

Práctica 12. Formularios

Diseña una página web, en la que se muestre un formulario similar al de la siguiente imagen, teniendo en cuenta lo siguiente:

- Los campos **nombre** y **correo electrónico** son obligatorios.
- Utiliza el atributo *placeholder* para el **domicilio**.
- El campo **País de origen** será un select, agrupando por continentes los países, y poniendo por defecto España.
- La **fecha de nacimiento** será tipo date, el **correo** tipo email, y el **teléfono** tipo tel.
- Se pueden elegir varios **dispositivos electrónicos**.
- Para el campo de **pelí favorita**, utiliza un datalist, de forma que te permita elegir una de las propuestas o añadir una nueva

Rellena el siguiente cuestionario para formar parte del club de "Amigos de la informática"

Datos personales

Nombre:

Domicilio:

<calle, numero y localidad>

País de origen:

España

Fecha de nacimiento:

dd/mm/aaaa

Correo electrónico:

Teléfono:

Estado civil:

☐ Solter@ ☒ Casad@ ☐ Separad@ ☐ Viud@

Sexo:

☐ Hombre ☐ Mujer

Otra información de interés

Móvil

Tablet

Ordenador

Ebook

Seleccione los componentes electrónicos con los que habitualmente trabajas:

Seleccione tus actividades favoritas:

☐ Hacer deporte

☐ Salir con mis amigos

☐ Ver series, pelis,...

☐ Realizar rutas de senderiso

☐ Ir al cine

☐ Escuchar música

Explique brevemente el motivo por el que desea ingresar en nuestro club:

Elige o escribe tu peli favorita:

Enviar

Cancelar

Controles de HTML5 en formularios

Aprovecha para probar en este formulario, o en otro que te crees de pruebas, los nuevos controles que aporta HTML5. Debes probar al menos:

- El formulario se enviará a vuestra dirección de correo al pulsar el botón enviar.
- El etiquetado de la información de los controles, deberá hacerse usando la etiqueta *label* junto a su atributo *for*, en lugar de escribir el texto directamente.
- Prueba la etiqueta input en sus variantes de validación de datos, en concreto:
 - Input type="number".
 - Input type="email"
 - Input type="url"
 - Input type="date"
 - Input type="time"
 - Input type="datetime"
 - Input type="month"
 - Input type="search"
 - Input type="tel"
 - Input type="range", con saltos de 1 en 1 y de 3 en 3.
 - Input type="color"
 - Input type="image", funcionará como el botón "submit"
 - Prueba de la etiqueta datalist. Prueba también el truco que hay en los apuntes para cuando los navegadores no soportan esta etiqueta.
 - El atributo "required" en algún dato de entrada.
 - El atributo "multiple" en una entrada de texto.
 - El atributo "autocomplete" activado y desactivado.
 - El atributo "pattern", creando campos de entrada con expresiones regulares para validar casos como los siguientes:
 - Teléfono (9 dígitos comenzando por 9, 8, 7 o 6)
 - Referencia (3 letras mayúsculas seguidas de 4 números)
 - DNI (8 dígitos y una letra)
 - El atributo "placeholder".
 - Probar los atributos max, min, range, step, por ejemplo una entrada tipo número que empiece en 10 vaya hasta 100 y salte de 5 en 5.

La tabla siguiente contiene la lista completa de metacaracteres y su comportamiento en el contexto de las expresiones regulares:

Carácter	Descripción
\	Marca el siguiente carácter como un carácter especial, un literal, una referencia inversa o un valor de escape octal. Por ejemplo, 'n' coincide con el carácter "n". '\n' coincide con un carácter de nueva línea. La secuencia '\\' coincide con "\" y \"(\" coincide con "(".
^	Coincide con la posición al principio de la cadena de entrada. Si la propiedad Multiline del objeto RegExp está establecida, ^ coincide también con la siguiente posición a '\n' o '\r'.
\$	Coincide con la posición al final de la cadena de entrada. Si la propiedad Multiline del objeto RegExp está establecida, \$ coincide también con la posición anterior a '\n' o '\r'.
*	Coincide cero o más veces con el carácter o subexpresión anterior. Por ejemplo, zo* coincide con "z" y con "zoo". * equivale a {0,}.
+	Coincide una o más veces con el carácter o subexpresión anterior. Por ejemplo, 'zo+' coincide con "zo" y con "zoo", pero no coincide con "z". + equivale a {1,}.
?	Coincide una vez o ninguna con el carácter o subexpresión anterior. Por ejemplo, "da(do)?" coincide con "da" en "da" o en "dado". ? equivale a {0,1}
{n}	<i>n</i> es un entero no negativo. Coincide exactamente <i>n</i> veces. Por ejemplo, 'o{2}' no coincide con la 'o' de "doy", pero coincide con las dos oes de "cooperar".
{n,}	<i>n</i> es un entero no negativo. Coincide como mínimo <i>n</i> veces. Por ejemplo, 'o{2,}' no coincide con la "o" de "sol" pero coincide con todas las oes de "goooooo!". 'o{1,}' equivale a 'o+'. 'o{0,}' equivale a 'o*'.
{n,m}	<i>M</i> y <i>n</i> son enteros no negativos donde $n \leq m$. Coincide <i>n</i> veces como mínimo y <i>m</i> veces como máximo. Por ejemplo, "o{1,3}" coincide con las tres primeras oes en "goooooo!". 'o{0,1}' equivale a 'o?'. Tenga en cuenta que no se puede incluir un espacio entre la coma y los números.
?	Cuando este carácter va inmediatamente después de cualquier otro cuantificador (*, +, ?, {n}, {n,}, {n,m}), el modelo coincidente es no expansivo. Un modelo no expansivo coincide lo menos posible con la cadena buscada, mientras que el modelo expansivo predeterminado coincide tanto como sea posible con la cadena buscada. Por ejemplo, en la cadena "oooo", 'o+?' coincide con una sola "o", mientras que 'o+' coincide con todas las oes.
.	Coincide con cualquier carácter excepto con "\n". Para coincidir con cualquier carácter incluido '\n', utilice un modelo como '[\s\S]'.
(modelo)	Una subexpresión que coincide con el <i>modelo</i> y captura la coincidencia. La coincidencia capturada se puede recuperar de la colección Matches resultante mediante las propiedades \$0...\$9. Para que coincidan los paréntesis (), utilice \"(' o '\)\".
(?:modelo)	Una subexpresión que coincide con el <i>modelo</i> , pero que no captura la coincidencia, es decir, la coincidencia no se almacena para un posible uso posterior. Esto resulta útil cuando se combinan partes de un modelo con el carácter disyuntivo "or" (). Por ejemplo, 'industr(?:y ies)' es una expresión más económica que 'industry industries'.
(?=modelo)	Una subexpresión que realiza una búsqueda anticipada positiva, la cual coincide con la cadena en cualquier punto donde empiece una cadena que coincida con el <i>modelo</i> . Ésta es una coincidencia sin captura, es decir, la coincidencia no se captura para un posible uso posterior. Por ejemplo, 'Windows (?:=95 98 NT 2000)' coincide con "Windows" en "Windows 2000" pero no con "Windows" en "Windows 3.1". Las búsquedas anticipadas no consumen caracteres, es decir, después de que se produce una coincidencia, la búsqueda de la siguiente comienza inmediatamente después de la última coincidencia, no después de los caracteres que componen la cadena de búsqueda anticipada.

(?! <i>modelo</i>)	Una subexpresión que realiza una búsqueda anticipada negativa, la cual coincide con la cadena de búsqueda en cualquier punto donde empiece una cadena que no coincida con el <i>modelo</i> . Ésta es una coincidencia sin captura, es decir, la coincidencia no se captura para un posible uso posterior. Por ejemplo, 'Windows (?!95 98 NT 2000)' coincide con "Windows" en "Windows 3.1" pero no coincide con "Windows" en "Windows 2000". Las búsquedas anticipadas no consumen caracteres, es decir, después de que se produce una coincidencia, la búsqueda de la siguiente comienza inmediatamente después de la última coincidencia, no después de los caracteres que componen la cadena de búsqueda anticipada.
x y	Coincide con x o con y. Por ejemplo, 'z food' coincide con "z" o con "food". '(z f)ood' coincide con "zood" o con "food".
[xyz]	Un juego de caracteres. Coincide con cualquiera de los caracteres incluidos. Por ejemplo, '[abc]' coincide con 'a' en "plan".
[^xyz]	Un juego de caracteres negativo. Coincide con cualquier carácter no incluido. Por ejemplo, '[^abc]' coincide con la 'p' de "plan".
[a-z]	Un intervalo de caracteres. Coincide con cualquier carácter del intervalo especificado. Por ejemplo, '[a-z]' coincide con cualquier carácter alfabético en minúsculas dentro del intervalo de la 'a' a la 'z'.
[^a-z]	Caracteres de un intervalo negativo. Coincide con cualquier carácter que no esté incluido en el intervalo especificado. Por ejemplo, '[^a-z]' coincide con cualquier carácter que no esté incluido en el intervalo de la 'a' a la 'z'.
\b	Coincide con el límite de una palabra; es decir, la posición entre una palabra y un espacio. Por ejemplo, 'er\b' coincide con 'er' en "ceder", pero no coincide con 'er' en "verbo".
\B	Coincide con una posición que no es un límite de palabra. 'er\B' coincide con 'er' en "verb", pero no coincide con 'er' en "never".
\cx	Coincide con el carácter de control indicado por x. Por ejemplo, \cM equivale a Control-M o un carácter de retorno de carro. El valor de x debe estar incluido en el intervalo A-Z o a-z. En caso contrario, se asume que c es un carácter 'c' literal.
\d	Coincide con un dígito. Equivale a [0-9].
\D	Coincide con un carácter que no sea un dígito. Equivale a [^0-9].
\f	Coincide con un carácter de avance de página. Equivale a \x0c y \cL.
\n	Coincide con un carácter de nueva línea. Equivale a \x0a y \cJ.
\r	Coincide con un carácter de retorno de carro. Equivale a \x0d y \cM.
\s	Coincide con cualquier carácter de espacio en blanco incluidos espacio, tabulación, avance de página, etc. Equivale a [\f\n\r\t\v].
\S	Coincide con cualquier carácter que no sea un espacio en blanco. Equivale a [^\f\n\r\t\v].
\t	Coincide con un carácter de tabulación. Equivale a \x09 y \cI.
\v	Coincide con un carácter de tabulación vertical. Equivale a \x0b y \cK.
\w	Coincide con cualquier carácter de palabra incluido el subrayado. Equivale a '[A-Za-z0-9_]'.
\W	Coincide con cualquier carácter que no pertenezca a una palabra. Equivale a '[^A-Za-z0-9_]'.
\xn	Coincide con n, donde n es un valor de escape hexadecimal. Los valores de escape hexadecimales deben tener exactamente dos dígitos de longitud. Por ejemplo, '\x41' coincide con "A". '\x041'

	equivale a '\x04' & '"'. Admite la utilización de códigos ASCII en las expresiones regulares.
<code>\num</code>	Coincide con <i>num</i> , donde <i>num</i> es un entero positivo. Una referencia de vuelta a las coincidencias capturadas. Por ejemplo, '(.)\1' coincide con dos caracteres idénticos consecutivos.
<code>\n</code>	Identifica un valor de escape octal o una referencia inversa. Si <code>\n</code> va precedido por como mínimo <i>n</i> subexpresiones capturadas, <i>n</i> es una referencia inversa. En caso contrario, <i>n</i> es un valor de escape octal si <i>n</i> es un dígito octal (0-7).
<code>\nm</code>	Identifica un valor de escape octal o una referencia inversa. Si <code>\nm</code> va precedido por como mínimo <i>nm</i> subexpresiones capturadas, <i>nm</i> es una referencia inversa. Si <code>\nm</code> va precedido por <i>n</i> capturas como mínimo, <i>n</i> es una referencia inversa seguida de una <i>m</i> literal. Si no existe ninguna de las condiciones anteriores, <code>\nm</code> coincide con el valor de escape octal <i>nm</i> cuando <i>n</i> y <i>m</i> son dígitos octales (0-7).
<code>\nml</code>	Coincide con el valor de escape octal <i>nml</i> cuando <i>n</i> es un dígito octal (0-3) y <i>m</i> y <i>l</i> son dígitos octales (0-7).
<code>\un</code>	Coincide con <i>n</i> , donde <i>n</i> es un carácter Unicode expresado mediante cuatro dígitos hexadecimales. Por ejemplo, <code>\u00A9</code> coincide con el símbolo de copyright (©).