**getElementsByTagName:**

Este método se utiliza para obtener todos los elementos de un documento con un nombre de etiqueta específico. Devuelve una colección de nodos que pueden ser accedidos como un array.

**Ejemplo:** Supongamos que tienes un documento HTML con varias etiquetas **<p>** y quieres acceder a ellas:

Html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>getElementsByTagName Example</title>

</head>

<body>

<p>Paragraph 1</p>

<p>Paragraph 2</p>

<p>Paragraph 3</p>

<script src="script.js"></script>

</body>

</html>

En tu archivo JavaScript (**script.js**), puedes usar **getElementsByTagName** de la siguiente manera:

// Obtener todas las etiquetas <p>

var paragraphs = document.getElementsByTagName("p");

// Iterar sobre la colección de elementos y mostrar el contenido de cada párrafo

for (var i = 0; i < paragraphs.length; i++) {

console.log(paragraphs[i].textContent);

}

**childNodes:**

El método **childNodes** devuelve una colección de los nodos secundarios de un nodo dado, como un NodeList.

**Ejemplo:** Supongamos que tienes un elemento div con varios nodos hijos y quieres acceder a ellos:

Html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>childNodes Example</title>

</head>

<body>

<div id="parent">

<p>Child 1</p>

<span>Child 2</span>

<div>Child 3</div>

</div>

<script src="script.js"></script>

</body>

</html>

En tu archivo JavaScript (**script.js**), puedes utilizar **childNodes** para acceder a los nodos hijos:

Obtener el elemento padre

var parent = document.getElementById("parent");

// Obtener todos los nodos hijos del elemento padre

var children = parent.childNodes;

// Iterar sobre la colección de nodos hijos y mostrar el tipo y contenido de cada nodo

for (var i = 0; i < children.length; i++) {

console.log("Tipo:", children[i].nodeType, "Contenido:", children[i].textContent);

**querySelector:**

Este método permite seleccionar el primer elemento que coincida con un selector CSS especificado. Devuelve solo el primer elemento que cumple con el selector.

**Ejemplo:** Supongamos que tienes un documento con varios elementos de clase "example" y quieres acceder al primero:

Html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>querySelector Example</title>

</head>

<body>

<div class="example">Element 1</div>

<p class="example">Element 2</p>

<span class="example">Element 3</span>

<script src="script.js"></script>

</body>

</html>

En tu archivo JavaScript (**script.js**), puedes utilizar **querySelector** de la siguiente manera:

// Obtener el primer elemento con la clase "example"

var firstElement = document.querySelector(".example");

// Mostrar el contenido del primer elemento

console.log(firstElement.textContent);