

Técnico Superior en Desarrollo de Aplicaciones Web

Desarrollo Web en Entorno Cliente

Práctica: Validación básica de formularios y EXPREG

Lee bien todo lo que se indica en este enunciado y procura respetarlo.

Si tienes alguna duda o crees que hay alguna incoherencia, avisa al profesor para que te la solucione.

Normas de Realización

- a) El ejercicio se realizará en clase en el tiempo indicado para ello. No se va a dejar tiempo para acabarlo en casa.
- b) Puedes consultar tus apuntes pero NO fuentes de internet para realizar el ejercicio.
- c) No se pueden utilizar herramientas que generen código para completar la práctica. Si se detecta dicho uso, se considerará plagio.
- d) Se proporcionan los archivos necesarios para la realización de la practica: `index.html`, y `estilos.css`. Así como la estructura de directorios que se debe utilizar.
- e) **Está prohibido** añadir o quitar código de los archivos `.css`. En los archivos `.html` solo se puede añadir la etiqueta necesaria para enlazar archivos `.js` y otra cosa que se explicará en el enunciado más adelante.
- f) Sólo se permite usar las herramientas del lenguaje que se han dado hasta este momento. Por tanto, no se permiten uso de librerías externas ni de etiquetas de formulario de HTML5. (*Recuerda: yo evalúo competencias, no sólo conocimientos*).

El no cumplimiento de alguno de los apartados antes indicados provocará una penalización de un punto en la nota final (una penalización por cada apartado incumplido).

Copia o plagio

Tal y como se indica en los estatutos del centro, está totalmente prohibido copiar código de otro/a compañero/a y/o de internet, incluidas herramientas que generan código. Si se detecta y se demuestra este tipo de comportamiento, la calificación de la actividad será de 0 puntos.

Entrega

El ejercicio puntuará para la evaluación práctica del 2o Trimestre de la asignatura.

- La estructura de directorios proporcionada con todos los archivos necesarios para el funcionamiento correcto de la práctica deben meterse en una carpeta cuyo nombre será: **Apellido1_Apellido2_Nombre**. Ej:
Gomez_Luque_Rafael
- Al finalizar el ejercicio debes entregar un archivo comprimido en .zip .rar o .7z con la carpeta creada en el punto anterior. Sube ese archivo a la tarea indicada en el campus dentro de la asignatura.
- **Recuerda:** se puntúa lo que se entrega en la plataforma.

Enunciado

La práctica consiste en usar el formulario que se facilita con este enunciado para implementar las funcionalidades que se piden, crear una validación conforme se indican en el apartado reglas de validación y realizar una serie de acciones cuando el formulario esté correctamente validado.

1. Funcionalidades

- Lo que el usuario ponga en el campo *email*, debe copiarse automáticamente al campo *Copia Email*. Se puede usar una de estas dos formas:
 - Conforme el usuario va escribiendo en *email*, se va escribiendo en *Copia Email*.
 - Cuando el campo *email* pierda el foco, copia el contenido en el campo *Copia Email*.
- Cuando un usuario marque alguna afición, esta debe colocarse en el textarea que hay justo debajo (el campo *afiText*). Si el usuario la desmarca, debe desaparecer. Si hay varias aficiones marcadas, deben aparecer una debajo de otra.

2. Reglas de validación

El formulario estará correctamente validado **SI Y SOLO SI** se cumplen las siguientes reglas:

- Todos los campos de texto son obligatorios, esto es, no pueden quedar vacíos (*nombre, clave, repite clave, respuesta, edad y captcha*)
- La longitud mínima de la clave debe ser de 8 caracteres.
- El campo clave debe coincidir con el campo donde se repite la clave.
Si el usuario falla en la condición anterior, se debe borrar el contenido de ambos campos.
- El contenido del campo donde va la clave no puede coincidir con el contenido del campo usado para el nombre.
- El campo donde se indican la pregunta debe tener seleccionada alguna opción de su lista que no sea la primera.
- La longitud mínima del campo donde va la respuesta debe ser 6 caracteres.
- Al menos un botón radio debe estar marcado.
- El campo edad solo puede contener números y estar en un rango válido que va desde 3 hasta 99 años.

- El valor del campo de la palabra de seguridad debe coincidir con el código **QGPHJD** (todo en mayúsculas)
- Se deben aceptar las condiciones de uso.

IMPORTANTE:

- Dado que no está permitido modificar el archivo .html, no se pueden usar atributos de HTML5 para solucionar las condiciones indicadas. Hay que programar todas las acciones con JavaScript. *Recuerda: Tengo que valorar tus habilidades como programador usando Javascript.*
- Salvo para verificar si un campo está vacío, en este apartado, no se pueden usar expresiones regulares para ninguna de las condiciones que se indican. CUIDADO: No se admite que el campo solo tenga espacios en blanco.

3. Programación

La función de validación que se debe realizar puede hacerse de varias formas. La más sencilla (y por la que se dará la mínima puntuación) es que, tras pulsar el botón ENVIAR, se validen todos los campos y se indiquen los errores correspondientes.

Si el alumno quiere realizar la validación de otra forma, puede hacerlo (y se puntuará en consecuencia tal y como se indica al final del enunciado), siempre que sea más compleja de realizar que la forma básica indicada. *Consulta con el profesor si tienes dudas.*

4. Control de errores

En el caso de que **falle alguna de las reglas** antes comentadas, el formulario no será válido. Para reflejar dicho estado se debe usar CSS dinámico a través de Javascript.

Para ello al campo erróneo (o a los campos erróneos) se le pondrá el borde en rojo y con grosor 2px (menos a los checkbox y a los radios) y aparecerá un mensaje flotante indicando el error cuyo id debe ser *errores* (mensaje creado con etiquetas de forma dinámica, NO usar alert u otra ventana modal).

Para cualquiera que sea la opción elegida, tras ejecutarse el aviso de error se debe devolver el foco al campo que produce el error o al primer campo que produce el error (solo en el caso que sea posible).

5. Envío

Cuando **todas las reglas sean válidas**, el formulario se moverá a la página `resultado.html`,

Dicha página debes crearla tú: tan solo debes colocar un H1 con el texto: *Datos Enviados* y enlazarle la misma hoja de estilos que tiene el formulario para que se vea correctamente.

Completa el código del formulario para que funcione el envío de forma correcta usando el método GET.

6. Mejoras

En este apartado se debe mejorar la validación haciendo uso de Expresiones Regulares. Para ello debe usarse sentencias/funciones de Javascript, no está permitido colocarlas en el HTML.

Sin borrar lo que ya tienes programado del apartado 3 (comentalo y NO lo borres). Añade el código necesario para usar Expresiones regulares en:

- La longitud mínima de la clave debe ser de 8 caracteres.
- La longitud mínima del campo donde va la respuesta debe ser 6 caracteres.
- El campo edad solo puede contener números y estar en un rango válido que va desde 3 hasta 99 años.
- El contenido del campo Email debe tener un email válido: obligatorio que aparezca al menos un punto después de la arroba.

Cómo se califica:

- Funcionalidad 1: *hasta 0,75 punto*
- Funcionalidad 2: *hasta 1 punto*
- Realización de la función básica de validación : *hasta 3 puntos*.
- Mejora en la función de validación: *hasta 1 puntos más*.
- Control de errores: *hasta 2 puntos*.
- Crear web resultado: *0,5 puntos*
- Arreglar formulario para envío: *0,25 puntos*
- Dos primeras expresiones regulares: *0,25 puntos cada una*.
- Tercera expresión regular: *0,5 puntos*
- Expresión regular email: *0,5 puntos*