



# Trabajo Práctico N°4 Prác. Profesionalizante

Estudiante: Puebla, Leonardo Gabriel

## 2.5. Averigüe y explique da manera concisa los siguientes tópicos de JS

<b>getElementById</b>	Este es un método que se utiliza para seleccionar un elemento del DOM (Documento de Objeto del Modelo) utilizando su ID único. Por ejemplo, <code>document.getElementById('miElemento')</code> seleccionaría un elemento con el ID "miElemento".
<b>getElementsByName</b>	Este método se utiliza para seleccionar elementos del DOM por su atributo "name". Devuelve una colección de elementos que tienen el mismo nombre. Por ejemplo, <code>document.getElementsByName('miNombre')</code> seleccionaría todos los elementos con el atributo "name" igual a "miNombre".
<b>getElementsByTagName</b>	Este método se utiliza para seleccionar elementos del DOM por su nombre de etiqueta (por ejemplo, "div", "p", "a"). Devuelve una colección de todos los elementos con la etiqueta especificada. Por ejemplo, <code>document.getElementsByTagName('div')</code>

	seleccionaría todos los elementos "div" en el documento.
<b>querySelector[All]</b>	Estos métodos permiten seleccionar elementos del DOM utilizando selectores CSS. <b>querySelector</b> devuelve el primer elemento que coincide con el selector, mientras que <b>querySelectorAll</b> devuelve una lista de todos los elementos que coinciden con el selector.
<b>for-in versus for-of</b>	Se utiliza para iterar sobre las propiedades enumerables de un objeto, mientras que <b>for-of</b> se utiliza para iterar sobre elementos de estructuras de datos iterables como arrays o cadenas.
<b>Object.freeze</b>	Este método se utiliza para hacer que un objeto en JavaScript sea inmutable, lo que significa que sus propiedades no se pueden modificar, agregar o eliminar después de congelar el objeto.
<b>let versus var</b>	Son palabras clave utilizadas para declarar variables en JavaScript. La diferencia clave es que <b>let</b> tiene un ámbito de bloque, mientras que <b>var</b> tiene un ámbito de función. En la mayoría de los casos, se prefiere <b>let</b> debido a su ámbito de bloque más predecible.
<b>Array.every, some</b>	Estos métodos se utilizan para realizar pruebas en los elementos de un array. <b>every</b> verifica si todos los elementos cumplen una condición, mientras que <b>some</b> verifica si al menos un elemento cumple la condición.
<b>DOMContentLoaded</b>	Es un evento que se dispara cuando se ha cargado completamente el contenido HTML de una página, antes de que se hayan cargado recursos externos como imágenes.
<b>innerHTML, innerText</b>	se utiliza para obtener o establecer el contenido HTML de un elemento, incluyendo etiquetas HTML, mientras que <b>innerText</b> se utiliza para obtener o establecer el contenido de texto de un elemento, sin incluir etiquetas HTML.
<b>Atributos con prefijos # y _</b>	Atributos con prefijos <b>#</b> y <b>_</b> : En JavaScript, los atributos con un prefijo <b>#</b> y <b>_</b> no tienen un significado especial. Son simplemente nombres de atributos válidos que pueden utilizarse en objetos y

	elementos del DOM. Sin embargo, suelen usarse en variables 'privadas'.
<b>Accessors</b>	En JavaScript, los accessors son métodos especiales <code>get</code> y <code>set</code> que se utilizan para definir propiedades de un objeto y controlar su comportamiento de lectura y escritura.
<b>NaN, null, undefined</b>	representa "Not-a-Number" y se utiliza para representar resultados matemáticos inválidos. <code>null</code> se utiliza para representar la ausencia intencional de valor, mientras que <code>undefined</code> se utiliza para representar la ausencia no intencional de valor o la falta de inicialización de una variable.