Tema 4 – Uso del dom – ejercicio 3 – eventos y formularios

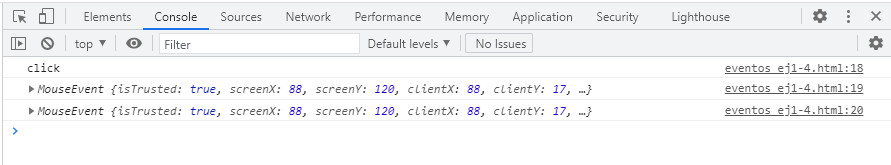
Documenta con explicaciones, fragmentos de código y capturas las siguientes tareas.

Eventos

Crea un fichero t4\_ej3\_eventos.html en el que incluyas las respuestas a las siguientes tareas:

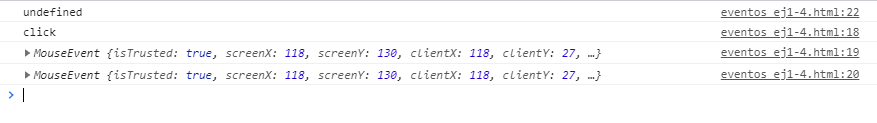
1. Añade un botón y registra una función manejadora para el evento “click”. Declara el objeto event como parámetro de la función manejadora, y comprueba mostrando por consola, que el evento recibido por parámetro es igual al objeto window.event (contiene el ultimo evento lanzado en la ventana). Adjunta capturas de la consola.

.Son iguales



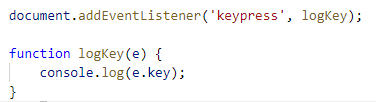
1. Desde el manejador anterior muestra con console.dir el objeto sobre el que se dispara el evento, es decir el botón. ¿Qué objeto es?. Adjunta captura.

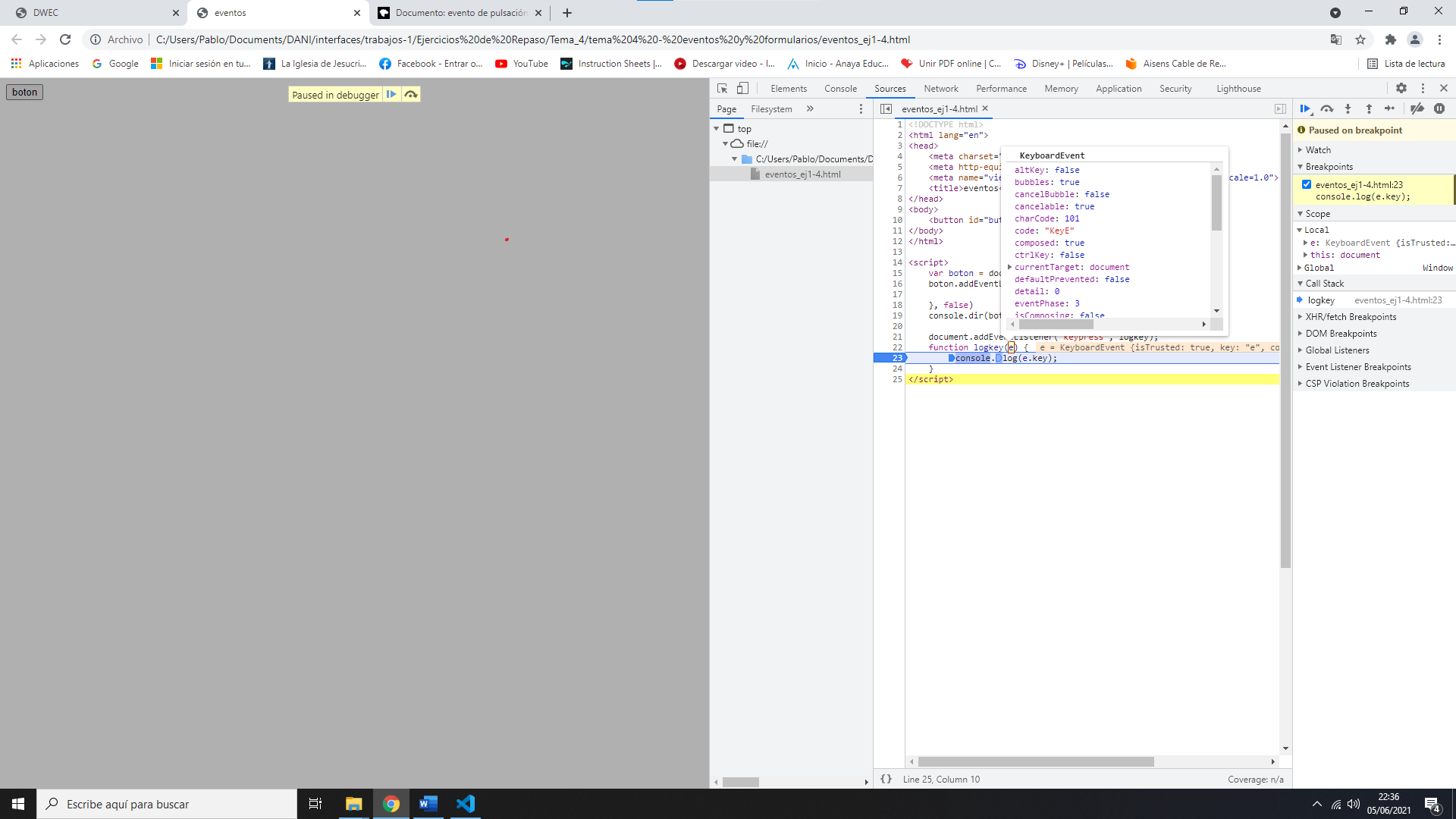
.Un objeto click



1. ¿Qué realiza el siguiente código? ¿Qué contiene el parámetro **e**?

es un evento que actúa al presionar la tecla “e” del teclado y en este caso solo lo muestra por consola





1. Comprueba el burbujeo de eventos: crea un nodo contenedor con un manejador del evento “click” que muestre un alert(“Has hecho click en el contenedor”).

Luego añade 50 botones al contenedor. Y para cada botón un manejador, para que al pulsar sobre cada uno de ellos muestre un alert diferente, en el que aparece su número de botón. Comprueba que se ejecutan ambas alertas al pulsar cualquiera de los botones:

Por ejemplo, si pulsamos sobre el primer botón saldrán 2 alertas:

* + “Hola, soy el botón 1”
  + “Has hecho click en el contenedor”

Formularios

Crea un fichero t4\_ej3\_formularios.html en el que incluyas las respuestas a las siguientes tareas:

1. A partir del siguiente HTML muestra por console.dir() el objeto formulario y su action.

<body onload="inicio()">

    <div id="menulateral">

        <form id="contactar" name="contactar" action="prueba.php" >

            <input type="text" id="mail" >

            <input type="text" id="nombre" >

            <input type="text" id="apellidos" >

            <label for="checkConforme">Está conforme en compartir esta información</label>

            <input type="checkbox" id="checkConforme">

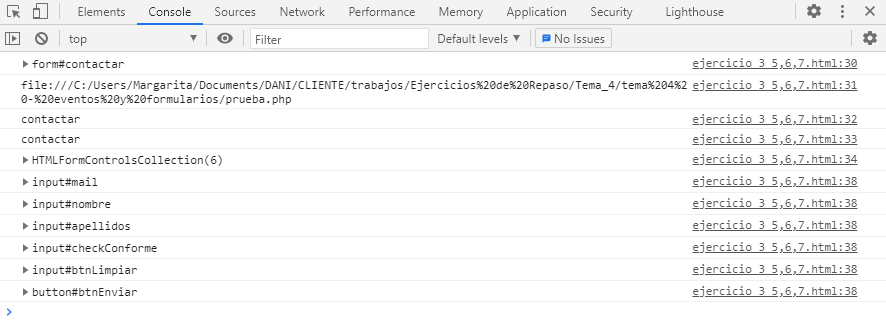
            <input type="button" id="btnLimpiar" value="Limpiar">

            <button id="btnEnviar">Enviar</button>

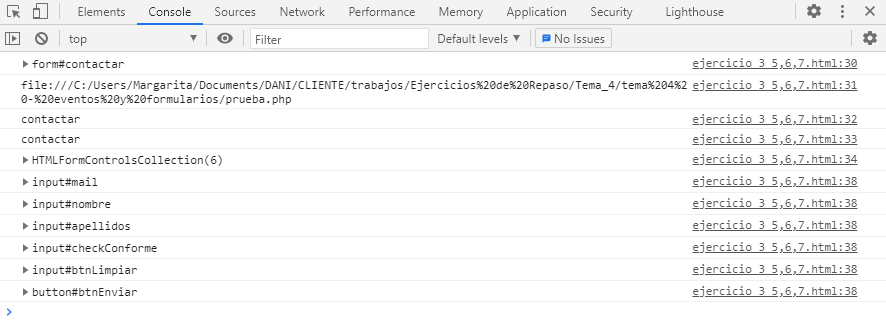
        </form>

    </div>

</body>

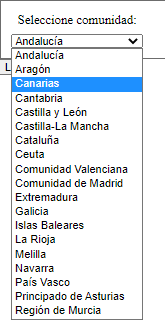


1. Recorre todos los elementos del formulario mostrándolos por consola.



1. Añade un manejador (addEventListener) al botón “limpiar” que recorra todos los elementos del form e inicialice a cadena vacía sólo los elementos de tipo input “text”.
2. Simulemos que hemos hecho una petición al servidor y éste nos ha devuelto el siguiente array de comunidades autónomas. Implementa el código necesario para cargar todos los elementos del array como opciones de un input de tipo select.

var comunidades= ["Andalucía", "Aragón", "Canarias", "Cantabria", "Castilla y León", "Castilla-La Mancha", "Cataluña", "Ceuta", "Comunidad Valenciana", "Comunidad de Madrid", "Extremadura", "Galicia", "Islas Baleares", "La Rioja", "Melilla", "Navarra", "País Vasco", "Principado de Asturias", "Región de Murcia"];



Pista: Necesitaremos recorrer el array e ir creando en tiempo de ejecución, elementos de tipo “option” usando los métodos vistos para la creación de nodos del DOM.

Validacion de formularios

1. Responde las siguientes preguntas:
   * ¿Si, usamos JavaScript para validar un formulario, será necesario también validar esos datos en el lado del servidor?

No porque la validación de los campos de un formulario en el frontend se realiza antes del envío de los datos al servidor

* + ¿Qué pasaría con nuestras validaciones si por ejemplo desactivamos JavaScript en el navegador?
  + ¿Qué pasaría si alguien programa una copia de nuestro formulario en otro servidor web para enviar datos a nuestro servidor sin validarlos previamente?

1. Valida los campos de texto del formulario anterior, usa expresiones regulares en aquellos campos que creas necesario:
   * Nombre: obligatorio y solo debe aceptar letras, no números y una longitud máxima de 11 caracteres.
   * Apellidos: obligatorio y solo debe aceptar letras, no números y una longitud máxima de 50 caracteres.
   * Mail: No obligatorio. Pero, si se introduce, ha de validarse un formato de email correcto. Por ejemplo: [aaaaa@bbbbb.ccc](mailto:aaaaa@bbbbb.ccc).
   * Check conforme: obligatorio
   * Seleccione comunidad: no obligatorio.
   * Botón de enviar: ha de permanecer deshabilitado hasta que todos los elementos obligatorios del formulario se hayan completado. Si al pulsarlo alguno de los elementos no cumple la validación su borde debe marcarse en rojo indicando en un mensaje (por ej: bajo el elemento, o a un lado) que ese elemento tiene un formato incorrecto.