

JavaScript Avançat: Formularis.

IES Son Ferrer

Formularis

Una de les raons per les que es va inventar JS va ser la necessitat de validar les dades dels formularis directament en el navegador de l'usuari per **evitar recarregar la pàgina** quan l'usuari cometia errors omplint-lo.

L'aparició de les aplicacions AJAX ha rellevat el tractament de formularis com la principal activitat de JS. Ara, el principal ús de JS és el de les comunicacions asíncrones amb els servidors i el de la manipulació dinàmica de les aplicacions (manipular el DOM).

Formularis

Quan es carrega una pàgina web, el navegador crea automàticament un array anomenat **forms** que conté la referència a tots els formularis de la pàgina:

```
document.forms[0];
```

Es poden accedir als elements dels formularis (quadres de text, botons, llistes desplegable, etc.). Per obtenir el primer element:

```
document.forms[0].elements[0];
```

Per obtenir el darrer element:

```
document.forms[0].elements[document.forms[0].elements.length-1];
```

Formularis

Accedir d'aquest forma té un problema: En les pàgines dinàmiques el contingut va canviant i, per tant, l'array de formularis. Una millor forma d'accedir als formularis és a través del seu nom (atribut **name**) o a través del seu atribut **id**:

```
let formulariPrincipal = document.formulari;  
let formulariSecundari = document.altre_formulari;
```

```
<form name="formulari" >
```

```
...
```

```
</form>
```

```
<form name="altre_formulari" >
```

```
...
```

```
</form>
```

Formularis

Òbviamment, també es pot accedir als formularis i als seus elements utilitzant les funcions DOM d'accés directe als nodes. El següent exemple utilitza l'habitual funció `document.getElementById()` per accedir de forma directa a un formulari i a un dels seus elements:

```
let formulariPrincipal = document.getElementById("formulari");  
let primerElement = document.getElementById("element");
```

```
<form name="formulari" id="formulari" >  
  <input type="text" name="element" id="element" />  
</form>
```

Formularis

Propietats útils dels elements dels formularis:

type: indica el tipus d'element del que es tracta. Per als elements de tipus `<input>` (`text`, `button`, `checkbox`, etc.) coincideix amb el valor del seu atribut **type**. Per a les llistes desplegable normals (element `<select>`) el seu valor és `select-one`, el que permet diferenciar-les de les llistes que permeten seleccionar diversos elements alhora i el tipus és `select-multiple`. Finalment, en els elements de tipus `<textarea>`, el valor de **type** és `textarea`.

form: és una referència directa al formulari al qual pertany l'element. Així, per accedir al formulari d'un element, es pot utilitzar `document.getElementById("id_del_element").form`

Formularis

Propietats útils dels elements dels formularis:

name: obté el valor de l'atribut **name** de HTML. És de només lectura (no es pot modificar).

value: permet llegir i modificar el valor de l'atribut **value** de HTML. Per als camps de text (`<input type="text">` i `<textarea>`) obté el text que ha escrit l'usuari. Per als botons obté el text que es mostra al botó. Per als elements *checkbox* i *radiobutton* no és molt útil, com es veurà més endavant.

Formularis

Esdeveniments més utilitzats amb els formularis:

onclick : esdeveniment que es produeix quan es fa clic amb el ratolí sobre un element. Normalment s'utilitza amb qualsevol dels tipus de botons que permet definir HTML (`<input type="button">`, `<input type="submit">`, `<input type="image">`).

onchange: esdeveniment que es produeix quan l'usuari canvia el valor d'un element de text (`<input type="text">` o `<textarea>`). També es produeix quan l'usuari selecciona una opció en una llista desplegable (`<select>`). Per a alguns elements, inclòs `<input type = "text">`, aquest esdeveniment de canvi no s'activa fins que el control perd el focus.

Formularis. Utilitats.

Obtenir els valors dels camps dels formularis:

text/texarea:

```
<input type="text" id="text" />
let valor = document.getElementById("text").value;
<textarea id="myTextarea">Text que es mostra</textarea>
let valor = document.getElementById("myTextarea").value;
```

radiobutton (només es pot seleccionar una opció):

```
<input type="radio" value="si" name="pregunta" id="pregunta_si"/> SI
<input type="radio" value="no" name="pregunta" id="pregunta_no"/> NO
<input type="radio" value="nsnc" name="pregunta" id="pregunta_nsnc"/> NS/NC
```

El següent codi permet determinar si cada *radiobutton* ha estat seleccionat o no:

```
let elements = document.getElementsByName("pregunta");
for (let i=0; i<elements.length; i++) {
    alert(" Element: " + elements[i].value + "\n Seleccionat: " + elements[i].checked);
}
```

Formularis. Utilitats.

Obtenir els valors dels camps dels formularis:

checkbox (es pot seleccionar més d'una opció):

`<input type="checkbox" value="condicions" name="condicions" id="condicions"/>` He llegit i accept les condicions

`<input type="checkbox" value="privacitat" name="privacitat" id="privacitat"/>` He llegit la política de privacitat

Utilitzant la propietat `checked`, és possible comprovar si cada *checkbox* ha estat seleccionat:

```
let element = document.getElementById ("condicions");  
alert("Element:" + element.value + "\n Seleccionat:" + element.checked);
```

```
element = document.getElementById ("privacitat");  
alert("Element:" + element.value + "\n Seleccionat:" + element.checked);
```

Formularis. Utilitats.

Obtenir els valors dels camps dels formularis:

select:

```
<select id="opcions" name="opcions">
  <option value="1">Primer valor</option>
  <option value="2">Segon valor</option>
  <option value="3">Tercer valor</option>
  <option value="4">Quart valor</option>
</select>
```

En general, el que es vol és obtenir el valor de l'atribut **value** de l'opció (**<option>**) seleccionada per l'usuari. Per obtenir el valor seleccionat es faran servir les següents propietats:

options, és un array creat automàticament pel navegador per a cada llista desplegable i que conté la referència a totes les opcions d'aquesta llista. D'aquesta manera, la primera opció d'una llista es pot obtenir mitjançant **document.getElementById("id_de_la_llista").options[0]**.

selectedIndex, quan l'usuari selecciona una opció, el navegador actualitza automàticament el valor d'aquesta propietat, que guarda l'índex de l'opció seleccionada. L'índex fa referència a l'array **options** creat automàticament pel navegador per a cada llista.

Formularis. Utilitats.

Obtenir els valors dels camps dels formularis:

select:

```
// Obtener la referencia a la lista
```

```
let llista = document.getElementById("opcions");
```

```
// Obtener la referència a la llista
```

```
let indexSeleccionat = llista.selectedIndex;
```

```
// Amb l'índex i la matriu "options", obtenir l'opció seleccionada
```

```
let opcioSeleccionada = llista.options[indexSeleccionat];
```

```
// Obtener el valor i el text de l'opció seleccionada
```

```
let textSeleccionat = opcioSeleccionada.text;
```

```
let valorSeleccionat = opcioSeleccionada.value;
```

```
alert("Opció seleccionada: " + textSeleccionat + "\n Valor de l'opció: " + valorSeleccionat);
```

Formularis. Utilitats.

Obtenir els valors dels camps dels formularis:

select:

Abreujades les instruccions de la diapositiva anterior:

```
let llista = document.getElementById("opcions");  
// Obtenir el valor de l'opció seleccionada  
let valorSeleccionat = llista.options[llista.selectedIndex].value;  
// Obtenir el text que mostra l'opció seleccionada  
let valorSeleccionat = llista.options[llista.selectedIndex].text;
```

Formularis. Utilitats.

Obtenir els valors dels camps dels formularis:

select:

És important és no confondre el valor de la propietat `selectedIndex` amb el valor corresponent a la propietat `value` de l'opció seleccionada. En l'exemple anterior, la primera opció té un `value` igual a `1` . No obstant això, si se selecciona aquesta opció, el valor de `selectedIndex` serà `0` , ja que és la primera opció de l'array `options` (i els arrays comencen a comptar els elements en el nombre `0`).

Formularis. Utilitats.

Resumint, un element `<select>` té 3 propietats importants:

- 1) `select.options`: la col·lecció de subelements `<option>`,
- 2) `select.value`: el valor de la `<option>` seleccionada actualment,
- 3) `select.selectedIndex`: el número de la `<option>` seleccionada actualment.

Aquestes proporcionen tres maneres diferents d'establir un valor per a un `<select>`:

- Cercar l'element `<option>` corresponent i assignar `true` a `option.selected`.
- Assignar a `select.value` el valor.
- Assignar a `select.selectedIndex` el número de l'opció.

Formularis. Utilitats.

```
<select id="opcions">
  <option value="1">Primer valor</option>
  <option value="2">Segon valor</option>
  <option value="3">Tercer valor</option>
  <option value="4">Quart valor</option>
</select>
```

```
<input type="submit" value="Enviar">
```

```
<script>
  let seleccio = document.getElementById('opcions');

  // Per seleccionar el tercer element ho podem fer de 3 maneres:

  // 1
  seleccio.options[2].selected = true;

  // 2
  seleccio.selectedIndex = 2;

  // 3
  seleccio.value = '3';

  // Per no tenir cap element seleccionat, lo millor és:
  seleccio.selectedIndex = -1;
</script>
```


Formularis. Utilitats.

Establir el focus en un element:

En programació, quan un **element està seleccionat i es pot escriure directament en ell o es pot modificar** alguna de les seves propietats, es diu que té el **focus** del programa.

En fer clic repetidament la tecla **TABULADOR** sobre una pàgina web, els diferents elements (enllaços, imatges, camps de formulari, etc.) van obtenint el focus del navegador.

Si en una pàgina web el formulari és l'element més important, p. e. el quadre de text d'un cercador, es considera **una bona pràctica d'usabilitat assignar automàticament el focus al primer element del formulari quan es carrega la pàgina.**

Formularis. Utilitats.

Establir el focus en un element:

Per assignar el focus a un element de HTML, s'utilitza la funció `focus()`.

```
<form id="formulari" action="#">  
    <input type="text" id="primer" />  
</form>
```

```
document.getElementById("primer").focus();
```

Formularis. Utilitats.

Establir el focus en un element:

Amb html5 també tenim la possibilitat d'usar **autofocus**, un atribut booleà que quan està present en un element, fa que aquest rebi el *focus* automàticament en carregar la pàgina.

```
<form>
```

```
  Nom: <input type="text" name="nom" autofocus><br>
```

```
  Cognoms: <input type="text" name="cognoms"><br>
```

```
  <input type="submit">
```

```
</form>
```

Formularis. Utilitats.

Establir el focus en un element:

Per assignar el focus al primer element del primer formulari de la pàgina:

```
if (document.forms.length > 0) {  
  for (let i=0; i < document.forms[0].elements.length; i++) {  
    let camp = document.forms[0].elements[i];  
    if (camp.type != "hidden") {  
      camp.focus();  
      break;  
    }  
  }  
}
```

Formularis. Utilitats.

Evitar la tramesa duplicada d'un formulari:

Un dels problemes habituals amb l'ús de formularis web és la possibilitat que l'usuari faci clic dues vegades seguides sobre el botó "Submit" (p. e. si la connexió de l'usuari és massa lenta o la resposta del servidor se fa esperar).

Per aquest motiu, una bona pràctica en el disseny d'aplicacions web sol ser la de **desactivar el botó d'enviament després del primer clic**:

```
<form id="formulari" action="#">  
  ...  
  <input type="button" value="Enviar"  
    onclick="this.disabled=true; this.value='Enviando...';  
    this.form.submit()" />  
</form>
```

Formularis. Utilitats.

Limitar el nombre de caràcters d'un textarea:

Fins a HTML5 textarea no tenia la propietat `maxlength` tal qual la té text. En HTML5 ha estat inclosa:

```
<input type="text" maxlength="10">  
<textarea maxlength="20"></textarea>
```

Fins a HTML5 ho havies de programar. És una bona pràctica per repassar conceptes.

Amb alguns esdeveniments (com `onkeypress` , `onclick` i `onsubmit`) es pot evitar el seu comportament normal si es retorna `false`. Si un esdeveniment torna el valor `true` , el seu comportament és l'habitual.

Si per exemple es torna el valor `false` a l'esdeveniment `onkeypress` , la tecla premuda per l'usuari no es té en compte. Si es retorna `false` en l'esdeveniment `onclick` d'un element com un enllaç, el navegador no carrega la pàgina indicada per l'enllaç.

Formularis. Utilitats.

Restringir els caràcters permesos en un quadre de text:

Pot ser útil de vegades definir que només es puguin entrar certs caràcters en un quadre de text.

Es pot fer un script que permeti/impedeixi el comportament habitual de l'esdeveniment `onkeypress`. Quan es fa clic en una tecla, es comprova si el caràcter d'aquesta tecla es troba dins dels caràcters permesos per a aquest element `<input>`.

Formularis. Utilitats.

Restringir els caràcters permesos en un quadre de text:

```
<label for="num">Només números:</label>
<input id="num" type="text" onkeydown="return permet(event, 'num')" />
<br><br>
<label for="num">Només lletres:</label>
<input id="car" type="text" onkeydown="return permet(event, 'car')" />
<br><br>
<label for="num-car">Només lletres:</label>
<input id="num-car" type="text" onkeydown="return permet(event, 'num-car')" />
```

```
<script>
function permet(ev, permesos) {
  // Variables que defineixen els caràcters permesos
  let numeros = "0123456789";
  let caracters = "abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ";
  let numeros_caracters = numeros + caracters;
  let tecles_especials = ["Backspace", "ArrowLeft", "ArrowRight", "Delete"];
  // 8 = BackSpace, 46 = Supr, 37 = fletxa esquerra, 39 = fletxa dreta
  // Seleccionar els caràcters a partir del paràmetre de la funció
  switch(permesos) {
    case 'num':
      permesos = numeros;
      break;
    case 'car':
      permesos = caracters;
      break;
    case 'num-car':
      permesos = numeros_caracters;
      break;
  }
  // Obténir la tecla premuda
  let caracter = ev.key;

  // Comprovar si la tecla premuda és alguna de las tecles especials
  // (tecles de borrar i fletxes horitzontals)
  let especial = false;
  for (let t in tecles_especials) {
    if (caracter == tecles_especials[t]) {
      especial = true;
      break;
    }
  }

  // Comprovar si la tecla premuda se troba entre els caràcters permesos
  // o si és una tecla especial
  return permesos.indexOf(caracter) != -1 || especial;
}
```


Formularis. Validació.

Una de les principals utilitats de JS és la validació de formularis. Normalment, la validació d'un formulari consisteix en cridar a una funció de validació quan l'usuari clica sobre el botó “submit”. En aquesta funció, es comproven si els valors que ha introduït l'usuari compleixen les restriccions imposades per l'aplicació. **Això es fa abans d'enviar el formulari al servidor.**

Notificar els errors de forma immediata mitjançant JS **millora l'experiència d'usuari** i ajuda a reduir la càrrega de processament al servidor.

Formularis. Validació.

Tot i que hi ha tantes possibles comprovacions com a elements de formulari diferents, algunes de les **comprovacions més habituals** són:

- que s'ompli un camp obligatori
- que se seleccioni el valor d'una llista desplegable
- que l'adreça d'email indicada sigui correcta
- que la data introduïda sigui lògica
- que s'hagi introduït un nombre on sigui requerit etc.

Formularis. Validació.

Codi JS bàsic necessari per incorporar la validació a un formulari:

```
<form action="" method="" id="" name="" onsubmit="return validacio()">  
  ...  
</form>
```

El funcionament d'aquesta tècnica de validació es basa en el comportament de l'esdeveniment `onsubmit` de JS. Igual que altres esdeveniments com `onclick` i `onkeypress` l'esdeveniment `onsubmit` varia el seu comportament en funció del valor que es retorna (`false` → no enviar el formulari, `true` → enviar el formulari).

Formularis. Validació.

Validar un camp de text obligatori:

```
valor = document.getElementById("camp").value;  
if (valor == null || valor.length == 0 || /^\\s+$/.test(valor) ) {  
    return false;  
}
```

La tercera part de la condició (`/^\\s+$/.test(valor)`) obliga que el valor introduït per l'usuari no només estigui format per espais en blanc.

Aquesta comprovació es basa en l'ús de "expressions regulars". Veurem per damunt el que són les expressions regulars en unitats posteriors. Una possible solució alternativa és fent servir el mètode `trim()` de string.

Formularis. Validació.

Validar un camp de text amb valors numèrics:

```
valor = document.getElementById("campo").value;  
if ( isNaN(valor) ) {  
    return false;  
}
```

Recordem exemples de `isNaN()`:

```
isNaN(3);           // false  
isNaN("3");         // false  
isNaN(3.3545);      // false  
isNaN(32323.345);   // false  
isNaN(+23.2);       // false  
isNaN("-23.2");     // false  
isNaN("23a");       // true  
isNaN("23.43.54");  // true
```

Formularis. Validació.

Validar que s'ha seleccionat una opció d'una llista:

```
index = document.getElementById("opcions").selectedIndex;  
if (index == null || index == 0) {  
    return false;  
}
```

```
<select id="opcions">  
  <option value="">- Selecciona un valor -</option>  
  <option value="1">Primer valor</option>  
  <option value="2">Segon valor</option>  
  <option value="3">Tercer valor</option>  
</select>
```

Formularis. Validació.

Validar una adreça de e-mail:

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

Aquesta regla valida la majoria d'emails vàlids d'usuari (però no tots al 100%).

Formularis. Validació.

Validar un número de DNI:

```
valor = document.getElementById("camp").value;

let lletres =
['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) !== lletres[(valor.substring(0,8)) % 23]) {
    return false;
}
```


Formularis. Validació.

Validar un número de telèfon:

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

Formularis. Validació.

Validar un número de telèfon:

L'expressió regular comprova si el valor indicat és una successió de nou nombres consecutius.

Nombre	Expressió regular	Format
900900900	/^\d{9}\$/	9 xifres seguides
900-900-900	/^\d{3}-\d{3}-\d{3}\$/	9 xifres agrupades de 3 en 3 i separades per guions
900 900900	/^\d{3}\s\d{6}\$/	9 xifres, les 3 primeres separades per un espai
900 90 09 00	/^\d{3}\s\d{2}\s\d{2}\s\d{2}\$/	9 xifres, les 3 primeres separades per un espai, les següents agrupades de 2 en 2
(900) 900900	/^\(\d{3}\)\s\d{6}\$/	9 xifres, les 3 primeres tancades per parèntesi i un espai de separació respecte de la resta
+34 900900900	/^\+\d{2,3}\s\d{9}\$/	Prefix internacional + seguit de 2 o 3 xifres), espai en blanc i 9 xifres consecutives

Formularis. Validació.

Validar que un checkbox ha estat seleccionat:

Si s'ha de seleccionar un checkbox de forma obligatòria:

```
element = document.getElementById("camp");  
if( !element.checked ) {  
    return false;  
}
```

Si es tracta de comprovar que tots els checkbox han estat seleccionats:

```
formulari = document.getElementById("formulari");  
for (let i=0; i<formulari.elements.length; i++) {  
    let element = formulario.elements[i];  
    if (element.type == "checkbox") {  
        if (!element.checked) {  
            return false;  
        }  
    }  
}
```

Formularis. Validació.

Validar que un radiobutton ha estat seleccionat:

Recordau que només se pot seleccionar un radiobutton d'un mateix grup.

```
opcions = document.getElementsByName("opcions");
let seleccionat = false;
for (let i=0; i<opcions.length; i++) {
    if (opcions[i].checked) {
        seleccionat = true;
        break;
    }
}

if (!seleccionat) {
    return false;
}
```

Formularis. Validació.

Validació de formularis nadiua amb HTML5:

```
<input title="Se necessita un nom" type="text" name="nom" required/>
```

Nom:



Email:

Se necessita un nom

Enviar

```
<input type="text" pattern="^[a-zA-Z0-9.!#$%&'*/+=?^_`{|}~-]+@[a-zA-Z0-9-]+(?:\.[a-zA-Z0-9-]+)*$" name="email" required/>
```

Nom: alice

Email: ddasdf

Feu que coincideixi amb el format sol·licitat.

Bibliografia

- http://www.w3schools.com/js/js_validation.asp
- http://www.w3schools.com/js/js_validation_api.asp
- <https://javascript.info/forms-controls>