

Guida ai Form HTML

Spiegazione del funzionamento dei form contenuti nel file moduliHTML2026.zip

Introduzione

Il file .zip fornito contiene file HTML, tutti esempi didattici sui form, più un'immagine (willc.gif) e un file phpinfo.php. Di seguito la spiegazione di ciascuno, raggruppati per argomento.

Form base

base.html

Form minimale con `action="tab.php"`, metodo di default **GET**. Un solo campo testo (`name=tabellina`), bottone submit e reset.

base-post.html

Stesso schema ma con `METHOD=POST` esplicito, e `action` che punta a un eseguibile CGI locale (`echo.exe` su 127.0.0.1) — tipico di ambienti di test Windows/CGI d'epoca.

esempio.html e provamask.html

Form anagrafica completa: nome, cognome, sesso (radio `mf`), data di nascita, email, note, dentro una `<table>` per il layout. Contengono anche un campo **hidden** (`AZIONE=S`) che viene inviato sempre, invisibile all'utente — usato tipicamente per dire allo script server-side quale azione eseguire. Nei due file html `action` punta a due eseguibili CGI diversi (`cgi.exe` e `provamask.exe`).

mask.html

Non è un vero form (manca l'apertura tag funzionante), ma una “maschera di ritorno”: mostra come una pagina di risposta potrebbe visualizzare i valori ricevuti (`$nome`, `$cogn`, ecc. — sintassi tipo template lato server).

Singoli tipi di input

- **checkbox.html**: una checkbox con `CHECKED`. Nota: se non è spuntata, il browser non invia affatto quella variabile al server.
- **radio.html**: gruppo di radio button con lo stesso `name="case"` — solo uno selezionabile per volta, uno preselezionato con `CHECKED`.
- **select.html e select2.html**: menu a tendina (`<select>`) con opzioni numeriche o testuali, per generare una tabellina o scegliere un colore.
- **SelectSenzaSubmit.html**: usa `onchange="this.form.submit()"` sul `<select>`, quindi il form si invia automaticamente non appena l'utente sceglie un'opzione, senza pulsante submit.
- **textarea.html**: area di testo multilinea (`<textarea>`), utile per messaggi lunghi.
- **password.html e password-get-banale.html**: campo password mascherato. Nota: `SIZE` controlla solo la larghezza visiva; nel secondo file `MAXLENGTH` controlla che non si possa digitare oltre il limite previsto.
- **hidden.html**: unico campo `TYPE=HIDDEN`, invisibile ma sempre inviato — utile per passare dati fissi al server.
- **input image.html**: input `TYPE=IMAGE`, immagine cliccabile che funge da bottone di invio (invia anche le coordinate del click).

- **ismap.html**: non è un form, ma una mappa immagine lato server (**ISMAP**) — cliccando sull'immagine si passano le coordinate all'URL nel link.
- **bottone.html**: form senza **action** (non invia dati da nessuna parte), con un bottone semplice (**TYPE=Button**) che richiama una funzione JavaScript locale (**onclick="funzione()"**) invece di inviare il form.

HTML5

input-type-html5.html

Vetrina dei nuovi tipi di input HTML5: **color**, **date**, **datetime-local**, **email**, **month**, **number**, **range**, **search**, **tel**, **time**, **url**, **week**. Alcuni hanno **required** (impedisce l'invio se vuoti) e uno ha anche **pattern** (regex di validazione, qui per forzare un URL che inizi con http:// o https://).

input-type-in-grigino.html

Solo uno snippet isolato: un input testo con **placeholder**, il testo grigio che scompare al focus.

tutto.html

Contiene **due form identici** per confronto diretto: uno con **METHOD=GET** e uno con **METHOD=POST**, entrambi verso **phpinfo.php**. Ogni form include quasi tutti i tipi di campo visti sopra insieme: hidden, testo, password, checkbox, radio, textarea, select e input image. È pensato per mostrare la differenza pratica tra GET (dati visibili nell'URL) e POST (dati nel corpo della richiesta) usando **phpinfo.php** come pagina di debug per vedere cosa arriva al server.

Sintesi

Il materiale copre progressivamente ogni elemento di form HTML — testo, password, checkbox, radio, select, textarea, hidden, image, i nuovi tipi HTML5 — e le differenze tra GET/POST, con un esempio di invio (submit) automatico via JavaScript.