

Ejercicio 3: JavaScript

Fecha de entrega: 23 de diciembre de 2024

1- JavaScript y formularios

Descarga el archivo *formulario.zip* en una carpeta de tu ordenador y extrae los dos archivos que contiene, *javascript.html* y *enviado.html*, en esa carpeta. Abre el archivo *javascript.html* en tu navegador y pulsa el botón Enviar. Deberás ver la página *enviado.html*. La siguiente imagen muestra el formulario implementado en el fichero *javascript.html*.

JavaScript y formularios

Nombre: ?

Edad: - +

Código Postal:

Sexo: ☐ Hombre ☐ Mujer ☐ Otro

Colectivo:

Titulación: ☐ Grado ☐ Máster ☐ Doctorado

Para poder manejar los controles del formulario debemos conocer el nombre que se le ha dado a cada uno. Todos ellos se encuentran dentro del formulario que se llama, sencillamente, *formulario*. La siguiente tabla muestra los nombres de los distintos controles:

Tipo de control	Texto adjunto/interno	Nombre del control
Cuadro de texto	Nombre:	nombre
Botón	?	ayuda
Botón	-	menos
Cuadro de texto (centro)	Edad:	edad
Botón	+	mas
Cuadro de texto	Código Postal:	cp
Botones de radio	Sexo:	sexo
Lista desplegable	Colectivo:	colectivo
Casilla de verificación	Grado	grado
Casilla de verificación	Máster	master
Casilla de verificación	Doctorado	doctorado
Área de texto		valores
Botón	Comprobar	comprobar
Botón	Desactivado	desactivado

La página enviado.html simplemente indica que se ha enviado el formulario y nos servirá para comprobar el efecto de pulsar el botón Enviar (si se envía o no).

Se requiere realizar los siguientes ejercicios:

1. Al cargarse la página se inicialice el cuadro de texto de edad con el valor 21.
2. Al cargarse la página se inicialice el cuadro de texto del código postal con el valor 00000.
3. Cuando se pulse el botón “-”, se requiere que se disminuya en una unidad el valor del campo de texto de edad y cuando se pulse el botón “+” se aumente en una unidad ese valor.
4. Como el formulario es sólo para mayores de edad, se requiere que no se pueda disminuir el cuadro de edad por debajo de los 18.
5. El botón Desactivado, a pesar de su nombre se puede pulsar, se requiere que cuando se ponga el puntero del ratón sobre ese botón se muestre al usuario un cuadro de mensaje que le diga que ese botón no se puede pulsar.

6. Cuando pulse el botón “?” de la derecha del nombre, se requiere pedir al usuario nombre y apellidos (por separado), para poner el nombre completo en el cuadro de texto.
7. Se requiere realizar algunas comprobaciones antes de enviar el formulario. Para empezar, se requiere implementar una función *enviar()* que mostrará un cuadro de confirmación al usuario para que confirme (o no) que lo quiere enviar.
8. Antes de enviar los datos del formulario se requiere que se haya seleccionado una opción en los botones de radio de *sexo*. En el caso de no haberlo seleccionado se requiere informar al usuario e impedir el envío de los datos.
9. Antes de enviar los datos del formulario se requiere que se haya seleccionado una opción en la lista desplegable *colectivo*. En el caso de no haberlo seleccionado se requiere informar al usuario e impedir el envío de los datos.
10. Antes de enviar los datos del formulario se requiere comprobar que el código postal contiene exactamente cinco caracteres. Si no es así, lo notificará al usuario y pondrá el foco en ese control y lo seleccionará.
11. Se requiere comprobar que en el código postal sólo haya dígitos. Si no es así, lo notificará al usuario y pondrá el foco en ese control y lo seleccionará.
12. Cuando pulse el botón “Comprobar” se requiere que se muestre en el área de texto los valores actuales de los campos del formulario.

JavaScript y formularios

Nombre: ?

Edad:

Código Postal:

Sexo: ☒ Hombre ☐ Mujer ☐ Otro

Colectivo:

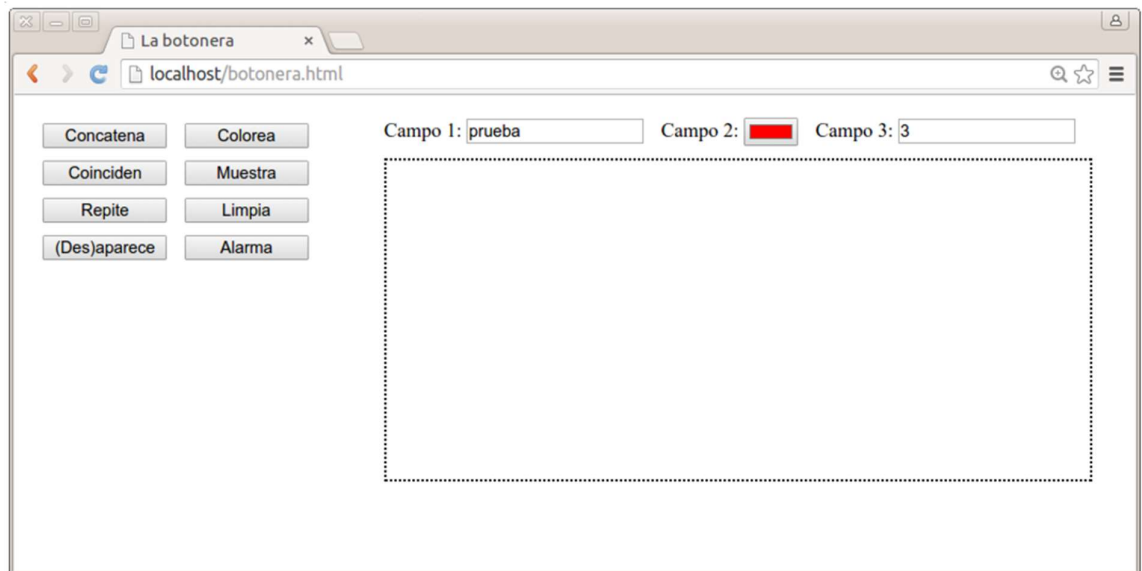
Titulación: ☒ Grado ☒ Máster ☐ Doctorado

```
Nombre = Luis Hernández Yáñez
Edad = 28
Código postal = 00000
Sexo = hombre
Colectivo = pdi
Grado = Sí
Máster = Sí
Doctorado = No
```

2- JavaScript e Interacciones

Descarga el archivo *Botonera.zip* que encontrarás en el espacio virtual de la asignatura y descomprímelo. Dentro de la carpeta ya estarán creados los ficheros “*botonera.html*”, “*botonera.css*” y “*botonera.js*”. Los dos primeros contienen el código que define la estructura de la página y **que no deberían ser modificados (para enlazar los eventos utilizar el método “*addEventListener*”)**. La página consta de los siguientes elementos:

- 8 botones de acción.
- 3 campos de texto: *campo1* de tipo *text*, *campo2* de tipo *color* y *campo3* de tipo *number*.
- Un bloque con identificador *principal*.
- Un párrafo para mostrar un *mensaje*.



El fichero “*botonera.js*” debe implementar las siguientes interacciones:

1. Cuando el ratón entre en *principal* se establecerá como color de fondo *beige*.
2. Cuando el ratón salga de *principal* se establecerá como color de fondo *white*.
3. Cuando se pulse en *principal* se escribirá un mensaje con las coordenadas de la pulsación (respecto a *principal*, no a la pantalla). Para ello, utiliza la propiedad *innerHTML*.
4. Cuando se esté escribiendo en el *campo1*, se copiará el valor escrito en *mensaje* (utiliza el evento *keypress*). Para obtener el valor de *campo1* utiliza la propiedad *value*.
5. Cuando se pulse el *boton01*, se concatenará el contenido de los tres campos de texto utilizando como separador el caracter “@” y se mostrará el resultado en *mensaje*.

6. Cuando se pulse el *boton02*, se asignará a *mensaje* el color de fondo indicado en *campo2*.
7. Cuando se pulse el *boton03*, se debe mostrar en *mensaje* si los valores de *campo1* y *campo3* coinciden.
8. Cuando se pulse el *boton04*, se debe realizar la misma concatenación que con el *boton01* pero, en vez de mostrarlo en *mensaje*, se añadirá un nuevo párrafo a *principal* que contenga el mensaje.
9. Cuando se pulse el *boton05*, se repite el texto de *campo1* tantas veces como se indique en *campo3*. Al igual que en el caso anterior, el texto se añadirá como un nuevo párrafo a *principal*.
10. Cuando se pulse el *boton06*, se eliminan todos los párrafos contenidos en *principal*. Para ello, haz un bucle que borre consecutivamente el primer hijo *firstChild* hasta que esta propiedad sea null.
11. Cuando se pulse el *boton07*, se cambia la visibilidad de *principal*. Es decir, si está visible se oculta y viceversa. Para ello debes preguntar por la propiedad de estilo *visibility* cuyos valores pueden ser *visible* o *hidden*.
12. Cuando se pulse el *boton08*, se creará un contador de tiempo que muestre un mensaje, transcurridos tantos segundos como se indique en *campo3*. Para ello, utiliza la función *setTimeout* teniendo en cuenta que en esta función el tiempo se indica en milisegundos.