



# Javascript



Impariamo Javascript

Giulio Angiani  
I.I.S. "Blaise Pascal" - Reggio Emilia



# Client-side validation form



# Il controllo dell'input

Javascript è spesso usato per controllare la correttezza **formale** dei dati inviati da utenti su moduli web

L'invio dei dati via POST è **subordinato** al passaggio di vari controlli

Per esempio:

- Completezza dei dati richiesti
  - i campi obbligatori devono essere compilati
- Correttezza formale dei dati
  - un campo email deve avere almeno una @ ed un punto
  - un campo codice fiscale deve essere di 16 caratteri
- Controllo di accettazione di clausole
  - l'utente deve accettare le regole di gestione dati o privacy

Username

Username is required

newuser@gmail.com

.....

.....

Register

# Primo esempio

Come fare per controllare che nella form i dati siano corretti?

**Spostare** l'invio dalla **form** ad una **funzione javascript**

Sia dato il codice:

```
<form class='mt-3' method=POST action='check.php'>  
  <input type='text' class='rounded w-75 p-2 m-3' name='usr' placeholder='username'>  
  <input type='password' class='rounded w-75 p-2 m-3' name='pwd' placeholder='password'>  
  <br>  
  <input type='submit' class='btn-success rounded ' value='Log In'>  
</form>
```

HTML

- il bottone **invia** diventa un **button**
- inserisco un **id** nella form
- sul click invochiamo la funzione di controllo **checkData()**

```
<form id='frm' class='mt-3' method=POST action='check.php'>  
  <input type='text' class='rounded w-75 p-2 m-3' name='usr' placeholder='username'>  
  <input type='password' class='rounded w-75 p-2 m-3' name='pwd' placeholder='password'>  
  <br>  
  <input type='button' class='btn-success rounded ' value='Log In' onClick="checkData()">  
</form>
```

HTML



# Client-side validation form

La funzione **checkData** è deputata ai controlli e all'eventuale submit del form

```
<script>
  function checkData() {
    // qui inserisco i controlli da fare

    // se tutti i controlli vanno a buon fine forzo il post via javascript
    document.getElementById('frm').submit();
  }
</script>
```

JS

dove:

- **document.getElementById('frm')** selezione la form nel documento
- il metodo **submit()** dell'oggetto selezionato permette l'invio dei dati



# Client-side validation form

La funzione **checkData** controlla che siano valorizzati sia il campo **usr** che il campo **pwd**

JS

```
<script>
function checkData() {
    // qui inserisco i controlli da fare
    // se non ho id posso selezionare per nome...
    // ma potrebbero esserci molti elementi con lo stesso nome, quindi devo
    // scegliere quale usare...
    // in questo caso il primo elemento con quel nome...
    var campouser = document.getElementsByName('usr')[0];
    if (campouser.value == '') {
        alert("Il campo username non può essere vuoto");
        campouser.focus(); // sposto il cursore nel campo da compilare
        campouser.style.borderColor = 'red';
        campouser.style.border = '1px solid';
        return false; // esco qui dalla funzione per non arrivare all'istruzione di submit
    }

    // se tutti i controlli vanno a buon fine forzo il post via javascript
    document.getElementById('frm').submit();
}
</script>
```



# Client-side validation form

Esercizio:

Modificare il file csvf01.html per controllare anche il campo **password**

5 minuti...



# Client-side validation form

Altro campo di applicazione è la **registrazione di un utente** su un sito web

Si consideri una form come quella in figura:

## Registrazione utente

nome\*

cognome\*

email\*

genere\*

☐ uomo ☐ donna

fascia di età

--- seleziona --- ▼

privacy\*

☐

Registrati



# Client-side validation form

La validazione non è la stessa per i diversi tipi di campi delle form

- per i campi **input** e **select** si considera l'attributo **value**
- per i campi **checkbox** si considera l'attributo **checked**
- per il campo **email** si considera l'attributo **validity.valid**
- per i campi **radio** conviene usare l'accesso via **form**



# Client-side validation form

a questo codice HTML

HTML

```
<input type='text' class='rounded w-100 p-2 m-3' id='nome' name='nome'>
<input type='text' class='rounded w-100 p-2 m-3' id='cognome' name='cognome'>
<input type='email' class='rounded w-100 p-2 m-3' id='email' name='email'>

<input type='radio' name='gender' value='M'>uomo
<input type='radio' name='gender' value='F'>donna

<select class='rounded w-100 p-2 m-3' id='age' name='age'>
  <option value=''> --- seleziona --- </option>
  <option value='1'> 0-20 </option>
  <option value='2'> 21-40 </option>
  <option value='3'> 41-60 </option>
  <option value='4'> 61 e oltre </option>
</select>

<input type='checkbox' value='1' class='rounded w-25' id='privacy' name='privacy'>

<input type='button' class='btn-success rounded ' value='Registrati' onClick="checkData()">
```



# Client-side validation form

corrisponde questa funzione di controllo

JS

```
function checkData() {  
    // qui inserisco i controlli da fare  
    if (document.getElementById("nome").value == "") {  
        alert("campo nome obbligatorio"); return;  
    }  
    if (document.getElementById("cognome").value == "") {  
        alert("campo cognome obbligatorio"); return;  
    }  
    if (document.getElementById("email").value == "") {  
        alert("campo email obbligatorio"); return;  
    }  
    if (document.getElementById("email").validity.valid == false) {  
        alert("campo email non valido"); return;  
    }  
    // campo gender - ho più elementi - posso usare la form  
    if (document.getElementById('frm').gender.value == "") {  
        alert("campo gender non selezionato"); return;  
    }  
    // campo privacy - controllo se selezionato  
    if (document.getElementById("privacy").checked == false) {  
        alert("campo privacy obbligatorio"); return;  
    }  
    document.getElementById('frm').submit();  
}
```



# Client-side validation form

Esercizio:

Il file **csfv\_esercizio.html** contiene l'html della pagina mostrata sotto. Scrivere il codice della funzione **inviarerecensione** per controllare la corretta compilazione dei campi presenti nella form.

## Lascia la tua recensione

Nome *	<input type="text"/>
Cognome *	<input type="text"/>
Luogo di nascita	<input type="text"/>
Provincia (sigla max 3 lettere)	<input type="text"/>
Email *	<input type="text"/>
Punteggio *	<input type="radio"/> 1 <input type="radio"/> 2 <input type="radio"/> 3 <input type="radio"/> 4 <input type="radio"/> 5
Do il consenso al trattamento dei dati *	<input type="checkbox"/>
<input type="button" value="Invia la mia recensione"/>	

20 minuti...



# Client-side validation form

Una soluzione:

JS

```
var nome=document.getElementById("f_nome").value;
var cognome=document.getElementById("f_cognome").value;
var email=document.getElementById("f_email").value;
var punti=document.getElementById("frm").punti.value;
if (nome==" || cognome==" || email==" || punti==" ) {
    alert("i campi con asterisco vanno compilati");
    return false;
}
// provincia
if (document.getElementById("f_provincia").value.length > 3) {
    alert("Max 3 caratteri per la provincia");
    return false;
}
// privacy
if (document.getElementById("f_privacy").checked == false) {
    alert("assicurarsi di aver dato il consenso al trattamento dei dati");
    return false;
}
if (document.getElementById("f_email").validity.valid == false) {
    alert("assicurarsi di aver scritto correttamente l'email");
    return false;
}
```





# User interaction



# Interazione con utente

Altro uso di JS è semplificare e guidare l'interazione dell'utente con la pagina web

Alcuni esempi sono:

- **segnalazione** di errore o di informazioni di contesto
- richiesta di **conferma** di una operazione
- acquisizione di **dati** da tastiera



Questa pagina dice

Hai fatto un click sul bottone!

OK

Questa pagina dice

Sei sicuro di voler cancellare questo elemento?

Annulla OK

Questa pagina dice

Inserisci il tuo nome!

Annulla OK

ref:  
[https://www.w3schools.com/jsref/met\\_win\\_alert.asp](https://www.w3schools.com/jsref/met_win_alert.asp)  
[https://www.w3schools.com/jsref/met\\_win\\_confirm.asp](https://www.w3schools.com/jsref/met_win_confirm.asp)  
[https://www.w3schools.com/jsref/met\\_win\\_prompt.asp](https://www.w3schools.com/jsref/met_win_prompt.asp)



# Interazione con utente

L'idea è di legare un elemento ad una funzione javascript che esegue le istruzioni di segnalazione, conferma o acquisizione dati

```
<button value='info' onClick='showinfo()'>info</button>  
<button value='info' onClick='showconferma()'>cancella</button>  
<button value='info' onClick='showprompt()'>inserisci</button>
```

HTML

```
function showinfo() {  
    alert("Hai fatto un click sul bottone!");  
}  
function showconferma() {  
    if (confirm("Sei sicuro di voler cancellare questo elemento?")) {  
        alert("Hai confermato!");  
    }  
    else {  
        alert("Non hai confermato!");  
    }  
}  
function showprompt() {  
    var n = prompt("Inserisci il tuo nome!");  
    alert("Ciao, " + n);  
}
```

JS

The image shows three sequential browser dialog boxes. The first is an alert box titled 'Questa pagina dice' with the message 'Hai fatto un click sul bottone!' and an 'OK' button. The second is a confirm box titled 'Questa pagina dice' with the message 'Sei sicuro di voler cancellare questo elemento?' and 'Annulla' and 'OK' buttons. The third is a prompt box titled 'Questa pagina dice' with the message 'Inserisci il tuo nome!' and an input field, with 'Annulla' and 'OK' buttons at the bottom.





# Interazione con utente

Un uso frequente è la richiesta di conferma alle operazione di cancellazione

- nei link si passa come parametro l'ID dell'elemento da cancellare

```
<!-- <a href='cancella.php?idriga={$arr["id"]} '>cancella</a> -->  
<a href='#' onClick="confermaCancellazione('{$arr["id"]}')">cancella</a>;
```

HTML

```
function confermaCancellazione(idelemento) {  
    if (confirm('Sei sicuro di voler cancellare questa riga?')) {  
        // chiama la pagina di cancellazione passando l'id corretto (*)  
        window.location = 'cancella.php?idriga=' + idelemento;  
    }  
    else {  
        // cosa faccio se non confermo....  
    }  
}
```

JS

(\*) la proprietà **location** di **window** permette di cambiare la pagina via javascript



# Client-side validation form

Esercizio:

Riprendiamo il file **ricerca.php** del progetto di gestione degli studenti e modifichiamo il link di **cancellazione** aggiungendo il controllo di conferma via javascript

15 minuti...





Giulio Angiani  
I.I.S. "Blaise Pascal" - Reggio Emilia