

I Form: risposte a domande frequenti

Autore: **Lorenzo Forti** (lorenzo.forti@gmail.com)

Com'è possibile utilizzare un'immagine per inviare un form?

```
<form name="saibal" action="....." method="post">
<input name="prova" type="text" size="25" class="campiform">
<a href="javascript:document.saibal.submit()">  </a>
</form>
```

Al posto di "saibal" dovete utilizzare il nome che avete dato al form. Se per esempio lo avete chiamato "prova" il collegamento dell'immagine diventerà:

```
<a href="javascript:document.prova.submit()"></a>
```

Ovviamente è possibile utilizzare un'immagine anche per resettare un modulo:

```
<form name="saibal" action="....." method="post">
<input name="prova" type="text" size="25" class="campiform">
<a href="javascript:document.saibal.reset()"></a>
</form>
```

Infine ricordo che è possibile sostituire l'immagine con del semplice testo.

Come posso dare il focus automaticamente ad un campo una volta caricata la pagina?

Ci sono diversi metodi, ma il più veloce e breve che conosco è questo:

```
<script language="JavaScript">
<!--
document.saibal.prova.focus();
//-->
</script>
```

Va inserito direttamente dopo il tag </form>. "saibal" e "prova" vanno sostituiti rispettivamente con il nome del form e del campo.

Posso utilizzare un form in html per ricevere dei messaggi? Posso anche creare una pagina di ringraziamento?

Premetto subito che la soluzione migliore, in questi casi, è quella di usare linguaggi lato server come CGI (il conosciutissimo Formmail), ASP o PHP. Se non potete utilizzare questi metodi sappiate che c'è una soluzione, anche se non sempre affidabile.

Innanzitutto, visto che utilizziamo HTML che è lato client, molto dipenderà dall'utente. Infatti basterà che il visitatore non abbia ben configurato il programma di posta e l'email non arriverà. Inoltre il funzionamento è limitato ad Outlook Express: chiunque abbia un altro programma di posta vedrà aprirsi una normalissima email pronta da spedire.

Se tutto questo vi può andar bene vi scrivo un classico esempio di modulo:

```
<form name="modulo" method="post" action="mailto:vostroindirizzo@provider.it"
ENCTYPE="text/plain">
<input name="prova" type="text" size="25" class="campiform">
<input type="submit" name="submit" value="Clicca & Godi">
</form>
```

È importante inserire la stringa ENCTYPE="text/plain" per essere sicuri che riceveremo un messaggio leggibile.

Se volete una pagina di conferma dopo che l'utente vi ha inviato il modulo c'è un piccolo espediente anche se tecnicamente non è possibile. Prima però facciamo una breve precisazione.

Nei moduli che utilizzano linguaggi lato server la pagina di conferma viene mostrata se, effettivamente, l'invio dell'email è andato a buon fine... in poche parole c'è un controllo reale sui dati inviati. Questo controllo non può essere fatto nei moduli html (che abbiamo detto essere lato client) ma, solamente vezzo estetico, può essere effettuato un redirect ad una pagina di ringraziamento. Il reindirizzamento, però, avviene sempre e comunque indipendentemente dall'esito positivo o negativo dell'invio. Può accadere quindi che, anche se l'utente non è riuscito ad inviare nulla, venga mostrata una bella pagina con scritto: "Grazie per avermi mandato un messaggio. Sei un amico!!".

Se non avete paura di essere insultati dai vostri utenti per questi possibili problemi ecco come fare un modulo che reindirizza ad una pagina predefinita dopo il SUBMIT:

```
<form name="modulo" method="post" action="mailto:vostroindirizzo@provider.it"
ENCTYPE="text/plain" onSubmit="location.href='pagina.htm'">
<input name="prova" type="text" size="25" class="campiform">
<input type="submit" name="submit" value="Clicca & Godi">
</form>
```

Non dovrete fare altro che inserire l'url (relativo o assoluto) della pagina nella stringa:

```
onSubmit="location.href='pagina.htm' "
```

È possibile limitare i caratteri inseribili di una textarea?

Ovviamente sì... e sempre grazie a javascript. Ecco come sarà strutturata la pagina:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">

<html>
<head>
  <title>Limita Textarea</title>
<SCRIPT LANGUAGE="JavaScript">
<!-- Begin
function textCounter(field, countfield, maxlimit) {
if (field.value.length > maxlimit)
field.value = field.value.substring(0, maxlimit);
else
countfield.value = maxlimit - field.value.length;
}
// End -->
</script>
```

```
</head>
<body>

<form name="saibal" action="..." method="post">
<input name="prova" type="text" size="25">

<textarea name="tarea" rows="5" cols="20"
onKeyDown="textCounter(document.saibal.tarea,document.saibal.mostra,100);"
onKeyUp="textCounter(document.saibal.tarea,document.saibal.mostra,100);"></textarea>

<input readonly type="text" name="mostra" size="3" value="100"> Max Caratteri
<input type="submit" name="submit" value="Clicca & Godi">
</form>

</body>
</html>
```

Il javascript tra i tag <head> non va toccato; ciò che dobbiamo personalizzare sono le due stringhe da inserire nel tag textarea:

```
onKeyDown="textCounter(document.saibal.tarea,document.saibal.mostra,100);"
```

```
onKeyUp="textCounter(document.saibal.tarea,document.saibal.mostra,100);"
```

I valori in neretto sono quelli da personalizzare:

- **saibal** è il nome del form
- **tarea** è il nome della textarea
- **mostra** è il nome del piccolo box che contiene i caratteri rimanenti
- **100** è il numero dei caratteri da conteggiare. Il valore deve essere uguale in tutte e due le stringhe

È possibile rendere un campo non modificabile anche con Netscape 4.x?

Come sapete l'attributo "readonly" è valido solo per Explorer e, a partire dalla versione 6, anche per per Netscape. Ma ovviamente Navigator 4.x non è così generoso con noi quindi, ancora una volta, bisogna ricorrere a javascript.

Personalmente conosco due soluzioni: a voi la scelta di quella che vi piace di più.

La prima, proposta da Massimo61 sul nostro forum, è decisamente veloce e utilizza l'istruzione onFocus="this.blur()" in questa maniera:

```
<form name="saibal" action="..." method="post">
<input name="prova" type="text" size="25" onFocus="this.blur()">
<input type="submit" name="submit" value="Clicca & Godi">
</form>
```

La seconda è più lunga ma ugualmente funzionale:

```
<form name="saibal" action="..." method="post">
```

```
<input type="text" name="prova" value="non modificabile" onFocus="(this.disabled); {
this.value = saveValue; setTimeout('document.saibal.prova.blur()',1); } ">
<input type="submit" name="submit" value="Clicca & Godi">
</form>
```

E, dopo il tag di chiusura del form, inseriamo questo breve codice:

```
<script language="JavaScript">
<!--
document.nomeform.text.disabled = true;
saveValue="non modificabile";
//-->
</script>
```

Anche in questo caso ricordo di personalizzare le stringhe document.saibal.prova con il nome del proprio modulo e del proprio campo.

Non ho di certo esaurito l'argomento ma spero di aver dato dei buoni spunti per dimostrare che i moduli, alla fine, non sono così difficili da gestire come si crede.

Con qualche nozione in più di css, javascript e html possiamo rendere un modulo perfettamente visibile anche con Netscape 4.x e vi assicuro che, se siete riusciti a superare lo stress creato da quello che alcune persone chiamano browser, siete pronti per affrontare qualsiasi cosa.