«Talento Tech»

Front End Js

CLASE 1

Buenos Aires aprende
Agencia de Habilidades para el Futuro

Buenos Aires Ciudad

Buenos Aires aprende
Agencia de Habilidades para el Futuro

Clase N° 1 Conceptos Básicos de HTML

Temario:

1. **Conceptos básicos sobre Full Stack**
   - Explicación general de Full Stack y las áreas que abarca.
2. **Diferencias entre Front-End y Back-End**
   - Comparación entre las responsabilidades y tecnologías de Front-End y Back-End.
3. **Herramientas y tecnologías utilizadas en cada área**
   - Tecnologías clave en Front-End (HTML, CSS, JS, etc.) y Back-End (Bases de Datos, Servidores, APIs, etc.).
4. **Instalación y configuración de Visual Studio Code**
   - Introducción a Visual Studio Code como herramienta principal de desarrollo.
   - Paso a paso para la instalación y configuración básica de Visual Studio Code.
5. **Introducción a HTML**
   - ¿Qué es HTML y cuál es su función en el desarrollo web?
   - Diferencias entre una página web y un sitio web.
   - Importancia del estándar HTML5.
6. **Etiquetas Semánticas en HTML**
   - ¿Qué son las etiquetas semánticas?
   - ¿Para qué sirven?
   - Ejemplos y uso de etiquetas semánticas (`<header>`, `<nav>`, `<main>`, `<footer>`, `<section>`, `<article>`, `<aside>`).
   - Encabezados (`<h1>`, `<h2>`, `<h3>`, …)
   - Párrafos (`<p>`)
   - Negrita (`<strong>`)
   - Itálica (`<em>`)
   - Tachado (`<s>`)

Buenos Aires aprende
Agencia de Habilidades para el Futuro

Conceptos Básicos sobre Full Stack

Conceptos Básicos sobre Full Stack

¿Qué es Full Stack?

El término "Full Stack" se refiere a la 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).

Un desarrollador o desarrolladora Full Stack tiene conocimientos en varias tecnologías y herramientas que le permiten construir aplicaciones completas, abarcando desde la experiencia del usuario hasta la gestión de datos y la lógica del servidor. Comprende el flujo completo de desarrollo de una aplicación web, lo que le permite tener una visión global del proyecto y la capacidad de solucionar problemas que puedan surgir en cualquier parte del stack.

Diferencias entre Front-End y Back-End

Diferencias entre Front-End y Back-End

Front-End:

El Front-End es la parte de una aplicación web que interactúa directamente con la persona, todo lo que ella ve. El desarrollo Front-End involucra la creación de interfaces de usuario utilizando tecnologías como HTML, CSS y JavaScript.

- **HTML:** proporciona la estructura básica de la página web.
- **CSS:** se utiliza para aplicar estilos a la estructura HTML, como colores, fuentes y distribución en la página.
- **JavaScript:** añade interactividad a la página, permitiendo que los elementos respondan a acciones de la persona, como clics y desplazamientos.

Back-End:

Buenos Aires aprende
Agencia de Habilidades para el Futuro

El Back-End es la parte de una aplicación web que gestiona la lógica del servidor, las bases de datos y las interacciones entre el servidor y el cliente. Se encarga de procesar solicitudes, ejecutar operaciones en la base de datos y devolver respuestas al Front-End.

- **Servidor Web:** recibe las solicitudes del cliente (navegador) y las procesa. Esto puede involucrar la ejecución de scripts, la gestión de sesiones de usuario y la interacción con bases de datos.
- **Base de Datos:** aquí es donde se almacenan los datos de la aplicación, como la información de usuarios/as, productos, y transacciones.
- **Lenguajes de Programación:** El Back-End suele desarrollarse utilizando lenguajes como Python, Java, PHP, Node.js, entre otros.

Herramientas y Tecnologías en Front-End:

Herramientas y Tecnologías en Front-End:

- **HTML, CSS, y JavaScript:** las tecnologías fundamentales para la creación de la interfaz de usuario.
- **Frameworks y Librerías:** herramientas como React, Angular o Vue.js ayudan a construir aplicaciones más complejas y dinámicas.
- **Preprocesadores de CSS:** Sass o Less permiten escribir CSS de manera más eficiente y organizada.
- **Control de Versiones:** Git es esencial para el control de versiones, permitiendo a los desarrolladores realizar un seguimiento de los cambios en su código.

Buenos Aires aprende
Agencia de Habilidades para el Futuro

Herramientas y Tecnologías en Back-End:

Herramientas y Tecnologías en Back-End:

- **Lenguajes de Programación:** Python, Java, PHP, Ruby, Node.js.
- **Bases de Datos:** MySQL, PostgreSQL, MongoDB, SQL Server.
- **Frameworks:** Django, Ruby on Rails, Express.js.
- **Servidores:** Apache, Nginx, o servidores en la nube como AWS y Azure.
- **APIs y RESTful Services:** Creación de APIs para interactuar con el Front-End y otras aplicaciones.

Instalación y Configuración de Visual Studio Code

Instalación y Configuración de Visual Studio Code

Paso 1: Descarga e Instalación

- Visitá el sitio oficial de Visual Studio Code: https://code.visualstudio.com/
- Descargá la versión correspondiente a tu sistema operativo (Windows, macOS, Linux).
- Seguí las instrucciones de instalación en pantalla. El proceso es sencillo y no requiere configuración especial.

Paso 2: Configuración Básica

- **Abrir Visual Studio Code:** Una vez instalado, abrí el editor.
- **Configurar Preferencias:**
  - Ve a `File > Preferences > Settings`.
  - Desde aquí, podés personalizar el tema, fuentes, y otros aspectos visuales.

Tema 1: Introducción a HTML

Introducción a HTML

¿Qué es HTML?

HTML (HyperText Markup Language) es el lenguaje de marcado estándar utilizado para crear y estructurar contenido en la web. Define cómo se organizan los elementos en una página web y cómo deben mostrarse estos elementos en los navegadores.

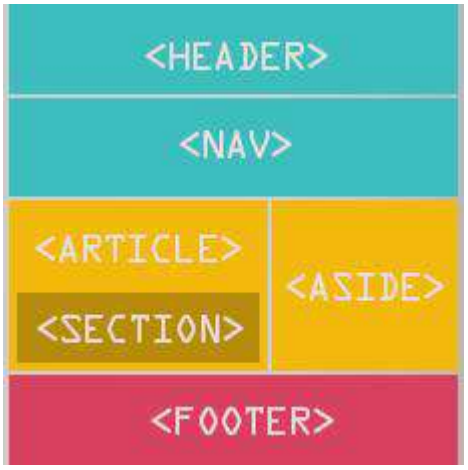
¿Cómo funciona HTML?

Buenos Aires aprende
Agencia de Habilidades para el Futuro

HTML permite a los y las desarrolladores crear la estructura básica de una página web. El navegador interpreta el código HTML y lo renderiza en pantalla para que las personas puedan ver y navegar por la página.

Tema 2: Etiquetas Semánticas en HTML

Etiquetas Semánticas en HTML



¿Qué son las Etiquetas Semánticas?

Las etiquetas semánticas en HTML5 son aquellas que describen el propósito de diferentes partes de una página web, tanto para los navegadores como para los motores de búsqueda. Estas etiquetas mejoran la accesibilidad y la optimización de la página, facilitando la comprensión del contenido.

¿Para qué sirven las Etiquetas Semánticas?

Las etiquetas semánticas ayudan a estructurar mejor el contenido y permiten que los navegadores y motores de búsqueda interpreten más fácilmente la jerarquía y la importancia de la información en la página.

Ejemplos y Uso de Etiquetas Semánticas:

- `<header>`: define el encabezado de una página o sección. Contiene elementos como logotipos, títulos y menús de navegación.
- `<nav>`: representa una sección de navegación con enlaces a otras partes de la página o a diferentes páginas.
- `<main>`: contiene el contenido principal del documento, que es único y esencial para la página.

- `<footer>`: define el pie de página, donde generalmente se encuentran los derechos de autor, enlaces adicionales y contactos.
- `<section>`: define una sección temática en el contenido que agrupa elementos relacionados.
- `<article>`: representa una pieza autónoma de contenido, como un artículo de un blog o una noticia.
- `<aside>`: contiene información adicional que puede estar relacionada con el contenido principal, como anuncios o enlaces secundarios.

Paso 2: Creá la estructura principal del documento:

Creá la siguiente estructura dentro de tu documento index.html

```
<!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 estructurales

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>
```

Paso 3: Resultado Final

Siguiendo los dos pasos anteriores tendrás como resultado la estructura básica principal para seguir construyendo tu proyecto HTML.

```
<!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>
     <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>

</body>
</html>
```

Tema 3: Etiquetas Básicas más comunes en HTML

Etiquetas Básicas más comunes en HTML

Encabezados (<h1> - <h6>)

Los encabezados son utilizados para definir títulos y subtítulos en una página web. HTML soporta seis niveles de encabezados, donde <h1> es el más importante y <h6> el menos relevante.

Ejemplo:

```
<h1>Este es el encabezado principal</h1>

<h2>Este es un subtítulo</h2>

<h3>Este es otro subtítulo</h3>
```

Buenos Aires aprende
Agencia de Habilidades para el Futuro

Párrafos (<p>)

La etiqueta <p> se utiliza para definir párrafos de texto. Es una de las etiquetas más utilizadas en HTML para organizar contenido textual.

Ejemplo:

```
<p>Esto es un párrafo. Aquí podés escribir cualquier texto que desees mostrar en tu página.</p>
```

Negrita ()

La etiqueta se utiliza para resaltar texto con importancia, mostrándolo en negrita.

Ejemplo:

```
<p>Este es un <strong>texto en negrita</strong>.</p>
```

Itálica ()

La etiqueta se utiliza para enfatizar texto, mostrándolo en cursiva.

Ejemplo:

```
<p>Este es un <em>texto en cursiva</em>.</p>
```

Tachado (<s>)

La etiqueta <s> se utiliza para mostrar texto tachado, generalmente para indicar que algo ha sido eliminado o corregido.

Ejemplo:

```
<p>Este es un <s>texto tachado</s>.</p>
```

Buenos Aires aprende
Agencia de Habilidades para el Futuro

Integración de Etiquetas Básicas en el Proyecto:

- En el `<header>`: agrega un título principal usando `<h1>`, seguido de un subtítulo con `<h2>`.
- En el `<main>`: incluye un párrafo de introducción con `<p>`, y utiliza `<strong>`, `<em>`, y `<s>` dentro de los textos según sea necesario.
- En el `<footer>`: podés repetir algunos estilos de texto o añadir un párrafo con contacto.

Ejemplo Integrado:

```
<html lang="es">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <title>Mi Primer Proyecto</title>
</head>
<body>
<header>
    <h1>Bienvenidos a Mi Proyecto</h1>
    <h2>Este es un subtítulo en el header</h2>
</header>
<nav>

</nav>
<main>
    <p>Este es un párrafo de introducción sobre el proyecto. Aquí podés
usar <strong>negrita</strong>, <em>cursiva</em>, y <s>tachado</s>.</p>
</main>
<footer>
    <p>Este es el pie de página con la información de contacto.</p>
</footer>
```

```
</body>
</html>
```

Ejercicio Práctico #1:

Crear la estructura básica del proyecto, incluyendo las etiquetas `<header>`, `<main>`, `<nav>` y `<footer>`. Dentro de `<header>`, incluye un título `<h1>` con el nombre del proyecto.

Para resolver este ejercicio, debés seguir estos pasos:

1. **Abrir Visual Studio Code:**
   - Abre Visual Studio Code y creá un nuevo archivo.
   - Guarda el archivo con el nombre index.html.
2. **Escribir la estructura básica del documento HTML:**
   - A continuación, debés escribir el código HTML que conformará la estructura básica de tu proyecto. El código debe incluir las etiquetas <header>, <nav> <main>, y <footer>.
3. **Guardar y visualizar en el navegador:**
   - Guardá los cambios y abre el archivo index.html en tu navegador para ver la estructura básica de la página web.

Ejercicio Práctico #2:

Agregar un archivo `README.md` en el proyecto, explicando brevemente de qué tratará la página que se va a desarrollar.

Para resolver este ejercicio, debés crear un archivo `README.md` en tu proyecto con una breve descripción del mismo.

1. **Crear el archivo `README.md`:**
   - Dentro del mismo directorio donde guardaste `index.html`, creá un nuevo archivo llamado `README.md`.
2. **Escribir la descripción del proyecto:**

- Abrí el archivo `README.md` en Visual Studio Code y escribí una breve descripción del proyecto.

Ejemplo del contendio 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.

Buenos Aires aprende

Agencia de Habilidades para el Futuro

Buenos Aires Ciudad