

MOBILE DEVELOPER

The background is a solid blue color with a faint, light-blue network diagram. This diagram consists of numerous small circular nodes connected by thin, light-blue lines, creating a complex web-like structure. A large, thin white arc is positioned in the upper right quadrant of the image. In the bottom right corner, there is a small, solid dark-blue circular shape.

Modulo web

Sommario

- Il modulo web.
- Il tag Form.
- I TAG di Input.

Il modulo web

I moduli web noti a tutti sono realizzati mediante il tag `<form>` di HTML.

Esso definisce l'inizio di un modulo web, in cui possono essere inseriti i campi di input ed i pulsanti per l'invio dei valori ad un'altra pagina.

```
<form action="page.html" method="post" name="form1">  
...  
</form>
```

Attributi del TAG <form>

name : viene specificato il nome del form;

action: Indica la pagina target a cui saranno inviati i valori dei campi di input, magari dopo la validazione per mezzo del submit.

method: Metodo di invio dei dati_GET o _POST.

enctype: Specifica il metodo di encoding dei dati una volta effettuato il submit.

Campi di input

Le input rappresentano le caselle di input dove l'utente può inserire del testo digitato dalla tastiera. I suoi attributi sono:

name: nome della campo di input.

size: definisce la lunghezza della casella.

value: definisce il valore.

type: definisce il tipo di casella di input.

INPUT TYPE: text

E' il classico campo di input testuale, adoperato maggiormente nei form.

```
<form name="form1" action="page.html" method="GET">  
<input type="text" name="inputText">  
  
</form>
```

INPUT TYPE: radio button

Sono caselle di scelta esclusive a cui viene associato lo stesso nome ad ogni singolo elemento per far comprendere al browser quali sono le opzioni che fanno parte dello stesso gruppo.

```
<form name="form1" action="page.html" method="GET">  
1 <input type="radio" name="scelta" value="1"/>  
2 <input type="radio" name="scelta" value="2"/>  
3 <input type="radio" name="scelta" value="3"/>  
</form>
```


INPUT TYPE: checkbox

Le checkbox sono raggruppate come le radio button, ma consentono una scelta multipla di opzioni di elementi dello stesso gruppo.

```
<form name="form1" action="page.html" method="GET">  
1 <input type="checkbox" name="scelta" value="1"/>  
2 <input type="checkbox" name="scelta" value="2"/>  
3 <input type="checkbox" name="scelta" value="3"/>  
</form>
```

La Combo box

Il classico menu a tendina, detto anche combo box, che può essere a scelta singola oppure multipla. Ciò dipende dal valore dell'attributo size applicato al tag <select>

```
<form name="form1" action="page.html" method="GET">  
<select name="InputSelect">  
    <option value="1">prima voce</option>  
    <option value="2">seconda voce</option>  
</select>  
</form>
```

La Textarea

Altro casella di input, molto adoperata nel form, che permette di inserire ampie descrizioni testuali. Il tag è `<textarea>`.

```
<form name="form1" action="page.html" method="GET">  
<textarea name="inputTextarea" rows="5" cols="10">Scrivere qui  
</textarea>  
</form>
```

INPUT TYPE: submit

Il campo di input che rappresenta un bottone per l'invio dei dati alla pagina target. Di solito è posto nella parte finale del form.

```
<form name="form1" action="page.html" method="GET">  
<input type="submit" name="Invia">  
  
</form>
```

INPUT TYPE: File

Nel moduli web è consentito l'invio al server di file, anche noto con il termine di upload.

Il form che accoglie questo tipo di input deve avere un enc-type particolare per il metodo POST.

```
<form name="form1" action="page.html" methd="GET">  
<input type="submit" name="Invia">  
  
</form>
```

Nuovi TAG di Input ed attributi

HTML5 introduce nel classico modulo web, detto form 2.0:

- 13 nuovi tipi di TAG di input;

- 5 attributi che è possibile specificare nel loro corpo.

Supporto ai nuovi TAG HTML5

La verifica del supporto dei browser per i nuovi TAG di input avviene mediante il seguente codice javascript.

```
Var input = document.createElement("input");  
input.setAttribute("type","range");  
if(newinput.type == "range"){  
    return true;  
}  
else{  
    return false;  
}
```

INPUT TYPE: Color picker

Un componente avanzato per la selezione del colore.

```
<input type="color" >
```

```
<form name="form1" action="page.html" method="GET">  
<input type="color" name="inputColor">  
  
</form>
```


INPUT TYPE: date

Un calendario per la selezione della data.

```
<form name="form1" action="page.html" method="GET">  
<input type="date" name="inputDate">  
  
</form>
```

INPUT TYPE: datetime

Un calendario con l'aggiunta di un controllo per la selezione di ore, minuti e secondi con indicazione del fuso orario (UTC).

```
<form name="form1" action="page.html" method="GET">  
<input type="datetime" name="inputDateTime">  
  
</form>
```

INPUT TYPE: datetime-local

Simile all'attributo datetime precedente, ma senza riferimento ad alcun fuso orario.

```
<form name="form1" action="page.html" method="GET">  
<input type="datetime-local" name="inputDateTime">  
  
</form>
```

INPUT TYPE: email

Un componente specifico per l'inserimento di indirizzi di posta elettronica.

```
<form name="form1" action="page.html" method="GET">  
<input type="email" name="inputEmail">  
  
</form>
```

INPUT TYPE: month

Un calendario che consente di specificare la coppia mese-anno.

```
<form name="form1" action="page.html" method="GET">  
<input type="month" name="inputMonth">  
  
</form>
```

INPUT TYPE: number

Un controllo specifico per i valori numerici.

```
<form name="form1" action="page.html" method="GET">  
<input type="number" name="inputNumber">  
  
</form>
```

INPUT TYPE: range

Una barra orizzontale sulla quale l'utente può far scorrere un cursore per indicare la selezione in un intervallo.

```
<form name="form1" action="page.html" method="GET">  
<input type="range" id="range" name="range">  
<output for="range" id="output"></output>  
</form>
```

INPUT TYPE: search

Un campo di ricerca.

```
<form name="form1" action="page.html" method="GET">  
<input type="search" name="search" results="5" autosave="saved-  
searches">  
  
</form>
```


INPUT TYPE: tel

Un campo utile a indicare numerici telefonici.

```
<form name="form1" action="page.html" method="GET">  
<input type="tel" name="inputTel">  
  
</form>
```

INPUT TYPE: time

Un controllo per l'inserimento di ore, minuti e secondi.

```
<form name="form1" action="page.html" method="GET">  
<input type="time" name="inputTime">  
  
</form>
```

INPUT TYPE: url

Un campo che ospita un indirizzo web.

```
<form name="form1" action="page.html" method="GET">  
<input type="url" name="inputUrl">  
  
</form>
```

INPUT TYPE: week

Un calendario che permette all'utente di selezionare una settimana dell'anno.

```
<form name="form1" action="page.html" method="GET">  
<input type="week" name="inputWeek">  
  
</form>
```

INPUT ATTRIBUTE: autofocus

Assegna automaticamente il focus ad un elemento del form.

```
<form name="form1" action="page.html" method="GET">  
  <input type="text" autofocus name="inputText">  
  <input type="text" name="inputText1">  
  
</form>
```

INPUT ATTRIBUTE: Pattern

Definisce regole, anche complesse, il cui formato del dato richiesto debba attenersi.

```
<form name="form1" action="page.html" method="GET">  
<input type="text" pattern="\w{3}" name="inputText">  
  
</form>
```

INPUT ATTRIBUTE: placeholder

Introduce il testo segnaposto per offrire un esempio di come il dato debba essere proposto.

```
<form name="form1" action="page.html" method="GET">  
<input type="search" placeholder="Inserire qui il testo" name="keyword">  
  
</form>
```

INPUT ATTRIBUTE: required

Identifica un elemento come obbligatorio ai fini dell'invio del form.

```
<form name="form1" action="page.html" method="GET">  
<input type="text" required name="inputText">  
  
</form>
```


DATA ATTRIBUTE

I data-attribute sono degli attributi personalizzabili utilizzabili con il formato JSON coppia «chiave=valore». La loro sintassi è

```
<div data-opacity="0.5">  
  <p data-testo="data">Informazione principale</p>  
  <p data-info="detail">dettaglio</p>  
</div>
```

DATA ATTRIBUTE: scopo

Il W3C al riguardo dei data-attribute afferma che essi sono destinati a memorizzare dei dati privati alla pagina, per i quali non ci sono attributi o elementi appropriati.

DATA ATTRIBUTE: Javascript

Il valore di questi TAG è facilmente gestibile tramite javascript.
Esistono:

set/getAttribute() - Metodi setter e getter standard già visti per altri elementi HTML

```
var mese = elemento.getAttribute('data-mese'); // Gennaio
```

dataset – indica una struttura dati contenente tutti gli attributi data-*

```
var mese = elemento.dataset.mese; // Gennaio;
```

MOBILE DEVELOPER