

UNIVERSIDAD DE COSTA RICA

ESCUELA DE COMPUTACIÓN

Ingeniería de Software

Laboratorio 2.B: Introducción a Javascript

Esteban Chaves Obando (C12149)

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

Los data types de JS son:

- Number (Para números enteros y decimales)
- String (Para texto)
- Big int (Para números grandes)
- Boolean (True o False)
- Undefined (Se usa cuando se quiere declarar una variable pero sin valor)
- Symbol (Identificador único)
- Null (Para valores nulos)

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

En JS un objeto se crea como en java normal

```
const carro = {  
  Marca = 'TOYOTA'  
  modelo = 'prado'  
  placa = xxxxx  
  modelo: function () {  
    console.log ('El modelo seleccionado es: ${this.modelo}');  
  }  
};  
  
carro.modelo();
```

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

el scope de las variables en JS son:

- Global, el cual define que está disponible en todo el código.
- local, define que está dentro de una función.
- bloque, la cual define que es solo accesible dentro de un bloque.

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

Undefined: Es el valor por defecto de las variables que nunca se inicializan.

null: Es algo que yo asigno de manera intencional para indicar que algo no tiene valor.

5. ¿Qué es el DOM?

Document object model es la estructura donde existe la jerarquía de los objetos de una 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: usualmente obtiene un elemento por su ID
getElementById(id).

querySelector: devuelve el primer elemento que haga match con el selector css
querySelector(selector)

7. Investigue cómo se pueden crear nuevos elementos en el DOM usando Javascript. De un ejemplo

Se puede usar el document.createElement() y tambien el appendChild()

```
nuevo_elemento = document.createElement("P");
```

8. ¿Cuál es el propósito del operador this?

this hace referencia a los atributos de justamente el objeto con el que se está trabajando, un ejemplo es en el punto 2 para ver el modelo del carro, justo se usa this para señalar que queremos ver del objeto en el cual estamos.

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

Un promise es una operación la cual puede estar en 3 estados:

- Pending, está esperando
- fulfilled, se completó de manera exitosa
- rejected, fallo

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

Fetch es una API para hacer peticiones HTTP

```
fetch('http://example.com/movies.json')  
  .then(response => response.json())  
  .then(data => console.log(data));
```

ejemplo sacado de: https://developer.mozilla.org/es/docs/Web/API/Fetch_API/Using_Fetch

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

Usualmente se utiliza para manejar de una buena manera las promises

```
async function obtenerDatos() {  
  let response = await fetch("www.paginaxd.com");  
  let data = await response.json();  
  console.log(data);  
}  
  
obtenerDatos();
```

12. ¿Qué es un Callback? De un ejemplo

Un callback es una función que se pasa como argumento a otra función y se ejecuta después de que dicha función haya terminado su proceso.

```
function sum(a, b, c) {  
  let resultado = a + b;  
  c(resultado);  
}  
  
function mostrarResultado(resultado) {  
  console.log("El resultado es:", resultado);  
}  
  
op(5, 3, mostrarResultado);
```

13. ¿Qué es Clousure?

Clousure es una función que sabe dónde fue creada, incluso después de que el ámbito termine

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

En JS las cookies se crean mediante document.cookie se crea con el formato:

nombre = Esteban; expires = Thu, 10 Apr 2025 ; path = /home

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

En JavaScript, var, let y const se utilizan para declarar variables. La principal diferencia entre ellas es que var permite reasignar valores, mientras que let y const no.

Nombre	Ámbito	Reassignable	Redefinible
var	Funciones	Si	Si
let	Bloques	Si	No
const	Bloques	No	No