



## 4 . Interacción con el usuario

- Envío y validación -

# Envío y validación de formularios

## Validación de formularios

Debemos indicar, en la etiqueta del formulario, que realice la validación:

```
<form action="URL" method="post" name="formValidado" onsubmit="return validar()">  
    <!-- ... -->  
</form>
```

Podemos hacerlo con un `AddEventListener` (mejor):

```
document.getElementById("enviar").addEventListener("click", validar, false);
```



# Envío y validación de formularios

## Validación de formularios

**En primer lugar definimos una función “iniciar” que se va a cargar únicamente cuando el documento haya sido cargado completamente.**

```
window.onload = iniciar;  
function iniciar() {  
    document.getElementById("enviar").addEventListener("click", validar, false);  
}
```

**La función validar recibe el evento por defecto asociado al botón enviar. Validamos cada apartado llamando a las funciones correspondientes:**

```
function validar(eventopordefecto) {  
    if (  
        validarcampostexto() && validarMatricula() && validarProvincia() &&  
        confirm("¿Deseas enviar el formulario?")  
    )  
        return true;  
    else {  
        //Cancelamos el evento de envío por defecto asignado al botón  
        eventopordefecto.preventDefault();  
        return false; // Salimos de la función devolviendo false.  
    }  
}
```

# Envío y validación de formularios

## Validación de un campo como obligatorio

### Valida un campo de texto:

```
function validacion() {  
    valor = document.getElementById("campo").value  
;  
    if (valor == "" || valor.length == 0) {  
        alert("El campo no puede ser vacío");  
        return false;  
    }  
    return true;  
}
```



# Envío y validación de formularios

## Validación de un campo como obligatorio

**Valida todos los campos de texto del formulario:**

```
var formulario = objeto.form;
for (var i = 0; i < formulario.elements.length; i++) {
    if (formulario.elements[i].type == "text" &&
        formulario.elements[i].value == "") {
        alert(
            "El campo: " + formulario.elements[i].name + " no puede estar en blanco
        "
        );
        formulario.elements[i].className = "error";
        formulario.elements[i].focus();
        return false;
    }
}
```



# Envío y validación de formularios

## Validación de un campo de texto como numérico

### Ejemplo 1(teléfono):

```
function validaNum() {  
    valor = document.getElementById("telefono").value;  
    if (isNaN(valor)) {  
        alert("El campo tiene que ser numérico");  
        return false;  
    }  
    return true;  
}
```

### Ejemplo 2(edad):

```
if (formulario.elements[i].id == "edad") {  
    if (  
        isNaN(formulario.elements[i].value) ||  
        formulario.elements[i].value < 0 ||  
        formulario.elements[i].value > 105  
    ) {  
        alert(  
            "El campo: " +  
            formulario.elements[i].name +  
            " posee valores incorrectos o la edad <0 o >105"  
        );  
    }  
}
```



# Envío y validación de formularios

## Validación de un campo de fecha

```
function validaFecha() {  
    vardia = document.getElementById("dia").value;  
    varmes = document.getElementById("mes").value;  
    varano = document.getElementById("ano").value;  
    fecha = new Date(ano, mes, dia);  
    if (!isNaN(fecha)) {  
        return false;  
    }  
    return true;  
}
```



# Envío y validación de formularios

## Modificación de la apariencia de un campo erróneo

Podemos modificar la apariencia de un campo creando en css una clase error:

```
.error {  
  border: solid 2px #ff0000;  
}
```

Así, cuando un campo sea erróneo, le aplicamos dicha clase. Ejemplo:

```
if (  
  formulario.elements[i].type == "text" &&  
  formulario.elements[i].value == ""  
) {  
  alert(  
    "El campo: " + formulario.elements[i].name + " no puede estar en blanco"  
  );  
  formulario.elements[i].className = "error";  
  formulario.elements[i].focus();  
  return false;  
}
```





# Envío y validación de formularios

## Validación de un checkbox

### Ejemplo:

```
function validaCheck() {  
    elemento = document.getElementById("campoCondiciones");  
    if (!elemento.checked) {  
        return false;  
    }  
    return true;  
}
```



# Envío y validación de formularios

## EJERCICIO: U4T3

- A partir del formulario creado en la tarea U4T1, realiza las funciones necesarias para validarlo teniendo en cuenta:
  - Nombre del disco: 20 caracteres, obligatorio.
  - Grupo de música o cantante: 20 caracteres, obligatorio.
  - Año de publicación: 4 caracteres numéricos.
  - Tipo de música (podrá ser “rock”, “pop”, “punk” o “indie”): sin comprobación.
  - Localización: almacenará un número de estantería: vacío o numérico.
  - Prestado: sin comprobación.
- Tendrás que tener en cuenta, además:
  - Que utilizarás un método `window.onload`.
  - Que la validación se asignará al formulario mediante un `AddEventListener`.
  - Que los campos nombre del disco y grupo de música se validarán en la misma función (`campo20`).



# Envío y validación de formularios

## EJERCICIO: U4T4:

- Modifica las funciones de tal manera que, en caso de que se produzca un error en la validación, el campo implicado tenga el reborde rojo, y la etiqueta que lo acompaña también aparezca de color rojo.
- Para ello deberás crear las clases css necesarias (puedes incluirlas en el propio html si lo deseas) para que se visualice correctamente.
- Ten en cuenta que si el usuario mete correctamente el nombre, deberá volver a su color habitual.



# Expresiones Regulares

## Expresiones regulares

Se utiliza en datos que siguen un patrón concreto.

Ejemplos de expresiones regulares:

Código postal.

Teléfono (sobre  
todo con  
prefijos...)

DNI.

Tarjeta de  
extranjería.

Código de  
cuenta del  
banco.

Correo  
electrónico.

Identificadores  
de usuario.



# Expresiones Regulares

## Caracteres especiales

Car.	Descripción	Detalle
<b>^</b>	<b>Principio de entrada o línea.</b>	Este carácter indica que las cadenas deberán comenzar por el siguiente carácter. Si este fuera una “a” minúscula como indicamos en el punto anterior la expresión regular sería, <b>^a</b> .
<b>\$</b>	<b>Fin de entrada o línea.</b>	Indica que la cadena debe terminar por el elemento precedido al dólar.
<b>*</b>	<b>El carácter anterior 0 o más veces.</b>	<b>*</b> El carácter anterior 0 o más veces. El asterisco indica que el carácter anterior se puede repetir en la cadena 0 o más veces.
<b>+</b>	<b>El carácter anterior 1 o más veces.</b>	El símbolo más indica que el carácter anterior se puede repetir en la cadena una o más veces.
<b>?</b>	<b>El carácter anterior una vez como máximo.</b>	El símbolo interrogación indica que el carácter anterior se puede repetir en la cadena cero o una vez
<b>.</b>	<b>Cualquier carácter individual.</b>	El símbolo punto indica que puede haber cualquier carácter individual salvo el de salto de línea.



# Expresiones Regulares

## Caracteres especiales

Car.	Descripción	Detalle
$x y$	$X$ ó $y$	La barra vertical indica que puede ser el carácter $x$ o el $y$ .
$\{n\}$	$n$ veces el carácter anterior	El carácter anterior a las llaves tiene que aparecer exactamente $n$ veces.
$\{n,m\}$	Entre $n$ y $m$ veces el carácter anterior	El carácter anterior a las llaves tiene que aparecer como mínimo $n$ y como máximo $m$ veces.
$[abc]$	Cualquier carácter de los corchetes.	En la cadena puede aparecer cualquier carácter que este incluido en los corchetes.
$[\wedge abc]$	Un carácter que no esté en los corchetes	En la cadena pueden aparecer todos los caracteres que no estén incluidos en los corchetes.



# Expresiones Regulares

## Caracteres especiales

Car.	Descripción	Detalle
<code>\b</code>	Fin de palabra	Este símbolo indica que tiene que haber un fin de palabra o retorno de carro.
<code>\B</code>	No fin de palabra	El símbolo <code>\B</code> indica cualquiera que no sea un límite de palabra.
<code>\d</code>	Cualquier carácter dígito	Este símbolo indica que puede haber cualquier carácter numérico, de 0 a 9.
<code>\D</code>	Carácter que no es dígito	Este símbolo indica que puede haber cualquier carácter siempre que no sea numérico.
<code>\f</code>	Salto de página	Este símbolo indica que tiene que haber un salto de página
<code>\n</code>	Salto de línea	Este símbolo indica que tiene que haber un salto de línea.
<code>\s</code>	Cualquier espacio en blanco	Este símbolo indica que tiene que haber un carácter individual de espacio en blanco: espacios, tabulaciones, saltos de página o saltos de línea.
<code>\S</code>	Cualquier carácter que no sea blanco	Este símbolo indica que tiene que haber cualquier carácter individual que no sea un espacio en blanco.
<code>\t</code>	Cualquier tabulación	Este símbolo indica que tiene que haber cualquier tabulación.
<code>\w</code>	Cualquier alfanumérico	Este símbolo indica que puede haber cualquier carácter alfanumérico.
<code>\W</code>	Cualquier carácter no alfanumérico	Este símbolo indica que puede haber cualquier carácter que no sea alfanumérico.



# Expresiones Regulares

## Validación de un DNI

```
function validaDNI() {  
    valor = document.getElementById("dni").value;  
    varletras = [  
        "T", "R", "W", "A", "G", "M", "Y", "F", "P", "D", "X", "B",  
        "N", "J", "Z", "S", "Q", "V", "H", "L", "C", "K", "E", "T",  
    ];  
    if (!/^d{8}[A-Z]$/.test(valor)) {  
        return false;  
    }  
    if (valor.charAt(8) != letras[valor.substring(0, 8) % 23]) {  
        return false;  
    }  
    return true;  
}
```





# Expresiones Regulares

## Validación de un correo electrónico

```
function validaEmail() {  
    valor = document.getElementById("campo").value;  
    if (!/\w+([-+.']\w+)*@\w+([-.\]\w+)*\.\w+([-.\]\w+)/.test(valor)) {  
        return false;  
    }  
    return true;  
}
```



# Expresiones Regulares

## Validación de un teléfono

```
function validaTelefono() {  
    valor = document.getElementById("telefono").value;  
    if (!/^\\d{9}$/.test(valor)) {  
        return false;  
    }  
    return true;  
}
```



# Expresiones Regulares



## EJERCICIO - U4T5

Walter White y Jesse Pinkman tienen que controlar las bolsas de producto azul que cocinan para distribuir en Albuquerque. Para ello tienen que crear un formulario que almacene la información de cada bolsa y validarlo teniendo en cuenta lo siguiente:

- Fecha de creación: obligatorio y con formato dd/mm/aaaa.
- Cocinero: será un nombre en clave formado por dos letras en mayúscula, un símbolo y cuatro dígitos (ej. WW\$1234)
- Destinatario: estará formado por dos o tres letras mayúsculas correspondientes al estado, un guión bajo, el nombre de la ciudad en minúsculas, dos puntos, y el código de distrito de 4 dígitos (ej. NM\_albuquerque:1234).
- Gramos: será un número del 100 al 5000.
- Composición: estará formado por una cantidad en gramos seguida de dos conjuntos de una o dos letras seguidas o no de un número. (ej. 200gC3OH7)
- Número de cuenta de EEUU: ver diapositiva siguiente:



# Expresiones Regulares



## EJERCICIO - U4T5

Supongamos que un número de cuenta estadounidense tiene el siguiente formato:

- Dos letras: suponemos que el valor de cada letra es del 1 al 26 (no hay ñ ni ll).
- Dos dígitos: debe corresponderse con la suma de la primera letra y la segunda: en caso de que sea menor que 10 se pone el 0 delante.
- Un guión.
- Doce dígitos de cuenta.
- Un guión.
- Dos dígitos de control: los dos primeros deben ser la suma de los 6 primeros dígitos de la cuenta dividido entre 6 y extrayendo solamente su parte entera; y los dos últimos exactamente igual, pero con los 6 siguientes.
- Si el número está correcto se colocará en un campo de texto al lado del anterior, pero sin guiones: solamente los números y las letras.



# Validación en HTML5

Podemos validar directamente en el HTML. Por ejemplo, establecer patrones (**pattern**), y que sea obligatorio (**required**):

```
<input
  type="text"
  name="nombre"
  id="nombre"
  maxlength="15"
  pattern="[A-Za-z]{2,15}"
  title="Introduce entre 2 y 15 letras"
  required
/>
```

```
<input type="number" name="edad" id="edad" min="18" max="100" required />
```



# Validación en HTML5

## Más ejemplos:

```
<input type="text" name="telefono" id="telefono"
      pattern="[0-9]{9}"
      title="Numero de 9 cifras"
      required
/>
```

```
<input type="number" name="edad" id="edad"
      min="18"
      max="100"
      required
/>
```



# Validación en HTML5

Atributo	Valor	Puede actuar sobre...	Descripción
<b>minlength</b>	<u>número</u>	Campos de texto	Establece la longitud mínima del texto requerida.
<b>maxlength</b>	<u>número</u>	Campos de texto	No permite escribir textos superiores a <u>número</u> caracteres.
<b>min</b>	<u>número</u>	Campos numéricos	Establece el número mínimo permitido.
	<u>fecha</u>	Campos de fecha	Establece la fecha mínima permitida.
	<u>hora</u>	Campos de hora	Establece la hora mínima permitida.
<b>max</b>	<u>número</u>	Campos numéricos	Establece el número máximo permitido.
	<u>fecha</u>	Campos de fecha	Establece la fecha máxima permitida.
	<u>hora</u>	Campos de hora	Establece la hora máxima permitida.
<b>step</b>	<u>número</u>	Campos numéricos	Establece el salto de números permitido. Por defecto, 1.
	<u>fecha</u>	Campos de fecha	Establece el salto de días permitido. Por defecto, 1.
	<u>hora</u>	Campos de hora	Establece el salto de segundos permitido. Por defecto, 1.
<b>required</b>		Campos en general	Campo obligatorio. Se debe rellenar para enviar formulario.
<b>disabled</b>		Campos en general	Campo desactivado. No se puede modificar. No se envía.
<b>readonly</b>		Campos en general	Campo de sólo lectura. No se puede modificar. Se envía.



**END**



**prof.jduran@iesalixar.org**