



DESARROLLO WEB EN ENTORNO CLIENTE

· INTERACCIÓN CON EL USUARIO: EVENTOS Y
FORMULARIOS

5. INTERACCIÓN CON EL USUARIO: EVENTOS Y FORMULARIOS

5.2. UTILIZACIÓN DE FORMULARIOS DESDE CÓDIGO

Eventos DOM

DOM

Evento	Descripción
DOMSubtreeModified	cuando añadimos o eliminamos nodos en el subárbol de un elemento o documento.
DOMNodeInserted	cuando añadimos un nodo hijo a un nodo padre.
DOMNodeRemoved	cuando eliminamos un nodo que tiene nodo padre.
DOMNodeRemovedFromDocument	cuando eliminamos un nodo del documento.
DOMNodeInsertedIntoDocument	cuando añadimos un nodo al documento.

5. INTERACCIÓN CON EL USUARIO: EVENTOS Y FORMULARIOS

5.2. UTILIZACIÓN DE FORMULARIOS DESDE CÓDIGO

Elementos de un formulario:

- `<html>`
 - `<head>`
 - `<title>Mi formulario</title>`
 - `</head>`
 - `<body>`
 - `<h3>Formulario</h3>`
 - `<form action= "pagina.php" method="post">`
 - `//Elementos`
 - `</form>`
 - `</body>`
 - `</html>`
-
- **Action:** es la url de la página a la que redirige el formulario.
 - **Method:** puede ser POST o GET, dependiendo del enmascaramiento de los datos enviados.



5. INTERACCIÓN CON EL USUARIO: EVENTOS Y FORMULARIOS

5.2. UTILIZACIÓN DE FORMULARIOS DESDE CÓDIGO

Elementos de un formulario

Propiedad	Descripción
acceptCharset	Ajusta o devuelve el valor del atributo accept-charset en un formulario.
action	Ajusta o devuelve el valor del atributo Action en un formulario.
enctype	Ajusta o devuelve el valor del atributo enctype en un formulario.
length	Devuelve el número de elementos de un formulario.
method	Ajusta o devuelve el valor del atributo method en un formulario.
name	Ajusta o devuelve el valor del atributo name en un formulario.
target	Ajusta o devuelve el valor del atributo target en un formulario.

`<form>`

Método	Descripción
reset()	Resetea un formulario.
submit()	Envía un formulario.

`form.elements[]`

- Devuelve un array con todos los input de un formulario.



5. INTERACCIÓN CON EL USUARIO: EVENTOS Y FORMULARIOS

5.2. UTILIZACIÓN DE FORMULARIOS DESDE CÓDIGO

Elementos de un formulario

`<input>`

Propiedad	Descripción
Type	Puede ser, a su vez: <ul style="list-style-type: none">• Text (cuadro de texto).• Password (cuadro de contraseña).• Checkbox (casilla de verificación).• radio (opción de entre dos o más).• Submit (botón de envío del formulario).• Reset (botón de vaciado de campos).• File (botón para buscar ficheros).• Hidden (campo oculto para, el usuario no lo visualiza en el formulario)• Image (botón de imagen en el formulario).• Button (botón del formulario).
Name	Asigna un nombre al elemento. Es necesario para que el servidor pueda trabajar con él.
Value	Inicializa el valor del elemento.

5. INTERACCIÓN CON EL USUARIO: EVENTOS Y FORMULARIOS

5.2. UTILIZACIÓN DE FORMULARIOS DESDE CÓDIGO

Elementos de un formulario

Propiedad	Descripción
Size	Tamaño inicial. En campos text y password se refiere al número de caracteres.
Maxlength	Número máximo de caracteres que pueden contener text y password.
Checked	Para elementos checkbox y radio. Indica si el checkbox está marcado; o cuál de los botones de radio está pulsado.
Disabled	El elemento aparece deshabilitado. El dato no se envía al servidor.
Readonly	Bloquea el contenido de control: el valor del elemento no se puede modificar (solo lectura).
Src	Solo se utiliza para asignar la url de una imagen que se colocará en como botón.
Alt	Descripción del elemento, que se visualiza si nos ponemos con el ratón sobr él.
Required	Indica si el campo debe ser completado antes de enviar el formulario.

`<input>`

Método	Descripción
select()	Selecciona el contenido de un campo de texto.

5. INTERACCIÓN CON EL USUARIO: EVENTOS Y FORMULARIOS

5.2. UTILIZACIÓN DE FORMULARIOS DESDE CÓDIGO

Cuadro de texto. Ejemplo:

- `< input type="text" name="nombre"/>`

Cuadro de texto. Ejemplo:

- `< input type="password" name="contrasena" />`

Campo oculto. Ejemplo:

- `< input type="hidden" name="campoOculto" value="valor"/>`



5. INTERACCIÓN CON EL USUARIO: EVENTOS Y FORMULARIOS

2. UTILIZACIÓN DE FORMULARIOS DESDE CÓDIGO

Casilla de verificación. Ejemplo:

- `<p>Frutas</p>`
- `<input name="fresa" type="checkbox" value="fresa"/>Fresa
`
- `<input name="platano" type="checkbox" value="platano"/>Plátano
`
- `<input name="manzana" type="checkbox" value="manzana"/>Manzana
`

Radio button. Ejemplo:

- `<p>Frutas</p>`
- `<input name="fruta" type="radio" value="fresa"/>Fresa
`
- `<input name="fruta" type="radio" value="platano"/>Plátano
`
- `<input name="fruta" type="radio" value="manzana"/>Manzana
`



5. INTERACCIÓN CON EL USUARIO: EVENTOS Y FORMULARIOS

2. UTILIZACIÓN DE FORMULARIOS DESDE CÓDIGO

Botón de envío. Ejemplo:

- `<input name="enviar" type="submit" value="Enviar formulario"/>`

Botón de reset. Ejemplo:

- `<input name="enviar" type="reset" value="Borrar formulario"/>`

Botón de imagen. Ejemplo:

- `<input name="enviar" type="image" src="imagen.jpg"/>`

Botón. Ejemplo:

- `<input name="enviar" type="button" value="Botón de prueba"/>`

Selector de ficheros. Ejemplo:

- Fichero adjunto: `<input type="file" name="fichero" />`



5. INTERACCIÓN CON EL USUARIO: EVENTOS Y FORMULARIOS

5.2. UTILIZACIÓN DE FORMULARIOS DESDE CÓDIGO

Elementos de un formulario

`<label>`

Propiedad	Descripción
<code>for</code>	Indica a qué otro elemento del formulario está asociada.

Label. Ejemplo:

```
<label for =“nombre”>Nombre</label>
```

```
<input type=“text” name=“nombre” />
```

5. INTERACCIÓN CON EL USUARIO: EVENTOS Y FORMULARIOS

5.2. UTILIZACIÓN DE FORMULARIOS DESDE CÓDIGO

Elementos de un formulario

`<textarea>`

Propiedad	Descripción
Name	Asigna un nombre al elemento. Es necesario para que el servidor pueda trabajar con él.
Rows	Numero de líneas del textarea.
Cols	Ancho del textarea.
Maxlength	Número máximo de caracteres que pueden contener text y password.
Checked	Para elementos checkbox y radio. Indica si el checkbox está marcado; o cuál de los botones de radio está pulsado.
Disabled	El elemento aparece deshabilitado. El dato no se envía al servidor.
Readonly	Bloquea el contenido de control: el valor del elemento no se puede modificar (solo lectura).
Required	Indica si el campo debe ser completado antes de enviar el formulario.

Textarea. Ejemplo:

```
<textarea name="comentarios" rows="4" cols="30"
></textarea>
```

5. INTERACCIÓN CON EL USUARIO: EVENTOS Y FORMULARIOS

5.2. UTILIZACIÓN DE FORMULARIOS DESDE CÓDIGO

Elementos de un formulario

`<select>` y `<option>`

Propiedad	Descripción
Name	Asigna un nombre al elemento. Es necesario para que el servidor pueda trabajar con él.
Disabled	El elemento aparece deshabilitado. El dato no se envía al servidor.
Multiple	Permite marcar más de una fila visible. Requiere del atributo size.
Size	En caso de que múltiple esté marcado, muestra cuántas filas estarán visibles.
Required	Indica si el campo debe ser completado antes de enviar el formulario.

Dentro del elemento `<select>` se encuentra la etiqueta `<option>` con el campo `value` para almacenar el valor y en su interior lo que se visualizará.



5. INTERACCIÓN CON EL USUARIO: EVENTOS Y FORMULARIOS

5.2. UTILIZACIÓN DE FORMULARIOS DESDE CÓDIGO

Elementos de un formulario

`<fieldset>` y `<legend>`

Propiedad	Descripción
Name	(Fieldset) Asigna un nombre al elemento.
Align	(Legend) Permite elegir la alineación del texto del legend sobre el fieldset.

Fieldset se utiliza para agrupar elementos del formulario mediante una línea y permite asignar un título para esta agrupación gracias a la etiqueta `<legend>`

Fieldset. Ejemplo:

```
<form>
  <fieldset>
    <legend>Datos:</legend>
    Nombre: <input type="text" size="30"><br>
    Email: <input type="text" size="30"><br>
  </fieldset>
</form>
```



5. INTERACCIÓN CON EL USUARIO: EVENTOS Y FORMULARIOS

5.2. UTILIZACIÓN DE FORMULARIOS DESDE CÓDIGO

Desplegable. Ejemplo:

- `<p>Comida: </p>`
 - `<select name="comida">`
 - `<option value="pasta">Pasta</option>`
 - `<option value="carne">Carne</option>`
 - `<option value="pescado">Pescado</option>`
 - `<option value="postre">Postre</option>`
 - `</select>`

Desplegable con elección múltiple. Ejemplo:

- `<p>Comida: </p>`
 - `<select multiple size="3" name="comida[]">`
 - `<option value="pasta">Pasta</option>`
 - `<option value="carne">Carne</option>`
 - `<option value="pescado">Pescado</option>`
 - `<option value="postre">Postre</option>`
 - `</select>`



5. INTERACCIÓN CON EL USUARIO: EVENTOS Y FORMULARIOS

5.2. UTILIZACIÓN DE FORMULARIOS DESDE CÓDIGO

Ejemplo de formulario:

```
<html>
<head>
  <title>Formulario</title>
</head>
<body>
  <h1>Formulario</h1>
  <form action="procesar.php" method="post">
    <table>
      <tr><td>Nombre: </td><td><input type="text" name="nombre"/></td></tr>
      <tr><td>Password: </td><td><input type="password" name="psw"/></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" checked/></td></tr>
      <tr><td>Fichero: </td><td><input type="file" name="upfile" /></td></tr>
      <tr><td>Hidden: </td><td><input type="hidden" name="escondido" value="sorpresa" /></td></tr>
    </table>
  </form>
</body>
</html>
```

Formulario

Nombre:

Password:

Sexo : ☒ Hombre ☐ Mujer

Mayor de 18: ☒

Fichero: Examinar... No se ha seleccionado ningún archivo.

Hidden:

Comida:

Bebidas:

5. INTERACCIÓN CON EL USUARIO: EVENTOS Y FORMULARIOS

5.2. UTILIZACIÓN DE FORMULARIOS DESDE CÓDIGO

```
<tr><td>Comida: </td>
<td>
<select name="comida">
  <option value="pasta">Pasta</option>
  <option value="carne">Carne</option>
  <option value="pescado">Pescado</option>
  <option value="postre">Postre</option>
</select>
</td></tr>
<tr><td>Bebidas: </td>
<td>
<select multiple size="3" name="bebida[]">
  <option value="agua">Agua</option>
  <option value="vino">Vino</option>
  <option value="cerveza">Cerveza</option>
</select>
</td></tr>
</table>
<p>
  <input type="submit" value="Enviar"/>
  <input type="reset" value="Borrar"/>
</p>
</form>
</body>
</html>
```

Formulario

Nombre:

Password:

Sexo : ☒ Hombre ☐ Mujer

Mayor de 18: ☒

Fichero: No se ha seleccionado ningún archivo.

Hidden:

Comida:

Bebidas:

5. INTERACCIÓN CON EL USUARIO: EVENTOS Y FORMULARIOS

5.3. MODIFICACIÓN DE APARIENCIA Y COMPORTAMIENTO

Seleccionar un formulario

- Disponemos del siguiente formulario:
 - `<form id="miFormulario" name="miFormulario" action="siguiente.php"...>`

`getElementById()`

- `var formulario = document.getElementById("contactar");`

`getElementsByName()`

- `var formularios = document.getElementsByName("form");`
- `var primerform = formularios[0];`

O en una sola línea:

- `var primerform = document.getElementsByName("form")[0];`



5. INTERACCIÓN CON EL USUARIO: EVENTOS Y FORMULARIOS

5.3. MODIFICACIÓN DE APARIENCIA Y COMPORTAMIENTO

Seleccionar un formulario

- Disponemos del siguiente formulario:
 - `<form id="miFormulario" name="miFormulario" action="siguiente.php"...>`

`forms[0]`

- `var formularios = document.forms; //Devuelve un array de formularios`
- `Var primerform = formularios [0];` 0 en una sola línea:
- `Var formularios = document.forms[0];`

`forms["nombre"]`

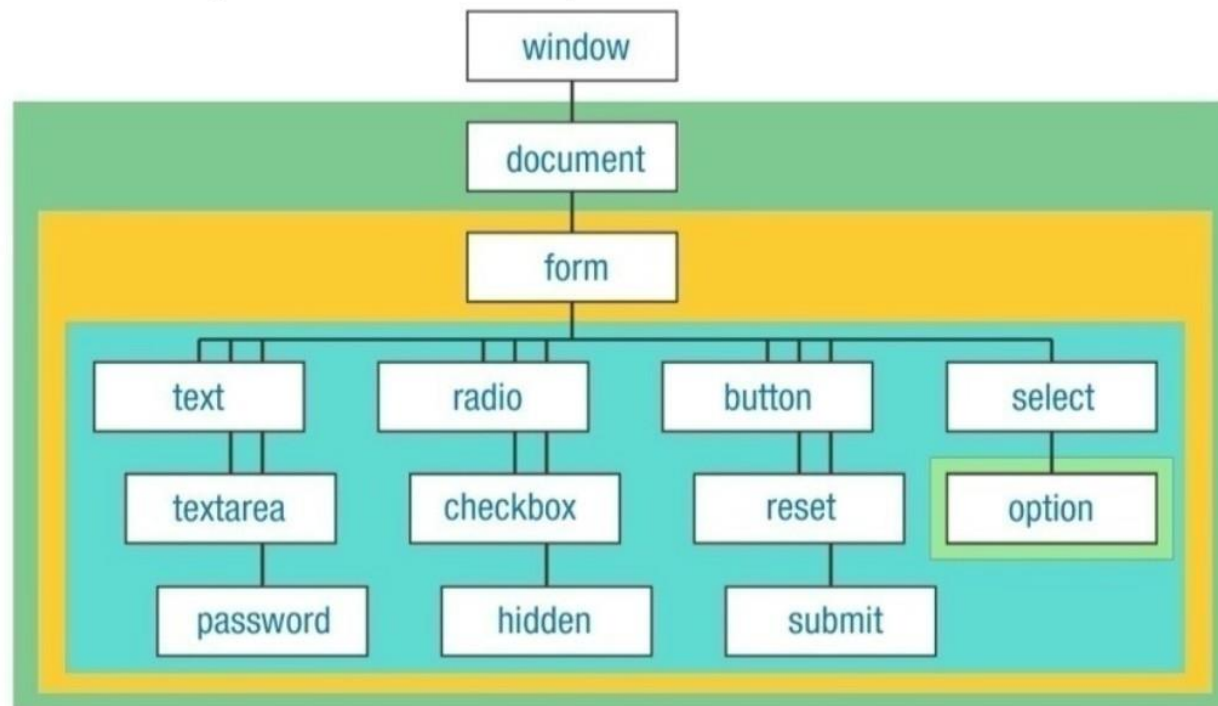
- `var formularios = document.forms; //Devuelve un array de formularios`
- `Var primerform = formularios ["contactar"];` 0 en una sola línea:
- `Var formularios = document.forms["contactar"];`



5. INTERACCIÓN CON EL USUARIO: EVENTOS Y FORMULARIOS

5.3. MODIFICACIÓN DE APARIENCIA Y COMPORTAMIENTO

El formulario como contenedor de objetos:



5. INTERACCIÓN CON EL USUARIO: EVENTOS Y FORMULARIOS

5.3. MODIFICACIÓN DE APARIENCIA Y COMPORTAMIENTO

Selección de objetos de un formulario

- Recuerda que...

`form.elements[]` devuelve un array con todos los input de un formulario.

`getElementById()` devuelve un elemento que tenga un id determinado.

`getElementsByTagName()` devuelve un array de elementos de un tipo.



5. INTERACCIÓN CON EL USUARIO: EVENTOS Y FORMULARIOS

5.3. MODIFICACIÓN DE APARIENCIA Y COMPORTAMIENTO

Selección de objetos de un formulario

- En este ejemplo:
- `<form id="formularioBusqueda" action="siguiente.php">`
 - `<p>`
 - `<input type="text" id="entrada" name="cEntrada">`
 - `<input type="submit" id="enviar" name="enviar" value="Buscar...">`
 - `</p>`
- `</form>`
- Para hacer referencia al input de tipo texto podemos hacerlo mediante:
 - `document.getElementById("entrada")`
 - `document.formularioBusqueda.cEntrada`
 - `document.formularioBusqueda.elements[0]`
 - `document.forms["formularioBusqueda"].elements["cEntrada"]`
 - `document.forms["formularioBusqueda"].cEntrada`



5. INTERACCIÓN CON EL USUARIO: EVENTOS Y FORMULARIOS

5.3. MODIFICACIÓN DE APARIENCIA Y COMPORTAMIENTO

Objetos input de tipo texto

- Es recomendable utilizar un id igual que el name para poder referirnos a ellos con `getElementById`.
- Podemos referirnos a cualquiera de sus propiedades para cambiarlas.
- Ejemplo:
 - `<input type="text" name="nombre" id="nombre"/>`
 - `document.getElementById("nombre").value="Valor nuevo"; //Cambia el valor del texto`

Objetos input de tipo checkbox

- La propiedad `value` es el texto asociado al objeto.
- El formulario solamente envía los datos si el checkbox está marcado.
- Ejemplo:
 - `<input type="checkbox" id="cantidad" name="cantidad" value="100"/>`
 - `document.getElementById("cantidad").checked=true; //Marca el checked`



5. INTERACCIÓN CON EL USUARIO: EVENTOS Y FORMULARIOS

5.3. MODIFICACIÓN DE APARIENCIA Y COMPORTAMIENTO

Objetos input de tipo radio

- Cuando utilizamos varios radiobutton con el mismo nombre, se crea un array de objetosinput radio.
- Podemos consultar el número de radio button con `formulario.gruporadio.length`;
- Podemos consultar si un radio button está marcado con `formulario.gruporadio[num].checked` y devolverá true o false.
- Ejemplo:
 - ```
for(var i=0; i<document.formulario.frutas.length;i++){
 if(document.formulario.frutas[i].checked)//Comprueba que es true
 alert(document.formulario.ac tores [i].val ue);
}
```

#### Objetos select

- La propiedad `selectedIndex` devuelve el índice de la opción que ha sido seleccionada.
- Cada opción tiene dos propiedades accesibles: `text` y `value` que indican el valor visible y el valor interno, respectivamente.
- Ejemplo:
  - `<option value="ZA">Zamora</option>`
  - `formulario.nombreCampoSelect.options[n].text;` //Devuelve Zamora
  - `formulario.nombreCampoSelect.options[n].value;` //Devuelve ZA



## 5. INTERACCIÓN CON EL USUARIO: EVENTOS Y FORMULARIOS

### 5.3. MODIFICACIÓN DE APARIENCIA Y COMPORTAMIENTO

#### Usando this

- Utilizamos this para hacer referencia al objeto que llama a un evento.
- Ejemplo:
  - `<input type="text" name="nombre" id="nombre" value="Infanta" onclick="identificar(this)"/>`
  - `function identificar(objeto){`
    - `var nombre=objeto.name;`
    - `var valor=objeto.value;`
    - `var id=objeto.id;`
    - `var tipo=objeto.type;`
    - `alert("Datos del objeto: "+nombre+ " - "+valor+ " - "+id+ " - "+tipo);`
  - `}`



## 5. INTERACCIÓN CON EL USUARIO: EVENTOS Y FORMULARIOS

### 5.3. MODIFICACIÓN DE APARIENCIA Y COMPORTAMIENTO

#### Eventos

- Los eventos se ejecutan cuando el usuario interactúa con la página (haciendo clic, pulsando una tecla, moviendo el ratón, etc.) o bien se tratan de eventos no relacionados con acciones del usuario (como onload, que sucede cuando la página se carga).
- Un poco de historia:

Incompatibilidades de navegadores: había que comprobar si los métodos eran soportados.

Modelo de registros de eventos en línea: se añade como atributo a la etiqueta html:  
`<a href="..." onClick=accion/>`

Modelo de registro de eventos tradicional: los eventos son una propiedad de los elementos:  
`elemento.onclick=accion;`

Modelo avanzado de registro de eventos del W3C: utiliza `addListener()`;

## 5. INTERACCIÓN CON EL USUARIO: EVENTOS Y FORMULARIOS

### 5.3. MODIFICACIÓN DE APARIENCIA Y COMPORTAMIENTO

#### Modelo de registro de eventos en línea

- El evento se añade como atributo de una etiqueta html.
- Se ejecuta primero el script y luego la acción por defecto.
- Ejemplos:
  - 1) `<a href="siguiente.html" onclick="alert('Has pulsado en el enlace')">Pulsa aquí</a>`
  - 2) `<a href="siguiente.html" onclick="alertar()">Pulsa aquí</a>`
    - `function alertar(){alert("Has pulsado el enlace");}`
- Para evitar que el navegador ejecute acción por defecto:
  - `<a href="siguiente.html" onclick="alertar(); return false;">Pulsa aquí</a>`  
De este modo no ejecutaría la acción por defecto: redirigir a la página siguiente.html.
- Para preguntar si ejecutar o no la acción por defecto:
  - `<a href="siguiente.html" onclick="return preguntar()">Pulsa aquí</a>`
  - `function preguntar(){  
 return confirm("¿Desea realmente ir a esa dirección?");  
}`

## 5. INTERACCIÓN CON EL USUARIO: EVENTOS Y FORMULARIOS

### 5.3. MODIFICACIÓN DE APARIENCIA Y COMPORTAMIENTO

¿Por qué no utilizar el modelo de registro de eventos en línea?

**No separa la programación de la estructura.**

Si la función redirige a una página no se ejecutan las demás tareas.

Al ejecutar la acción por defecto se pueden perder los objetos que hay en memoria (ej. Al cambiar de página).



## 5. INTERACCIÓN CON EL USUARIO: EVENTOS Y FORMULARIOS

### 5.3. MODIFICACIÓN DE APARIENCIA Y COMPORTAMIENTO

#### Modelo de registro de eventos tradicional

- El evento se convierte en una propiedad del elemento:
- Ejemplos:
  - `elemento.onclick = acción;`
  - `function accion() {...}`
- Para eliminar un gestor de eventos de un elemento:
  - `elemento.onclick=null;`
- Para llamar al gestor de eventos cuando queramos:
  - `elemento.onclick();`
- No usamos paréntesis, porque no necesitamos que devuelva el resultado.  
• ~~`elemento.onclick=accion();`~~ → `elemento.onclick=accion;`



## 5. INTERACCIÓN CON EL USUARIO: EVENTOS Y FORMULARIOS

### 5.3. MODIFICACIÓN DE APARIENCIA Y COMPORTAMIENTO



#### Modelo de registro de eventos avanzado del W3C

- Utiliza un evento `addEventListener()` con tres argumentos:
  - Tipo de evento.
  - Función a ejecutar.
  - Valor booleano para elegir la fase de captura (`true`) o burbujeo (`false`).
    - `elemento.addEventListener('evento', función, false | true)`
    - Ej:  
`document.getElementById("mienlace").addEventListener('click', alertar, false);`
- **¡¡Ojo!! No lleva “on” delante del nombre del evento.**



## 5. INTERACCIÓN CON EL USUARIO: EVENTOS Y FORMULARIOS

### 3. MODIFICACIÓN DE APARIENCIA Y COMPORTAMIENTO

#### Modelo de registro de eventos avanzado del W3C

- Eliminar un evento de un elemento:
  - `elemento.removeEventListener();`
- Crear más de un evento para el mismo elemento:
  - `document.getElementById("mienlace").addEventListener('click', alertar, false);`
  - `document.getElementById("mienlace").addEventListener('click', avisar, false);`
  - `document.getElementById("mienlace").addEventListener('click', chequear, false);`
- Crear un evento anónimo (no está asociado a ninguna función):
  - `element.addEventListener('click', function () {`
    - `this.style.backgroundColor = '#cc0000'; }, false)`



## 5. INTERACCIÓN CON EL USUARIO: EVENTOS Y FORMULARIOS

### 5.4. 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);`



## 5. INTERACCIÓN CON EL USUARIO: EVENTOS Y FORMULARIOS

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





## 5. INTERACCIÓN CON EL USUARIO: EVENTOS Y FORMULARIOS

### 5.4. 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;
}
```

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;
 }
}
```



## 5. INTERACCIÓN CON EL USUARIO: EVENTOS Y FORMULARIOS

### 5.4. 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");
 }
}
```



## 5. INTERACCIÓN CON EL USUARIO: EVENTOS Y FORMULARIOS

### 5.4. ENVÍO Y VALIDACIÓN DE FORMULARIOS

#### Validación de un campo de fecha:

Ejemplo:

```
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;
}
```



## 5. INTERACCIÓN CON EL USUARIO: EVENTOS Y FORMULARIOS

### 5.4. 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;
}
```



## 5. INTERACCIÓN CON EL USUARIO: EVENTOS Y FORMULARIOS

### 5.4. ENVÍO Y VALIDACIÓN DE FORMULARIOS

#### Validación de un checkbox:

Ejemplo:

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



## 5. INTERACCIÓN CON EL USUARIO: EVENTOS Y FORMULARIOS

### 5.5. EXPRESIONES REGULARES

#### Expresiones regulares:

- Se utiliza en datos que siguen un patrón concreto.
- Ejemplos de expresiones regulares:

Código postal.

Teléfono (sobre  
todo con  
prefijos...)

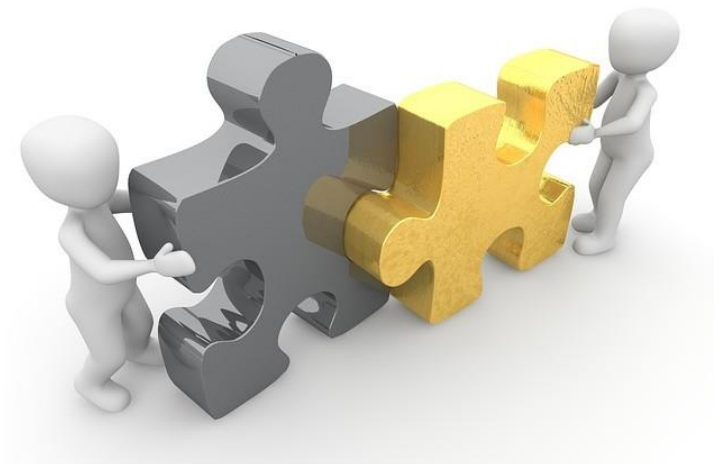
DNI.

Tarjeta de  
extranjería.

Código de  
cuenta del  
banco.

Correo  
electrónico.

Identificadores  
de usuario.



## 5. INTERACCIÓN CON EL USUARIO: EVENTOS Y FORMULARIOS

### 5.5. EXPRESIONES REGULARES

#### Caracteres especiales

| Car. | Descripción                               | Detalle                                                                                                                                                                                    |
|------|-------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ^    | Principio de entrada o línea.             | 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 sería, ^a. |
| \$   | Fin de entrada o línea.                   | Indica que la cadena debe terminar por el elemento precedido al dólar.                                                                                                                     |
| *    | El carácter anterior 0 o más veces.       | * El carácter anterior 0 o más veces. El asterisco indica que el carácter anterior se puede repetir en la cadena 0 o más veces.                                                            |
| +    | El carácter anterior 1 o más veces.       | El símbolo más indica que el carácter anterior se puede repetir en la cadena una o más veces.                                                                                              |
| ?    | El carácter anterior una vez como máximo. | El símbolo interrogación indica que el carácter anterior se puede repetir en la cadena cero o una vez                                                                                      |
| .    | Cualquier carácter individual.            | El símbolo punto indica que puede haber cualquier carácter individual salvo el de salto de línea.                                                                                          |



## 5. INTERACCIÓN CON EL USUARIO: EVENTOS Y FORMULARIOS

### 5.5. 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.         |
| [^abc] | Un carácter que no esté en los corchetes | En la cadena pueden aparecer todos los caracteres que no estén incluidos en los corchetes. |





## 5. INTERACCIÓN CON EL USUARIO: EVENTOS Y FORMULARIOS

### 5.5. EXPRESIONES REGULARES

#### Caracteres especiales

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

