TEST DE REAPSO UT6 DWEC.

1. ¿Qué método selecciona un elemento por su ID en el DOM?

```
let elemento = document._____('miId');
```

a) getElementById
b) querySelector
c) getElementsByClassName
d) getElementsByTagName

---

2. ¿Qué propiedad devuelve el tipo de nodo de un objeto DOM?

```
let tipoNodo = nodo._____;
```

a) nodeType
b) nodeName
c) nodeValue
d) nodeContent

---

3. ¿Qué método selecciona todos los elementos de una clase específica en el DOM?

```
let elementos = document._____('nombreClase');
```

a) getElementsByClassName
b) querySelectorAll
c) getElementById
d) getElementsByTagName

---

4. ¿Cómo se selecciona el primer elemento que coincide con un selector CSS?

```
let elemento = document._____(selectorCSS);
```

a) querySelector
b) querySelectorAll
c) getElementsByClassName
d) getElementById

---

5. ¿Qué método elimina un atributo de un nodo del DOM?

```
nodo._____('atributo');
```

a) removeAttribute
b) setAttribute
c) getAttribute
d) toggleAttribute

---

6. ¿Cómo se añade una clase a un elemento utilizando classList?

```
elemento.classList._____('miClase');
```

a) add
b) remove
c) toggle
d) contains

---

7. ¿Qué método crea un nuevo nodo de tipo elemento en el DOM?

```
let nuevoNodo = document._____('div');
```

a) createElement
b) createTextNode
c) appendChild
d) replaceChild

---

8. ¿Qué método devuelve todos los nodos hijos de un elemento?

```
let hijos = elemento._____;
```

a) childNodes
b) children
c) childElementCount
d) childContent

---

9. ¿Cómo se obtiene el texto de un nodo ignorando etiquetas HTML?

```
let texto = elemento._____;
```

a) textContent
b) innerHTML
c) nodeValue
d) innerText

---

10. ¿Qué método permite insertar un nodo antes de otro existente?

```
padre._____(nuevoNodo, nodoReferencia);
```

a) removeChild
b) insertBefore
c) appendChild
d) replaceChild

---

11. ¿Qué método selecciona el primer nodo que coincide con un selector CSS y devuelve null si no existe?

```
let nodo = document._____(selector);
```

a) getElementsByClassName
b) querySelectorAll
c) querySelector
d) getElementById

---

12. ¿Qué propiedad devuelve el nodo padre de un nodo?

```
let padre = nodo._____;
```

a) parentNode
b) previousSibling
c) rootNode
d) nextSibling

---

13. ¿Qué método devuelve un array de los nodos hijos de un elemento con su respectivo texto o etiquetas?

```
let hijos = elemento._____;
```

a) children
b) childNodes
c) innerHTML
d) textContent

---

14. ¿Cómo eliminas un nodo hijo específico del DOM?

```
padre._____(nodoHijo);
```

a) deleteChild
b) removeChild
c) clearChild
d) replaceChild

---

15. ¿Qué método permite alternar una clase en un elemento?

```
elemento.classList._____('miClase');
```

a) contains
b) remove
c) toggle
d) add

---

16. ¿Qué propiedad se usa para acceder al primer nodo hijo de un elemento?

```
let primerHijo = elemento._____;
```

a) childNodes
b) parentNode
c) firstChild
d) firstElementChild

---

17. ¿Qué método devuelve todos los nodos que coinciden con un selector CSS específico?

```
let nodos = document._____(selector);
```

a) querySelectorAll
b) querySelector
c) getElementsByTagName
d) getElementsByClassName

---

18. ¿Cómo puedes contar el número de hijos de tipo elemento dentro de un nodo?

```
let contador = nodo._____;
```

a) childNodes
b) innerHTML
c) children
d) childElementCount

---

19. ¿Qué método crea un nodo de tipo texto en el DOM?

```
let texto = document._____('Este es un texto');
```

a) createTextNode
b) appendChild
c) createElement
d) replaceChild

---

20. ¿Qué propiedad devuelve el nombre de la etiqueta de un nodo?

```
let nombre = nodo._____;
```

a) nodeName
b) nodeValue
c) tagName
d) nodeType

---

21. ¿Cómo verificas si un nodo contiene una clase específica?

```
let existeClase = nodo.classList._____('miClase');
```

a) add
b) toggle
c) remove
d) contains

---

22. ¿Qué método permite reemplazar un nodo hijo por otro?

```
padre._____(nuevoNodo, nodoViejo);
```

a) replaceChild
b) appendChild
c) removeChild
d) insertBefore

---

23. ¿Qué propiedad devuelve el siguiente nodo hermano de un elemento?

```
let siguiente = nodo._____;
```

a) childNodes
b) previousSibling
c) nextSibling
d) parentNode

---

24. ¿Qué método se utiliza para comprobar si un atributo existe en un nodo?

```
let tieneAtributo = nodo._____('nombreAtributo');
```

a) toggleAttribute
b) hasAttribute
c) getAttribute
d) setAttribute

---

25. ¿Qué propiedad permite obtener el contenido HTML completo de un elemento?

```
let contenido = nodo._____;
```

a) textContent
b) innerHTML
c) innerText
d) nodeValue

---

26. ¿Cómo accedes al último nodo hijo de un elemento?

```
let ultimoHijo = elemento._____;
```

a) parentNode
b) childNodes
c) lastChild
d) lastElementChild

---

27. ¿Qué método elimina todos los nodos hijos de un elemento?

```
while (nodo.firstChild) { nodo._____(nodo.firstChild); }
```

a) appendChild
b) clearChild
c) removeChild
d) deleteChild

---

28. ¿Qué método permite convertir una NodeList en un array?

```
let array = Array.from(nodoLista);
```

a) map
b) [...NodeList]
c) toArray
d) Array.from

---

29. ¿Qué propiedad devuelve todos los estilos calculados de un elemento?

```
let estilos = window._____(elemento);
```

a) getStyles
b) currentStyle
c) computedCSS
d) getComputedStyle

---

30. ¿Cómo se accede al valor de un atributo específico de un nodo?

```
let valor = nodo._____('atributo');
```

a) getAttribute
b) setAttribute
c) hasAttribute
d) removeAttribute

---

31. ¿Qué código crea un botón con texto "Click aquí" y lo añade a un contenedor en el DOM?

```
let boton = document.createElement('button');
boton.textContent = 'Click aquí';
let contenedor = document.getElementById('contenedor');
contenedor.appendChild(boton);
```

a) Reemplaza el contenedor por el botón creado.
b) Elimina todos los nodos del contenedor y añade el botón.
c) Crea un nodo de texto pero no lo añade al DOM.
d) Crea un botón con texto y lo añade al contenedor especificado.

---

32. ¿Qué hace este código en el DOM?

```
let nuevoParrafo = document.createElement('p');

nuevoParrafo.textContent = 'Este es un párrafo nuevo.';

document.body.insertBefore(nuevoParrafo,
document.body.firstChild);
```

a) Añade un párrafo al inicio del cuerpo del documento.
b) Lanza un error porque firstChild no es válido.
c) Añade un párrafo al final del cuerpo del documento.
d) Reemplaza el primer nodo del cuerpo por el nuevo párrafo.

---

33. ¿Qué resultado genera este código al ejecutarse?

```
let lista = document.createElement('ul');
for (let i = 1; i <= 3; i++) {
    let item = document.createElement('li');
    item.textContent = `Elemento ${i}`;
    lista.appendChild(item);
}
document.body.appendChild(lista);
```

a) Lanza un error porque item no está definido fuera del bucle.
b) Crea una lista desordenada con tres elementos y la añade al cuerpo.
c) Añade solo el último elemento de la lista al cuerpo.
d) Reemplaza todo el contenido del cuerpo con una lista vacía.

---

34. ¿Qué hace este código al ejecutarse?

```
let lista = document.getElementById('miLista');
if (lista.firstChild) {
    lista.removeChild(lista.firstChild);
}
```

a) Añade un nuevo elemento al inicio de la lista.
b) Lanza un error si la lista no tiene hijos.
c) Elimina el primer nodo hijo de la lista si existe.
d) Elimina todos los nodos hijos de la lista.

---

35. ¿Qué genera el siguiente código?

```
let div = document.createElement('div');
div.innerHTML = '<strong>Texto en negrita</strong>';
document.body.appendChild(div);
```

a) Crea un div vacío y lo añade al cuerpo.
b) Lanza un error porque innerHTML no acepta HTML.
c) Reemplaza todo el contenido del cuerpo con el div creado.
d) Crea un div con contenido HTML y lo añade al cuerpo.

36. ¿Qué hace este código al ejecutarse?

```
let elemento = document.createElement('p');

elemento.textContent = 'Texto del párrafo';

document.body.appendChild(elemento);

setTimeout(() => {

    document.body.removeChild(elemento);

}, 3000);
```

a) Añade un párrafo al cuerpo y lo mantiene indefinidamente.
b) Añade un párrafo, pero nunca lo elimina del cuerpo.
c) Lanza un error porque setTimeout no puede modificar el DOM.
d) Añade un párrafo al cuerpo y lo elimina después de 3 segundos.

---

37. ¿Qué hace este código en el DOM?

```
let nodo = document.createElement('span');

nodo.textContent = 'Este es un texto en span.';

document.body.replaceChild(nodo, document.body.firstChild);
```

a) Reemplaza el primer nodo del cuerpo con el nodo span creado.
b) Lanza un error porque replaceChild no es válido en document.body.
c) Añade un nodo span al final del cuerpo.
d) Elimina todos los nodos del cuerpo y añade el nodo span.

---

38. ¿Qué hace el siguiente código?

```
let encabezado = document.createElement('h1');

encabezado.textContent = 'Bienvenido a mi página';

document.body.prepend(encabezado);
```

a) Lanza un error porque prepend no es un método válido.
b) Añade un encabezado H1 con texto al inicio del cuerpo del documento.
c) Añade un encabezado H1 con texto al final del cuerpo del documento.
d) Reemplaza el contenido del cuerpo con el encabezado.

---

39. ¿Qué efecto tiene este código en el DOM?

```
let lista = document.querySelector('ul');

lista.innerHTML = '<li>Elemento 1</li><li>Elemento 2</li>';
```

a) Añade dos elementos al final de la lista existente.
b) Reemplaza todos los elementos de la lista con dos nuevos elementos.

c) Lanza un error si la lista ya tiene hijos.
d) No realiza cambios en la lista si está vacía.

---

40. ¿Qué hace este código al ejecutarse?

```
let div = document.getElementById('contenedor');

let parrafo = document.createElement('p');

parrafo.textContent = 'Párrafo dinámico';

div.appendChild(parrafo);

div.removeChild(parrafo);
```

a) Lanza un error porque no se puede eliminar un nodo recién añadido.
b) Añade un párrafo pero no lo elimina del contenedor.
c) Reemplaza el contenido del contenedor con el párrafo.
d) Añade y luego elimina un párrafo dinámico en el contenedor especificado.

---

41. ¿Qué genera el siguiente código?

```
let input = document.createElement('input');

input.type = 'text';

input.placeholder = 'Introduce tu nombre';

document.body.appendChild(input);
```

a) Lanza un error porque no se especifica un formulario padre.
b) Crea un campo de texto vacío pero no lo añade al DOM.
c) Crea un campo de texto con un marcador de posición y lo añade al cuerpo.
d) Crea un botón con texto.

---

42. ¿Qué efecto tiene este código en el DOM?

```
let lista = document.createElement('ul');

['Elemento A', 'Elemento B', 'Elemento C'].forEach(texto => {

    let item = document.createElement('li');

    item.textContent = texto;

    lista.appendChild(item);

});

document.body.appendChild(lista);
```

a) Lanza un error porque forEach no es compatible con arrays.
b) Añade solo el último elemento de la lista al cuerpo.
c) Crea una lista desordenada con tres elementos y la añade al cuerpo.
d) Crea una lista vacía y la añade al cuerpo.

---

43. ¿Qué hace este código en el DOM?

```
let boton = document.createElement('button');
boton.textContent = 'Eliminar párrafo';
boton.onclick = () => {
    let parrafo = document.getElementById('miParrafo');
    if (parrafo) parrafo.remove();
};
document.body.appendChild(boton);
```

a) Crea un botón que elimina un párrafo con ID "miParrafo" si existe.
b) Lanza un error porque remove no es un método válido.
c) Elimina directamente todos los párrafos del cuerpo.
d) Añade un párrafo con el texto "Eliminar párrafo".

---

44. ¿Qué efecto tiene este código?

```
let tabla = document.createElement('table');
for (let i = 0; i < 3; i++) {
    let fila = tabla.insertRow();
    for (let j = 0; j < 2; j++) {
        let celda = fila.insertCell();
        celda.textContent = `Fila ${i + 1}, Celda ${j + 1}`;
    }
}
document.body.appendChild(tabla);
```

a) Crea una tabla vacía y la añade al cuerpo.
b) Lanza un error porque insertCell no es un método válido.
c) Reemplaza todo el contenido del cuerpo con la tabla.
d) Crea una tabla de 3 filas y 2 columnas con contenido dinámico y la añade al cuerpo.

---

45. ¿Qué genera este código?

```
let enlace = document.createElement('a');
enlace.href = 'https://example.com';
enlace.target = '_blank';
enlace.textContent = 'Abrir enlace en nueva pestaña';
document.body.appendChild(enlace);
```

a) Lanza un error porque no se permite el atributo target.
b) Añade un enlace sin funcionalidad al cuerpo.
c) Crea un enlace que se abre en una nueva pestaña y lo añade al cuerpo.
d) Crea un enlace pero ignora el atributo target.

---

46. ¿Qué hace este código en el DOM?

```
let imagen = document.querySelector('#miImagen');

imagen.setAttribute('alt', 'Descripción alternativa');

imagen.src = 'nueva-imagen.jpg';
```

a) Elimina todos los atributos existentes de la imagen.
b) Actualiza el texto alternativo y la fuente de una imagen con ID "miImagen".
c) Lanza un error si la imagen no tiene atributos previos.
d) Reemplaza la imagen por un texto alternativo.

---

47. ¿Qué hace este código en el DOM?

```
let lista = document.querySelector('ul');

lista.childNodes.forEach(nodo => {

    if (nodo.nodeType === 3) {

        lista.removeChild(nodo);

    }

});
```

a) Reemplaza los nodos de texto con elementos vacíos.
b) Elimina todos los nodos de texto de la lista seleccionada.
c) Lanza un error porque childNodes no es iterable.
d) Elimina todos los elementos de la lista seleccionada.

48. ¿Qué hace este código en el DOM?

```
let contenedor = document.getElementById('contenedor');

contenedor.innerHTML = '<h2>Título dinámico</h2>';
```

a) Reemplaza el contenido del contenedor con un título dinámico.
b) Añade un título al final del contenedor.
c) No realiza cambios si el contenedor está vacío.
d) Lanza un error si el contenedor tiene contenido previo.

---

49. ¿Qué efecto tiene este código en el DOM?

```
let imagen = document.createElement('img');

imagen.src = 'imagen.jpg';
```

```
imagen.alt = 'Descripción de la imagen';

document.body.appendChild(imagen);
```

a) Lanza un error porque faltan los atributos obligatorios.
b) Añade una imagen vacía al cuerpo del documento.
c) Reemplaza el contenido del cuerpo con la imagen creada.
d) Crea una imagen con atributos src y alt, y la añade al cuerpo del documento.

---

50. ¿Qué hace este código al ejecutarse?

```
let boton = document.querySelector('#miBoton');

boton.remove();
```

a) Elimina el botón con ID 'miBoton' del DOM si existe.
b) Cambia el texto del botón pero no lo elimina.
c) Lanza un error si el botón no existe.
d) Reemplaza el botón con un nodo vacío.

---

51. ¿Qué genera el siguiente código?

```
let enlace = document.createElement('a');

enlace.href = 'https://example.com';

enlace.textContent = 'Visita este sitio';

document.body.appendChild(enlace);
```

a) Crea un enlace que se abre en una nueva pestaña.
b) Añade solo el texto del enlace sin funcionalidad.
c) Lanza un error porque no se puede crear un enlace dinámico.
d) Crea un enlace al sitio especificado y lo añade al cuerpo del documento.

---

52. ¿Qué hace este código al ejecutarse?

```
let parrafo = document.getElementById('parrafo');

parrafo.style.color = 'blue';

parrafo.style.fontWeight = 'bold';
```

a) Cambia solo el color del párrafo, ignorando el peso de la fuente.
b) Aplica estilos CSS para cambiar el color a azul y el texto a negrita en el párrafo.
c) Lanza un error porque no se puede acceder directamente al estilo.
d) Elimina todos los estilos existentes en el párrafo.

---

53. ¿Qué efecto tiene este código?

```
let lista = document.createElement('ul');
```

```
for (let i = 0; i < 2; i++) {
    let item = document.createElement('li');
    item.textContent = `Elemento ${i + 1}`;
    lista.appendChild(item);
}
document.body.appendChild(lista);
```

a) Añade solo el último elemento de la lista al cuerpo.
b) Lanza un error porque item no está definido fuera del bucle.
c) Crea una lista desordenada con dos elementos y la añade al cuerpo.
d) Crea una lista vacía y la añade al cuerpo.

---

54. ¿Qué hace este código?

```
let tabla = document.createElement('table');
let fila = tabla.insertRow();
let celda = fila.insertCell();
celda.textContent = 'Celda de la tabla';
document.body.appendChild(tabla);
```

a) Reemplaza el contenido del cuerpo con la tabla creada.
b) Crea una tabla vacía y la añade al cuerpo.
c) Lanza un error porque insertRow y insertCell no son válidos.
d) Crea una tabla con una fila y una celda, y la añade al cuerpo del documento.

---

55. ¿Qué hace este código en el DOM?

```
let div = document.createElement('div');
div.classList.add('miClase');
div.style.backgroundColor = 'red';
document.body.appendChild(div);
```

a) Crea un div con una clase y un fondo rojo, y lo añade al cuerpo del documento.
b) Lanza un error porque no se permite modificar estilos desde JavaScript.
c) Añade un div al cuerpo sin estilos ni clases.
d) Reemplaza el contenido del cuerpo con un div vacío.

---

56. ¿Qué hace este código al ejecutarse?

```
let lista = document.querySelector('ul');
lista.innerHTML += '<li>Nuevo elemento</li>';
```

a) Añade un nuevo elemento al inicio de la lista.
b) Lanza un error porque innerHTML no permite añadir contenido dinámico.
c) Reemplaza todos los elementos existentes en la lista con uno nuevo.
d) Añade un nuevo elemento al final de la lista.