

- **Porcentajes (%):** Una unidad relativa que se calcula en relación con otro valor, usualmente el elemento padre.

Ejemplo:

```
/* Fija el ancho del elemento al 50% del elemento padre */  
div {  
  width: 50%;  
}
```

- **Em (em):** Relativa al tamaño de fuente del elemento en el que se utiliza, o del elemento padre si se aplica a propiedades como width. Ejemplo:

```
/* Fija el tamaño del texto al doble del tamaño del texto del elemento padre */  
p {  
  font-size: 2em;  
}
```

- **Rem (rem):** Similar a em, pero siempre relativo al tamaño de fuente del elemento raíz (html).

Ejemplo:

```
/* Fija el tamaño del texto basado en el tamaño del elemento raíz */  
p {  
  font-size: 1.5rem;  
}
```

Unidades de Medida en CSS: Comparación y Uso



- **px:** Útil para detalles precisos, pero no escala bien.
- **%:** Excelente para diseños responsivos, pero puede complicarse con anidamiento.
- **em:** Útil para diseños escalables, pero puede complicarse con elementos anidados.
- **rem:** Ideal para un diseño uniforme y escalable basado en un tamaño de fuente raíz.

Diseño Responsivo y Media Queries

¿Qué es el Diseño Responsivo?

El diseño responsivo se refiere a la práctica de crear sitios web que se adaptan automáticamente al tamaño de la pantalla del dispositivo. Esto asegura una experiencia de usuario óptima sin importar cómo se accede al sitio web.

Importancia del Diseño Responsivo

- Mejora la experiencia de usuario.
- Beneficia el SEO.
- Amplía el alcance a usuarios de dispositivos móviles.

Recomendaciones Generales

- Utiliza unidades relativas como % y em para un diseño más flexible.
- Comienza con un diseño móvil primero ("Mobile First") y escala hasta pantallas más grandes.

Media Queries

Las Media Queries son una técnica de CSS que permite aplicar estilos específicos según las características del dispositivo, como la anchura, la altura o la orientación de la pantalla.

```
/* Estilos para pantallas mayores de 768px */
@media (min-width: 768px) {
  body {
    font-size: 18px;
  }
  h1 {
    font-size: 36px;
  }
}

/* Estilos para pantallas menores de 768px */
@media (max-width: 767px) {
  body {
    font-size: 16px;
  }
  h1 {
    font-size: 30px;
  }
}
```

Herramientas y Librerías Comunes

- **Bootstrap:** Es un marco (framework) de diseño de código abierto que se centra en simplificar el desarrollo de páginas web responsivas y móviles.
 - **¿Por qué es útil?:** Ofrece componentes pre-diseñados que aceleran el desarrollo y garantizan un diseño coherente.
- **Tailwind CSS:** Un framework de CSS de bajo nivel que permite crear diseños personalizados de forma eficiente.
 - **¿Por qué es útil?:** Permite más control sobre el diseño al utilizar clases de utilidad, optimizando el tiempo de desarrollo.
- **SASS (Syntactically Awesome Stylesheets):** Es un preprocesador de CSS que añade funcionalidades como variables, anidamiento y funciones.
 - **¿Por qué es útil?:** Facilita la organización del código y reutilización de estilos, haciendo más eficiente el mantenimiento del código.

Bueno, Vamo a Codea!!!!

Vamos a crear una página web sencilla que incluya un encabezado, una sección de contenido y un pie de página, usando HTML y CSS. La página debe ser responsiva y cambiar su diseño dependiendo del tamaño de la pantalla.



```
<!DOCTYPE html>
<html>
<head>
  <link rel="stylesheet" type="text/css" href="styles.css">
  <title>Mi Página Web</title>
</head>
<body>
  <header>
    <h1>Bienvenidos a Mi Página Web</h1>
  </header>
  <section>
    <h2>Contenido Principal</h2>
    <p>Este es el cuerpo de la página.</p>
  </section>
  <footer>
    <p>Copyright 2023</p>
  </footer>
</body>
</html>
```

```
/* Estilos para todos los dispositivos */
body {
  font-family: Arial, sans-serif;
}
header {
  background-color: #f2f2f2;
  padding: 20px;
}
section {
  margin: 20px;
}
footer {
  background-color: #f2f2f2;
  padding: 10px;
}

/* Estilos responsivos */
@media (max-width: 600px) {
  body {
    font-size: 16px;
  }
  header, section, footer {
    padding: 10px;
  }
}
```

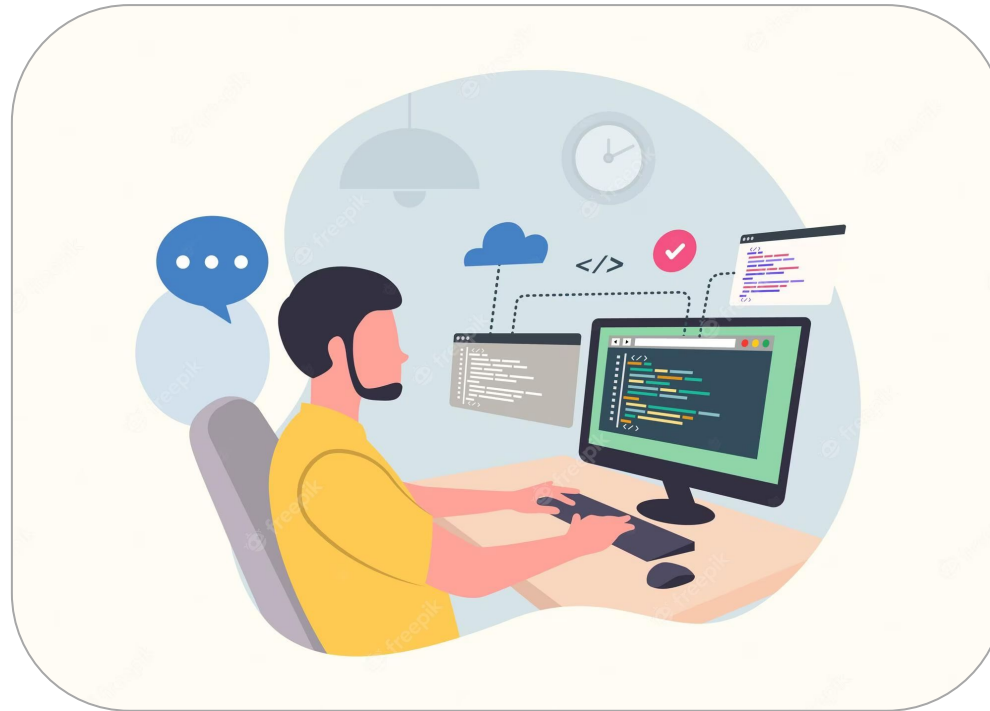
Próximos Pasos para Profundizar

- **Práctica, Práctica, Práctica:** La mejor manera de aprender es haciendo. Empiece a crear sus propios proyectos o contribuya en proyectos de código abierto.
- **Aprender JavaScript:** Es el siguiente paso lógico para añadir interactividad a sus sitios web.
- **Frameworks y Librerías:** A medida que se sienta más cómodo con HTML, CSS y JavaScript, explore frameworks populares como React, Angular o Vue.
- **Cursos Avanzados y Tutoriales:** Hay una cantidad infinita de recursos en línea para continuar su educación en desarrollo frontend.
- **Unirse a Comunidades:** Conéctese con otros desarrolladores para aprender, compartir recursos y mejores prácticas.



Actividad 2: Paso a Paso HTML + CSS

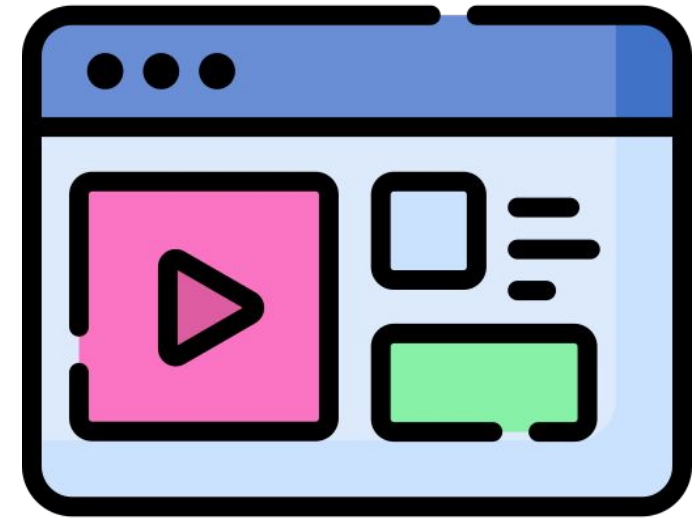
- Seguir las instrucciones de la actividad publicada en la UVE.



Próxima Clase

Introducción a JavaScript

- ¿Qué es JavaScript y por qué es importante?
- Sintaxis básica de JavaScript.
- Variables, tipos de datos y operadores.
- Estructuras de control de flujo.



MUCHAS TOTALES

ANDÉN
Centro de Innovación
y Emprendimientos Tecnológicos

SECRETARÍA DE
EXTENSIÓN
UNIVERSITARIA
UTN - FRC

SEU

UTN
Facultad Regional Córdoba

Agencia
**CÓRDOBA
JOVEN**



CÓRDOBA
entre todos