educación IT
CODE YOUR FUTURE

JavaScript desde cero

Módulo 4

HTML DOM

¿Qué es HTML?

HTML es el lenguaje de marcado más utilizado para la construcción de páginas web.

La sigla HTML significa *Hyper Text Markup Language* (*lenguaje de marcado de hipertexto*).

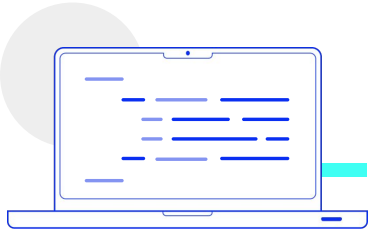
No es un lenguaje de programación, sino de **marcas con sentido**, a las que llamamos: ***etiquetas*** o ***Tags***. Las etiquetas le **indican al navegador qué es lo que debe mostrar en pantalla**.

Inicios de HTML

Este lenguaje de programación tuvo dos versiones, que nacieron a mediados de los 90’s, junto con la era de la Internet comercial. Luego, en 2008, nació la especificación (borrador) de lo que hoy se conoce como **HTML5**.

Esta versión, independiente de HTML, respetó la original del lenguaje pero se ocupó de **simplificar notablemente la estructura de etiquetas HTML originales,** haciéndolo más fácil de leer y aprovechar, además de mejorar su velocidad de carga en los navegadores web.

En 2014, llegó su primera versión final, y desde ese año, continúa incluyendo mejoras significativas, que **hacen que la web sea mejor y más fácil de programar**.



index.html

Todo proyecto web *frontend* debe contener, al menos, un documento HTML denominado: ***index.html***.

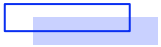
Este documento HTML contiene **dos secciones principales**:

- **<head>**
- **<body>**

<head>: almacena la información técnica que necesita el navegador web.

<body>: almacena los *tags* HTML que conformarán la estructura visible del documento, o sea, todo aquello que el usuario verá y con lo que va a interactuar.

Veamos un ejemplo en el siguiente *slide.*



```html
<!DOCTYPE html>
<html lang="es">
<head>
    <meta charset="UTF-8">
    <title>Document</title>
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
</head>
<body>
    <h1>Título principal</h1>
    <p>Lorem ipsum dolor, sit amet consectetur adipisicing elit.</p>
    <img src="images/logo.jpg" alt="Logotipo de la empresa">
</body>
</html>
```

Etiquetas (*tags*) HTML

Existen muchos otros *tags*. Aquí, solo dejamos un resumen de lo mínimo indispensable para usar en un documento HTML:

Etiqueta	Descripción
`<h1> hasta <h6>`	Permite generar párrafos del tipo título, con diferentes niveles de importancia; similar a los títulos aplicables en procesadores de texto.
`<p> <span> <strong> <em> <code>`	Permiten generar párrafos del tipo texto, con formato enriquecido, y representaciones de texto con fuente monoespaciada, entre otros.
`<img> <audio> <video>`	Permiten agregar elementos multimedia (imágenes, archivos de audio, archivos de video, respectivamente.

(Continúa en el siguiente slide).

Continuación:

Etiqueta	Descripción
`<div>`	Es un *tag* contenedor, que oficia de separador de contenido. Con éste se pueden generar cards HTML, encabezados y pié de páginas, menús laterales de navegación, etcétera. No aporta estructura visual alguna; se lo debe estilizar gráficamente con CSS.
`<input type="algo">`	Permiten crear cajas de texto para el ingreso de información. Por ejemplo; los formularios web están compuestos de varios *input types*. Su atributo `type` referencia el tipo de dato que se debe cargar.
`<button>`	Etiqueta que genera un botón en pantalla. Tiene una configuración básica que puede personalizarse en detalle aplicando estilos gráficos con el lenguaje CSS.

¿Qué es CSS?

CSS es el lenguaje de estilos gráficos que permite personalizar HTML en detalle, y generar todos los estilos gráficos que hacen lucir a las webs en la actualidad.

Es un lenguaje muy profundo y maduro, que aporta no solamente estética y color, sino también transiciones, animaciones, control de la estructura HTML según el dispositivo donde se cargue la web (*desktop, tablet, mobile*), entre otras tantas maravillas propias del diseño.

Entornos de trabajo

También **existen *frameworks* y *microframeworks* CSS que permiten incorporar un archivo remoto, en el HTML, para cambiar por completo su estética**, sin que debamos conocer el lenguaje CSS.

El *framework* más popular es ***Bootstrap*** y sus alternativas más modernas: ***Bulma*** y ***Tailwind***.

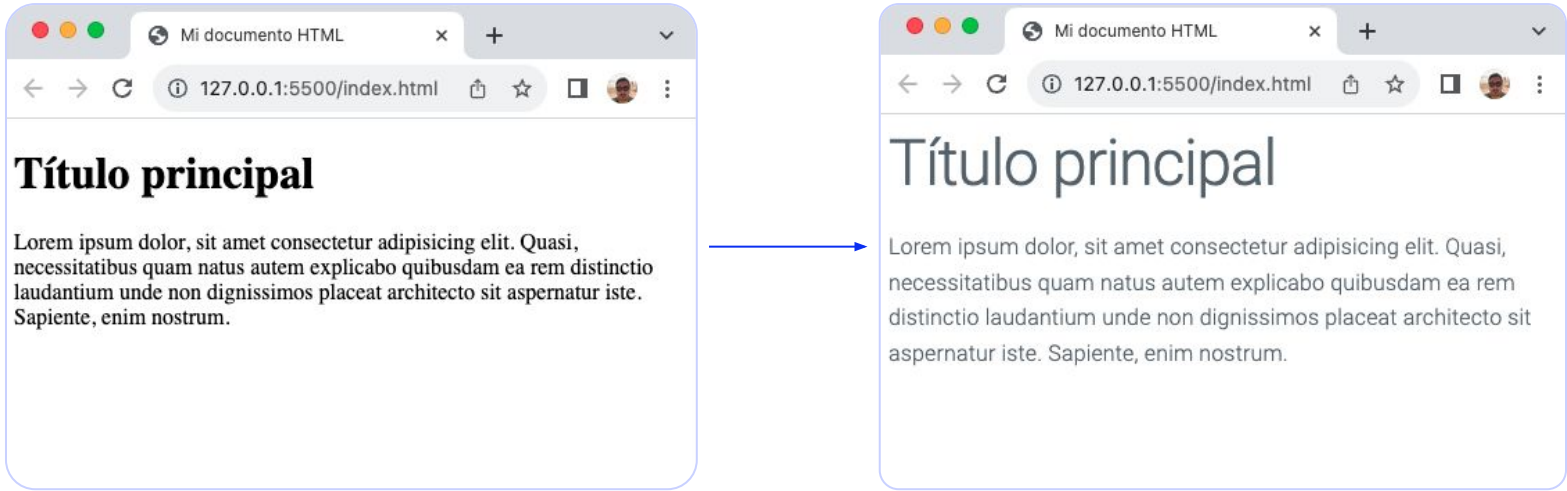
También hay muchas opciones de *microframeworks* CSS, entre ellas: ***Milligram***. Propone una estructura muy simple; con solo aplicarlo a nuestro documento HTML, la estética cambia sin esfuerzo alguno.

Ejemplo de uso de Milligram

En el documento HTML básico, se **agregan los links a la tipografía web y *microframework* Milligram**, que este comparte en su página principal, y se observará un cambio notable en el documento HTML sin esfuerzo adicional:

```
<!DOCTYPE html>
<html lang="es">
<head>
    <meta charset="UTF-8">
    <title>Mi documento HTML</title>
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Roboto:300,300italic,700,700italic">
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/milligram/1.4.1/milligram.css">
</head>
```

Veamos el antes y el después:



JavaScript y el DOM HTML

educación IT
CODE YOUR FUTURE

JavaScript y el DOM HTML

JavaScript cuenta con un **objeto propio**, denominado **`document`**, que brinda una **batería importante de herramientas (*métodos*)**, para poder acceder a elementos HTML, leer y/o modificar su contenido, generar nuevos elementos HTML, eliminar elementos HTML, ocultar y mostrar, entre otras tantas funcionalidades.

Para lograr esto de forma efectiva, se debe configurar el archivo JS dentro del documento HTML, con un **atributo** denominado **`defer`**.

```html
<!DOCTYPE html>
<html lang="es">
<head>
    <meta charset="UTF-8">
    <title>Document</title>
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <script defer src="js/main.js"></script>
</head>
<body>
    ...
</body>
</html>
```

defer

Cuando se referencia el archivo JS dentro de un documento HTML para que éste interactúe con el DOM, se debe declarar al archivo JS dentro del apartado **<head>** del documento HTML.

Además, se agrega un atributo en el *tag* **<script>**, denominado **defer**.

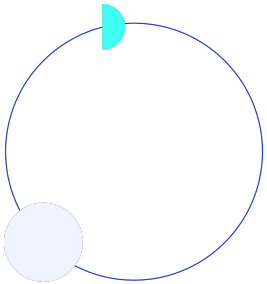
`defer` tiene el rol de **"esperar" que todo el documento HTML se cargue en el navegador web,** para luego recién permitir que el navegador web lea e interprete nuestro código JavaScript.

De esta forma, nos aseguramos de que todos los *tags* HTML con los que queremos interactuar, estén previamente representados en pantalla.

defer nació como propuesta en 2010, y terminó de integrarse en todos los navegadores web, en 2014.

Si bien se puede referenciar el *tag* `<script>` al final del documento HTML para conseguir el mismo objetivo, es mejor aplicar el atributo **defer**. Este atributo **aporta mejoras relacionadas al SEO para el posicionamiento de una web en buscadores** y, además, para estructurar todo archivo que complementa al documento HTML donde realmente deben ir: dentro del apartado **<head>**.

```
<head>
    ...
    <script defer src="js/main.js"></script>
</head>
```



¡Sigamos trabajando!