

# Llenguatges de Marques i Sistemes de Gestió d'Informació

## UD 5.2 Validació de formularis



IES Jaume II El Just  
Tavernes de la Valldigna  
Departament d'Informàtica  
Curs 2024-25

## Índex

<b>1</b>	<b>Lectura de formularis</b>	<b>2</b>
1.1	inputs . . . . .	2
1.2	checkbox . . . . .	2
1.3	radio buttons . . . . .	2
1.4	select . . . . .	3
<b>2</b>	<b>Validació de formularis</b>	<b>4</b>
2.1	Validació bàsica . . . . .	4
2.2	Validació de tipus . . . . .	5
2.3	Validacions de rangs i longituds . . . . .	6
2.4	Validació avançada . . . . .	6

## 1 Lectura de formularis

### 1.1 inputs

Ja hem vist com llegir els *inputs* generals (text, password, email, etc.):

```
let myTextValue = document.getElementById("myText").value;
```

De la mateixa manera llegim els *textarea*.

Recordar que tots els inputs es lliguen com a string i que hem de fer la validació i transformació a valor numèric en cas d'estar treballant amb nombres:

```
let myNumberValue = document.getElementById("myNumber").value;
if (!isNaN(myNumberValue)) {
  myNumberValue = Number(myNumberValue);
}
document.getElementById("demo").innerHTML = typeof myNumberValue;
```

### 1.2 checkbox

Per a llegir si un **checkbox** està marcat, podem utilitzar la propietat **checked**:

```
document.getElementById("myCheck").checked;
```

Recordar que cada *checkbox* ha de tindre un name diferent i pel mateix motiu, hauria de tindre un id únic.

També podem canviar l'estat del checkbox amb la mateixa propietat:

```
document.getElementById("myCheck").checked = true;
```

### 1.3 radio buttons

Recordem que els *radio buttons* d'un mateix grup han de tindre el mateix name i value diferent:

```
<input type="radio" name="size" value="S" id="s">
<label for="s">S</label>

<input type="radio" name="size" value="M" id="m">
<label for="s">M</label>

<input type="radio" name="size" value="L" id="l">
<label for="m">L</label>
```

Per a llegir el valor seleccionat, hem de llegir tots els elements del grup amb `querySelectorAll` i buscar amb un bucle l'element seleccionat:

```
function myFunction() {
  let radioButtons = document.querySelectorAll('input[name="size"]');
  let selectedSize;
  for (const radioButton of radioButtons) {
    if (radioButton.checked) {
      selectedSize = radioButton.value;
      break;
    }
  }
  document.getElementById("demo").innerHTML = selectedSize;
}
```

## 1.4 select

Llegir el valor seleccionat d'un select d'opció única és senzill: seleccionem el select pel seu id i llegim `value`:

```
<select name="sizes" id="sizes">
  <option value="s">S</option>
  <option value="m">M</option>
  <option value="l">L</option>
</select>

function myFunction() {
  selectedSize = document.querySelector('#sizes').value;
  document.getElementById("demo").innerHTML = selectedSize;
}
```

Si el select és d'**opció múltiple**, hem d'iterar amb un bucle buscant tots els elements seleccionats:

```
<select multiple name="foods" id="foods">
  <option value="paella">Paella</option>
  <option value="fideua">Fideuà</option>
  <option value="arrosbanda">Arròs a banda</option>
</select>

function selectFoods() {
  select = document.querySelector('#foods');
  selectedOptions = [];
  for (const option of select.options) {
    if (option.selected) {
```

```
        selectedOptions.push(option.value);
    }
    document.getElementById("demo2").innerHTML = selectedOptions;
}
}
```

## 2 Validació de formularis

En aquest apartat veurem la validació de les dades introduïdes en formularis HTML. Cal recordar que l'estàndard **HTML5** ja proporciona una validació bàsica per a alguns tipus de inputs (`type="email"`, `type="number"`, etc) així com altres restriccions (`maxlength`, `min`, `max`, `required`, etc). Però en aquests apunts farem tota la validació amb codi JavaScript, per tant el tipus dels inputs serà `type="text"` i no usarem cap altra restricció.

### 2.1 Validació bàsica

El primer pas serà validar els camps requerits. Això es pot detectar si el valor introduït és igual a la cadena buida (`""`).

Per exemple, si tenim el següent formulari:

```
<form action="#" onsubmit="">
  <label for="nom"><strong>Nom</strong></label>
  <input type="text" name="nom" id="nom">

  <button type="button" onclick="validar()">Enviar</button>
</form>
```

La funció `validar()` validaria que el input **nom** no és buit:

```
function validar() {
    let nom = document.getElementById("nom").value;
    if (nom == "") {
        alert("El nom no pot estar buit");
    }
}
```

L'ús de `alert` no és recomanat per a mostrar informació a l'usuari. Al seu lloc és millor fer-ho mostrant el text de l'error al costat, dalt o baix de l'input. Per a això podem usar un element de text que inicialment estarà buit i on podrem mostrar el text de l'error:

```
<form action="#" onsubmit="">
  <label for="nom"><strong>Nom</strong></label>
  <input type="text" name="nom" id="nom">
  <span class="error" id="nomErr"></span><br>

  <button type="button" onclick="validar()">Enviar</button>
</form>

<script>
function validar() {
  let nom = document.getElementById("nom").value;
  if (nom === "") {
    document.getElementById('nomErr').innerHTML = 'Camp requerit';
  }
}
</script>
```

En aquest exemple estariem mostrant l'error dins del span amb id “nomErr”. La classe “error” ens servirà per a assignar un estil CSS a tots els missatges d'error.

Una funció molt util és **trim()**, que s'utilitza per a eliminar els espais a l'inici i al final de una cadena. Així, si un usuari escriu només espais en blanc en un camp d'un formulari, els podem eliminar:

```
let camp = '   '; // Només espais
console.log(camp === ''); // false
camp = camp.trim();
console.log(camp === ''); // true
```

## 2.2 Validació de tipus

Podem utilitzar la funció **isNaN()** per a saber si un valor conté un número vàlid (recordem que tots els valors dels formularis es llegeixen com a strings). Aquesta funció converteix el valor a nombre i després comprova si el resultat és **NaN** (Not a number):

```
console.log(isNaN('Hola')); // true
console.log(isNaN('5')); // false
console.log(isNaN('5.4')); // false
console.log(isNaN('5,4')); // true, la conversió falla per la coma
console.log(isNaN('')); // false!!, el valor de la conversió és 0
```

## 2.3 Validacions de rangs i longituds

Podem validar que la cadena llegida tinga una longitud mínima o màxima usant la propietat **length** dels strings:

```
console.log('hola'.length); // 4
console.log('').length; // 0
console.log(' hola '.length); // 8, compte amb els espais en blanc!
console.log(' hola '.trim().length); // 4
```

Com podem veure, és convenient usar **trim()** per a eliminar espais a l'inici i al final.

En el cas de **valors numèrics**, podem comprovar fàcilment que estiguen entre un valor mínim i màxim:

```
if (Number(valor) < 0 || Number(valor) > 15) {
  console.log("El valor ha d'estar entre 0 i 15");
} else {
  console.log("Valor correcte");
}
```

Recordem que és convenient fer abans la comprovació numèrica i convertir els strings a valors numèrics.

En el cas de les **dates**, podem comparar-les directament si tenim en compte que tenen el format **yyyy-mm-dd** i que són strings:

```
dataActual = '2023-04-19';
dataLlegida = '2021-01-01';

if (dataLlegida === dataActual)
  console.log('La data llegida correspon a hui');

if (dataLlegida < dataActual)
  console.log('La data llegida és anterior a la data actual');

if (dataLlegida > dataActual)
  console.log('La data llegida és posterior a la data actual');
```

## 2.4 Validació avançada

Podem fer una validació més acurada utilitzant expressions regulars i els mètodes **test()** o **match()**. Per exemple, **test()** retorna true si l'expressió regular es troba a la cadena passada com a argument:

`expressioRegular.test(cadena)`

Per exemple, tenim l'expressió regular `/^\d*$/` que correspon amb una cadena que té només dígit, sense altres caràcters i una cadena llegida en la variable *valor*:

```
if ( /^\d*$/.test(valor) ) {  
    console.log('El valor llegit és un número');  
} else {  
    console.log('El valor llegit NO és un número');  
}
```