Come formattare e commentare il proprio codice?

È un dato di fatto: I programmatori sono persone concrete, di sostanza, che spesso preferiscono non approfondire questioni “futili” come aspetto, stile e formattazione del markup su cui lavorano.

In realtà la formattazione del codice fa si che quest’ultimo diventi più intuitivo - per te stesso e soprattutto per le persone con cui collabori - e l’utilizzo di commenti può aiutarti a comprendere le funzionalità del codice sviluppato anche a distanza di anni, rendendo di fatto più semplice la gestione dello stesso.

Ovviamente esistono diversi approcci nella formattazione del codice e se lavori da tempo nello sviluppo probabilmente avrai già perfezionato un tuo personale modo di lavorare e di gestire il markup. In questo articolo vedremo insieme qualche piccola *best practices* capace di ottimizzare il tuo lavoro e di renderlo più semplice e comprensibile.

Prima di tutto, occupiamoci della formattazione. Migliorare la leggibilità del codice è molto importante e per farlo possiamo avvalerci di alcune semplici linee guida. Niente di particolarmente difficile, basta seguire alcuni piccoli accorgimenti durante il processo di sviluppo.

Indentazione

Una delle prime linee guida riguarda sicuramente l’indentazione del codice.
Indentare il codice in modo corretto, **con qualsiasi linguaggio si lavori**, è una delle pratiche essenziali per rendere il codice visivamente più comprensibile.

Vediamo un esempio:

Come puoi vedere, in questo modo comprendere il codice che abbiamo analizzato è piuttosto difficile. Basterebbe indentarlo correttamente per renderlo molto più leggibile:

Per l’indentazione è consigliabile utilizzare gli spazi al posto delle tabulazioni così da aumentare la portabilità su sistemi e computer differenti. Qualunque editor ha un’impostazione che permette di sostituire il carattere di tabulazione con un certo numero di spazi, di solito 4 o 2.

Parentesi graffe

Esistono vari stili per l’utilizzo delle parentesi graffe, scegli tu quello che preferisci, non fa differenza per la leggibilità. I seguenti sono i più comunemente utilizzati:

- Le parentesi vengono allineate verticalmente:

- La parentesi di apertura viene allineata alla riga di codice a cui fa riferimento, quella di chiusura va su una nuova linea:

- Le parentesi vengono allineate verticalmente ed indentate:

Ne approfitto per segnalarti questa pagina di Wikipedia in cui puoi trovare molti stili di utilizzo delle parentesi graffe e la loro storia.

Variabili

Le variabili nel codice dovrebbero avere sempre un nome informativo, che faccia capire al volo a cosa servono o cosa contengono:

Da questo codice non si capisce a cosa realmente serva la variabile $var. Ok, sto facendo un controllo, ma di cosa? Un nome, un username, una password?

Decisamente meglio ora, non credi?

Attento però. Non abusare di questa regola, utilizzandola dove palesemente inutile: in un ciclo *for*, il contatore non ha bisogno di chiamarsi ***$contatoreDelMioCiclo***, basta un semplice ***$i***.

Scegli anche un metodo per nominare le variabili.

1. *CamelCase*: capitalizzare l'iniziale di ogni parola;
2. Caratteri minuscoli ed uso degli underscore (_) per dividere le parole;

Spazi bianchi

Utilizza gli spazi bianchi dove possono migliorare la leggibilità del codice.
Io personalmente li utilizzo:

- prima e dopo ogni parentesi;
- prima e dopo il carattere di unione (. nel caso di PHP o ad esempio + nel caso di Javascript);
- dopo ogni virgola;
- prima e dopo ogni carattere matematico o di confronto

Gli spazi bianchi comprendono anche le linee vuote e i ritorni a capo.
È corretto lasciare una linea vuota dopo la dichiarazione di una condizione, di un ciclo o di una funzione e prima della rispettiva chiusura:

È meglio scrivere su più righe la dichiarazione di un array con più di un elemento o la dichiarazione di una funzione, istruzione condizionale o ciclo con più condizioni:

Commenti

Non mi stancherò mai di ripeterlo, **commenta il tuo codice**, e fallo per due semplici motivi:

1. Quando andrai a rileggerlo dopo qualche anno, in pochi minuti saprai cosa fa e come funziona;
2. Altri programmatori che dovranno per un motivo o per un altro lavorare sul tuo codice non avranno problemi a comprenderlo.

Se devi modificare uno script scritto precedentemente, aggiorna anche i commenti relativi alle porzioni di codice modificate. Questo perché in futuro potresti trovarti a leggere informazioni errate e potresti fare modifiche che invece di migliorare il markup lo peggioreranno.

In ogni linguaggio esistono più stili per commentare il codice:

- Commenti su riga singola
- Commenti su più righe

Quando utilizzare l'uno o l'altro?

Il commento su riga singola va utilizzato per brevi annotazioni o descrizioni.

Come noti dall'ultima riga di codice, è molto comodo utilizzare i commenti a riga singola anche in corrispondenza delle parentesi di chiusura, per indicare a quale istruzione essi appartengono.

I commenti a riga multipla invece andrebbero utilizzati all'inizio di ogni file per indicare cosa contiene, prima di ogni dichiarazione di classe o funzione per indicare i parametri che accetta, il tipo di valore restituito, la sua descrizione ecc...

Per quanto mi riguarda utilizzo le linee guida di DocBlox per i commenti per due motivi:

1. Sono estremamente esplicativi;
2. Tramite quel software e i soli commenti, si può generare la documentazione del proprio codice.

Ricorda di **commentare il codice man mano che lo scrivete**: dopo sarà sicuramente troppo da commentare e inizierete a rimandare e rimandare e rimandare...

Ti segnalo infine gli standard di scrittura di Wordpress che trovo essere delle ottime linee guida da seguire.

Conclusioni

Con questo articolo non voglio costringerti a cambiare il tuo stile di programmazione, ma aiutarti a capire che è meglio scrivere correttamente il proprio codice, anche se all'inizio può essere noioso stare attenti all'indentazione, ad un certo stile, ai commenti ecc. Del resto quando inizierai a scrivere progetti con script di molte pagine e numerose classi, ti accorgerai che diventa assolutamente necessario saper ottimizzare il markup nel modo più usabile possibile.

E poi diciamoci la verità, è una grande soddisfazione completare uno script e ammirare la sua bellezza con tutti i commenti che lo accompagnano, le parentesi tutte al loro posto, le variabili esplicative...come si suol dire in certi casi, "*Code is poetry*", il codice è poesia, perciò tanto vale assicurarsi che la forma renda il più gradevole possibile la sostanza.

Sei d'accordo?