

## PRUEBA DE NIVEL

### 1. Demuestra que la siguiente cadena "geeksforgeeks" no es un palíndromo.

Podríamos definir el concepto de palíndromo como una cadena de caracteres que se recorren igual en el sentido normal de lectura como en el sentido contrario.

Para comprobar si una palabra o frase es o no es un palíndromo, podríamos utilizar la siguiente función:

```
//Cris Mironescu

function palindromo()
{
    var palabra=prompt("Añade tu palabra").toLowerCase();

    palabra=palabra.replace(/ /g, "");

    for (var i=0;i<palabra.length;i++){
        if(palabra[i]!=palabra[palabra.length-i-1]){
            return false;
        }
    }
    return true;
}

if(palindromo())
{
    alert("Esto es palíndromo");
}else{
    alert("Esto no es palíndromo")
}
```

## **2. Diferencias entre flexbox y grid.**

CSS Flexbox es un módulo de CSS que nos da una nueva forma de maquetación Web con un modelo de cajas flexible. Su contenido se va adaptando al modelo de caja que lo contiene y nos permite organizar elementos de forma vertical o horizontal. Esto quiere decir que flexbox es “unidimensional”.

CSS Grid es una nueva tecnología que incluye CSS que nos va a permitir maquetar sitios Web de manera sencilla y optimizada. Nos va a permitir alinear elementos en columnas y filas. A diferencia de Flexbox, Grid tiene dos dimensiones. Cuando definimos un layout con Grid lo que hace es crearnos una nueva cuadrícula (en función de las columnas y filas que hayamos definido).

Otra diferencia que podemos destacar es que la cuadrícula se define principalmente en el elemento padre en el caso de Grid y en Flexbox la mayor parte del diseño se define en el elemento child.

Además, Grid es más resistente. Aunque la flexibilidad de Flexbox podría ser también una fortaleza, en ocasiones, la forma en que se dimensiona un elemento flexible se vuelve bastante complicada.

### **3. Html**

#### **3.1. Crea un div**

```
document.createElement('div')  
let div = document.createElement ('div')
```

#### **3.2. Crea un svg**

```
document.createElementNS(http://www.w3.org/2000/svg)
```

#### **3.3. Agrega una clase al div creado en el punto 1**

```
div.classList.add('red')
```

#### **3.4. Agrega un estilo que cambie la propiedad color al punto 1**

```
div.style.setProperty('color', 'blue')
```

#### **3.5. Agrega un atributo id al punto 1**

```
div.setAttribute('id', 'cris')
```

#### **3.6. Crea un página en blanco y agrega al body el div con "hello world"**

```
text = document.createTextNode (Hello World)  
div.appendChild(text)  
document.body.appendChild(div)
```

#### **3.7. Crea un nuevo div y agregalo como hijo al punto 1 antes de "hello world"**

```
child = document.createElement('div')  
div.insertBefore(child,text)
```

#### **3.8. Dime los nodos que tiene el primer div**

```
div.childNodes
```

#### **3.9. Dime los elementos que tiene el primer div**

```
div.children
```

#### **3.10. Busca todos los div de la página**

```
document.querySelectorAll('div')
```

### **3.11. Busca los div que contienen la clase agregada al punto 3**

```
document.querySelectorAll('.red')  
document.getElementsByClassName('red')
```

### **3.12. Busca los div que tienen el id agregado en punto 5**

```
document.querySelectorAll('#cris')  
document.getElementById('cris')
```