

Come rendere dinamiche alcune sezioni di un sito web per ottimizzarne la manutenzione?

L'argomento che affronterò oggi potrebbe sembrarti banale e scontato ma a volte sono proprio le cose che consideriamo più ovvie a causarci enormi perdite di tempo: sono ancora in tanti, soprattutto chi è ancora alle prime armi, a commettere questo genere di mancanze.

Quando sei alle prese con **lo sviluppo un sito web** devi sempre prendere in considerazione il fatto che su quel sito tu, o chi per te, debba eseguirne **una manutenzione periodica**, anche banale, come potrebbe essere per esempio la modifica del nome di una voce del menù di navigazione (modifica che poi andrebbe apportata in tutte le pagine del sito in cui questa viene richiamata).

Hai mai pensato quanto tempo richiederebbe apportare una semplice modifica sul sito che stai sviluppando?

Se il sito non è sviluppato tenendo conto di questi accorgimenti anche una modifica apparentemente banale, come quella accennata in precedenza, potrebbe richiedere un'inutile perdita tempo, che ovviamente sarà proporzionata alle dimensioni del sito.

Pensa a un sito abbastanza grande composto da un centinaio di pagine che magari hai sviluppato qualche mese addietro. Il cliente adesso ti chiama per chiederti di modificare il semplice nome di una voce di menù la quale è richiamata in tutte le pagine del sito. **Hai idea di quanto tempo potresti perdere ad apportare questa semplice variazione?**

I problemi vanno eliminati sempre alla radice e per questo motivo è fondamentale prendere le **giuste precauzioni già in fase di sviluppo del sito** progettandolo con criterio e cognizione di quella che potrebbe essere in seguito una possibile attività di manutenzione.

A questo proposito ci vengono in aiuto i linguaggi di programmazione lato server, come il [php](#) per esempio, che permette di conferire dinamicità alle nostre pagine web.

Come ottimizzare il markup per rendere più semplice la manutenzione di un sito web?

Vediamo nel concreto come possiamo ottimizzare il markup di un sito web in fase di sviluppo e prendiamo come esempio il markup sviluppato per il [sito internet dell'asilo nido](#) che abbiamo realizzato nel corso della nostra guida "[Come realizzare un sito internet dalla A alla Z](#)".

Il Markup riguardante il menù di navigazione dell'asilo nido preso in considerazione è il seguente:

Questo markup andrebbe dunque ricopiato in tutte le altre pagine del sito nel punto in cui è richiamato il menù di navigazione.

Adesso, piuttosto che scrivere fisicamente questo markup su ciascuna pagina, **realizziamo un file php** che denominiamo “*function.php*” e sarà questo file a contenere il markup necessario a visualizzare il nostro menù di navigazione. Come?

Nel file “*function.php*” andiamo a creare una funzione in grado di produrre il markup interessato ogni qual volta questa sarà richiamata all’interno di una pagina.

Il codice della funzione che chiameremo “*nav()*” risulterà il seguente:

Non ho fatto altro che ricopiare il markup del menù di navigazione all’interno della funzione “*nav()*”.

Adesso basterà includere il file “*function.php*” all’inizio di ogni pagina del nostro sito web (per avere a disposizione su tutte le pagine la funzione “*nav()*”) e sostituire al markup originale che richiama il menù di navigazione proprio la funzione “*nav()*”.

Vediamo nel dettaglio come procedere. Con le righe di codice (evidenziate in grigio) sotto includiamo il file “*function.php*” all’interno delle nostre pagine:

Tramite la funzione php “*include_once*” stiamo inserendo tutto ciò che è contenuto nel file specificato (nel nostro caso “*function.php*”) all’interno della nostra pagina html.

Ovviamente le pagine del sito che conterranno questa funzione **andranno rinominate con estensione .php piuttosto che .htm** (il file “*index.htm*” diventerà “*index.php*”, ecc) in modo da permettere all’interprete php di riconoscere ed eseguire il codice php.

Fatto questo, possiamo eliminare, all’interno della nostra pagina, il markup che serviva a visualizzare il menù di navigazione e sostituirlo con la nostra funzione “*nav()*” creata in precedenza (che non fa altro che richiamare il markup interessato in modo dinamico), in questo semplice modo:

Quando l’interprete php processerà la nostra pagina, nel punto in cui è richiamata la funzione “*nav()*” verrà inserito in modo dinamico il markup html che abbiamo aggiunto all’interno della funzione.

In cosa consiste il vantaggio?

Con un sito sviluppato in questo modo, se il cliente ci dovesse chiedere di modificare una voce di menù del sito, esempio: “*eventi*” in “*news*”, basterà semplicemente aprire il file “*function.php*” e apportare una banale modifica alla funzione “*nav()*” cambiando la voce “*eventi*” con “*news*”.

Cinque secondi del nostro prezioso tempo contro alcuni minuti se avessimo progettato il sito in malo modo.

E se oltre al nome visualizzato avessimo dovuto cambiare anche il collegamento ipertestuale che puntava alla nuova pagina?

Sette secondi del nostro prezioso tempo contro diversi minuti se avessimo progettato il sito in malo modo.

E se il nostro sito fosse stato composto da un centinaio di pagine piuttosto che cinque pagine?

Sette secondi del nostro prezioso tempo contro diverse ore se avessimo progettato il sito in malo modo.

Conclusioni

E potrei andare avanti all’infinito, i casi delle modifiche banali da apportare potrebbero essere tantissimi, come per esempio aggiungere una ulteriore voce di menù (sempre che il layout grafico lo permetta), cambiare il collegamento ipertestuale a una pagina che è stata rinominata in funzione dell’ottimizzazione del sito per migliorarne l’indicizzazione sui motori, ecc.

Ti renderai conto da solo che **sono parecchie le sezioni di un sito che possono essere rese dinamiche**, come il footer, l’intestazione, il menù di navigazione e tutto ciò che generalmente è riprodotto con lo stesso markup in più pagine.

Prima di chiudere una piccola curiosità su cui confrontarci: **quali sezioni di un sito rendi generalmente dinamiche e con che risultati? E quale linguaggio di programmazione utilizzi?**