Laboratorio 3B – Henry Campos C21636

1. ¿Cuáles son los data types que soporta javascript?

Tiene tipos primitivos: string, numbers, booleans, undefined, null, symbol, bigInt. Y no primitivos, que serían los object.

2. ¿Cómo se puede crear un objeto en javascript? De un ejemplo

```
var carro = {
        color: “negro”,
        placa: 123456,
        parquear: function(){ return “parqueando”;}
};
```

3. ¿Cuáles son los alcances (scope) de las variables en javascript?

- Scope global: se declaran fuera de un bloque y son accesibles desde cualquier parte del código.
- Scope local: se declaran dentro de un bloque (una función), y solo se pueden utilizar dentro del bloque en que se definieron.

4. ¿Cuál es la diferencia entre undefined y null?

- Undefined: es el valor por defecto de cualquier variable declarada pero que no tiene nada asignado.
- Null: este se asigna intencionadamente y que indica que cierta variable no tiene valor o que este todavía no existe.

5. ¿Qué es el DOM?

Document Object Model (DOM) representa la estructura de un documento HTML, el cual permite que lenguajes como JavaScript puedan acceder a los elementos de cierta página web.

6. Usando Javascript se puede acceder a diferentes elementos del DOM. ¿Qué

hacen, que retorna y para qué funcionan las funciones getElement y

querySelector? Cree un ejemplo

- getElement: sirve para acceder a elementos del DOM por medio de un ID, su clase o su nombre de etiqueta, este devuelve el objeto que coincida con la solicitud. Ej: const elemento = document.getElementById("ID");

- querySelector: selecciona el primer elemento del DOM que coincide con un selector CSS. Ej: const elemento = document.querySelector(".elemento");

7. Investigue cómo se pueden crear nuevos elementos en el DOM usando

Javascript. De un ejemplo

Se puede utilizar la función del document que se llama “createElement(“X”)”, la cual sirve para crear un elemento HTML a partir de cierta etiqueta. Ej:

```
const elementoCreado = document.createElement("p");
nuevoElemento.textContent = "Párrafo";
document.body.appendChild(elementoCreado);
```

8. ¿Cuál es el propósito del operador this?

El operador this sirve para referir al objeto actual desde el que se está invocando, su valor depende únicamente del contexto de ejecución. Básicamente, es utilizado para referenciar los elementos pertinentes al objeto de la función con que se está trabajando.

9. ¿Qué es un promise en Javascript?De un ejemplo

Un objeto que representa la finalización o fallo de una operación y su resultado. Ej:

```
const promesa = new Promise((resolve, reject) => {
  let exito = true;
  if (exito) {
    resolve("Se realizó con éxito");
  } else {
    reject("Hubo un fallo");
  }
});
promesa.then(resultado => console.log(resultado)).catch(error =>
console.error(error));
```

10.¿Qué es Fetch en Javascript? De un ejemplo

Se utiliza para realizar solicitudes HTTP y manejar respuestas. Ej:

```
fetch(‘yo.com/index.html’)
  .then(response => response.json())
  .then(data => console.log(data))
  .catch(error => console.error('Error:', error));
```

11. ¿Qué es Asynch/Await en Javascript ? De un ejemplo

Async sirve para que una función retorne una promise, await espera la resolución de una promise como la función “fetch()”. Ej:

```
async function extraerInformacion() {
  try {
    const response = await fetch(‘yo.com/index.html’);
    const data = await response.json();
    console.log(data);
  } catch (error) {
    console.error('Error:', error);
  }
}
extraerInformacion ();
```

12.¿Qué es un Callback? De un ejemplo

Es una función que se pasa como argumento a otra función, lo cual permite que sea llamada dentro de la otra función. Ej:

```
function saludar(nombre, callback) {
  console.log('Hola ' + nombre);
  callback();
}
function despedir() {
  console.log('Adiós');
}
```

```
saludar('Juan', despedir);
```

13. ¿Qué es Clousure?

Es una función que recuerda el estado de las variables en su entorno léxico, incluso cuando la función se ejecuta fuera de ese entorno. Ej:

```
function crearContador() {
  let contador = 0;
  return function() {
    contador++;
    console.log(contador);
  };
}
const contar = crearContador();
contar();
contar();
```

14.¿Cómo se puede crear un cookie usando Javascript?

A través de JavaScript se puede hacer de la siguiente manera:

```
document.cookie = "usuario=user; expires=Fri, 31 Dec 2024 12:00:00 UTC; path=/";
```

El cual agrega la cookie al HTML.

15.¿Cuál es la diferencia entre var, let y const?

- var: tiene un alcance de función, es una variable a la que se le puede cambiar su valor.
- let: tiene un alcance de bloque, es una variable a la que se le puede cambiar su valor.
- const: tiene un alcance de bloque, es una constante que se almacena y no se puede cambiar su valor.