

Web design elements: i form (design)

Nella scorsa lezione, abbiamo visto quali sono gli elementi principali che compongono i form, e quali sono i loro principali attributi. In questa lezione vedremo come strutturare i form con XHTML, come applicare degli stili di base e quali sono i layout più diffusi.

In generale, è considerata pratica comune porre un elemento per riga, così da permettere agli utenti di seguire un percorso di completamento più semplice possibile e di concentrarsi su un solo elemento alla volta.

La scelta più importante è quella della posizione dell'etichetta. Le posizioni più utilizzate per questo elemento sono:

Sopra gli input (Ebay).

A sinistra degli input, allineate a sinistra (Google).

A sinistra degli input, allineate a destra (Yahoo).

La scelta dipenderà da vari aspetti: obiettivo del form, argomento delle richieste, velocità di completamento. Vedremo caso per caso quali sono i vantaggi di ogni posizione.

Form di registrazione: HTML

Prima, però, vediamo come strutturare un semplice form di registrazione in XHTML. Ecco il markup:

Vediamo quali sono i punti principali:

- Inserire un **id al form**, così poter applicare degli stili specifici in seguito.
- Racchiudere i vari elementi nell'elemento fieldset con relativa legenda, **raggruppandoli per aree di significato**. Nell'esempio ho raggruppato tutti gli elementi relativi alle informazioni personali e quelle relative all'indirizzo.
- Utilizzare un elemento block-level, come paragrafi, liste, div, per racchiudere ogni input. Nell'esempio ho utilizzato una lista ordinata per racchiudere i vari input (a parte il pulsante di invio, che è stato racchiuso in un paragrafo): la lista è una scelta personale, vanno benissimo anche i div o i paragrafi.
- Aggiungere l'**attributo tabindex** ad ogni input, così da sovrascrivere l'ordine di default imposto dal browser.

L'ultimo punto su cui voglio porre l'attenzione è l'utilizzo dell'elemento abbr per indicare i campi obbligatori: infatti l'asterisco è *esattamente* un'abbreviazione che indica "campo obbligatorio" e questo accorgimento rende il nostro codice più significativo.

Ecco [il form risultante](#), senza nessuno stile applicato. Come puoi vedere, **il form è ben strutturato e significativo**, nonostante la mancanza del foglio di stile.

Form di registrazione: CSS di base

Applichiamo alcuni stili di base, eliminando quelli di default del browser:

Ho azzerato margini e padding degli elementi più comuni (per questo esempio è bastato questo, per i tuoi lavori potresti considerare l'uso di un css di reset più completo); eliminato la numerazione crescente della lista ordinata e centrato il tutto.

Una scelta "sofferta" per l'elemento legend

L'aspetto dell'elemento legend è notoriamente difficile da modificare. Questo perché diversi browser lo trattano in modi diversi: alcuni come elemento interno al fieldset, altri come elemento esterno; e questo complica notevolmente le cose. Una rapida soluzione è quella di aggiungere uno span aggiuntivo (e ingiustificato) all'interno e applicare gli stili a quest'ultimo elemento:

Questa soluzione, per quanto efficace, **aggiunge del markup non significativo** all'interno del

documento XHTML: se riesci a convivere con la cosa, bene; se invece non riesci a dormire la notte pensando di esserti macchiato di un grave crimine (come ti capisco!), puoi ovviare utilizzando JavaScript o jQuery. Ad esempio questa riga di codice:

permette di ottenere lo stesso risultato (tramite [jQuery](#)) senza appesantire il documento HTML.

Siamo pronti quindi ad applicare gli stili relativi all'elemento legend, aggiungendo anche quelli relativi ai campi obbligatori e al pulsante di invio:

[Questo è il risultato](#) di questa prima fase del lavoro. Esaminiamo ora i pro e i contro dei vari metodi di allineamento delle etichette.

Etichette allineate superiormente

Dei tre metodi proposti, questo è quello più utilizzato e a buon ragione: è quello che **garantisce tempi di completamento più rapidi** da parte degli utenti. Il motivo è dato dalla vicinanza tra etichetta e campo d'input: con un solo sguardo è possibile associare i due elementi, processando più rapidamente il form. Un altro vantaggio di questo metodo è il maggior spazio a disposizione per l'etichetta che, anche se più lunga, non altera la struttura generale del form.

L'unico svantaggio è lo spazio verticale occupato: rispetto agli allineamenti laterali è quasi il doppio, dando l'impressione di un form più lungo da completare.

Il CSS per ottenere questo effetto è molto semplice:

Rendiamo tutti gli input e le etichette come block level, in modo da fargli occupare una riga per uno, ottenendo [questo risultato](#).

Etichette allineate a destra

Come nel caso precedente, la vicinanza tra etichetta e campo d'input, permette una rapida associazione tra i due elementi. Il problema è che, essendo le lingue occidentali lette da sinistra verso destra, viene ridotta di molto la possibilità di effettuare uno *scanning* veloce delle etichette, a causa delle varie rientranze nelle voci.

Il codice CSS per questa struttura è il seguente:

Applichiamo il floating alle etichette, avendo cura di applicare una larghezza fissa ed un piccolo margine destro, in modo da non farle risultare attaccate agli input. L'aggiunta della larghezza fissa fa sì che le etichette più lunghe vadano a capo, quindi è bene tenere in conto questo particolare. Infine, allineamo il testo a destra, ottenendo [questo risultato](#).

Etichette allineate a sinistra

Dei tre metodi, questo è quello che prevede i tempi di completamento più lunghi, a causa dei vari "salti" che bisogna fare per passare dall'etichetta all'input corrispondente. Questo potrebbe sembrare un aspetto negativo, ma non è così. Immagina di avere un form che colleziona dati non troppo familiari (ad esempio numero di carta di credito, indirizzo della filiale bancaria di appartenenza): questi tempi lunghi permettono al web designer di "frenare" gli utenti, in modo da considerare con attenzione i dati inseriti. Inoltre l'allineamento a sinistra garantisce un rapido *scanning* delle voci.

Il CSS relativo è identico al precedente, escluso l'allineamento del testo:

E questo è il [risultato finale](#).

Conclusioni

Dietro ogni scelta progettuale si nasconde un universo di dettagli che possono fare la differenza. In questo articolo abbiamo visto un esempio di come ciò sia vero. Se l'argomento ha stuzzicato il tuo interesse, ti consiglio di leggere [lo studio di Matteo Penzo](#) che espone con più precisione gli aspetti che abbiamo elencato. La prossima settimana vedremo come validare i dati "al volo" utilizzando jQuery e faremo insieme le ultime considerazioni su questo affascinante argomento.

- [I form: elementi di base](#)

- I form: struttura e aspetto
- [I form: jQuery e considerazioni](#)
- [Come realizzare un modulo contatti in PHP](#)