

## Gabriel Zúñiga Vega C28741

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

string, number, bigint, boolean, undefined, null, symbol, object

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

```
let perrito = {  
  nombre: "Juanito",  
  edad: 3,  
  comer: function() {  
    console.log (this.nombre + " está comiendo");  
  }  
};  
perrito.comer();
```

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

Global: Variable que se puede acceder desde cualquier parte del código.

Function: Variable declarada dentro de una función y que sólo es accesible dentro de esta.

Block: Variable declarada dentro de un bloque y que sólo es accesible dentro de este.

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

undefined: Una variable que ya fue declarada pero que todavía no tiene ningún valor

null: Representa una ausencia de valor intencional (la variable se declaró como null)

5. ¿Qué es el DOM?

El DOM es el Document Object Model y es la representación ya estructurada de un documento HTML con forma de árbol para poder manipular la página más dinámicamente mediante JS.

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: Obtiene un elemento por su id.

querySelector: Obtiene el primer elemento que coincida con el selector de CSS.

**Ejemplo:**

```
let elemento = document.getElementById("miElemento");
```

```
let boton = document.querySelector(".btn");
```

7. Investigue cómo se pueden crear nuevos elementos en el DOM usando Javascript. De un ejemplo  
Se puede usar `.createElement()`, `.appendChild()` o `.insertAdjacentHTML()`

**Ejemplo:**

```
let oracion = document.createElement("p");
oracion.textContent = "Esta es una oracion";
document.body.appendChild(oracion);
```

8. ¿Cuál es el propósito del operador `this`?  
El operador `this` hace referencia a la variable que pertenece al contexto actual.

9. ¿Qué es un promise en Javascript? De un ejemplo  
Es un objeto que sirve como operación asincrónica que puede tener éxito o fallar.

**Ejemplo:**

```
let promesa = new Promise((resolve, reject) => {
    setTimeout(() => resolve( "wujú!" ), 2000);
});
```

```
promesa.then(res=> console.log(res));
```

10. ¿Qué es Fetch en Javascript? De un ejemplo  
Es una API que sirve para hacer peticiones de tipo HTTP.

**Ejemplo:**

```
fetch("https://fotodepatitos.com");
```

11. ¿Qué es Async/Await en Javascript ? De un ejemplo  
Función que permite manejar código asíncrono de forma más fácil y clara.

**Ejemplo:**

```
async function obtenerFoto() {
    let respuesta = await fetch("https://fotodepatitos.com");
    let datos = await respuesta();
    console.log(datos);
}
obtenerFoto();
```

12. ¿Qué es un Callback? De un ejemplo  
Una función que se pasa como argumento a otra función para que se ejecute después de que la otra haya terminado.

**Ejemplo:**

```
function procesar(callback) {  
    console.log( "Estoy procesando..." )  
    callback();  
}
```

```
procesar(() => console.log("Listo"));
```

**13. ¿Qué es Closure?**

Los closures son los caracteres { o } que ayudan a mantener alcances globales o privados en funciones u objetos.

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

```
document.cookie = "username= Gabriel Zuniga";
```

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

Característica	var	let	const
Ámbito	global o local	bloque	bloque
Reasignación	Sí	Sí	No
Redeclaración	Sí	No	No
Hoisting	Se eleva con undefined	Se eleva pero no se inicializa	Se eleva pero no se inicializa
Cuándo usar	No recomendado por errores de hoisting o ámbito global	Cuando el valor va a cambiar	Cuando el valor no va a cambiar (constantes, objetos, arrays, etc.)