



FRONT-END JS

Clase n° 1 | Conceptos básicos de
HTML

¡Les damos la bienvenida!

 Vamos a comenzar a grabar la clase.



✨**Conozcamos al Staff**✨

✨ Staff Académico ✨



Maximiliano Torres

Rol: **TUTOR**

Descripción del rol: **Consultas administrativas, asistencia, acceso Aula Virtual, ...**

Contacto: **maximiliano.torres@bue.edu.ar**



Norberto Llopiz

Rol: **INSTRUCTOR**

Descripción del rol: **Dictado de clases, consultas temario, ...**


Contacto: **norberto.llopiz@bue.edu.ar**


Acuerdos de convivencia

 Puntualidad e Informar por mail o Discord inasistencias

 Mantener el micrófono apagado durante la clase.

 Para participar, levantar la mano o escribir en el chat.

 Mantener el respeto y los buenos modales, al hablar y utilizar el chat.

 No usar el chat para spam y cuestiones que no tengan que ver con la clase.

[Pautas de convivencia](#)

Metodología de cursada



Canales de comunicación

E-mail y Discord

Periódicamente nos estaremos comunicando al **mail con el que te inscribiste**, para que tengas toda la información importante del curso

Es muy importante sumarse al servidor de su comisión ya que enviaremos mensajes y compartiremos material a través de este medio.

discord.com

+ Invitación a Discord (solo la primera vez que ingresan): [Link](#)

Material de estudio: Aula Virtual

Link: aulasvirtuales.bue.edu.ar

Acceso:

- El usuario es tu número de DNI (sin puntos ni guiones).
- La contraseña es: Prueba!123

Si cuentan con una contraseña de otro curso puedes usarla para ingresar

Por **problemas** con el Aula Virtual: [Formulario](#)

Material de estudio: Clases grabadas

 Clases grabadas en YouTube: [Link](#)

Metodología de Evaluación

Nuestro objetivo es prepararte para enfrentar los desafíos del siglo XXI y facilitar tu inserción en el mercado laboral. Para lograrlo, hemos desarrollado un programa que enfatiza la ejercitación constante y el seguimiento continuo. A continuación, te explicamos cómo serás evaluado a lo largo de la cursada:



Ejercicios Prácticos



Cuestionario



Pre-Entrega



Proyecto Final



La información detallada de cada evaluación está disponible en el apartado “Introducción” dentro del Campus Virtual.

Proyecto Integrador Final

Obligatorio | Entregable

¿Cómo se construye el proyecto integrador?

Al final de la cursada, serás evaluado mediante la entrega de un **Proyecto Integrador**, que es fundamental para completar el curso y cumplir con los requisitos de egreso. Este proyecto se construirá de manera progresiva, combinando la resolución de **Ejercicios Prácticos** y la **Pre-Entrega de Proyecto** presentes a lo largo de la cursada.

Las consignas a evaluar en este proyecto final integrador estará constituido por la resolución de los **"Ejercicios Prácticos"** presentes a lo largo de la cursada en función del proyecto integrador.

After Class

El espacio "After Class" está diseñado para ofrecerte apoyo adicional y facilitar tu progreso durante la cursada. Aunque es opcional, te recomendamos que utilices este espacio para optimizar tu aprendizaje y el desarrollo de tu proyecto integrador.

✨ Beneficios de asistir:

Consultas y Asesoría: Aprovecha este tiempo para resolver cualquier duda o consulta que tengas sobre el contenido de las clases, ejercicios prácticos, o cualquier aspecto relacionado con tu proyecto integrador. Podrás recibir orientación más personalizada de los/as instructores y obtener aclaraciones que te ayudarán a comprender mejor los conceptos y mejorar tu desempeño.




Frecuencia: Una vez por semana en un día distinto y en la franja horaria de la cursada regular.

Enfoque inmersivo

En este curso utilizaremos un enfoque inmersivo como herramienta fundamental para que puedan experimentar los roles más solicitados en el mercado. Este curso va más allá de la enseñanza de programación; su objetivo es prepararlos para enfrentarse a problemas del mundo real en entornos laborales.

Este enfoque nos ayuda a:

- 1 Conectar con situaciones reales.**
- 2 Desarrollar pensamiento crítico.**
- 3 Hacer el aprendizaje más significativo.**
- 4 Vivir la experiencia de un profesional.**

 Cada ejercicio en el curso será como un capítulo de una historia. Ustedes serán los protagonistas, y su misión será utilizar las herramientas brindadas en las clases para superar los retos que se les presenten.

Índice

Clase 01

1. Conceptos básicos de HTML

- Conceptos básicos sobre Full Stack
- Instalación del software
- Explicación del proyecto final
- Introducción a HTML: Etiquetas semánticas
- Etiquetas básicas más comunes

Clase 02

2. Listas, Rutas, Multimedia, Tablas

- Listas y enlaces
- Rutas absolutas y relativas
- Elementos en bloque y en línea
- Multimedia: imágenes, video, audio, iframes
- Accesibilidad (etiqueta alt, subtítulos en video)
- Tablas

¿Qué es un Full Stack?

Full Stack: Visión General

Capacidad de un desarrollador para trabajar en todas las capas de una aplicación web, desde la interfaz de usuario (Front-End) hasta la lógica del servidor y la base de datos (Back-End).

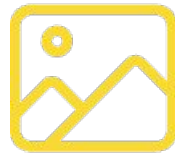
Áreas

Incluye Front-End y Back-End, cubriendo toda la estructura web.

Habilidades

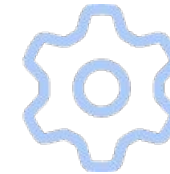
Requiere conocimientos en múltiples tecnologías y herramientas.

Front-End vs Back-End



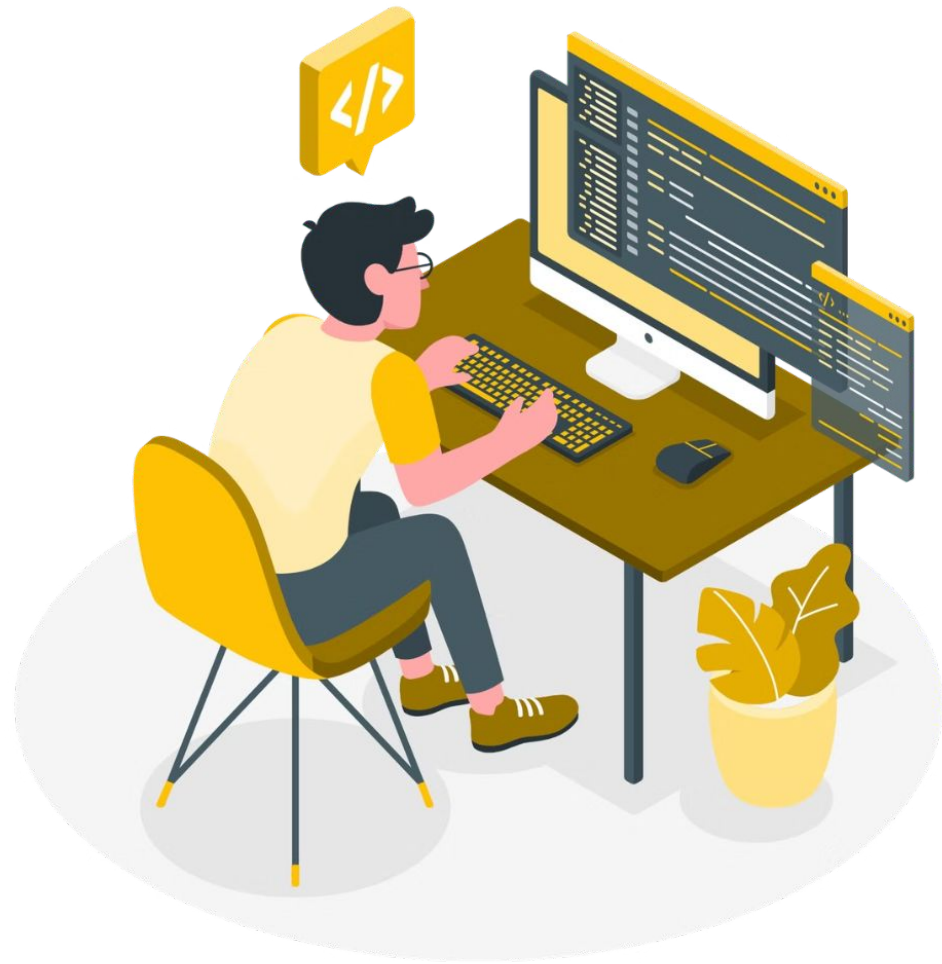
Front-End

- Interfaz de usuario visible.
- Parte de una aplicación web que interactúa directamente con la persona
- Utiliza HTML, CSS y JavaScript.



Back-End

- Lógica del servidor y bases de datos.
- Parte de una aplicación web que procesa solicitudes y ejecuta operaciones.
- Usa lenguajes como Python o Java, PHP, Node.js, entre otros.



Tecnologías Front-End

</> **HTML, CSS, y JavaScript**

Tecnologías fundamentales para la interfaz de usuario



Frameworks y Librerías

React, Angular, Vue.js ayudan a construir aplicaciones más complejas y dinámicas



Preprocesadores de CSS

Sass o Less permiten escribir CSS de manera más eficiente.

Tecnologías Back-End



Bases de Datos

MySQL, PostgreSQL, MongoDB, SQL Server.



Lenguajes de Programación

Python, Java, PHP, Ruby, Node.js.



Servidores

Apache, Nginx, servidores en la nube como AWS y Azure



Instalando Visual Studio Code

Visual Studio Code

Descarga

Visita code.visualstudio.com y descarga la versión para tu sistema operativo.

1

2

Instalación

Sigue las instrucciones en pantalla para instalar Visual Studio Code.

Configuración

Personaliza el editor según tus preferencias en
File > Preferences > Settings.

3

VER [TUTORIALES](#)

HTML

Introducción a HTML

HTML (HyperText Markup Language) es el lenguaje de marcado estándar para crear páginas web. HTML corresponde a la estructura principal de todas las páginas y sitios web.

Función

Estructura el contenido de una página web para su visualización en navegadores.

El navegador interpreta el código HTML y lo renderiza en pantalla para que las personas puedan ver y navegar por la página.



Etiquetas Semánticas en HTML

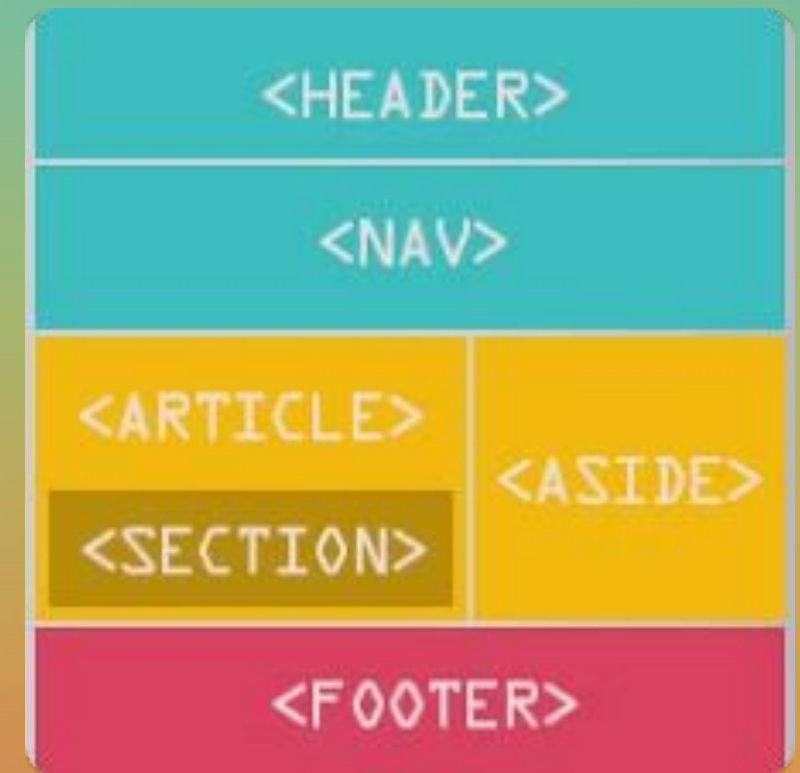
Son aquellas que describen el propósito de diferentes partes de una página web.

¿Para qué sirven?

Permiten estructurar mejor el contenido y que los navegadores interpreten fácilmente la jerarquía e importancia de la información en la página.

Ejemplos:

`<header>`, `<nav>`, `<main>`,
`<section>`, `<footer>`, entre
otras.



Etiquetas Semánticas Estructurales:

<header>

Encabezado de página o sección.
Incluye logotipos, títulos y navegación.

<nav>

Sección de navegación con enlaces.

<main>

Contenido principal del documento.

<footer>

Pie de página con derechos de autor y contactos.

<section>

Sección temática que agrupa elementos relacionados.

<article>

Contenido independiente, como un artículo o noticia.

<aside>

Información adicional, como anuncios o enlaces secundarios.

¡Vamos a la práctica!



Creá la estructura principal del documento:

1

<!DOCTYPE html>

Creá la estructura dentro de tu documento index.html, podes hacerlo con ayuda de emmet y el atajo `html:5`

2

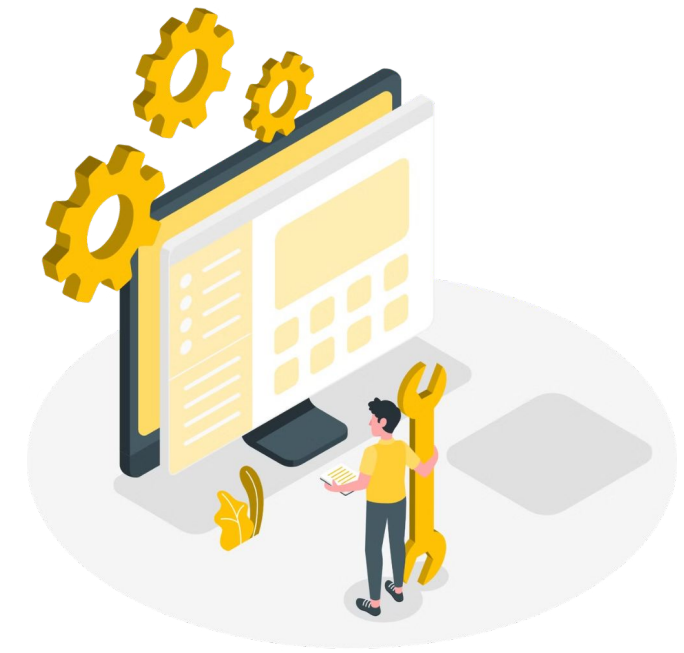
Agregá etiquetas semánticas estructurales

Dentro de la etiqueta `<body>` (`<header>`, `<nav>`, `<main>`, `<section>`, `<footer>`)

3

Resultado Final

Estructura básica principal para seguir construyendo tu proyecto HTML



Paso 1: Creá la estructura principal

- ❏ Creá la siguiente estructura dentro de tu index.html. En VSC si escribimos **!** y presionamos **Enter** este código se escribirá automáticamente.

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport"
content="width=device-width,
initial-scale=1.0">
    <title>Document</title>
</head>
<body>
</body>
</html>
```

Paso 2: Agregá etiquetas semánticas

📄 Dentro de la etiqueta `<body>`, agregá las siguientes etiquetas semánticas:

```
<header>
  <!-- Acá estamos dentro del header-->
</header>
<nav>
  <!-- Acá estamos dentro del nav-->
</nav>
<main>
  <!-- Acá estamos dentro del main-->
</main>
<footer>
  <!-- Acá estamos dentro del footer-->
</footer>
```

Etiquetas básicas

<Etiquetas Básicas más comunes en HTML>

Encabezados (<h1> - <h6>)

Son utilizados para definir títulos y subtítulos en base a la jerarquía e importancia.

Párrafos (<p>)

Se utiliza para definir párrafos de texto.

Negrita ()

Se utiliza para resaltar texto con importancia.

Itálica ()

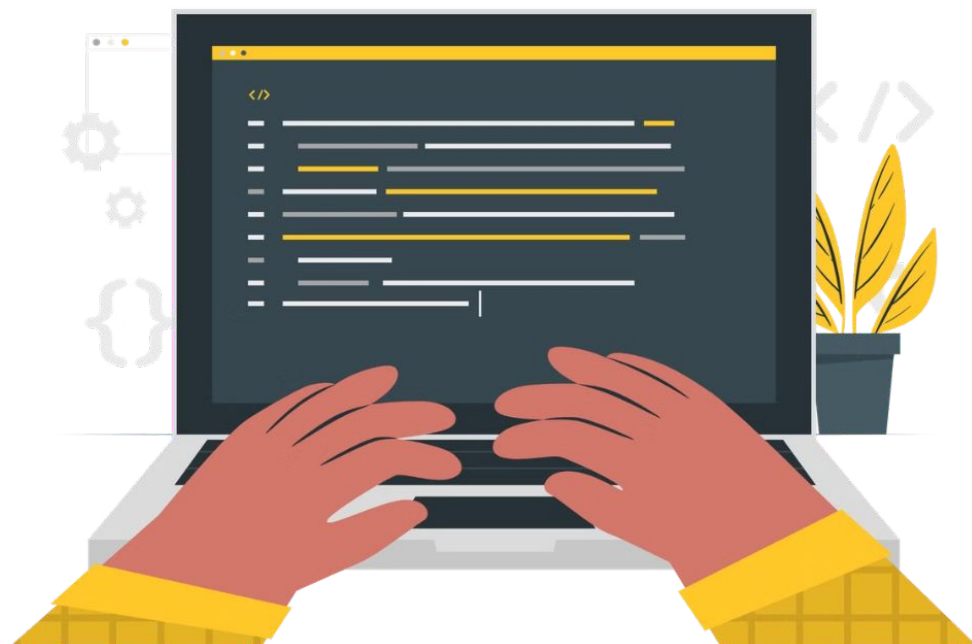
Se utiliza para enfatizar texto, mostrándolo en cursiva.

Tachado (<s>)

Se utiliza para mostrar texto tachado, generalmente para indicar que algo ha sido eliminado o corregido.

¡Vamos a la práctica!





Integración de Etiquetas Básicas en el Proyecto

1

En el `<header>`:

Agregá un título principal usando `<h1>`, seguido de un subtítulo con `<h2>`

2

En el `<main>`:

Crea una section e incluye un párrafo de introducción con `<p>`, y utiliza ``, ``, y `<s>` dentro de los textos según sea necesario.

3

En el `<footer>`:

Podés repetir algunos estilos de texto o añadir un párrafo con contacto.

Ejercicios Prácticos

Ejercicio Práctico #1

Optativo | No entregable

Estructura básica

Crear la estructura básica del proyecto, incluyendo las etiquetas `<header>`, `<nav>`, `<main>`, `<section>`, y `<footer>`.

Dentro de `<header>`, incluye un título `<h1>` y dentro de una `<section>` hace uso del resto de las etiquetas vistas hasta el momento.

Pasos a seguir:

- 1. Abrir Visual Studio Code:**
 - Abre Visual Studio Code y crea un nuevo archivo.
 - Guarda el archivo con el nombre `index.html`.
- 2. Escribir la estructura básica del documento HTML:**
 - Escribir el código HTML que conformará la estructura básica. El código debe incluir las etiquetas `<header>`, `<nav>`, `<main>`, y `<footer>`.
- 3. Guardar y visualizar en el navegador.**

Ejercicio Práctico #2

Optativo | No entregable

README.md

Agregar un archivo [README.md](#) en el proyecto, explicando brevemente de qué tratará la página que se va a desarrollar.

Para resolver este ejercicio:

Crear el archivo [README.md](#):

Dentro del mismo directorio donde guardaste index.html, creá un nuevo archivo llamado [README.md](#).

Escribir la descripción del proyecto:

Abrí el archivo [README.md](#) en Visual Studio Code y escribí una breve descripción del proyecto.

Ejemplo del contenido del README.md:

Nombre del Proyecto

Descripción:

Este proyecto es una página web básica desarrollada como parte de un curso de Front-End. La página está estructurada con HTML semántico y utiliza las etiquetas `<header>`, `<main>`, y `<footer>` para organizar el contenido. El objetivo es aprender a crear la estructura básica de una página web y prepararla para futuras mejoras con CSS y JavaScript.