

**Desarrollo 1:** Partiendo de estas líneas de código HTML:

```
<html>
<body>
  <div>Usuarios:</div>
  <ul>
    <li>Juan</li>
    <li>Pedro</li>
  </ul>
</body>
</html>
```

Escriba el código javascript necesario para acceder y realizar las siguientes operaciones:

- Partiendo del nodo **"document"**, acceda y almacene el nodo `<div>` en una variable.
- A continuación, modifique el texto del nodo `<div>` con el texto "Usuarios VIP:". **(Solo texto)**
- A continuación, modifique el contenido del nodo `<div>` con el siguiente contenido: `<strong>Super Usuarios: </strong>`
- ¿Partiendo del nodo `<div>` del apartado a) , ¿Cómo puedo acceder y almacenar en una variable el nodo `<ul>` ?
- ¿Partiendo del nodo del apartado d) escriba el nodo necesario para recorrer sus hijos (nodos `<li>`) y modificarles el contenido de los mismos por otros nombres diferentes (Pepa y Pepe)?

### // Ejercicio 1

```
let div = document.getElementsByTagName("div")[0];
div.textContent = "Usuarios VIP"; //Solo texto
div.innerHTML = "<strong> Super Usuarios: </strong>"; // HTML
let ulNode = div.nextElementSibling; //previousElementSibling

let primerHijo = ulNode.firstElementChild ;
primerHijo.textContent = "Pepa";
primerHijo.nextElementSibling.textContent = "Pepe";
```

**Desarrollo 2:** Complemente el siguiente código en el cual se implementa una función capaz de crear un nodo imagen, con su respectiva ruta y con un atributo **"alt"** con el texto "Error: Imagen no cargada" y por último, se añade este nodo imagen como hijo al nodo que se pasa por parámetro. Para su implementación debe de usar los métodos y propiedades estudiados para tal cometido:

```
function crearImagen( nodo , rutaImagen) {
  //Crear elemento imagen:
  let imagen =

  //Añadir la ruta de la imagen:

  //Añadimos un atributo "alt" con el texto "Error: Imagen no cargada":

  //Añadimos como hijo el nodo imagen al nodo pasado por parámetro:
}
crearImagen (document.body , "money.png"); //Ejemplo de uso
```

**Desarrollo 2: 0.5 puntos**

## //// Ejercicio 2

```
function crearImagen ( nodoPapi , rutaImg)
{
    let nodo = document.createElement("img");
    nodo.src = rutaImg;
    //nodo.setAttribute("src", rutaImg);
    nodo.setAttribute("alt", "La imagen no se ha podido cargar.");
    nodoPapi.appendChild(nodo);
}
```

**Desarrollo 3:** Añada al nodo **h1** y al nodo **h2** el evento **"onclick"**. Aplique a ambos nodos **la misma función callback** con el nombre que quieras. En dicha función, debes de implementar la lógica para detectar cual de los dos nodos es el que ha disparado la función. Muestra por consola un mensaje con el nombre del nodo que haya "triggeado" la función.

```
<body>
  <h1 id="eventos">Obtener información de un evento</h1>
  <h2 id="parrafo1">Párrafo 1</h2>
  <script>
    // Código aquí:
  </script>
</body>
```

## //// Ejercicio 3

```
//<h1 id="eventos">Obtener información de un evento</h1>
//<h2 id="parrafo1">Párrafo 1</h2>
```

```
document.getElementById("eventos").addEventListener("click", metodo1);
document.getElementById("parrafo1").addEventListener("click", metodo1);
```

```
function metodo1( event ) {
    let nodoClickeado = event.target;
}
```

**Desarrollo 4:** Dado el siguiente código, ¿Cómo podemos **"desacoplar"** (eliminar dependencias) el código para hacer genérica la función llamadaServidor y así poder usarla desde otro archivo, y tener de alguna forma, su resultado en el método desde el que se llama? Asegúrese de mostrar por consola también un posible error.

Código a desacoplar: *libreriaDatos.js*

```
function llamadaServidor(url)
{
    fetch(url)
    .then(response => {
        if (response.status == 200) {
            return response.json();
        }
        else {
            throw new Error(response.status);
        }
    })
    .then(json =>
        obtenerDatosAlumnos(json);
    )
}

function obtenerDatosAlumnos(json)
{
    console.log( json.nombre );
}
```

```
function llamadaServidor(url, callback) {
    fetch(url)
    .then(response => {
        if (response.status == 200) {
            return response.json();
        } else {
            throw new Error(response.status);
        }
    })
    .then(json => callback(json));
}
```

```
function obtenerDatosAlumnos(json) {
    console.log(json.nombre);
}
```