



HTML Validation API

Ing. Simone Giustetti
www.giustetti.net

Introdotta in HTML 5.

I controlli dei dati lato browser consentono di rilevare eventuali errori commessi dagli utenti, migliorando l'usabilità

Potenzialmente previene l'invio di dati errati ed il ritardo causato dal trasferimento via rete.

Non è esaustiva né sicura. **Una seconda serie di controlli deve essere eseguita sul server.**



Validazione delle Maschere

Fornisce un riscontro immediato in merito ai valori inseriti nei controlli delle maschere:

- “Il campo nome è obbligatorio”
- “Per cortesia utilizzare il formato AAAA-MM-GG per le date”
- “Per cortesia inserire un indirizzo e-mail valido”
- “La lunghezza minima di una password deve essere 8 caratteri”

Se i dati sono formattati correttamente, la maschera li trasmette al server, altrimenti mostra un messaggio di errore.



Giustificazioni per la Validazione

- Istruire gli utenti;
- Migliorare la fruizione del sito;
- Migliorare la formattazione dei dati;
- Migliorare le prestazioni;
- Ottenere dati corretti dagli utenti.



Giustificazioni per l'Uso della API

Tutti i browser eseguono una qualche forma di validazione predefinita basata sugli attributi degli elementi di pagina. **Il comportamento non è omogeneo.**

La API di validazione consente di personalizzare il comportamento dei browser, i riscontri che questi forniscono agli utenti ed i messaggi di errore.

Il codice JavaScript ha generalmente prestazioni peggiori della validazione implicita.



Validazione Mediante CSS

A tutti i controlli il cui contenuto differisce dalle condizioni di validazione viene assegnata la pseudo-classe CSS **:invalid**.

A tutti i controlli che passano le regole di validazione viene assegnata la pseudo-classe CSS **:valid**.

È possibile utilizzare le pseudo-classi per assegnare ai controlli una formattazione specifica quando i dati inseriti non sono validi.



Validazione Mediante CSS

```
input.textbox:invalid {  
    border: 2px dashed red;  
}
```

```
input.textbox:invalid:required {  
    background-image: linear-gradient(to right, pink,  
lightgreen);  
}
```

```
input.textbox:valid {  
    border: 2px solid black;  
}
```



Validazione Mediante JavaScript

La libreria JavaScript di validazione dei dati agisce sui seguenti controlli:

- <button>
- <fieldset>
- <input>
- <output>
- <select>
- <textarea>



Proprietà Aggiunte agli Elementi

La libreria JavaScript aggiunge le seguenti proprietà agli elementi di una pagina:

- **willValidate**: Rende vero se l'elemento verrà controllato prima dell'invio dei dati al server.
- **validationMessage**: Rende un messaggio di errore che può essere tradotto.
- **validity**: Rende lo stato di validità dell'elemento:
 - **patternMismatch**: Rende vero se il valore impostato non combacia con una espressione regolare
 - **tooLong**: Rende vero se la stringa inserita è troppo lunga



Proprietà Aggiunte agli Elementi

- ***tooShort***: Rende vero se la stringa inserita è troppo corta
- ***rangeOverflow***: Rende vero se il valore impostato supera il limite massimo
- ***rangeUnderflow***: Rende vero se il valore impostato è inferiore al minimo
- ***typeMismatch***: Rende vero se la e-mail oppure lo URL impostato non sono validi
- ***valid***: Rende vero se il valore impostato rispetta tutti i vincoli
- ***valueMissing***: Rende vero se non è stato inserito un campo obbligatorio



Funzioni di Controllo

La libreria JavaScript aggiunge le seguenti funzioni di validazione:

- **checkValidity()**: Rende vero se l'elemento non presenta problemi. Lancia un evento `invalid` in caso contrario;
- **reportValidity()**: Rende un elenco di campi contenenti valori non consentiti;
- **setCustomValidity(<messaggio>)**: Imposta un messaggio di errore personalizzato.



Esempio di Controllo

```
<form>
  <label for="indirizzo_mail">
    Inserire un indirizzo E-mail valido:
  </label>
  <input id="indirizzo_mail" name="indirizzo_mail"
    type="email" />
  <br />
  <button onclick="controllo()">Invia</button>
</form>
```

```
Function controllo() {
  const email = document.getElementById("indirizzo_mail");
  if( ! email.checkValidity ) {
    email.setCustomValidity( "L'indirizzo fornito non e'
      valido!" );
  } else {
    email.setCustomValidity( "" );
  }
}
```



Manuali

- https://www.w3schools.com/js/js_validation_api.asp
- https://developer.mozilla.org/en-US/docs/Learn/Forms/Form_validation



Informazioni & Licenze

LICENZA

Salvo dove altrimenti specificato grafica, immagini e testo della presente opera sono © Simone Giustetti. L'opera può essere ridistribuita per fini non commerciali secondo i termini della licenza:

Creative Commons Attribuzione - Non commerciale - Condividi allo stesso modo 4.0 Internazionale



È possibile richiedere versioni rilasciate sotto diversa licenza scrivendo all'indirizzo: studiosg@giustetti.net

TRADEMARK

- FreeBSD è un trademark di The FreeBSD Foundation.
- Linux è un trademark di Linus Torvalds.
- Macintosh, OS X e Mac OS X sono tutti trademark di Apple Corporation.
- MariaDB è un trademark di MariaDB Corporation Ab.
- MySQL è un trademark di Oracle Corporation.
- UNIX è un trademark di The Open Group.
- Windows e Microsoft SQL Server sono trademark di Microsoft Corporation.
- Alcuni algoritmi crittografici citati nella presente opera potrebbero essere protetti da trademark.

Si prega di segnalare eventuali errori od omissioni al seguente indirizzo: studiosg@giustetti.net

