

Cos'è il concetto di mantenibilità? E perché applicarlo?

Il concetto di mantenibilità, assolutamente irrinunciabile quando parliamo di programmazione, **deve trovare uno spazio importante anche nella realizzazione di un sito tradizionale**. Quando ci si ritrova (e ci si ritrova sempre prima o poi) a dover mettere le mani in un sito che abbiamo terminato sei mesi prima, potrebbe essere un vero e proprio incubo se non lo abbiamo progettato correttamente nella mantenibilità.

Di conseguenza potremmo perdere ore a cercare un file o a capire perché abbiamo scritto quella riga o dove agisce quella classe del foglio di stile.

Vale la pena dunque **prendersi il tempo per osservare alcune regole basilari** iniziando proprio da qualcosa che troppo spesso viene lasciato al caso: i files.

Perché dovrei definire i namespaces per i files?

Questo termine -rubato alla programmazione- significa semplicemente attribuire uno **standard uniforme ai nomi che utilizziamo**. Un esempio sbagliato potrebbe essere questo:

loginArea.html

customer_area.html

User-profile.html

Meglio scegliere un modello ed applicarlo a tutti i files.

login_area.html

customer_area.html

user_profile.html

Sempre parlando di files e namespace, pensa a come saranno visualizzati i files quando apri una cartella e fai in modo che **siano ordinati** come ti risulterà più comodo, faccio subito un esempio:

delete_user.php

add_user.php

modify_user.php

Nella mia logica (e sottolineo *mia* in quanto la mantenibilità è soggettiva, ognuno vi applica una sua logica personale) preferisco assegnare i nomi in questo modo:

user_delete.php

user_add.php

user_modify.php

Così quando aprirò la cartella che conterrà altri trecento files (mostrati in ordine alfabetico), saranno raggruppati per "tematiche". Come detto è una questione di preferenze; infatti qualcuno potrebbe obiettare che il primo modello è migliore se guardato da questa prospettiva:

add_user.php

add_page.php

add_section.php

È vero. L'importante è stabilire un criterio logico e mantenerlo in modo da rendere il più possibile **omogeneo il contenuto delle cartelle**. Ovviamente possiamo applicare lo stesso concetto anche per le immagini o per i file da scaricare. Ad esempio, nella cartella download non dovremmo trovare una situazione di questo genere:

catalogo_gennaio_2010.pdf

Catalogo_02_10.pdf

catalogo-mar-2010.pdf

La conseguenza, come negli altri casi, sarà una difficoltà a rintracciare i files. Pensate sempre che la cartella potrebbe diventare molto grande con il passare del tempo.

Perchè devo preoccuparmi di applicare una corretta indentazione?

Un codice bene indentato è molto leggibile e facilmente comprensibile:

Mentre pasticci di questo genere:

in una pagina magari lunga e complessa fanno perdere tempo in fase di aggiornamento/manutenzione.

Perchè dovrei perdere tempo a commentare il codice?

Inserire dei commenti nel codice che stiamo sviluppando, sia che si tratti di programmazione che di semplice markup (XHTML/CSS), è un'azione che troppo spesso viene trascurata. Inutile dire che, di questa dimenticanza, **paghiamo le conseguenze in fase di aggiornamento**.

Imponiamoci di commentare il codice. Lo so, quando stiamo realizzando un lavoro ci sembra tutto chiaro. Ma già dopo qualche settimana possiamo avere difficoltà ad individuare i punti sui quali agire. Commentiamo l'inizio e la fine delle varie sezioni. Soffermiamoci sui passaggi particolari, su quel frammento di codice che abbiamo messo per risolvere una compatibilità con Netscape 2.4.3.

Organizzare il foglio di stile

Nello sviluppo di layout moderni, dobbiamo spesso confrontarci con fogli di stile complessi; È dunque indispensabile organizzarli in modo molto ordinato. In questo ambito ci sono due scuole di pensiero: **ordine alfabetico** o **ordinamento per sezione** (header, navigazione, eccetera). Come ho già avuto modo di dire, non c'è un metodo giusto e un metodo sbagliato: l'importante è che voi vi riconosciate una logica e che essa venga rispettata sempre.

Inoltre, anche nei fogli di stile i commenti possono aiutarci a fare chiarezza ed a velocizzare un futuro intervento sugli stili del layout.

Prevediamo inoltre un'ottimizzazione finale. Se riguardiamo attentamente un foglio di stile, troveremo certamente delle righe ridondanti che possiamo eliminare, oppure un mancato sfruttamento delle caratteristiche di ereditarietà del CSS. Facciamo un esempio banale:

Questo codice è molto ridondante e non utilizza le caratteristiche dei CSS. Possiamo compattarlo in questo modo:

Una cosa che invece trovo scomoda (ma come sempre è personale) è il suddividere il foglio di stile in diversi file.

La navigazione: un elemento fondamentale che va studiato nella mantenibilità

In caso di aggiornamenti/aggiunte di contenuti, **un problema che si presenta frequentemente è quello della navigazione.**

Avevamo creato una bellissima navigazione orizzontale, prevista al massimo per sei voci. Ed ora arrivano decine di nuovi contenuti suddivisi in quattro nuove sezioni. Magari è quel cliente che quando ti ha contattato ti ha detto: "Ma sì, mi serve un sito con due o tre pagine. Sai, la presentazione, i contatti, e nella terza non so ancora bene cosa mettere. Non ti preoccupare.". E adesso siamo a 4TB di contenuti. È capitato a tutti almeno una volta. Quindi, dal momento che sappiamo che un sito è quasi sempre destinato a crescere (anche quando il cliente ci ha assicurato il contrario), quando è possibile cerchiamo di **progettare una navigazione estendibile e mantenibile.**

Immaginiamo già possibili sotto-menù, pensiamo a come potrebbe crescere e a come potrebbe essere modificata ed estesa senza perderci una settimana. E cosa fondamentale, separiamola dal contenuto (per favore non con i frame). Due righe di PHP bastano, e il loro utilizzo è relativamente semplice, anche se la programmazione non è il tuo forte.

In [questo articolo](#) trovi una spiegazione dettagliata di questa procedura.

Oltre alla navigazione, dovremo naturalmente immaginare anche la possibile crescita dei contenuti e di conseguenza come organizzarla in modo ordinato.

Se il sito non è gestito con un CMS, prevediamo delle sezioni ed eventualmente sottosezioni con relative cartelle e, come già detto precedentemente, l'applicazione dei namespace ai files sarà di grande aiuto.

Utilizzare le convenzioni e l'approccio semantico

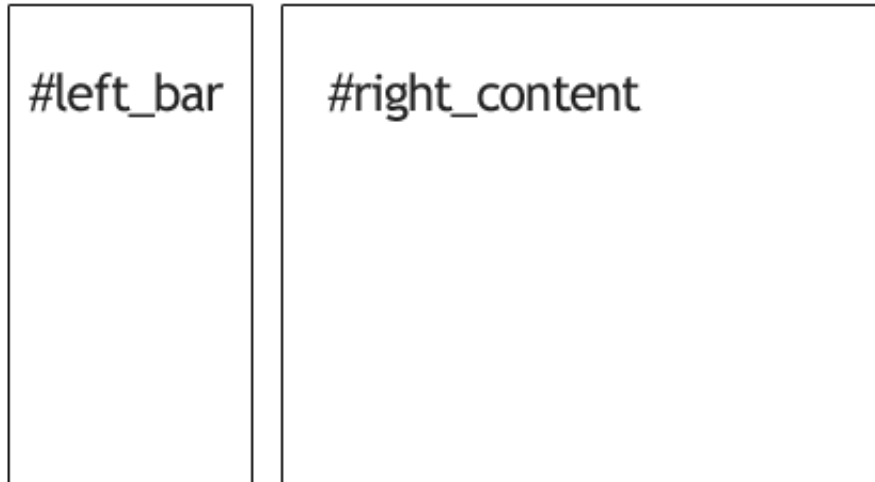
Io sono un programmatore, ambiente nel quale le convenzioni salvano la vita. La colonna di un database che contiene la chiave primaria, per convenzione si chiama "id". Posso chiamarla come voglio, posso chiamarla "*Ezechiele*", ma perché dovrei? Servirebbe solo a creare confusione.

Similmente, ad esempio, il div utilizzato per contenere i diversi elementi del documento, per convenzione si chiama "*container*" o "*wrapper*", quindi utilizziamo i nomi convenzionali, perché dovremmo inventarne altri?

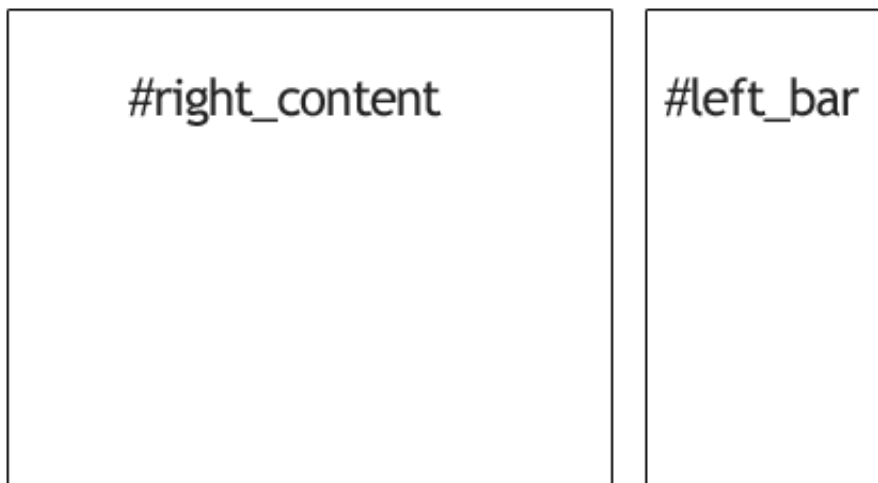
Sempre parlando di nomi arriviamo all'approccio semantico nel quale un esempio vale più di mille

parole:

Consideriamo questo schema di layout



Sembra andare benissimo. Ma cosa succederebbe se, in corso d'opera, decidessimo che la migliore impostazione fosse questa?



Ci troveremmo nella spiacevole situazione di dover modificare i nomi utilizzati nel foglio di stile. Con un approccio semantico invece non avremo nessun problema.

#main

#menu

Utilizzare i CMS

Quando i contenuti sono molti e/o gli aggiornamenti sono frequenti, considerare di basarsi su un CMS è d'obbligo. L'utilizzo di uno di questi software risolve in un colpo solo la maggior parte dei problemi di mantenibilità che abbiamo visto in questo articolo. Ma chiaramente anche in questo caso, oltre a scegliere il CMS giusto per le nostre esigenze, dovremo comunque prestare attenzione alla mantenibilità nella prima fase di impostazione.

Conclusione

Molto probabilmente ognuno di noi ha avuto modo di scontrarsi in più di un'occasione con problemi di mantenibilità; quanto esposto in questo articolo non costituisce di certo una grande novità ma i punti trattati possono comunque esserti utili per fare un po' di ordine durante la fase di progettazione del tuo prossimo sito.

E tu, quante volte ti sei maledetto per non aver tenuto conto della mantenibilità in un tuo sito?