

## Explicado con mis propias palabras:

- ¿Qué es el DOM?

El Dom es como la versión viva del código o estructura que se haya hecho del HTML que después el navegador crea para que pueda entender y pueda mostrar la página a y también para que lenguajes como Javascript o Css pueda cambiarla o interactuar en ella.

En palabras sencillas el dom es como un mapa que muestra todos los elementos de la página y el cómo están conectados.

- ¿En qué se diferencia del archivo HTML que escribimos?

Que el HTML es un archivo que es solo texto plano, un archivo con etiquetas que uno escribe.

Mientras que el dom es un archivo convertido a un modelo que el navegador entiende y que puede modificar en tiempo real.

Se podría decir que se puede ver de la siguiente manera, que el html es el plan original mientras que el dom es el modelo 3d que el navegador crea a partir de ese plan para mostrarlo y manipularlo.

- ¿Por qué decimos que el DOM es un “árbol de nodos”?

Primero se tiene que tener en cuenta que cuando se refiere a árbol en programación se refiere a una estructura que conecta cosas en niveles con ramas y hojas.

El DOM es un árbol porque cada etiqueta HTML es un nodo (una pieza o un punto) y estos nodos están conectados unos con otros en forma de árbol.

El nodo raíz es el `<html>`, que tiene ramas como el `<head>` y `<body>`, y estas tienen más nodos hijos, como `<div>`, `<p>`, y así se va.

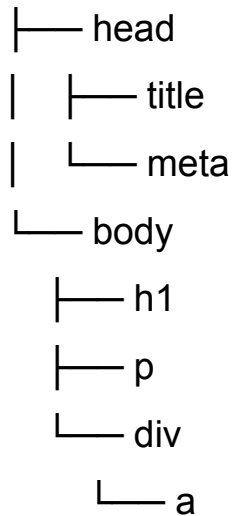
- ¿Cómo se relaciona el DOM con CSS y JavaScript?

**CSS** usa el DOM para cambiar cómo se ve la página (colores, tamaños, posiciones) apuntando a esos nodos y modificando su estilo.

Mientras **JavaScript** usa el DOM para cambiar o interactuar con la página (añadir texto, quitar elementos, responder a clics) modificando directamente esos nodos.

- Incluye un diagrama sencillo del árbol DOM de una página básica.

html



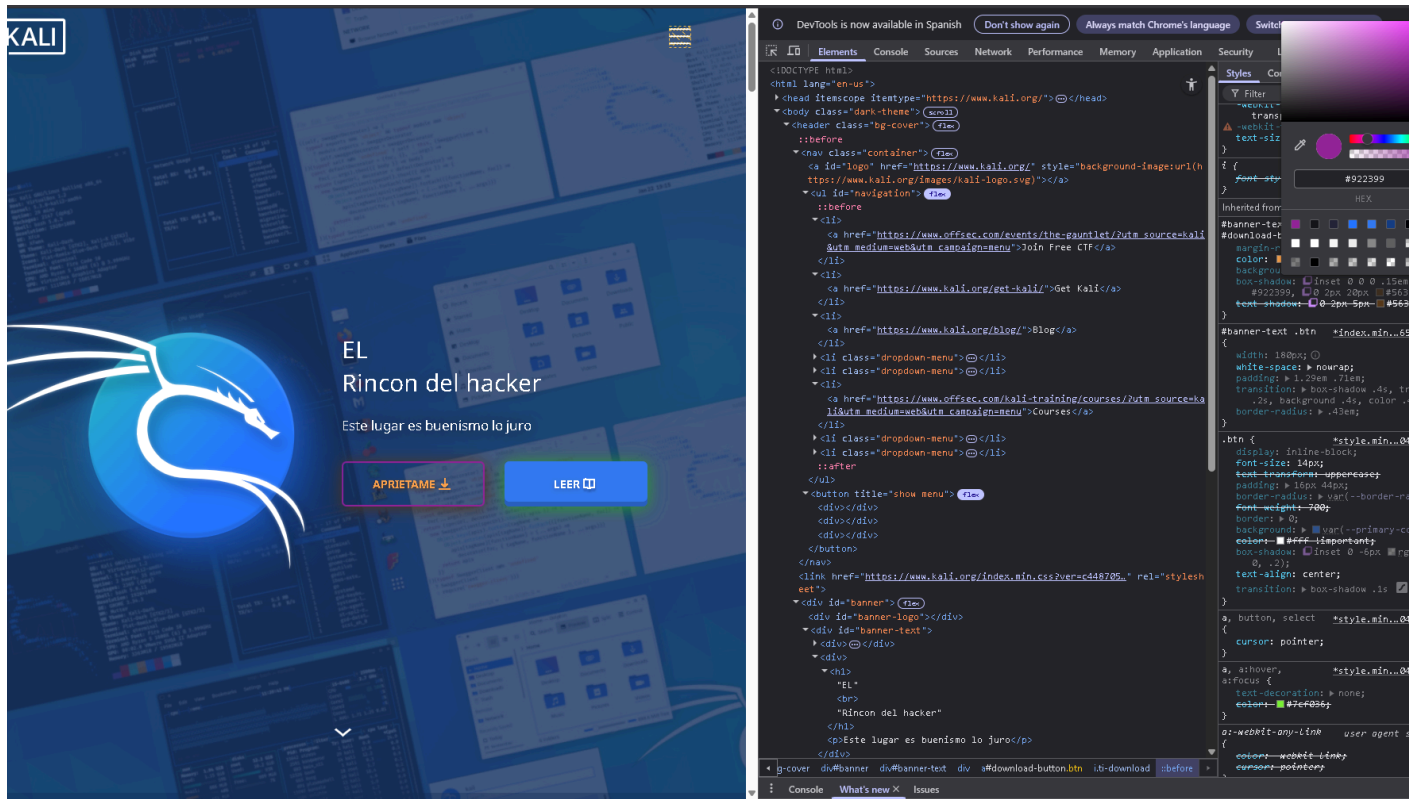
**Localiza:**

- La etiqueta <html>
- La etiqueta <head>
- La etiqueta <body>

**Busca un elemento sencillo, como un <h1> o un <p>.**

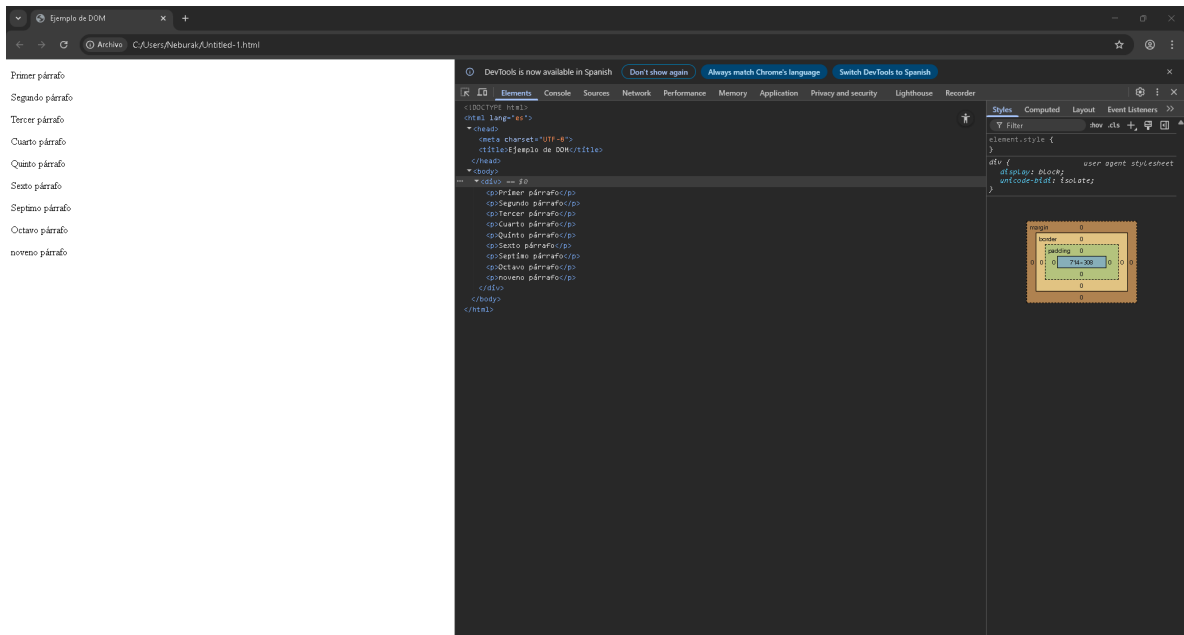
**Modifica el estilo desde el inspector:**

**Ejemplo: cambia su color (color: red;) o su tamaño (font-size: 30px;).**



(Cambie párrafos , los colores que los botones y el texto tenían )  
Las etiquetas típicas <html>,<head>, <body> **están en la parte de esta arriba.**

**Crea un archivo HTML sencillo con un <div> y varios <p>.**



Cuando abro el archivo HTML en el navegador y lo inspecciono, el DOM aparece igual al código que escribí porque el documento está bien estructurado, con todas las etiquetas correctamente cerradas. Esto se debe a que el `<html>`, `<head>`, `<body>`, el `<div>` y todos los párrafos tal como están escritos. Sin embargo, aunque parece igual, el navegador puede hacer pequeñas modificaciones automáticas que no siempre se notan a simple vista. Por ejemplo, si no hubiera incluido etiquetas como `<html>`, `<head>` o `<body>`, el navegador las habría agregado por mí. También corrige errores si olvidara cerrar una etiqueta y puede crear nodos de texto invisibles llamados `#text`, que aparecen por los espacios o saltos de línea entre elementos. En este caso no hubo diferencias porque el HTML estaba bien hecho, pero es importante destacar que el navegador siempre interpreta el código para asegurarse de que funcione correctamente, incluso si tiene errores.

# Reporte

## Resumen de la investigación.

El DOM (Document Object Model) es como la versión viva del código HTML que el navegador crea cuando abre una página web. Es una representación estructurada que permite que lenguajes como CSS y JavaScript interactúen con los elementos de la página.

Podría decirse que el HTML es el plano original (solo texto con etiquetas), mientras que el DOM es el modelo 3D interactivo que el navegador construye a partir de ese plano.

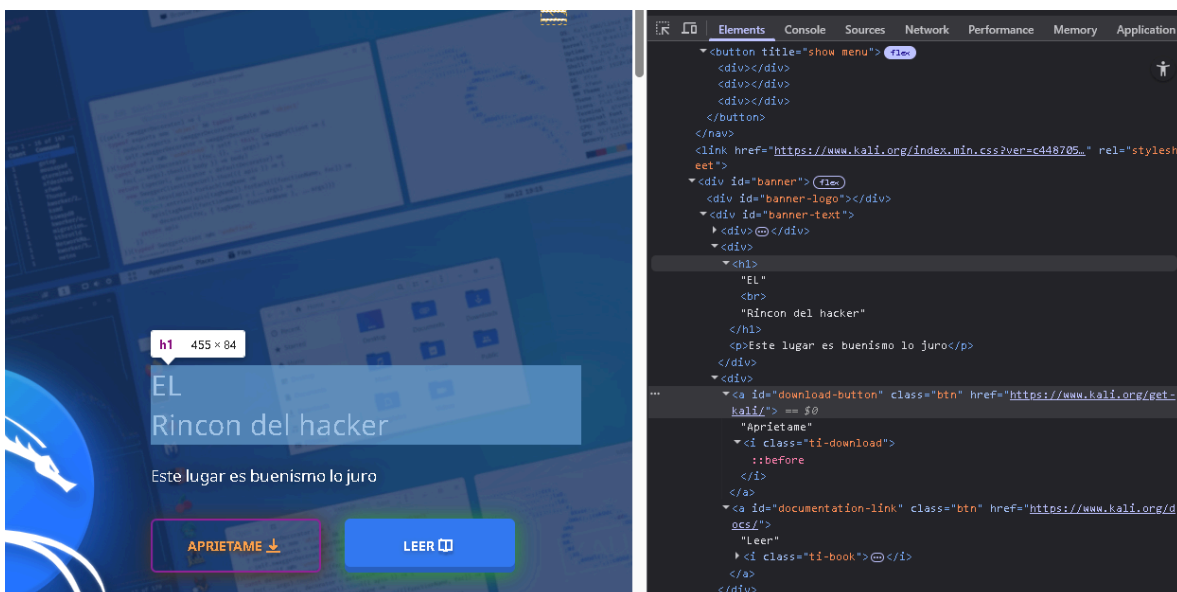
El DOM se organiza como un árbol de nodos, donde cada etiqueta HTML es un nodo conectado a otros.

CSS usa el DOM para cambiar la apariencia de los elementos, aplicando estilos como colores, tamaños o márgenes.

Ejemplo: cambiar el color de un texto con `color: red;` o su tamaño con `font-size: 30px;`.

JavaScript usa el DOM para cambiar el contenido o comportamiento de la página: puede agregar o eliminar elementos, cambiar textos, responder a clics o mover objetos.

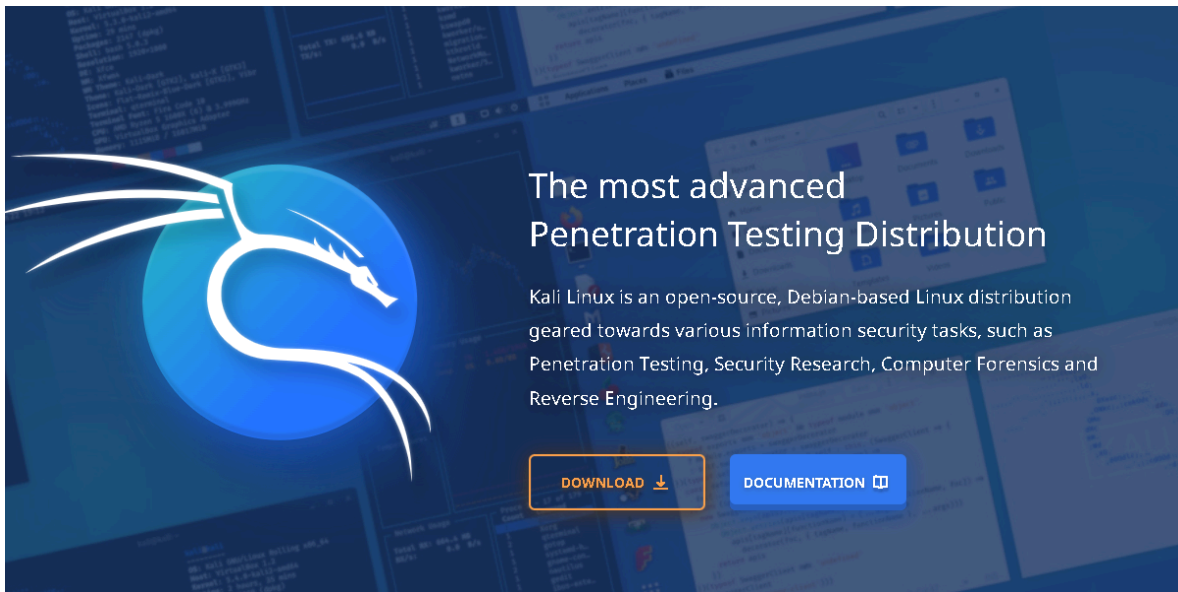
## Capturas de pantalla del DOM en el inspector.



Primer párrafo  
Segundo párrafo  
Tercer párrafo  
Cuarto párrafo  
Quinto párrafo  
Sexto párrafo  
Septimo párrafo  
Octavo párrafo  
novenio párrafo

```
DevTools is now available in Spanish
Elements Console Sources Net
<!DOCTYPE html> == $0
<html lang="es">
  <head>
    <meta charset="UTF-8">
    <title>Ejemplo de DOM</title>
  </head>
  <body>
    <div>
      <p>Primer párrafo</p>
      <p>Segundo párrafo</p>
      <p>Tercer párrafo</p>
      <p>Cuarto párrafo</p>
      <p>Quinto párrafo</p>
      <p>Sexto párrafo</p>
      <p>Septimo párrafo</p>
      <p>Octavo párrafo</p>
      <p>novenio párrafo</p>
    </div>
  </body>
</html>
```

## Ejemplo de un cambio de estilo antes y después.





## Conclusión:

Es importante entender el DOM porque CSS aplica sus estilos sobre los elementos del DOM, no directamente sobre el archivo HTML.

Conocer su estructura ayuda a identificar correctamente qué parte de la página se quiere modificar y cómo hacerlo.

Además, permite trabajar mejor con herramientas de desarrollo, inspeccionar errores visuales y entender cómo interactúan HTML, CSS y JavaScript dentro de una página web.(Fin)