

Ejercicios UT6

1. No incluido.
2. No incluido.
3. No incluido.
4. No incluido.
5. No incluido.
6. No incluido.
7. No incluido.
8. No incluido.
9. No incluido.
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 clic 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 clic 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 clic 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 clic 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. Razón: ¿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. No incluido.

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. Razona: visualiza dentro de la función asociada al evento click, el valor del atributo onclick del elemento HTML hiperenlace “*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 frase “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()`)