



## Lezione 8: Form Beyond JavaScript (READ ONLY)

# Introduzione

- ▶ Form sono uno degli esempi principali di interattività nel Web
- ▶ Elemento form con controlli come check box, radio button, submit button introdotti nel 1993
  - ▶ Rimasti identici per molti anni
- ▶ Dal 2005 circa inizia il lavoro su Web form 2.0 integrato in HTML5
  - ▶ Nuovi componenti prima sviluppati in JS e Flash adesso integrati direttamente
  - ▶ Facilitano componenti avanzati come slider, date picker
  - ▶ Supportano funzionalità native ad esempio per required field

# Introduzione

- ▶ HTML5 web form hanno introdotto nuovi elementi, input type, attribute e altre funzionalità
- ▶ Molte di queste funzionalità sono state usate per anni
  - ▶ Form validation, combo box, placeholder text, ...
- ▶ La differenza è che non c'è più necessità di utilizzare JavaScript per creare certi comportamenti, sono disponibili già nel browser
- ▶ Tutto quello di cui si ha bisogno è di inizializzare un attributo

# Introduzione

- ▶ HTML5 rende la gestione delle form più semplice per gli sviluppatori
- ▶ HTML5 rende la gestione delle form migliore anche per gli utenti
  - ▶ Con validazione client-side gestita nativamente dai browser, ci sarà più consistenza tra siti web
  - ▶ Siti web più performanti senza JavaScript ridondante

# Form: sintassi

- ▶ Elemento <form>
- ▶ Contiene un attributo id serve per applicare css
- ▶ Contiene un attributo action che specifica l'agente responsabile per il processing della form
  - ▶ Contiene la locazione a cui mandare i dati nella form
- ▶ Contiene un attributo method che specifica come i dati verranno trattati, formattati e comunicati

# Form: esempio

## ▶ contact\_form.html

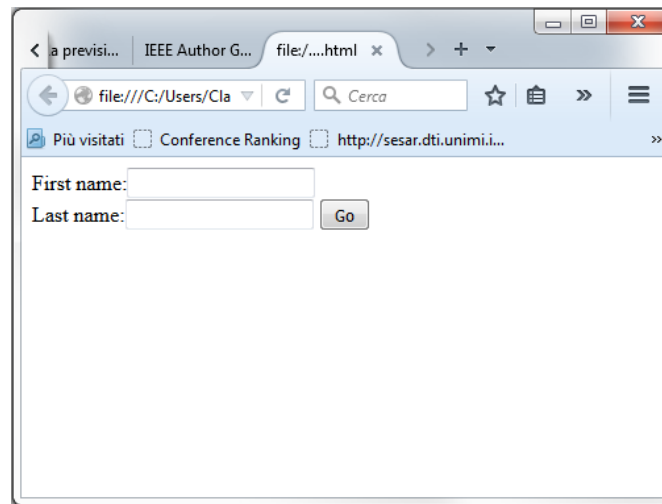
```
<form id="contactform" action="" method="post">
```

```
  First name:<input type="text" name=" " > <br />
```

```
  Last name:<input type="text" name=" " >
```

```
  <input type="submit" value="Go">
```

```
</form>
```



# Form: esempio

```
<form id="contactform" action="" method="post">  
  <label>  
    First name:<input type="text" name=" " >  
  </label><br />  
  <label>  
    Last name:<input type="text" name=" " >  
  </label>  
  <input type="submit" value="Go">  
</form>
```

- ▶ Label permette di selezionare la box semplicemente cliccando sul testo

# Form: esempio

▶ Fieldset.html

**<fieldset>**

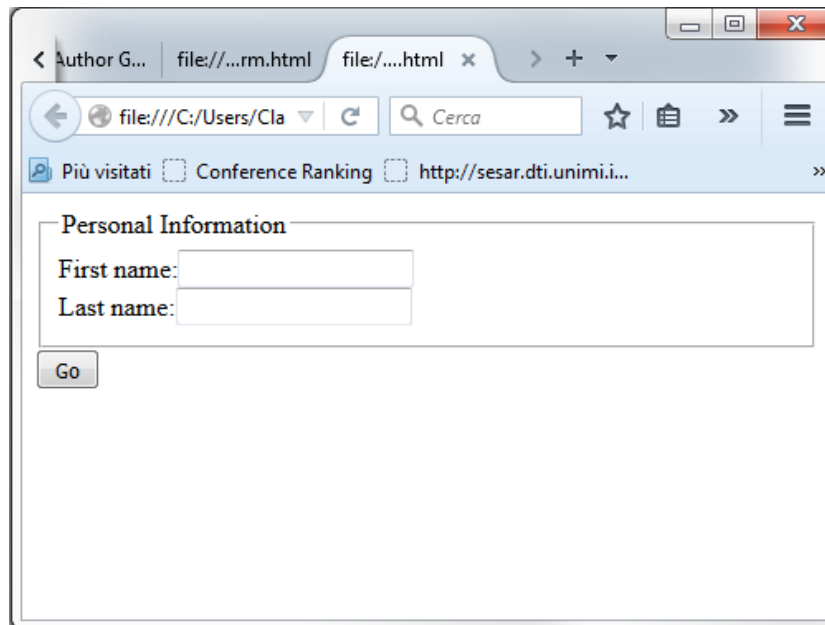
**<legend> Personal Information </legend>**

**<label>First name:<input type="text" name="firstname"> </label><br />**

**<label>Last name:<input type="text" name="lastname"> </label>**

**</fieldset>**

**<input type="submit" value="Go">**





# Form: esempio

## ▶ Fieldset2.html

```
<fieldset>
```

```
<legend>Additional  
Information</legend>
```

```
<p>Would you like to subscribe to  
the SmoothieWorld e-mail  
newsletter?</p>
```

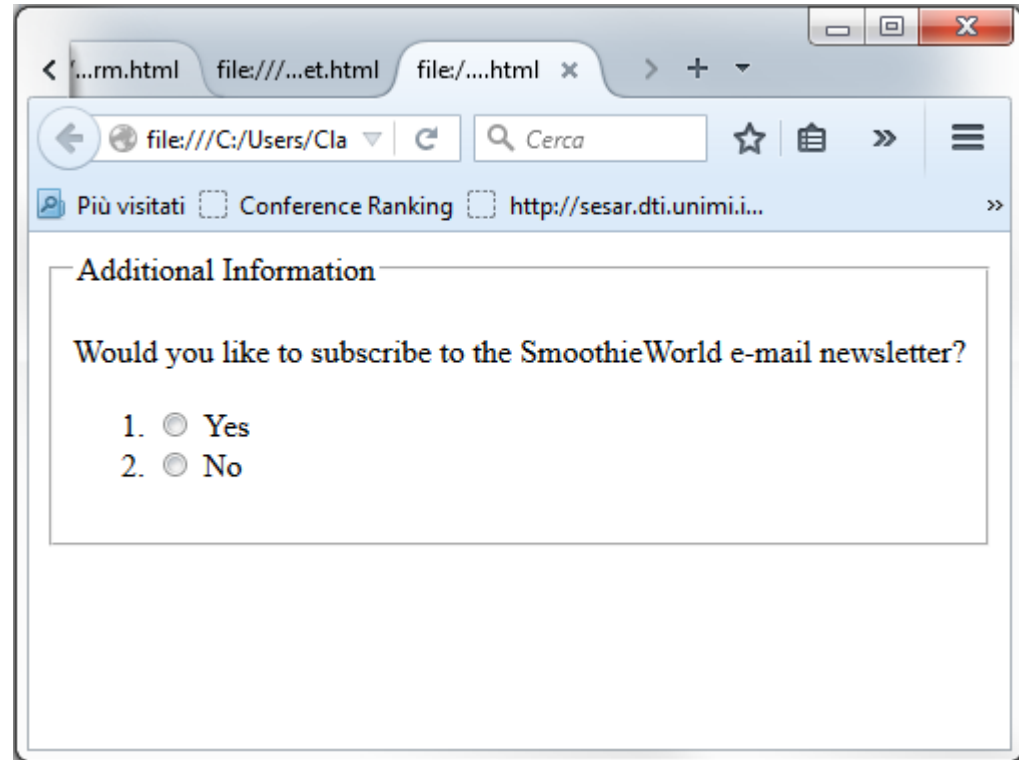
```
<ol>
```

```
<li> <input name="subscribe"  
type="radio"> <label>Yes</label>  
</li>
```

```
<li> <input name="subscribe"  
type="radio"> <label>No</label>  
</li>
```

```
</ol>
```

```
</fieldset>
```



# Form: esempio

- ▶ Aggiungere la label a un element di input

```
<label for="firstname">First name:<input id="firstname"  
type="text" name="firstname"></label>
```

```
<label for="lastname">Last name:<input id="lastname"  
type="text" name="lastname"> </label>
```

```
<input name="subscribe" id="yes" type="radio">
```

```
<label for="yes">Yes</label>
```

```
<input name="subscribe" id="no" type="radio">
```

```
<label for="no">No</label>
```

# Form: esempio

- ▶ Aggiungere una lista di elementi (data-list.html)

```
<fieldset>
```

```
<legend> Order free copies of smoothie recipes </legend>
```

```
<input type="text" list="smoothierecipes">
```

```
<datalist id="smoothierecipes">
```

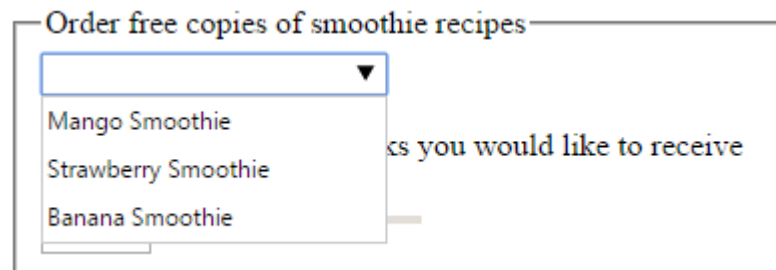
```
<option value="Mango Smoothie">Mango Smoothie</option>
```

```
<option value="Strawberry Smoothie">Strawberry Smoothie</option>
```

```
<option value="Banana Smoothie">Banana Smoothie</option>
```

```
</datalist>
```

...



Order free copies of smoothie recipes

Enter the number of copies you would like to receive

- Mango Smoothie
- Strawberry Smoothie
- Banana Smoothie

# Form: esempio

- ▶ Elemento data (data-list.html)
  - ▶ Date: `<input type="date">`
  - ▶ Data and time: `<input type="datetime">`

## Shipping Date

Date:

July, 2015 ▾

Sun	Mon	Tue	Wed	Thu	Fri	Sat
28	29	30	1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	1

- ▶ Date with constraints: `<input type="date" step="X" min="Y" max="Z" >`

# Tipi di input

- ▶ Tipi disponibili prima di HTML5
  - ▶ Button, checkbox, file, hidden, image, password, radio, reset, submit, text
- ▶ HTML5 fornisce input type con elementi UI data-specific e validazione dei dati nativa
- ▶ HTML5 supporta 13 nuovi input type
  - ▶ Search, email, url, tel, datetime, date, month, week, time, datetime-local, number, range, color

# <input>: range e checkbox

- ▶ HTML5 introduce diversi tipi di input
  - ▶ Email, numero di telefono, indirizzi web
- ▶ Range input (a slider nella pagina)
  - ▶ `<input type="range" min="0" max="100" value="50"/>`



- ▶ Check box
  - ▶ `<input type="checkbox" checked />`



- ▶ I valori sono definiti attraverso gli attributi

# <input>: search

- ▶ <input type> search
- ▶ Usato per campi di ricerca
- ▶ Search Google (search.html)

```
<form action="https://www.google.it/search"
method="get">
```

Search Google:

```
<input type="search" name="q">
```

```
<input type="submit">
```

```
</form>
```

## <input>: email

- ▶ <input type> email
- ▶ Tipo email (type="email") è usato per specificare uno o più indirizzi email
  - ▶ Supporta un attributo "multiple" che supporta l'inserimento di indirizzi email separate da virgole
- ▶ E-mail: <input type="email" name="usermail" />
- ▶ varie.html



# <input>: url

- ▶ <input type> url
- ▶ Tipo url (type="url") usato per campi di input che devono contenere indirizzi URL
  - ▶ Il valore di un campo url è validato automaticamente quando la form è sottomessa
- ▶ Add your homepage: <input type="url" name="homepage" />
- ▶ varie.html

# <input>: tel

- ▶ <input type> tel
- ▶ Tipo tel (type="tel") per specificare numeri di telefono
- ▶ Diversamente dai tipi url e email, tel non richiede una particolare sintassi/pattern
  - ▶ Lettere e numeri (o meglio ogni carattere a parte new line e carriage return) sono validi
- ▶ Telephone: <input type="tel" name="usrstel" />
- ▶ varie.html

## <input>: number

- ▶ <input type> number
- ▶ Per inserimento numeri
- ▶ Un campo di testo arricchito con due frecce (per increment e decrement)
- ▶ Permette di definire con attributi min, max, step
- ▶ <input type="number" min="" max="" step="">
- ▶ varie.html

# <input>: range

- ▶ <input type> range
- ▶ Tipo range (type="range") mostra uno slider nel browser
- ▶ Permette di definire attributi min, max, step
- ▶ Definisce un controllo per inserire un numero il cui valore esatto non è importante
- ▶ <label for="rating">On a scale of 1 to 10, my knowledge of HTML5 is:</label>  
<input type="range" min="1" max="10" name="rating" type="range">
- ▶ varie.html

## <input>: color

- ▶ <input type> color
- ▶ Tipo color (type="color") presenta all'utente un'interfaccia per la selezione di un colore
- ▶ Select your favorite color: <input type="color" name="favcolor" />
- ▶ varie.html

# <input>: keygen

- ▶ <input type> keygen
- ▶ Obiettivo elemento <keygen> è di fornire un meccanismo di autenticazione utente sicuro
- ▶ Elemento <keygen> fornisce un generatore di key-pair generator in una form
  - ▶ Quando la form è sottomessa, due chiavi, una pubblica e una privata, vengono generate
- ▶ Chiave private memorizzata localmente (browser key storage), chiave pubblica inviata al server
- ▶ Chiave pubblica utilizzate per generare un certificato del client e autenticare l'utente in futuro
- ▶ 

```
<form action="demo_keygen.asp" method="get">  
  Username: <input type="text" name="usr_name" />  
  Encryption: <keygen name="security" />  
  <input type="submit" />  
</form>
```
- ▶ Keygen.html
- ▶ [http://www.w3schools.com/tags/tryit.asp?filename=tryhtml5\\_keygen](http://www.w3schools.com/tags/tryit.asp?filename=tryhtml5_keygen)

# <input>: output

- ▶ <input type> output
- ▶ Elemento <output> presenta il risultato di un calcolo (come se fosse calcolato da uno script)

- ▶ <form  
oninput="x.value=parseInt(a.value)+parseInt(b.value)">0  
  <input type="range" name="a" value="50" />100  
  +<input type="number" name="b" value="50" />  
  =<output name="x" for="a b"></output>  
</form>

- ▶ output.html

0 50 100 + 50 =100

# Attributi: autocomplete

- ▶ `<form>` / `<input>`
- ▶ Attributo `autocomplete` specifica se una form o un input type devono avere la funzionalità di autocompletamento attiva
- ▶ Quando è attivato il browser completa automaticamente i valori sulla base dei valori usati in precedenza dall'utente
- ▶ 

```
<form action="action_page.php" autocomplete="on">  
  First name:<input type="text" name="fname"><br>  
  Last name: <input type="text" name="lname"><br>  
  E-mail: <input type="email" name="email" autocomplete="off"><br>  
  <input type="submit">  
</form>
```
- ▶ `autocomplete.html`



# Attributi: novalidate

- ▶ `<form>` / `<input>`
- ▶ Attributo `novalidate` è booleano
- ▶ Quando presente specifica che i campi di una form per l'immissione di dati non devono essere validati quando sottomessi
- ▶ `<form action="action_page.php" novalidate>`  
    E-mail: `<input type="email" name="user_email">`  
    `<input type="submit">`  
    `</form>`
- ▶ `novalidate.html`

# Attributi: autofocus

- ▶ `<input>`
- ▶ Attributo autofocus è booleano
- ▶ Quando presente specifica che elemento `<input>` deve ottenere il focus quando la pagina viene caricata
- ▶ Solo un elemento all'interno della form può avere l'attributo autofocus in una data pagina
- ▶ First name: `<input type="text" name="fname" autofocus>`

# Attributi: required

- ▶ `<input>`
- ▶ Attributo `required` è booleano
- ▶ Quando presente specifica che un elemento `<input>` **deve** essere definito correttamente
  - ▶ Se campo vuoto o invalido, la form non viene sottomessa e il focus passa al campo invalido
  - ▶ Associabile a ogni input type tranne `button`, `range`, `color`, and `hidden` (specificano un valore di default)
- ▶ `<label for="firstname">First name:<input id="firstname" type="text" name="firstname" placeholder="Enter Your First Name" required></label>`
- ▶ `required.html`

## Contact Us

The screenshot shows a 'Contact Us' form with a blue background. The form is divided into two main sections. The first section, titled 'Personal Information', contains four input fields: 'First name:' (with 'Jeremy' entered), 'Last name:' (with 'Osborn' entered), 'Email:' (empty), and 'Website:' (empty). The 'Email' field is highlighted with a red border, and a white tooltip box with a pointer to the field contains the text 'Please fill out this field.' Below this section is a question: 'Would you like to subscribe to the SmoothieWorld e-mail newsletters?' with two radio button options: 'Yes' and 'No'. At the bottom of the form is a 'Go' button.

# Attributi: form\* (estratto)

- ▶ `<input>`
- ▶ Attributo `formaction` specifica l'url del gestore dei dati quando la form è sottomessa
  - ▶ Sovrascrive attributo `action` di `<form>`
  - ▶ Usato con `type=submit` e `type=image`
- ▶ Attributo `formmethod` specifica il metodo HTTP da usare per spedire i dati all'url definita da `action`
  - ▶ Sovrascrive attributo `method` di `<form>`
  - ▶ Usato con `type=submit` e `type=image`
- ▶ Attributo `formnovalidate` impone di non validare i campi di una form quando sottomessi
  - ▶ Sovrascrive attributo `novalidate` di `<form>`
  - ▶ Usato con `type=submit`

# Attributi: form\* (estratto)

- ▶ Esempio formnovalidate (formnovalidate.html)
  - ▶ `<form action="action_page.php">`
    - E-mail: `<input type="email" name="userid"><br>`
    - `<input type="submit" value="Submit"><br>`
    - `<input type="submit" formnovalidate value="Submit without validation">`
    - `<input type="submit" formaction="action_page2.php" value="Submit to another page">`
  - ▶ `</form>`

# Attributi: list

- ▶ `<input>`
- ▶ Datalist permette di definire un insieme di opzioni autocomplete per un campo di testo (list.html)
- ▶ 

```
<input list="browsers">  
<datalist id="browsers">  
  <option value="Internet Explorer">  
  <option value="Firefox">  
  <option value="Chrome">  
  <option value="Opera">  
  <option value="Safari">  
</datalist>  
...
```

# Attributi: placeholder

- ▶ `<input>`
- ▶ Attributo placeholder visualizza un breve suggerimento all'interno dell'elemento form
  - ▶ Suggerisce all'utente cosa deve essere inserito in una form
- ▶ Il testo del placeholder sparisce quando il campo ottiene il focus e riappare se nessun dato viene inserito
- ▶ Funzionalità fornita per anni con JavaScript, supportato nativamente con HTML5 (placeholder.html)

# Attributi: readonly

- ▶ `<input>`
- ▶ Attributo `readonly` simile all'attributo `disable`
  - ▶ Rende impossibile l'editing da parte dell'utente
- ▶ A differenza di `disable`, il campo `readonly` può ricevere il focus e il suo valore essere sottomesso tramite una form
- ▶ `<input type="text" name="country" value="Italy" readonly>`