

Simulacro Examen 1ª Evaluación Parte 1 DWECC

Ejercicio 1

Diseñar una página web denominada **ejercicio1.html** que muestre en un mensaje de alerta los **5 primeros** números enteros positivos denominados “**curiosos**”. Se considera un número “curioso” aquel que coincide con la suma de los cubos de sus dígitos. El código, obligatoriamente, debe contar con una función booleana (**esCurioso()**) que reciba como parámetro un número y devuelva verdadero o falso según si cumple o no el requisito.

Son ejemplos de números curiosos: 1, 153, ...

La función debe almacenarse en un archivo externo denominado “funcionesExamen.js”.

Ejercicio 2

Crear una página web denominada **ejercicio2.html** que presente un formulario con el aspecto que se muestra en la imagen. Los datos del formulario serán procesados por la página `cambia_cuenta.php` que se entrega (observad los nombres de los campos en el script php).

Configuración de usuario

Inicio de sesión

¿Cuándo quieres que Windows te vuelva a pedir que inicies sesión? Cuando el PC se activa tras haber estado en suspensión ▼

Contraseña

Nueva contraseña
Vuelve a escribir la contraseña
Indicio de contraseña

Privacidad

☐ Mostrar detalles de la cuenta en la pantalla de inicio de sesión

Las opciones de inicio de sesión son: “Cuando el PC se activa tras haber estado en suspensión” o “Nunca”.

Antes del envío, debe validarse en el cliente teniendo en cuenta lo siguiente:

- La contraseña deberá tener un mínimo de 6 caracteres, empezar por una letra mayúscula y tener al menos un dígito. Si no cumple esta condición se emitirá el mensaje “La contraseña debe empezar por una letra mayúscula, tener un mínimo de 6 caracteres y contener, al menos, un dígito”. El campo tomará el foco y se seleccionará su contenido.
- La segunda contraseña debe coincidir con la primera. En caso contrario se emitirá el mensaje “Las contraseñas escritas no coinciden. Vuelve a intentarlo”. Tomará el foco la primera contraseña y se seleccionará su contenido.
- La frase de indicio de contraseña nunca podrá contener la propia contraseña, por lo que en caso erróneo se mostrará el mensaje “El indicio de contraseña no puede contener la contraseña”. Tomará el foco y será seleccionado su contenido.

Ejercicio 3

El Cifrado de Caesar es uno de los métodos de codificación conocidos más antiguos. Es muy simple, sólo se cambian las posiciones del alfabeto. La transformación se denomina ROTN, donde N es el valor de cambio de posiciones y ROT significa "ROTAR", porque es un cambio cíclico.

Por ejemplo, ROT2 significa que "a" se vuelve "c", "b" se vuelve "d" y al final "y" se vuelve "a", y "z" se vuelve "b". La cantidad de transformaciones posibles depende de la longitud del alfabeto. Para castellano sólo llega a 26 (ROT0 o ROT27 no cambian el texto original). Debido a eso, es un método muy débil, ya que se pueden revisar todas las combinaciones posibles de manera muy rápida.

Escribir la página **ejercicio3.html** que permita **descifrar** un **texto cifrado** en un **nivel de rotación** determinado según los caracteres de un **alfabeto** por el método descrito anteriormente. La página deberá mostrar, por tanto:

- Un menú desplegable con los diferentes alfabetos. Para este ejercicio sólo se tendrán en cuenta castellano e inglés (aunque en un futuro puedan añadirse más alfabetos y debería ser sencilla la modificación del código). En inglés las letras van de la A a la Z, en castellano de la A a la Z incluyendo la Ñ en su posición correcta. El valor por defecto será Castellano.
- Un área de texto donde el usuario insertará el texto a descifrar.
- Un elemento tipo number para determinar el nivel de rotado (el valor mínimo será 0 y el máximo dependerá del alfabeto elegido).
- Un botón para proceder al descifrado.
- Una capa donde mostrar el resultado, que aparecerá en color de texto rojo.

Cualquier carácter del texto a descifrar que no pertenezca al alfabeto elegido no será descifrado, pero se mantendrá en el resultado.

NOTA: Para mayor facilidad, es este ejercicio, se supone que el mensaje está escrito en mayúsculas y sin tildes ni ningún otro tipo de acento tipográfico.

Podéis probar en castellano con ROT4 el texto **WM OIIW IWXI XIBXS GSVVIGXEPIQXI IQ V SX4 IW UYI LE WEOMHS TIVJIGXS, IQLSVEFYIQE !**, a ver qué pasa (está en el archivo cifrado_ejemplo.txt).

Ejemplo de pantalla inicial	Ejemplo (castellano) con nivel de rotación 6 del texto KY RAE LAIÑQ, YUQU NGE WAK VKSYGX AS VUWAÑZU. YAKXZK !!
<div><h3>Cifrado César</h3><p>- Descifrar -</p><p>Alfabeto Castellano ▾</p><p>Texto a descifrar <input type="text"/></p><p>Nivel de Rotado <input type="text"/></p><p><input type="button" value="Descifrar"/></p></div>	<div><h3>Cifrado César</h3><p>- Descifrar -</p><p>Alfabeto Castellano ▾</p><p>Texto a descifrar <input type="text" value="KY RAE LAIÑQ, YUQU NGE WAK VKSYGX AS VUWAÑZU. YAKXZK !!"/></p><p>Nivel de Rotado <input type="text" value="6"/></p><p><input type="button" value="Descifrar"/></p><p>ES MUY FÁCIL, SOLO HAY QUE PENSAR UN POQUITO. SUERTE !!</p></div>

Mejora Ejercicio 3

Crear una página web denominada **ejercicio2_mejorado.html** que contemple la posibilidad de que los caracteres del texto a cifrar estén escritos tanto en mayúsculas como minúsculas como con acentos tipográficos (á, Ü, é, ...). Para ello es obligatorio añadir un método al objeto String mediante la propiedad **prototype** para convertir los caracteres a mayúsculas simples (caracteres propios del alfabeto) antes de proceder al descifrado.