

Form accessibile: come posizionare gli elementi e renderlo attraente?



Ci eravamo lasciati con gran suspense, curiosi di vedere come [il nostro brutto anatroccolo](#), di difficile comprensione, si sarebbe trasformato in uno splendido cigno... e ora sei pronto per seguirmi in questo processo di trasformazione attraverso il codice CSS?

L'(x)HTML per completare il nostro form

Prima di procedere dobbiamo aggiungere un altro po' di mark-up al nostro documento .html:

prendi quindi [il file di esempio che abbiamo realizzato due settimane fa](#), e inizia a **inserire il tag span all'interno di ogni tag label** che sia figlio diretto del primo , **in modo che ogni tag span contenga solo il testo** e non il campo input come di seguito.

Entrando nei **tag fieldset figli del fieldset principale** (che, ti ricordo, abbiamo usato per i radio-button e le check-box) invece dovrai **aggiungere ancora il tag span** a racchiudere il testo contenuto dal tag legend, e **raggruppare le varie scelte (tutte le label) all'interno di un tag paragrafo**

: vediamo un esempio.

Alla fine dovresti ottenere [il codice di questa pagina](#): man mano capirai il perché di questo mark-up aggiuntivo. D'ora in avanti, non toccheremo più il file html: non serve aggiungere classi, id né altro

mark-up.

CSS: posizioniamo gli elementi

Avrai notato, in alto nella pagina, i vari richiami (tramite il metodo import) ai fogli di stile CSS, vediamo in cosa consistono.

Reset

In genere, la prima cosa che faccio quando sviluppo un qualsiasi nuovo progetto, è **richiamare un foglio di stile di reset**. Questo foglio di stile ha il compito di **eliminare di fatto qualsiasi regola pre-impostata dal foglio di stile del browser** (ad esempio il margine sinistro e i “pallini” dell'elemento list

o, per restare in tema, toglie il bordo al fieldset) avendo la precedenza su di esso.

Questi file vengono messi a disposizione **gratuitamente** ed è possibile trovarne molteplici senza grandi differenze: [ricerca su Google](#). Io utilizzo quello di Yahoo, cui ho apportato alcune modifiche (ad esempio la dimensione del font a 62,5%, in modo da avere una facile corrispondenza tra misura pixel e em: 12px = 1.2em).

E' importante ricordarsi che **questo file va richiamato per primo, in quanto altrimenti sovrascriverebbe le regole precedenti**.

Larghezze

Il form è un elemento che non deve contenere molto testo: **le etichette sono generalmente brevi**, anche volendo inserire brevi descrizioni che aiutino l'utente nella compilazione si tratta il più delle volte di **una lunghezza che va dalla singola parola alla breve frase**.

Per questo motivo **non è necessario impostare una larghezza eccessiva**, che potrebbe anzi essere controproducente dal momento che gli elementi si posizionerebbero al suo interno in maniera quasi casuale. Per gli stessi motivi è **opportuno impostare una larghezza fissa** piuttosto che lasciare ai vari elementi la libertà di ridimensionarsi in base alla dimensione dei caratteri o della finestra del browser.

Personalmente, se si tratta di un form a una sola colonna evito di andare oltre i 400px, e in questo caso **ho impostato una larghezza di 380px direttamente al fieldset, in un nuovo file .css**.

Scendendo in profondità nel nostro form, troviamo **come figli diretti del fieldset** le varie etichette label e gli altri fieldset e adesso ti occuperai di questi, dandogli lo stesso stile.

A questi tag potremmo dare una larghezza di 370px, e un margine di 5px su ogni lato, come di seguito.

In questo modo però **la regola dei 5px di margine non viene rispettata dai browser, è per tanto necessario floatare gli elementi** tramite l'aggiunta del seguente codice.

Le etichette

Abbiamo inserito i testi delle nostre etichette **all'interno di tag span, figli di label**, sappiamo perciò che **tali testi sono selezionabili così: "label span"**.

Sappiamo anche che il tag legend all'interno dei fieldset **dovrebbe avere lo stesso aspetto delle altre etichette** e per questo gli daremo lo stesso stile.

Queste etichette dovranno essere floatate a sinistra, avere una larghezza di 150px ma allineare il testo a destra, **in modo che il testo non sia troppo distante dal campo input**, generando confusione.

Verticalmente, **il testo si allinea automaticamente in alto all'interno dell'elemento, perciò è meglio aggiungere un padding-top** che allinei in maniera migliore il testo rispetto al suo campo input.

Per evitare che poi i campi input siano proprio a ridosso del testo, aggiungiamo un margine destro che li separi.

Può essere che nel caos generale, non te ne sia reso conto: ma se utilizzi Firefox come browser, riscontrerai un problema con i tag legend, cui alcune versioni di Firefox **si rifiutano di riconoscere la larghezza** che abbiamo assegnato.

Se andassimo avanti in questa direzione, vedremo anche che, una volta assegnato lo stile ai campi input e ai tag paragrafo, questi ultimi non si allineerebbero come i primi, perché il tag legend li spingerebbe a capo.

Niente paura: **è proprio per questo motivo che abbiamo predisposto il tag span all'interno del tag legend**, in modo da dare a lui lo stile che ci interessa.

Modifichiamo quindi il codice precedente come segue, aggiungendo il tag span nel CSS.

In questo modo, **Firefox funziona a dovere, ma è Safari a fare l'antipatico**: infatti il tag legend ha, misteriosamente, un'altezza uguale a zero (o comunque non viene visualizzato a dovere), per cui **le etichette non sono visibili**.

Per ovviare a questo problema basta fare un mix tra le ultime due righe di codice presentato, cioè dare lo stile sia al tag fieldset legend, che al tag fieldset legend span.

Ora **le etichette si disporranno esattamente una sotto all'altra**, tutte con il testo allineato a destra. Anche i tag input saranno più o meno in posizione, eccezion fatta per ciò che è contenuto dal tag

, quindi le check-box e i radio-button: adesso ci occuperemo proprio di questo.

Gli input

Ai campi **input**, **textarea**, **select** e all'elemento **paragrafo daremo lo stesso stile**: una larghezza di 200px e il floating a sinistra.

Su Safari nessun problema, **su Firefox, il solito tag legend ci manda il paragrafo a capo a sinistra**, togliendolo dall'allineamento del resto del form.

Per risolvere il problema su Firefox, **floattiamo il tag**

a destra, al contrario degli altri: aggiungiamo quindi, al codice precedente quello che segue.

Su Safari il risultato è piacevole, **ma su Firefox lo spazio a destra della legend rimane bianco, spingendo in basso il contenuto del tag p.**

La cosa migliore sarebbe aggiungere un'altra proprietà al tag

: un margine negativo in alto, in modo che il contenuto torni su.

Ottimo, il problema **su Firefox è risolto** e il risultato è buono anche su IE8 e 7.

A dir la verità su IE7 ci sarebbe la necessità di aumentare un po' il margine negativo, in modo che anche qui la prima scelta si posizioni alla stessa altezza dell'etichetta. Ma dato che questo **non mina la comprensione ed usabile del form**, possiamo sorvolare.

Purtroppo su IE6 il **problema è leggermente più grave**, in quanto il paragrafo viene forzato ad andare a capo, posizionando quindi le varie scelte sotto al titolo “Registrati” e **facendo sì che il form perda in coerenza**.

Come fare a correggerlo? Commenti condizionali? Una preghiera?

No, la soluzione è molto più semplice e il risultato è perfetto, **per di più corregge anche il problema su IE7!** Ci basta aggiungere un margine sinistro al paragrafo, pari alla larghezza delle label, e quindi aumentare il margine in alto, dato che il paragrafo andrà in questo modo a capo in qualsiasi browser e perciò dovremo modificargli nuovamente la posizione, cambiamo allora la regola sui paragrafi nel seguente modo.

Le larghezze di Check-box e Radio-button

Avendo assegnato una larghezza ai campi input, **anche i radio-button e le check-box ne risentiranno**: pur conservando il loro aspetto naturale (rispettivamente cerchio e quadrato), su Safari occupano lo stesso spazio degli altri campi input, cioè i 200px di larghezza, spingendo il testo della loro etichetta più a destra.

Per questo motivo **dobbiamo dichiarare che i campi input all'interno di una label, all'interno di un elemento paragrafo non avranno una larghezza definita**, ma automatica.

In ogni modo sarà poi necessario **aggiungere un margine destro** agli stessi campi per evitare che i relativi testi si addossino ad essi.

Nota: avremmo potuto utilizzare delle dichiarazioni diverse per distinguere i radio-button e le check-box dall'input text, cioè scegliere in base al valore dell'attributo type dell'elemento input (selettore dell'attributo o selettore[attr]). Ad esempio assegnando la regola della larghezza solo a input[type=text]. Ma dovendo aggiungere le regole anche per i valori password, radio e checkbox, che **ci si aspetta siano coerentemente simili** (due a due) **all'interno dello stesso form**, mi è sembrato superfluo un approccio del genere, a prescindere dal fatto che IE6 non lo supporta.

Button

Ora che abbiamo posizionato tutto, la vista dei bottoni “Annulla” e “Invia” lì, sulla sinistra, mi sembra che sbilancino il quadro generale, **appesantendo troppo a sinistra e senza dare un valido supporto alla colonna di destra**.

Per cui aggiungerei un float a destra, e così noterai che **i bottoni sono troppo appiccicati alla textarea sovrastante e un po' troppo vicini tra loro**: per questo motivo aggiungiamo anche un margine di 5px.

L'aspetto del testo e legend

Fin'ora non abbiamo utilizzato nessuna proprietà font, e noterai che tutti i testi hanno il medesimo aspetto mentre **probabilmente ci potrebbe interessare la possibilità di distinguere il titolo del form “Registrati” da tutto il resto**.

Potresti racchiuderlo in un tag heading (h2, h3... a seconda dell'organizzazione della tua pagina – ricordati che hai caricato il reset.css e dovrai dare uno stile a questi elementi); oppure potresti decidere di **dare lo stile direttamente al tag legend**.

In questo secondo caso, dovresti distinguere il primo tag legend da quelli seguenti, qual è la differenza? Che **questo tag legend è figlio diretto di primo livello del fieldset unico figlio diretto di primo livello del form**.

Ma come si traduce questo albero genealogico?

Il selettore che indica i figli diretti di primo livello di un elemento è “>”, quindi la dichiarazione CSS sarebbe:

Questo selettore non è supportato da IE6. Ora, se il tuo sito ha problemi ben più gravi con questo browser, **è inutile fare virtuosismi per una situazione simile**. In caso contrario basterebbe eliminare il selettore “>” dalla regola precedente e aggiungere una dichiarazione font-size:100% alla regola legend scritta in precedenza.

Bisogna fare attenzione che **questa regola sia scritta nel CSS dopo quella che imposta il testo “Registrati” al 150%**, in quanto deve avere la precedenza.

CSS: rendiamo il form attraente

[Vedendo il risultato raggiunto](#), ti renderai sicuramente conto di come è cambiato il nostro form: **ha una forma, una solida base** su cui ora possiamo sbizzarrirci con colori, immagini e grafica.

Potremmo ad esempio aggiungere un bordino o addirittura **un doppio sfondo al form, sfruttando il tag form e il tag fieldset** (diverse tecniche CSS necessitano l'aggiunta di mark-up a questo scopo, ma in questo caso siamo avvantaggiati avendo già a disposizione due tag che descrivono lo

stesso spazio).

Con questo poco codice da aggiungere (o meglio, da implementare nel codice scritto fin'ora), il risultato è già diverso. [Vedi con i tuoi occhi](#).

Volendo, potremmo aggiungere qualche riga di CSS3 (non ancora supportati da tutti i browser) per arrotondare qualche angolo. Senza far ricorso a librerie javascript, **molti utenti vedranno comunque il form "spigoloso"...** ma **essendo solo un vezzo estetico** io non credo valga la pena **appesantire la pagina con del codice superfluo**. Se hai un browser che supporta gli angoli arrotondati CSS3 [puoi vedere qui il risultato](#).

Nota sui pulsanti: nel modificare le proprietà dell'elemento button ricordati di aggiungere sempre la dichiarazione "cursor:pointer;" altrimenti sarà visibile il puntatore standard del mouse, ovvero la consueta "freccetta".

Nota sui radio-button e check-box: esiste la possibilità di dare uno stile anche a questi due elementi, tramite librerie javascript, cambiando l'aspetto in maniera a mio parere troppo radicale, rendendone difficile il riconoscimento e quindi minando all'usabilità del form. Bisogna ricordarsi di togliere sempre i bordi associati ai vari input, perché altrimenti il bordi faranno da sgradevole cornice agli elementi in tutte le versioni di IE.

Considerazioni e anticipazioni

Ci è voluto più a dirlo che a farlo, non credi?

Il nostro form, **che era già accessibile dallo scorso articolo**, ora ha anche gli elementi posizionati in maniera piacevole e bilanciata. Abbiamo visto anche **le possibilità che abbiamo grazie ai CSS** per dargli un bell'aspetto generale con elementi decorativi e di contorno.

I problemi li abbiamo **riscontrati un po' con tutti i browser ma li abbiamo risolti (IE non si è comportato tanto peggio degli altri)**; il nostro codice è **standard, pulito e ottimizzato**: non abbiamo utilizzato **neanche una classe o un id** e abbiamo visto la **potenza dei selettori CSS** di tipo e discendenti.

Ma non finisce qui! Eh no... perché ora abbiamo solo **una bella statua** che non fa assolutamente nulla. La prossima volta vedremo **come implementare il controllo javascript di validazione sui campi**, in maniera accessibile ed usabile.

Tornando all'argomento di oggi... come è venuto fuori il tuo form? Avresti mai pensato di poter fare tutto questo **senza l'ausilio di div, class, id** e quant'altro?