

Laboratorio 11: Javascript y DOM

Enunciado

Esta práctica consiste en la aplicación de los conocimientos expuestos en clase de teoría sobre JavaScript y DOM (Document Object Model). En estas prácticas se utilizará JavaScript para modificar la página HTML estática y para reaccionar a determinados eventos del usuario. La práctica está dividida en cuatro ejercicios.

0. Preparación

Usando una herramienta para administrar Oracle, como SQL Developer, cree un nuevo usuario llamado “ACCESO” y ejecute los scripts “Definición de Datos” e “Inserción de datos” que podrás encontrar en la carpeta SQL del material proporcionado para crear la nueva base de datos. Si ya creó la base de datos en la práctica anterior no hace falta volver a crearla ahora.

1. ¡ Hola Mundo !

Tomando de partida el archivo `hola.html`, introducir un pequeño código en JavaScript que escriba en el documento la frase “¡Hola Mundo, ya sé JavaScript!”.

Una vez desarrollado este ejercicio, ampliar el script para que muestre a continuación la hora actual (usar la clase predefinida `Date`).

2. Cálculo de la fortaleza de una contraseña

El objetivo de este ejercicio es implementar en el archivo **`validación_cliente_alta_usuario.js`** las funciones Javascript necesarias para calcular la fortaleza de la contraseña que el usuario introduzca en el formulario e informar al mismo. Para ello, nos planteamos los siguientes pasos:

- Implementar una función que cuente los caracteres repetidos en la contraseña, de forma que ésta sea tanto más fuerte cuantos menos caracteres repetidos tenga. Seguiremos la siguiente fórmula:

$$\text{Fortaleza} = \frac{\text{Número de caracteres repetidos}}{\text{Longitud de la contraseña}}$$

- Evaluar la contraseña conforme el usuario vaya escribiendo caracteres en el campo correspondiente. Para ello debemos capturar el evento producido por la pulsación de teclas en el campo de la contraseña y llamar a la función implementada en el apartado anterior.
- Informar al usuario del resultado del cálculo. Por ahora lo que haremos será sacar por pantalla el valor numérico de la fortaleza junto al campo de contraseña, utilizando el espacio creado por la etiqueta **`span`** que aparece en el formulario.

Recordar incluir el archivo **`validación_cliente_alta_usuario.js`** en la cabecera de **`form_alta_usuario.php`** mediante la etiqueta *script*.

3. Validación en cliente

En este ejercicio nos planteamos la validación en cliente de aquellos aspectos del formulario que NO ha sido posible realizar con las herramientas ofrecidas por HTML5. Como ejemplo de esto trabajaremos en la validación de los campos de contraseña de manera que validemos dos aspectos:

Laboratorio 11: Javascript y DOM

1. La contraseña debe contener al menos una letra mayúscula, una minúscula y un dígito. Para ello debemos implementar la función *passwordValidation*, que devolverá un mensaje de error en caso de que la contraseña no cumpla con los requisitos pedidos, o una cadena vacía en caso contrario.
2. El contenido del campo de confirmación de la contraseña debe coincidir con el contenido del campo de contraseña. Para lo cual desarrollaremos una función que compare el contenido de los campos del formulario correspondientes a la contraseña y la confirmación de la contraseña, devolviendo un mensaje de error si son distintos, y la cadena vacía en caso contrario.

Para ello se nos proporciona en la práctica, dentro de la carpeta **js**, el archivo **validación_cliente_alta_usuario.js**, donde debemos implementar ambas validaciones.

Adicionalmente, debemos implementar la función **validarFormulario**, que muestre al usuario los posibles mensajes de error provocados por la validación, utilizando la función **setCustomValidity** sobre los campos afectados. Además, esta función deberá devolver “cierto” si la validación es correcta y “falso” en caso contrario.

Por último, la función **validarFormulario** deberá ser invocada cuando el usuario pulse el botón de “enviar” del formulario (gestionando de forma conveniente el evento **submit** en el formulario).

4. Javascript + CSS

En este ejercicio nos planteamos informar al usuario sobre la fortaleza de su contraseña de forma algo más intuitiva, utilizando Javascript para cambiar el estilo CSS del campo de password según la fortaleza del mismo.

Se pide implementar una función en el archivo **validación_cliente_alta_usuario.js** que invoque a la función del ejercicio 2 para calcular la fortaleza de la contraseña introducida por el usuario y luego utilice los estilos definidos en el archivo **biblio.css** según éste valor, de forma que consideremos tres niveles de seguridad en la contraseña:

- 1) Contraseña débil: si NO cumple la validación ó la fortaleza es menor a 0.4
- 2) Contraseña media: si la fortaleza está entre 0.4 y 0.7
- 3) Contraseña fuerte: si la fortaleza es mayor que 0.7

Por último, tendremos que manejar el evento apropiado en **form_alta_usuario.php**, para que se invoque a esta función cuando resulte más conveniente.

NOTAS:

- Ejercicio 3: Para el punto 1) nos puede resultar útil la función *test*, que devuelve cierto si una cadena cumple con una expresión regular dada y falso en caso contrario, con el siguiente prototipo:

EXPREG.test(CADENA)

- Ejercicio 4: recordar que en Javascript existe el atributo *className*, que sirve para consultar/modificar la clase CSS del elemento correspondiente.