## ENTREGA: 07-01-21

## Al final de la teoría de formularios hay 2 ejercicios: el primero lo validáis con validación clásica, y el segundo aplicáis las etiquetas HTML5 para validación y customizáis los mensajes de error.

## Ejercicios

### **Ejercicio 1: Validación básica/clásica de un formulario mediante código javascript:**

Dado el siguiente código html:

|  |
| --- |
| <!DOCTYPE html> <html> <head>  <title>Formulario</title>  <script src="validar.js"></script>  <style>  .error {  border: solid 2px #FF0000;  }  </style> </head>  <body>  <h1>Formulario</h1>  <form action="procesar.php" method="post" id="miFormulario">  <table>  <tr>  <td>Nombre: </td>  <td>  <input type="text" name="nombre" id="nombre" />  </td>  </tr>  <tr>  <td>Telefono: </td>  <td>  <input type="text" name="telefono" id="telefono" />  </td>  </tr>  <tr>  <td>Fecha de nacimiento: </td>  <td>  <input type="text" name="dia" size="2" id="dia" />  <input type="text" name="mes" size="2" id="mes" />  <input type="text" name="ano" id="ano" size="4" />  </td>  </tr>  <tr>  <td>Sexo : </td>  <td>  <input type="radio" name="sexo" value="H" checked /> Hombre  <input type="radio" name="sexo" value="M" /> Mujer  </td>  </tr>  <tr>  <td>Mayor de 18:</td>  <td>  <input type="checkbox" name="mayor" id="mayor" checked/>  </td>  </tr>  </td>  </tr>  </table>  <p>  <input type="submit" value="Enviar" id="enviar" />  <input type="reset" value="Borrar" id="borrar" />  </p>  </form> </body>  </html> |

Debemos de realizar una validación antes del envío del formulario mediante código javascript que realice las siguientes operaciones:

* Una método **iniciar()** el cuál se llame cuando el DOM se haya formado y que sea la encargada de añadir un evento “onclick” al botón “enviar” del formulario y llame a un método que nos determine la validación global del formulario.
* Un método **validar()** el cuál se encargue de comprobar que todos los campos validan. Si todos los campos están bien, antes de dar el OK, mostrar un mensaje de confirmación para que el usuario decida si enviar o no el formulario por última vez. Si algún campo no valida, debe de eliminar la acción del botón “submit”.
* **Métodos de validaciones de campos:**
  + Comprobar que el campo “nombre” no sea vacío.
  + Comprobar que el campo “teléfono” sea un número (da igual el formato).
  + Comprobar que los campos “día”, ”mes” y “ano” formen una fecha válida.
  + Comprobar que el checkbox de mayor de 18 años esté chequeado o no.

### 

### **Ejercicio 2: Validación avanzada de un formulario mediante código javascript + etiquetas HTML5:**

Dado el siguiente código html:

|  |
| --- |
| <html> <head>  <title>VALIDACIÓN DE FORMULARIOS CON HTML5</title> </head> <body>  <h1>Formulario</h1>  <form action="procesar.php" method="post" id="miFormulario">  <table>  <tr>  <td>Nombre\*: </td>  <td>  <input type="text" name="nombre" id="nombre" required/>  </td>  </tr>  <tr>  <td>Edad\*: </td>  <td>  <input type="number" name="edad" id="edad" />  </td>  </tr>   <tr>  <td>Telefono\*: </td>  <td>  <input type="text" name="telefono" id="telefono" />  </td>  </tr>  </table>  <p>  <input type="submit" value="Enviar" id="enviar" />  <input type="reset" value="Borrar" id="borrar" />  </p>  <p id="mensajeError"></p>  </form> </body> </html> |

1. Añade las etiquetas HTML5 necesarias para que el campo de texto **“nombre”** tenga una longitud máxima de 15 caracteres, y sea capaz de validar una expresión regular que acepte solo letras (mayúsculas y/o minúsculas) y tenga una longitud de 2 a 15 letras. También, si el usuario sitúa el ratón encima del campo por unos segundos debe de aparecer el mensaje: “Introduce entre 2 y 15 letras”. Además el campo debe de ser “obligatorio” de cumplimentar.
2. Añade la etiqueta HTML5 necesaria para que el campo numérico **“edad”** tenga un valor numérico mínimo de 18 y máximo de 100. También de ser “obligatorio” de cumplimentar.
3. Añade la etiqueta HTML5 necesaria para que el campo de texto **“teléfono”** sea capaz de validar una expresión regular en la que solo se puedan utilizar números y de una longitud de 9 dígitos. Si el usuario sitúa el ratón por unos segundos sobre el campo teléfono, aparecerá el mensaje “Número de 9 cifras”. Además el campo debe de ser “obligatorio” de cumplimentar.
4. Mediante el uso del método **checkValidity**() (mira en los apuntes) y las distintas propiedades **“validity”** debemos de comprobar y capturar todos aquellos errores producidos por las validaciones HTML5 y personalizar los mensajes de error según el error producido. Los mensajes de error se mostrarán al final de la pantalla en la etiqueta con id **“**mensajeError**”.**
5. Vamos a añadir 2 nuevos métodos: **mostrarError( elemt , mensajeError ) y borrarError( elemt )**. Estos métodos reciben por parámetro un objeto al cual aplicarán o borrarán una clase de CSS que bordea el campo de rojo o lo elimina. La clase a aplicar es:

|  |
| --- |
| <style>  .error {  border: solid 2px #FF0000;  } </style> |
|  |

1. Investiga cómo añadirle al formulario un objeto de tipo “<input type="range"> y que tenga como valor inicial 50, el rango sea entre 1 y 100, y el paso (como cambian los valores) sea de 5 en 5.