

Gestire i Form

[Pagina principale di Javascript](#)

La gestione dei form è uno dei principali usi di JavaScript.

La gestione dei form consiste nel controllare se un campo è compilato o meno oppure se in un dato campo è inserito un valore giusto.

I form servono a passare dati da una pagina all'altra, le pagine che ricevono i dati utilizzano il valore dell'attributo name dei tag del form. Per questo per fare controlli sui campi utilizzeremo l'attributo name e la proprietà value.

L'attributo name

```
<form id="myForm">
  <label>Nome:</label>
  <input type="text" name="nome">
  <input type="button" value="Vai" onclick="controlla()">
</form>

<script>
  function controlla(){
    let nome = document.querySelector('#myForm input[name=nome]').value
    console.log(nome)
  }
</script>
```

Vedere se un campo è stato compilato

Campi di testo

I campi testo sono i campi nei quali l'utente può scrivere:

- password
- textarea
- textfield

Il contenuto di questi campi è sempre una stringa, anche se inseriamo al loro interno dei valori numerici. Se abbiamo bisogno di controllare se in uno di questi campi ci sia un valore numerico, bisogna effettuare delle conversioni di tipo.

Per controllare se un campo è compilato si controlla la lunghezza della stringa corrispondente al suo valore.

Controllare i campi di testo

```
<form id="cdt">
```

```
<label for="campoDiTesto">Campo di testo</label><br>
<input type="text" name="campoDiTesto"><br>
<label for="pwd">Password</label><br>
<input type="password" name="pwd"><br>
<label for="textarea">Area di testo</label><br>
<textarea name="textarea"></textarea>
<input type="button" value="Vai" onclick="controlla()">
</form>

<script>
  function controlla(){
    let campoDiTesto = document.querySelector('#cdt
input[name=campoDiTesto]').value;
    let pwd = document.querySelector('#cdt input[name=pwd]').value;
    let textarea = document.querySelector('#cdt
textarea[name=textarea]').value;
    if(campoDiTesto.length==0){
      alert('Inserire un valore nel campo');
      //metto il cursore nel campo
      document.querySelector('#cdt input[name=campoDiTesto]').focus();
    } else if(pwd.length==0){
      alert('Inserire un valore nel campo');
      //metto il cursore nel campo
      document.querySelector('#cdt input[name=pwd]').focus();
    } else if(textarea.length==0){
      alert('Inserire un valore nel campo');
      //metto il cursore nel campo
      document.querySelector('#cdt textarea[name=textarea]').focus();
    } else {
      alert('Il form è compilato correttamente');
    }
  }
</script>
```

Select

L'utente non può scrivere nulla all'interno dei menù a tendina, può solamente selezionare una voce tra quelle presenti. Se non si seleziona nulla la voce di default è la prima della tendina, quindi si usa mettere un valore specifico nella tendina per controllare se l'utente ha selezionato qualcosa.

Controllo di una tendina

```
<form id="sel">
  <select name="lavoro">
    <option value="---">Scegli un lavoro</option>
    <option value="impiegato">Impiegato</option>
    <option value="disoccupato">Disoccupato</option>
    <option value="pIva">Libero professionista</option>
  </select>
  <input type="button" value="Vai" onclick="controlla()">
```

```
</form>
<script>
    function controlla(){
        let lavoro = document.querySelector('#sel
select[name=lavoro]').value;
        if (lavoro=='---'){
            alert('Scegliere un lavoro');
        }
    }
</script>
```

Checkbox

Anche il checkbox è un campo nel quale l'utente non può scrivere, è possibile controllare se è spuntato o meno:

Controllo di un checkbox

```
<form id="cb">
    <label for="cbName">Checkbox</label>
    <input type="checkbox" name="cbName" onchange="controlla()">
</form>
<script>
    function controlla(){
        let cbName = document.querySelector('#cb input[name=cbName]');
        if (cbName.checked){
            alert('Il checkbox è spuntato');
        } else {
            alert('Il checkbox non è spuntato');
        }
    }
</script>
```

Radio

Il radio serve per le scelte opzionali. Anche nel radio non è possibile inserire valori ma si può controllare se è spuntato o meno.

Radio button

```
<form id="radio">
    <p>Qual è il tuo piatto preferito?</p>
    <input type="radio" name="preferito" value="pasta">Pasta<br>
    <input type="radio" name="preferito" value="carne">Carne<br>
    <input type="radio" name="preferito" value="pesce">Pesce<br>
    <input type="radio" name="preferito" value="verdura">Verdura<br>
    <input type="button" value="Vai" onclick="controlla()">
</form>
<script>
```

```
function controlla(){
    let radio = document.querySelectorAll('#radio
input[name=preferito]');
    let chk = false;
    let val;
    for(i=0;i<radio.length;i++){
        if(radio[i].checked){
            chk = true;
            val = radio[i].value;
        }
    }
    if (chk){
        alert('Hai sceto ' + val);
    } else {
        alert('Scegli un valore');
    }
}
</script>
```

Controllare se un campo contiene un numero

Per controllare se un campo contiene un valore numerico, basta fare riferimento solo ai campi di testo, quanto detto varrà anche i campi password e textarea.

parseInt

La funzione **parseInt()** prende in input una stringa e restituisce un valore intero se la stringa può essere convertita oppure **NaN**. NaN sta per Not a Number ed è il valore tornato da parseInt() quando non riesce ad eseguire la conversione.

parseInt()

```
let str1 = "123";
let num1 = parseInt(str1); //num vale 123

let str2 = "123abc";
let num2 = parseInt(str2); //num vale 123

let str3 = "abc123";
let num3 = parseInt(str3); //num vale NaN
```

Quando la stringa inizia con dei numeri tramite parseInt() viene convertita in un numero, se la stringa con delle lettere la stringa viene convertita in NaN.

Per controllare se una stringa contiene o meno un valore numerico, insieme a parseInt() di può usare la funzione isNaN().

isNaN

La funzione **isNaN()** torna true se il parametro passato è NaN, falso altrimenti.

isNaN()

```
let str1 = "123";
let num1 = parseInt(str1); //num vale 123
isNaN(str1) //vale false

let str2 = "abc123";
let num2 = parseInt(str2); //num vale NaN
isNaN(str2) //vale true
```

Campo numerico

Vediamo il codice per controllare se un campo di un form è numerico:

Campo numerico

```
<form id="num">
  <input type="text" name="stringNum">
  <input type="button" value="Vai" onclick="controlla()">
</form>
<script>
  function controlla(){
    let stringNum = document.querySelector('#num
input[name=stringNum]').value;
    stringNum = parseInt(stringNum);
    if(isNaN(stringNum)){
      alert('Inserire un valore numerico');
      document.querySelector('#num input[name=stringNum]').focus();
    } else {
      document.querySelector('#num').submit(); //invia il form
    }
  }
</script>
```

parseFloat

La funzione **parseFloat()** è molto simile a parseInt(), solo che converte una stringa in un numero con la virgola, anche in questo caso se la conversione non è possibile il risultato sarà NaN.

parseFloat

```
let str1 = "123.23";
let num1 = parseFloat(str1); //num vale 123.23
```

```
let str2 = "123.23abc";  
let num2 = parseInt(str2); //num vale 123.23  
  
let str3 = "abc123.23";  
let num3 = parseInt(str3); //num vale NaN
```

Validare una email

La validazione di un indirizzo email consiste nel controllare se l'email è scritta nel modo corretto. Con javascript non è possibile sapere se una email esiste o meno.

Ecco una funzione che controlla la correttezza dell'indirizzo email:

Validare una email

```
function ValidateEmail(input) {  
    var validRegex = /^[a-zA-Z0-9.!#$%&'*/+=?^_`{|}~-]+@[a-zA-Z0-9-  
]+(?:\.[a-zA-Z0-9-]+)*$/;  
    if (input.value.match(validRegex)) {  
        alert("Indirizzo email valida!");  
        return true;  
    } else {  
        alert("Indirizzo email non valida!");  
        return false;  
    }  
}
```

From:
<https://www.wiki.webtocom.com/> - **WTC wiki**

Permanent link:
<https://www.wiki.webtocom.com/doku.php?id=page:form>

Last update: **2024/04/09 10:40**

