

Introduzione

Il World Wide Web, quel meraviglioso luogo in cui possiamo compiere qualsiasi cosa ci passi per la testa – dalla semplice navigazione di un sito di notizie all'utilizzo di una complessa applicazione di e-commerce – sta, come di consueto fanno tutte le tecnologie moderne, cambiando, e tale mutamento sta avvenendo a una velocità ancora più elevata del passato.

Ciò è dovuto all'impulso e alla spinta, da un lato, da parte degli organismi che si occupano di standardizzare le tecnologie web attraverso la scrittura e definizione di documenti di specifiche (*in primis* il W3C di Timothy John Berners-Lee ma anche il WHATWG Group che sta dietro a HTML5), e dall'altro lato, da parte degli sviluppatori dei moderni browser che cercano in tutti i modi di aggiornare tali *user agent* in modo da poterli dotare delle nuove e potenti caratteristiche definite all'interno delle specifiche.

Questa continua ricerca e innovazione è fondamentale e produttiva sia per il semplice utente che usa il browser come mero mezzo di navigazione, sia per noi sviluppatori che possiamo utilizzare i nuovi strumenti e le relative API per scrivere applicazioni web sofisticate prima impensabili (player multimediali, carrelli della spesa con funzionalità *drag and drop*, programmi per il disegno, videogame e così via), dotate quindi di tutte quelle moderne caratteristiche che si trovano nelle consuete applicazioni *standalone* che girano sul desktop dei vari sistemi operativi.

Il browser è assurto da semplice strumento per la navigazione dei contenuti a una sorta di mini sistema operativo al cui interno eseguire le applicazioni web moderne.

Tutto questo ha un prezzo da pagare che è quantificabile in un cambiamento di mentalità nell'approccio allo sviluppo di applicazioni e nel tempo che è necessario per imparare tutte le tecnologie principali che oggi ruotano attorno al Web.

Per quanto concerne il cambiamento di mentalità, uno sviluppatore moderno dovrebbe applicare, laddove possibile, le seguenti regole.

- Non legarsi più a implementazioni proprietarie delle tecnologie web ma guardare e utilizzare solo quelle standard che consentono di dotare le applicazioni di *feature* che nel momento in cui saranno implementate nei browser saranno tra di essi interoperabili.
- Strutturare il codice a *layer*, ossia separare nettamente la parte strutturale (HTML) dalla parte di presentazione (CSS) e dalla parte del codice (JavaScript). Ciò consente, infatti, di programmare le applicazioni in modo professionale e scalabile evitando, per esempio, di scrivere codice JavaScript o regole CSS direttamente nei tag HTML.

- Utilizzare le tecnologie lato server (PHP, Java EE, ASP.NET...) per il loro scopo primario (la gestione della sicurezza, dei dati e così via) e non per compiere operazioni comunemente effettuabili lato client come, per esempio, quelle inerenti la parte di interazione (quante applicazioni web ci sono ancora dove un menu a tendina viene popolato dopo un *postback* su un server?).

Partendo da queste considerazioni, questo libro è stato strutturato in modo che l'apprendimento risulti essere graduale e sistematico: i concetti sono esposti in modo che, per esempio, nella parte su HTML5 non è presente alcun riferimento a CSS3 o JavaScript. Inoltre, non sono trattate API proprietarie ma solo quelle degli standard web che sono state implementate, nel momento in cui questo libro viene scritto, almeno da un browser. Una considerazione è d'obbligo: data la mutabilità degli standard che non hanno ancora raggiunto la versione definitiva, alcune delle proprietà, dei metodi, degli attributi di HTML5, CSS3 e JavaScript potrebbero subire cambiamenti a cui in queste pagine non è possibile dare seguito. In ogni caso ciò non deve spaventare perché è una conseguenza normale e ravvisabile in tutte quelle tecnologie, come quelle legate al Web, ad alta velocità di innovazione e cambiamento.

ATTENZIONE

I linguaggi HTML5 e CSS3 non sono in tutto e per tutto nuove tecnologie. Essi si rifanno alle specifiche che li hanno preceduti, ovvero HTML4.01 e CSS2. Non potrebbe del resto essere altrimenti: solo in questo modo è possibile sviluppare nuovi e più potenti standard garantendo allo stesso tempo la retrocompatibilità con quanto fino a oggi in uso. Il lettore non deve quindi stupirsi di ritrovare nelle prossime pagine anche costrutti a lui già noti, documentati insieme ad altri in tutto e per tutto nuovi. Per esempio, l'elemento `p` è parte di HTML5 quanto `canvas`. Questo libro è stato concepito per offrire uno spaccato il più completo e pratico possibile di HTML5 e CSS3, evitando di limitarsi alle funzionalità più innovative. In queste pagine il lettore troverà quindi una trattazione che parte dalle basi e potrà approcciare lo studio di queste tecnologie anche senza conoscere HTML4.01 e CSS2.

Una nota finale è necessaria per quanto riguarda HTML5 e CSS3. Lo scopo di questo libro è quello di documentare tali standard a prescindere dall'ampiezza di supporto dei browser. Tranne dove diversamente indicato, gli esempi di codice riportati sono stati testati con il browser Mozilla Firefox perché è quello che, più degli altri, implementa con attenzione gli standard web. Questo non significa però che HTML5 e CSS3 non possano funzionare con browser diversi da Firefox. Per tenersi aggiornati sul supporto che i browser offrono a HTML5 e CSS3 si può fare riferimento a fonti come <http://caniuse.com/> oppure <http://www.quirksmode.org/dom/html5.html> (per HTML5) e <http://www.quirksmode.org/css/contents.html> (per CSS3), senza dimenticare su Wikipedia [http://en.wikipedia.org/wiki/Comparison_of_layout_engines_\(HTML5\)](http://en.wikipedia.org/wiki/Comparison_of_layout_engines_(HTML5)) e [http://en.wikipedia.org/wiki/Comparison_of_layout_engines_\(Cascading_Style_Sheets\)](http://en.wikipedia.org/wiki/Comparison_of_layout_engines_(Cascading_Style_Sheets)).

Organizzazione del libro

Il libro è organizzato nei capitoli elencati di seguito.

- Capitolo 1, “Il layer strutturale: l'HTML5 ”: introduciamo i concetti di base e gli elementi fondamentali per la strutturazione di una pagina web con HTML5. Trat-

tiamo infatti del testo, delle liste, delle tabelle, degli URL, delle immagini e così via fino a esaminare nel dettaglio nuovi tipi di widget inseribili nei form e i nuovi elementi semantici.

- Capitolo 2, “Il layer di presentazione: CSS3”: mostriamo in modo approfondito come utilizzare i fogli di stile, elemento imprescindibile per la definizione di regole di visualizzazione degli elementi HTML. Illustriamo tutte le tipologie di selettori e pseudo-classi e spieghiamo gli importanti concetti di organizzazione a cascata, ereditarietà e box model. Infine, passiamo in rassegna tutte le proprietà utilizzabili come quelle per i font, per il testo, per le tabelle, per il box model e così via.
- Capitolo 3, “Il layer di interazione: JavaScript”: esaminiamo il versatile e potente linguaggio di programmazione JavaScript, “motore” delle moderne applicazioni web, trattando sia i costrutti principali come le variabili, gli array, gli operatori, le strutture di controllo e altro, sia quelli più complessi come le funzioni, le *closure*, gli oggetti, i prototipi, l’ereditarietà, il polimorfismo e così via. Infine, sottolineiamo che gli argomenti sono tutti aggiornati alle nuove caratteristiche dello standard di riferimento del linguaggio, ossia alla versione ECMAScript 5 del giugno 2011.
- Capitolo 4, “Programmazione lato client”: dopo aver studiato i “mattoni” fondamentali per la costruzione di un’applicazione web, entriamo nel vivo della programmazione lato client analizzando le API del BOM (*Browser Object Model*), del DOM (*Document Object Model*), della gestione degli eventi e dello scripting dei CSS e di Ajax. Chiudiamo il capitolo con un’introduzione a jQuery, una potente libreria JavaScript che è diventata lo standard *de facto* per la scrittura di applicazioni web moderne e potenti.
- Capitolo 5, “Applicazioni web”: impariamo a utilizzare degli elementi HTML e delle API straordinarie e innovative che consentono di implementare applicazioni web che possono “mimare”, per caratteristiche e complessità, quelle scritte normalmente per gli ambienti desktop. Analizziamo infatti l’elemento *canvas*, gli elementi multimediali *audio* e *video*, il *drag and drop*, la geolocalizzazione, i *Web Workers* (i thread), il *Web Storage*, le applicazioni web offline e i microdata.

Convenzioni utilizzate

Il libro è diviso in capitoli. Ogni capitolo è numerato in ordine progressivo e denominato significativamente nel suo obiettivo didattico (per esempio Capitolo 5, “Applicazioni web”). I capitoli sono poi suddivisi in paragrafi, all’interno dei quali possiamo avere dei blocchi di testo o di immagini, ausiliari alla teoria e denominati come segue.

- **Listato NrCapitolo.NrProgressivo Descrizione** per i listati del codice sorgente.
- **Snippet NrCapitolo.NrProgressivo Descrizione** per i frammenti di codice sorgente.
- **Sintassi NrCapitolo.NrProgressivo Descrizione** per la sintassi di un costrutto.
- **Output NrCapitolo.NrProgressivo Descrizione** per l’output di un frammento di codice.
- **Figura NrCapitolo.NrProgressivo Descrizione** per una figura in generale ma anche per l’output dei listati delle applicazioni web.
- **Tabella NrCapitolo.NrProgressivo Descrizione** per una tabella.

Per esempio, il blocco denominato *Listato 1.8, File ElementiStrutturali.html* indica il listato di codice numero 8 del Capitolo 1 avente come descrizione il file `ElementiStrutturali.html`. Infine, per quanto attiene ai listati, i punti di sospensione (...) indicano che, in quel passaggio, alcune parti del listato sono state omesse. Le medesime parti sono presenti nei relativi file allegati al libro.

Codice sorgente e annotazioni finali

All'indirizzo <http://www.apogeeonline.com/libri/9788850331314/scheda> è possibile scaricare un archivio ZIP contenente i listati di ogni capitolo e tutte le relative risorse (immagini, video e così via).

ATTENZIONE

La maggior parte dei listati e degli snippet di codice sono stati testati con il browser Firefox della Mozilla Organization perché è quello che, più degli altri, ha implementato e implementa con certissima attenzione gli standard delle tecnologie web. In altri casi si è invece fatto riferimento ad altro browser (come Chrome e Opera) per evidenziare come l'applicazione di una particolare proprietà, elemento o script, vengano da essi gestite e interpretate. Il browser utilizzato è di norma specificato nella didascalia di riferimento.