**Examen: Validación y Gestión de Eventos en un Formulario HTML**

**Instrucciones Generales:**

1. Utiliza únicamente el API nativo de JavaScript para resolver este examen.
2. No uses bibliotecas externas como jQuery o frameworks.
3. Debes crear un archivo JavaScript separado llamado formValidation.js y vincularlo al archivo HTML proporcionado.
4. Asegúrate de comentar tu código para explicar las funciones y validaciones que implementas.

### **Parte 1: Validación de Campos del Formulario**

Implementa las siguientes validaciones en un formulario utilizando **HTML, CSS y JavaScript**.

Además, debes utilizar las funciones **checkValidity()** y **setCustomValidity()** para gestionar la validación personalizada de los campos, asegurándote de proporcionar mensajes de error adecuados cuando la entrada no sea válida.

#### **Validaciones requeridas:**

1. **Nombre (text):**
   * Este campo es obligatorio y debe contener al menos 3 caracteres.
   * Usa setCustomValidity() para mostrar un mensaje si el campo no cumple con la validación.
2. **Correo Electrónico (email):**
   * Valida que el correo tenga un formato correcto (por ejemplo, usuario@dominio.com).
   * Aplica checkValidity() para verificar si el valor ingresado es válido.
3. **Teléfono (tel):**
   * Debe contener solo números y tener una longitud de 10 dígitos.
   * Aplica una expresión regular para la validación y usa setCustomValidity() en caso de error.
4. **Edad (number):**
   * Debe ser un número entre 18 y 100.
   * Implementa checkValidity() para asegurarte de que el número esté dentro del rango permitido.
5. **Género (select):**
   * Este campo es obligatorio, y el usuario debe elegir una opción que no sea la predeterminada.
   * Implementa setCustomValidity() para indicar que una opción válida debe ser seleccionada.
6. **Habilidades (checkbox):**
   * Asegúrate de que al menos haya una habilidad seleccionada.
   * Usa setCustomValidity() para evitar que el formulario se envíe si no hay casillas marcadas.
7. **¿Por qué quieres unirte? (textarea):**
   * Si el campo se completa, no debe superar los 200 caracteres.
   * Usa setCustomValidity() para notificar si se excede el límite de caracteres.
8. **Términos y condiciones (checkbox):**
   * Asegúrate de que el check esté marcado. SI no, no se permitirá el envío del formulario
   * Usa setCustomValidity() esta situación.

**Parte 2: Gestión de Eventos**

Implementa las siguientes funcionalidades relacionadas con eventos:

1. **Prevención del envío del formulario:**
   * Si algún campo no pasa la validación, evita que el formulario se envíe y muestra un mensaje de error al usuario.
2. **Eventos en tiempo real:**
   * Valida los siguientes campos mientras el usuario escribe o selecciona valores:
     + Nombre, Correo Electrónico y Edad.
   * Muestra mensajes de validación en tiempo real debajo de los campos.
3. **Manejo de errores globales:**
   * Si el formulario tiene varios errores, muestra un mensaje consolidado al final indicando todos los campos que necesitan correcciones.
4. **Restablecer formulario:**
   * Cuando el usuario haga clic en un botón "Restablecer", limpia todos los campos del formulario y elimina los mensajes de validación.
5. **Mostrar caracteres restantes en tiempo real:**
   * Para el campo "**¿Por qué quieres unirte?**", muestra debajo la cantidad de caracteres restantes mientras el usuario escribe. El límite es de 200 caracteres.

**Entregables:**

1. Archivo formValidation.js con la solución del examen.
2. Explicación breve (en comentarios del código) de cómo implementaste cada validación y evento.

**Instrucciones:** Resuelve los siguientes ejercicios utilizando el HTML del CV proporcionado. Usa JavaScript nativo para completar cada tarea.

### **Ejercicios Prácticos de Manejo del DOM con HTML de CV de Homer Simpson**

**Instrucciones:** Resuelve los siguientes ejercicios utilizando el HTML proporcionado. Usa JavaScript nativo para completar cada tarea.

1. **Selección por SRC:** Selecciona la imagen de perfil utilizando su atributo src y cambia su borde a un color azul #007bff. Debe ocurrir con el evento click sobre la imagen.
2. **Selección por Clase:** Selecciona todos los elementos de la lista en la sección "Habilidades Tecnológicas" y cambia su color de texto a rojo (#ff0000). Debe ocurrir con el evento mouseover sobre el título Habilidades Tecnológicas.
3. **Selección con querySelector:** Usa querySelector para seleccionar el nombre "Homer J. Simpson" y cambia su texto a "Homer Jay Simpson". Debe ocurrir con el evento dblclick sobre el título principal (h1).
4. **Selección con querySelectorAll:** Selecciona todos los elementos de la sección "Cursos de Formación" y agrega la clase highlighted-course a cada uno. Debe ocurrir con el evento click sobre el elemento Taller de Motivación Personal.
5. **Cambio de Contenido con textContent:** Selecciona el primer elemento de la lista en la sección de "Experiencia Laboral" y cambia su descripción a "Experto en Seguridad Nuclear y Prevención de Crisis". Debe ocurrir con el evento click sobre el elemento Técnico en Seguridad Nuclear.
6. **Cambio de Contenido con innerHTML:** Selecciona el último elemento de la lista en la sección de "Estudios" y reemplaza su contenido con:  
   <strong>Licenciatura en Ciencias Nucleares</strong> - Universidad de Springfield (2025).  
   Debe ocurrir con el evento click sobre el elemento Instituto Secundario de Springfield.
7. **Cambio de Atributos con setAttribute:** Selecciona el enlace de correo electrónico y cambia su contenido a "homersimpson@planta-nuclear.com". Debe ocurrir con el evento mouseover sobre el elemento Ocupación: Técnico en Seguridad Nuclear.
8. **Cambio de Estilos con style:** Selecciona el contenedor principal (.cv-container) y cambia su fondo a un color gris claro #f8f9fa. Debe ocurrir con el evento dblclick sobre el título Estudios.
9. **Agregar Elemento con appendChild:** Crea un nuevo elemento li dentro de la sección "Habilidades Tecnológicas" con el contenido:  
   <strong>Optimización del tiempo de descanso</strong> - Maestro en siestas estratégicas.
10. **Asociar Evento de Clic:** Agrega un evento de click al título principal (h1) que cambie su color de texto a verde (#008000).
11. **Detener la Propagación de Evento:** Agrega un evento de clic a cada elemento de la lista en "Experiencia Laboral" que registre un mensaje en la consola. Luego, asegúrate de que, al hacer clic en un elemento, el evento no se propague a la lista principal.

**Entregables:**

1. Archivo manejoDom.js con la solución del examen.
2. Explicación breve (en comentarios del código) de cómo implementaste cada validación y evento.

### **Rúbrica de Evaluación - Examen de Validación y Gestión de Eventos en un Formulario HTML**

### **Parte 1: Validación de Campos del Formulario (75 puntos)**

|  |  |  |
| --- | --- | --- |
| **Criterio** | **Descripción** | **Puntuación** |
| **Nombre** | Requiere al menos 3 caracteres y usa setCustomValidity() para mostrar mensajes personalizados. | 10 |
| **Correo Electrónico** | Verifica formato válido (ejemplo: [usuario@dominio.com](mailto:usuario@dominio.com)) y usa checkValidity(). | 10 |
| **Teléfono** | Solo acepta números, con una longitud de 10 dígitos, validado con expresión regular y setCustomValidity(). | 10 |
| **Edad** | Valor numérico entre 18 y 100, validado con checkValidity(). | 10 |
| **Género** | No debe permitir la opción predeterminada, con validación en setCustomValidity(). | 10 |
| **Habilidades** | Al menos una casilla debe estar seleccionada. Usa setCustomValidity(). | 10 |
| **¿Por qué quieres unirte?** | No debe superar los 200 caracteres. Muestra error con setCustomValidity(). | 10 |
| **Términos y condiciones** | Requiere ser marcado para enviar. Usa setCustomValidity(). | 5 |

### **Parte 2: Gestión de Eventos (50 puntos)**

|  |  |  |
| --- | --- | --- |
| **Criterio** | **Descripción** | **Puntuación** |
| **Prevención del envío del formulario** | Si algún campo es inválido, evita el envío y muestra mensajes de error. | 10 |
| **Validación en tiempo real** | Nombre, Correo Electrónico y Edad se validan mientras el usuario escribe. | 10 |
| **Mensajes de error globales** | Si hay múltiples errores, se muestra un mensaje consolidado al final del formulario. | 10 |
| **Restablecer formulario** | El botón "Restablecer" limpia todos los campos y elimina mensajes de validación. | 10 |
| **Mostrar caracteres restantes** | Se muestra en tiempo real el número de caracteres restantes en "¿Por qué quieres unirte?". | 10 |

### **Parte 3: Ejercicios de Manejo del DOM (25 puntos)**

|  |  |  |
| --- | --- | --- |
| **Criterio** | **Descripción** | **Puntuación** |
| **Manipulación del DOM** | Se ejecutan correctamente las modificaciones de atributos, estilos y contenido (setAttribute(), textContent). | 10 |
| **Uso correcto de eventos** | Implementa correctamente click, mouseover, dblclick, etc. | 10 |
| **Propagación y Delegación de Eventos** | Se usa stopPropagation() correctamente y se maneja delegación de eventos. | 5 |

**Puntuación Máxima:** 150 puntos