**VALIDAZIONE DEL FORM -** [**https://www.w3schools.com/js/js\_validation.asp**](https://www.w3schools.com/js/js_validation.asp)

Uno degli utilizzi più frequenti di JavaScript è nell’ambito della validazione dei campi di un form:

* Riduce il carico delle applicazioni server side filtrando l’input
* Riduce il ritardo in caso di errori di inserimento dell’utente
* Consente di introdurre dinamicità all’interfaccia web

Generalmente si valida un form in due momenti:

1. Durante l’inserimento utilizzando l’evento ***onChange()*** sui vari controlli
2. Al momento del ***submit*** di un ***form.*** Intercettare l’evento ***onClick()*** del bottone di ***submit*** o l’evento ***onSubmit()*** del form, assegnare all’attributo il valore *«return validationFunc()»*

La funzione di validazione deve:

* Controllare se il valore di un dato campo corrisponde a quanto ci si aspetta, restituendo true e mandando il form alla servlet.
* In caso negativo, dare un messaggio di errore, restituendo false e non mandando il modulo.

Se un campo modulo (***fname***) è vuoto, questa funzione avvisa un messaggio e restituisce false, per impedire che il modulo venga inviato:

function validateForm() {  
  var x = document.forms["myForm"]["fname"].value;  
  if (x == "") {  
    alert("Name must be filled out");  
    return false;  
  }  
}

La funzione può essere chiamata quando il modulo viene inviato:

<form name="myForm" action="/action\_page.php" **onsubmit="return validateForm()"** method="post">  
Name: <input type="text" name="fname">  
<input type="submit" value="Submit">  
</form> <https://www.w3schools.com/js/tryit.asp?filename=tryjs_validation_js>

***Convalida automatica del modulo HTML***

La convalida del modulo HTML può essere eseguita automaticamente dal browser.

Se un campo modulo (***fname***) è vuoto, l' attributo ***required*** impedisce che questo modulo venga inviato:

<form action="/action\_page.php" method="post">  
  <input type="text" name="fname" **required**>  
  <input type="submit" value="Submit">  
</form> <https://www.w3schools.com/js/tryit.asp?filename=tryjs_validation_html>

Molto spesso, lo scopo della convalida dei dati è quello di garantire un corretto inserimento dell'utente.

**La convalida del lato server** viene eseguita da un server Web, dopo che l'input è stato inviato al server.

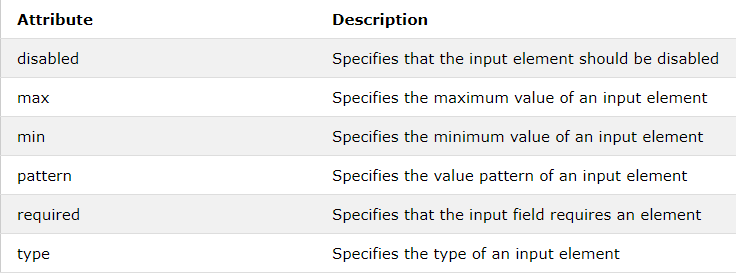
**La convalida lato client** viene eseguita da un browser Web, prima che l'input venga inviato a un server Web.

***Convalida vincoli HTML***

La convalida del vincolo HTML si basa su:

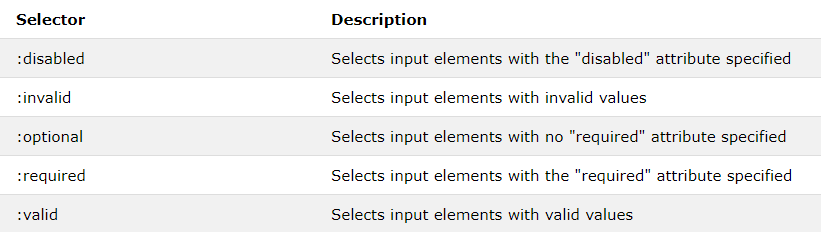
* **Attributi di input HTML di** convalida del vincolo
* **Selezione** vincoli **CSS Pseudo selettori**
* **Proprietà e metodi DOM di** convalida del vincolo

**Attributi di input HTML di convalida vincoli:**



<https://www.w3schools.com/html/html_form_attributes.asp>

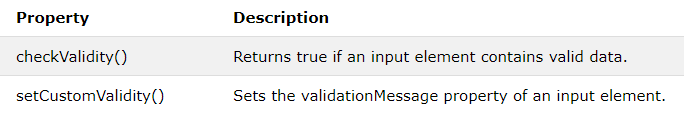
**Selettori Pseudo CSS di convalida vincoli:**



<https://www.w3schools.com/css/css_pseudo_classes.asp>

**API di convalida JavaScript -** [**https://www.w3schools.com/js/js\_validation\_api.asp**](https://www.w3schools.com/js/js_validation_api.asp)

**Metodi DOM Validazione vincoli:**



Se un campo di input contiene dati non validi, visualizza un messaggio:

<input id="id1" type="number" min="100" max="300" required>  
<button onclick="myFunction()">OK</button>  
<p id="demo"></p>  
<script> function myFunction() {  
  var inpObj = document.getElementById("id1");  
   if (!inpObj.checkValidity()) {  
     document.getElementById("demo").innerHTML = inpObj.validationMessage;  
   }  
}  
</script> <https://www.w3schools.com/js/tryit.asp?filename=tryjs_validation_check>

**JavaScript Espressioni regolari -** [**https://www.w3schools.com/js/js\_regexp.asp**](https://www.w3schools.com/js/js_regexp.asp)

***Un'espressione regolare*** è una sequenza di caratteri che forma un ***modello di ricerca***, che può essere utilizzato per la ricerca del testo e le operazioni di sostituzione del testo.

Le espressioni regolari possono essere utilizzate per eseguire tutti i tipi di **ricerca** di **testo** e operazioni di **sostituzione del testo** .

Sintassi: /pattern/modifiers;

var patt = /w3schools/i;

**/w3schools/ i**   è un'espressione regolare.

**w3schools**   è un pattern (da utilizzare in una ricerca).

**i**   è un modificatore (modifica la ricerca senza distinzione tra maiuscole e minuscole).

In JavaScript, le espressioni regolari vengono utilizzate con i due **metodi stringa**:

* Il metodo ***search()*** utilizza un'espressione per cercare una corrispondenza e restituisce la posizione della corrispondenza

con una stringa:

var str = "Visit W3Schools!";  
var n = str.search("W3Schools");

<https://www.w3schools.com/js/tryit.asp?filename=tryjs_string_search>

con una espressione regolare:

var str = "Visit W3Schools";  
var n = str.search(/w3schools/i);

<https://www.w3schools.com/js/tryit.asp?filename=tryjs_regexp_string_search>

* Il metodo ***replace()*** restituisce una stringa modificata in cui il modello viene sostituito.

con una stringa:

var str = "Visit Microsoft!";  
var res = str.replace("Microsoft", "W3Schools");

<https://www.w3schools.com/js/tryit.asp?filename=tryjs_string_replace>

con una espressione regolare:

var str = "Visit Microsoft!";  
var res = str.replace(/microsoft/i, "W3Schools");

<https://www.w3schools.com/js/tryit.asp?filename=tryjs_regexp_string_replace>