

Ejercicios UT6

1 . Crea una página HTML con este input tipo text:

```
<input type="text" id="apellidos" name="apellidos" placeholder="apellido1  
apellido2"/>
```

- a) Añade un botón llamado “Imprime atributos del input” que al pulsarlo muestre en el documento HTML el nombre y valor de cada uno de los atributos de este elemento input. Utiliza la propiedad attributes. Pincha este botón
- b) Añade un botón llamado “Cambia tipo a number utilizando el método setAttribute”. Al hacer clic a este botón se cambiará el tipo de input de text a number. Utiliza el método setAttribute(). Pincha este botón y luego el botón imprime.
- c) Añade un botón más llamado “Cambia tamaño a 100 utilizando el metodo setAttribute”. Al hacer clic a este botón se cambiará el tamaño de la caja de texto a 100. Utiliza el método setAttribute(). Pincha este botón y luego el botón imprime.
- d) Añade un botón más llamado “Cambia placeholder a 40” utilizando el metodo setAttribute”. Al hacer clic a este botón se cambiará el placeholder a “40”. Utiliza el método setAttribute(). Pincha este botón y luego el botón imprime.
- e) Añade otro botón más llamado “Eliminar el atributo size” que elimine el atributo size con el método removeAttribute(). Pincha este botón y luego el botón imprime.

2 . Crea un documento HTML con los siguientes elemento (3 inputs y 1 botón):

Nombre:

Apellidos:

Edad:

Al pulsar al botón se listarán todos los valores introducidos por el usuario en los input. Utiliza el método `getElementsByTagName ()` y un bucle for para recorrer todos los elementos de tipo “input”

3 . Crea una página HTML con el siguiente elemento radio-button:

```
<form id="formulario">  
  <input type="radio" id="c_madrid" name="ciudad" value="Madrid">  
  <label for="c_madrid">Madrid</label><br>  
  
  <input type="radio" id="c_sevilla" name="ciudad" value="Sevilla">  
  <label for="c_sevilla">Sevilla</label><br>  
  
  <input type="radio" id="c_cuenca" name="ciudad" value="Cuenca">  
  <label for="c_cuenca">Cuenca</label>  
</form>
```

Además crea un botón que al pulsarlo lista los atributos value de cada elemento html con nombre ciudad utilizando `getElementByName()` y bucle `for`.

4. Crea una página HTML con el siguiente elemento párrafo:

```
<p id="Parrafo" title="Titulo"></p>
```

Recorre los atributos de este elemento párrafo con la propiedad `attributes` y un bucle `for`.

5. Crea una nueva página HTML con este elemento:

```
<p title="Texto de un párrafo">Esto es un ejemplo de <b>texto HTML<br />
que puedes tener</b> en tu documento</p>
```

En este ejercicio vamos a modificar los nodos textos de este elemento HTML “p”, para ello vamos a recorrer los nodos de este elemento utilizando `childNodes` y modificando sus nodos textos con `nodeValue`

Con un botón llamado “Añade HOLA a en tu documento”, y con otro botón llamado “Añade HOLA a que puedes tener”.

6. Crea una nueva página HTML, con un párrafo como éste:

```
<p id="demo">Parrafo creado en HTML.</p>
```

En este ejercicio vamos a crear un nuevo elemento HTML “p” y todos sus nodos hijos (elementos y textos) utilizando `createElement()`, `createTextNode()` `appendChild()`

Y con un botón llamado “Crear párrafo”, que al pulsarlo cree el siguiente párrafo desde JavaScript de la siguiente forma:

- Cree 3 elementos: p, b y br.
- Crear al elemento p el atributo title “Parrafo creado desde JavaScript”.
- Crear los siguientes 4 textos:
 1. Con JavaScript se
 2. pueden realizar
 3. un montón
 4. de cosas sobre el documento.
- Se asigna al elemento b los siguientes nodos hijos y por este orden:
 - El texto 2 (pueden realizar).
 - El elemento br creado al principio.
 - El texto 3 (un montón).
- Se asigna al elemento p los siguientes nodos hijos y por este orden:
 - El texto 1 (Con JavaScript se).
 - El elemento b anterior.
 - El texto 4 (de cosas sobre el documento).
- Asigna al elemento párrafo demo el elemento p que has creado.

¿Qué resultado da? Dibuja el árbol DOM

7. A la aplicación anterior crea los siguientes botones.
 1. “Reemplazar br por un nuevo b” que haga lo siguiente:
 - a) Cree un elemento .
 - b) Cree un nodo texto con valor “Retorno de carro”.
 - c) Añada este nodo texto anterior al elemento b.
 - d) Reemplace el elemento
 por este nuevo elemento
 2. “Eliminar elemento nuevo b” que elimine el anterior elemento b
8. A la aplicación anterior añade un botón llamado “Recorrer”. Al pulsarlo se ha de recorrer desde el comienzo hasta el final todos los nodos del párrafo. Si se encuentra con un nodo hijo que tiene hijos a su vez también ha de recorrerlos.
9. Crea una aplicación web con:
 1. Un input de tipo number llamado “columnas”.
 2. Un input de tipo number llamado “filas”.
 3. Un botón llamado “Crear tabla” que cree una tabla con el número de filas y el número de columnas indicados en los inputs anteriores
10. Ejercicio de utilización de modelo de registro de eventos en línea. Crea un programa HTML que:
 1. Muestre un hipervínculo a otra página HTML.
 2. Al hacer click sobre este hipervínculo se debe ejecutar una función JavaScript llamada avisar(). Esta función avisar debe sacar una alerta informando de que se va a abrir una página HTML muy pesada.
11. Ejercicio de utilización de modelo de registro de eventos en línea evitando que se ejecute la acción del objeto HTML desde código HTML. Crea un programa HTML que:
 1. Muestre un hipervínculo a otra página HTML.
 2. Al hacer click sobre este hipervínculo se debe ejecutar una función JavaScript llamada avisar(). Esta función avisar debe sacar una alerta informando de que el enlace no funciona, y, por tanto, no se puede ejecutar el hipervínculo.
12. Ejercicio de utilización de modelo de registro de eventos en línea. Crea un programa HTML que:
 1. Muestre un hipervínculo a otra página HTML.
 2. Al hacer click sobre este hipervínculo se debe ejecutar una función JavaScript llamada confirmar(). Esta función confirmar debe informar al usuario que la página que desea abrir es pesada, y pedirle continuación de si quiere abrirla. Si el usuario le da a Aceptar la página ha de abrirse y si le da a Cancelar no.
13. Ejercicio de utilización de modelo de registro de eventos en línea evitando que se ejecute desde código JavaScript. Crea un programa HTML que:
 1. Muestre un hipervínculo. Este hipervínculo se cargará al hacer click si el usuario es mayor de edad, si no es mayor de edad saldrá un mensaje diciendo no puedes visitar esta página por ser menor de edad.
 2. Para ello debes utilizar una función JavaScript asociada al evento onclick. Será la propia función JavaScript la que pedirá al usuario que facilite su edad.

3. Razona: ¿Qué se ejecuta antes el evento o la acción del objeto (la carga de la página HTML para mayores de edad?

14. Ejecuta este código y analiza qué hace el operador this en los eventos en línea que aparecen:

```
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="content-type" content="text/html; charset=utf-8">
<title>Uso de la palabra reservada this</title>
<script type="text/javascript">
function identificar(objeto){
    var atrName=objeto.name;
    var atrId=objeto.id;
    var atrValue=objeto.value;
    var atrType=objeto.type;

    alert("Datos del campo pulsado:\n\nName: "+atrName+"\nID: "+atrId+"\nValue:
"+atrValue+"\nType: "+atrType);
}
</script>

</head>

<body>
<h1>Trabajando con this</h1>
<form id="formulario" action="pagina.php">
<p>

<label for="nombre">Nombre: </label>
<input type="text" name="nombre" id="nombre" value="Constantino"
onclick="identificar(this)"/>

<label for="apellidos">Apellidos: </label>
<input type="text" name="apellidos" id="apellidos" value="Veiga Perez"
onclick="identificar(this)"/>

<label for="edad">Edad: </label>
<input type="password" name="edad" id="edad" value="55"
onclick="identificar(this)"/>

<label for="pais">Pais: </label>
España <input type="radio" name="pais" id="pais1" value="ES"
onclick="identificar(this)"/>

Francia <input type="radio" name="pais" id="pais2" value="FR"
onclick="identificar(this)"/>

</p>
Haga clic en cada uno de los campos para ver más información.

</form>
</body>
</html>
```

15. Realizar el Ejercicio 1 de los Ejercicios completos unidad trabajo 4 (el de la calculadora) con this.

16. Ejercicio de utilización de modelo de eventos en línea con el objeto this:

1. Crea un programa HTML con un hipervínculo con identificar “*enlace*” a la página del Ayuntamiento de Alcobendas (<https://www.alcobendas.org/es/portal.do?IDM=27&NM=1>)
2. En este hipervínculo HTML introduce el atributo *onclick* para que llame a una función llamada *informar*, que avise mediante una ventana de alerta que se va a abrir un enlace, indicando la URL del hipervínculo. NOTA: La función *informar* conseguirá la URL porque recibirá el objeto HTML *enlace* como argumento utilizando *this*
17. Repite el ejercicio 10, pero con el modelo de registros de eventos tradicional (separando la estructura HTML de la codificación JavaScript).
18. Repite el ejercicio 14 pero utilizando el modelo tradicional de registro de eventos.
19. Repite el ejercicio 16 con el método tradicional.
20. Repite el ejercicio 16 con el método avanzado W3C. Razón: visualiza dentro de la función asociada al evento click, el valor del atributo *onclick* del elemento HTML hipervínculo “*enlace*” ¿Qué aparece?, ¿por qué?
21. Repite el ejercicio 16 utilizando una función anónima.
22. Repite el ejercicio 16 utilizando una función anónima y función flecha.
23. Crea un programa HTML que haga lo siguiente utilizando los métodos `addEventListener` y `removeEventListener`:
 1. Tenga un párrafo con el siguiente contenido: “Si pasas por encima de mí con el ratón (sin hacer click) se escribirá más abajo Eres el/la mejor. Si haces click sobre mí dejará de ocurrir esto.”
 2. Cada vez que se pase por encima de este párrafo con el ratón (evento `mouseover`) aparece más abajo la frase “Eres el/la mejor” (esta frase se irá acumulando cada vez que se pase el botón por encima, es decir, no desaparece lo que ya esté escrito en la página)
 3. Al hacer click sobre el párrafo ese comportamiento de `mouseover` se eliminará, y se borrarán todas las frases “Eres el/la mejor” que haya escritas.
 4. Tenga un botón que al hacer click sobre el botón elimine el evento `mouseover` del párrafo, y que borre todas las frases “Eres el/la mejor”
24. Utilizando el método `preventDefault` crea un programa HTML que haga lo siguiente:
 1. Tenga un elemento `div` vacío.
 2. Un formulario con 1 `checkbox` que pregunte al usuario si come al día fruta 2 veces o más.
 3. De forma que cuando se pulse el `checkbox`:
 - a) No se marque.
 - b) Aparezca en el `div` el texto “NO me lo creo” en tamaño normal.
 - c) Tras pulsar por segunda vez el `checkbox` este texto se hace un poco más grande.
25. Crea un programa HTML con lo siguiente:
 1. Un botón y un enlace.
 2. Se deben capturar los eventos `load` de la ventana, `click` del enlace, y `click` del botón, asociándoles la función `esCancelable()`.
 3. Esta función va a hacer lo siguiente:
 - a) Sacar por pantalla en qué elemento se ha producido el evento (utiliza `this`).

- b) Sacar por pantalla si ese evento es cancelable o no (utiliza la propiedad `cancelable` de los eventos)
- c) Intentar cancelar la acción por defecto (utiliza `preventDefault()`)

26. Haz este ejercicio https://www.w3schools.com/jsref/tryit.asp?filename=tryjsref_element_addeventlistener_capture. Saca conclusiones.

27. Escribir una aplicación en JavaScript que simule un pequeño tablero de dibujo.

Para ello tendrás que dibujar una tablero de 30x30 celdas de 10x10px cada una. Para realizar el tablero de dibujo tendrás que emplear obligatoriamente los métodos de creación de nodos del DOM. Una vez generado el tablero lo meterás dentro de un elemento `<div>` identificador "zonadibujo". Cuando se crea la tabla se le da a border el valor 1px, y se le asigna la clase "tablerodibujo". Esta clase está definida en el fichero css que se facilita (la URL está más abajo)

Además tendrás una paleta con 5 colores de dibujo (que ya está creada y se facilita con el código .html)

Se facilita un fichero .html y un fichero .css con los estilos que tendrás que utilizar (<https://aulavirtual32.educa.madrid.org/ies.lapaz.alcobendas/mod/resource/view.php?id=29360>). La programación de la aplicación JavaScript la tendrás que realizar en un fichero .js adicional.

El código .js tiene que seguir el estandar de W3C para la gestión de eventos, creación de elementos HTML, etc.

La forma de funcionamiento de la aplicación será la siguiente:

- En el código HTML hay programado una tabla inicial con identificador "paleta" con 5 colores, 1 color en cada celda. Haremos clic en alguno de los 5 colores de la paleta y se le asignará a la celda de ese color la clase "seleccionado". Solo puede haber 1 color con clase "seleccionado".
- Una vez seleccionado un color de la paleta, haremos un clic en una celda del tablero de dibujo (que se pintará del color activo en la paleta) y desde ese momento al mover el ratón por el tablero pintará del color activo todas las celdas por las que vayamos pasando el ratón. En el momento que volvamos a hacer clic en otra celda dejará de pintar.
- Podremos escoger un color diferente y repetir el proceso, incluso sobre celdas que ya han sido pintadas.
- Para borrar una celda pintaremos con el color blanco de la paleta.
- Cada vez que el pincel esté activado se mostrará un mensaje debajo de la paleta de colores indicando : PINCEL ACTIVADO o PINCEL DESACTIVADO.

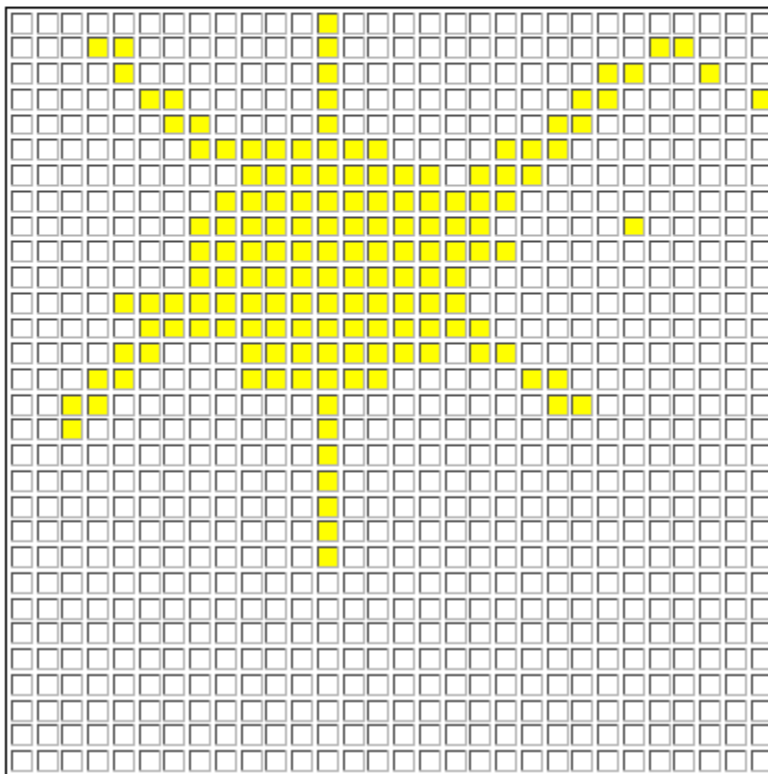
Captura de pantalla del aspecto de la aplicación:

TABLERO DE DIBUJO EN JAVASCRIPT

Haga click en un color para comenzar a pintar



Haga click en cualquier tecla para activar/desactivar el Pincel.



28. Modifica el programa anterior para que haga lo siguiente (para practicar fase de burbujeo y de captura):

1. Cuando se haga click en el tablero de dibujo para activar el pincel, antes de la activación del pincel se debe preguntar al usuario si desea agrandar el tablero. Si contesta que sí, se debe agrandar el tablero 2 filas. Después de esto se debe activar el pincel pero no antes

NOTA: Se debe registrar este evento click en el tablero de dibujo, no en las celdas. Por lo que tendremos 2 eventos click en elementos anidados:

- a) El nuevo click del tablero que preguntará si se agranda o no el tablero.
- b) El evento click en todas las celdas que activa el pintado y que ya está programado en el ejercicio anterior.

29. Incorpora este código HTML a una página web.

Programa el código JavaScript para que pulsar la tecla `Intro` dentro de cualquiera de los inputs se mueve el foco al input siguiente incluyendo el botón. Recorre los nodos del formulario utilizando el método `nextSibling()`

Al llegar al final del formulario (el botón) se debe pasar el foco al primer input del formulario.

```

<form name="formulario" id="formulario">
  <label for="nombre">Nombre:</label>
  <input type="text" id="nombre" name="nombre" />

  <label for="apellidos">Apellidos:</label>
  <input type="text" id="apellidos" name="apellidos" />

  <label for="provincia">Provincia:</label>
  <input type="text" id="provincia" name="provincia" />

  <input type="button" id="enviar" value="Enviar" />
</form>

```

NOTA: tened en cuenta que el formulario es un nodo padre de los siguientes nodos y por este orden:

- Nodo elemento label nombre.
- Nodo texto que es un enter que hay detrás del label. Los retornos de carro, espacios en blanco etc. que hayan entre elementos los considera un nodo texto.
- Nodo elemento input nombre.
- Nodo texto que es un enter que hay detrás del input nombre.
- Nodo elemento label apellidos.
- Nodo texto que es un enter que hay detrás del label.
- Nodo elemento input apellidos.
- Nodo texto que es un enter que hay detrás del input apellidos.
- Nodo elemento label provincia.
- Nodo texto que es un enter que hay detrás del label.
- Nodo elemento input provincia.
- Nodo texto que es un enter que hay detrás del input provincia.
- Nodo elemento input button.
- Nodo texto que es un enter que hay detrás del input button.

30. Igual que el anterior pero el avance del foco es hacia atrás. Cuando llegue al primer input debe pasar el foco al botón. Recorre los nodos del formulario utilizando el método `previousSibling()`

31. Modifica el ejercicio 26 para que se imprima también en qué fase está el evento en cada momento que se ejecuta su manejador (utiliza la propiedad `eventPhase`).

32. Modifica el ejercicio 26 para evitar que el evento burbujee (en el primer bloque `div-p`). Utiliza el método `stopPropagation()`

33. Descarga el ejercicio resuelto que hay en el aula virtual sobre `client`, `page`, `offset` y `screen` y saca conclusiones.

34. Dado esta página HTML, programa en JavaScript para que el párrafo siga al ratón cuando cuando el ratón se mueve.

NOTA: para ello tienes que modificar la propiedad `style.left` y `style.top` del párrafo


```

<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Mover elemento</title>
</head>
<body>
  <p id="demo" style="position: absolute;">Holis</p>

<script>

</script>
</body>
</html>

```

35. Modifica el programa del ejercicio 33 para que solo te muestre las coordenadas si está pulsada la tecla Ctrl (utiliza la propiedad ctrlKey del evento, en este caso mouseover)
36. Crea una página HTML con un párrafo. Registra en ese párrafo los 2 eventos siguientes:
 1. El evento mouseover de forma que cuando se produzca indique de qué elemento viene el ratón (utiliza relatedTarget del evento mouseover)
 2. El evento mouseout de forma que cuando se produzca indique a qué elemento se ha ido el ratón (utiliza relatedTarget del evento mouseover)
37. Modifica el ejercicio 28 para que cuando el usuario no quiera agrandar no se active la pintura.
38. Modifica el ejercicio 29 resolviéndolo utilizando el método nextElementSibling. La diferencia entre nextElementSibling y nextSibling, es que nextElementSibling devuelve el siguiente nodo que sea de tipo element, nextSibling devuelve el siguiente nodo que sea de tipo text o de tipo element.
39. Modifica el ejercicio 30 resolviéndolo utilizando el método previousElementSibling. La diferencia entre previousElementSibling y previousSibling, es que previousElementSibling devuelve el nodo anterior que sea de tipo element, previousSibling devuelve el nodo anterior que sea de tipo text o de tipo element
40. Modifica el ejercicio 29 resolviéndolo sin utilizar nextSibling ni nextElementSibling, puedes utilizar el objeto this.
41. jkjlñk
42. kjñklj