

RECUPERACIÓN Javascript. 14 de junio de 2024

Como sabes, muchas apps de bancos utilizan un teclado virtual para que un usuario introduzca su PIN sin que se le pueda leer las pulsaciones del teclado.

NOTA: todas las funciones pedidas deben estar en un fichero externo

1. Crea una función Javascript llamada *comprobarPIN(pinEscrito, pinEsperado)* que:
 - a. recibe dos parámetros de entrada (PIN escrito y PIN esperado). De momento no debemos preocuparnos de dónde llegan estos datos.
 - b. compruebe que los datos son válidos (PIN escrito es un número de 4 cifras).
 - c. compruebe que el número escrito coincide con el esperado
 - d. la función debe devolver un boolean que indique si el PIN era correcto.
 - e. La función debe estar en un fichero externo.
2. Crea una función llamada *anadirMensaje(mensaje, tipo)* que:
 - a. reciba dos parámetros de entrada:
 - i. el texto del mensaje a escribir en la página
 - ii. de qué tipo es (si es de error, o de éxito)
 - b. El div llamado “*aviso*” será el que contenga el texto mensaje y se mostrará bajo el teclado
 - c. Este div debe convertirse a un componente bootstrap del tipo *badge-pill*
 - d. El color de fondo del componente variará según el tipo de mensaje indicado. (error –rojo- o éxito –verde-)
 - e. Solo debe mostrarse un mensaje. Por lo que, si se llama a la función varias veces, el mensaje debe machacarse

Aquí tienes una captura de un mensaje de error:



3. Crea las funciones que se indican a continuación. Estas funciones deben estar en el mismo fichero externo y, **de momento, estarán vacías.**
 - a. Crea una función llamada *botonNuméricoPulsado()* que sea llamada cuando el ratón o cualquier otro tipo de puntero haya dejado de pulsarse (*pointerup*) sobre cualquiera de los botones de números.
 - b. Crea una función llamada *botonComprobarPulsado()* que sea llamada cuando se pulse en el botón “Comprobar”.

- c. Crea otra función llamada `botonLimpiarPulsado()` que sea llamada cuando se pulse en el botón amarillo “Limpiar datos”.

Ayuda 1: para ahorrar tiempo y código, valora hacer una búsqueda de todos los botones y recorrerlos con un array

4. Ahora vamos a completar la función `botonNuméricoPulsado()`. Esta función debe:
 - a. leer el número pulsado
 - b. añadirlo al final del número almacenado en la pantalla (es un campo tipo 'password')
 - c. si se han alcanzado los 4 dígitos:
 - i. no se añade el número
 - ii. se muestra un mensaje de error (fondo rojo), usando el método del ejercicio 2.
5. Ahora vamos a completar la función `botonComprobarPulsado()`. Esta función debe:
 - a. Leer el campo que está escrito en la pantalla (es un campo tipo “password”)
 - b. Llamar al método `comprobarPIN` creado en el ejercicio 1. Supón que el pin del usuario que hay que comprobar es 1234.
 - c. Si el resultado es correcto, se muestra bajo el teclado un mensaje de éxito (de color verde), usando el método del ejercicio 2.
 - d. Si el resultado es incorrecto, se muestra bajo el teclado un mensaje de error (de color rojo), usando el método del ejercicio 2.

Ayuda 2: recuerda que puedes usar `parseInt(String)` para convertir el texto leído en un campo de formulario en un número

6. Ahora vamos a completar la función `botonLimpiarPulsado()`. Esta función debe vaciar los campos del formulario, dejándolos en blanco y, si se ha mostrado algún mensaje de resultado, eliminarlo.

INSTRUCCIONES DE ENTREGA

- Debes escribir todo el código javascript en un fichero independiente llamado `apellidos_nombre.js`, que se guarde en una carpeta diferente a la carpeta donde está el HTML.
- No debes tocar nada del código HTML entregado, solo para incorporar el script

CRITERIOS DE EVALUACIÓN

Esta prueba evalúa los siguientes Criterios de Evaluación:

- C.E.10.2 Se han creado programas sencillos para insertarlos en un sitio web usando funciones Javascript básicas (10p).
 - Inserción del código javascript (1p)
 - Sintaxis básica (3p)
 - Uso de bucles y estructuras de control de flujo (ejercicios 1, 2, 3, 4 y 5) (3p)
 - Operaciones con números y strings (ejercicios 1, 2, 4 y 5) (3p)
- C.E.10.3 Se han creado funciones Javascript, vinculadas a eventos del navegador, que acceden al DOM para manipular el contenido de una página web desde Javascript. (12p)
 - Búsqueda de elementos en el DOM (ejercicios 2, 3 y 6) (3p)
 - Gestión de eventos (ejercicio 3) (3p)
 - Modificación de las clases de un elemento (ejercicios 2 y 6) (3p)
 - Acceso y modificación del contenido de un elemento (ejercicios 2, 4, 5 y 6) (3p)