

Marco Avvenuti
Mario G.C.A. Cimino

Laboratori di

programmazione Web

HTML, CSS, Javascript e PHP

web
site 

McGraw-Hill

web
site 

www.mcgraw-hill.it
www.ateneonline.it/avvenuti

Laboratori di

programmazione Web

HTML, CSS, Javascript e PHP

**Una collezione di esercitazioni
di laboratorio per insegnare
a progettare e sviluppare
un'applicazione Web**

Marco Avvenuti è professore associato di Sistemi di elaborazione delle informazioni presso la Facoltà di Ingegneria dell'Università degli Studi di Pisa.

Mario G.C.A. Cimino è assegnista di ricerca presso il Dipartimento di Ingegneria dell'Informazione dell'Università degli Studi di Pisa.

€ 24,00 (i.i.)

ISBN 978-88-386-6293-5



9 788838 662935

Copyright © 2011 The McGraw-Hill Companies, S.r.l.
Publishing Group Italia
Via Ripamonti, 89 – 20139 Milano

McGraw-Hill



A Division of the McGraw-Hill Companies

I diritti di traduzione, di riproduzione, di memorizzazione elettronica e di adattamento totale e parziale con qualsiasi mezzo (compresi i microfilm e le copie fotostatiche) sono riservati per tutti i Paesi.

Nomi e marchi citati nel testo sono generalmente depositati o registrati dalle rispettive case produttrici.

Le fotocopie *per uso personale* del lettore possono essere effettuate nei limiti del 15% di ciascun volume/fascicolo di periodico dietro pagamento alla SIAE del compenso previsto dall'art. 68, commi 4 e 5, della legge 22 aprile 1941 n. 633.

Le riproduzioni effettuate per finalità di carattere professionale, economico o commerciale o comunque *per uso diverso da quello personale* possono essere effettuate a seguito di specifica autorizzazione rilasciata da AIDRO, Corso di Porta Romana 108, 20122 Milano, e-mail segreteria@aidro.org, sito web www.aidro.org.

Publisher: Paolo Roncoroni
Development Editor: Filippo Aroffo
Produzione: Donatella Giuliani
Sviluppo Web: Daniela Cipollone
Realizzazione editoriale: a cura degli Autori
Grafica di copertina: FeellItalia, Milano
Stampa: Cromografica Europea, Rho (MI)

ISBN 978-88-386-6293-5

Printed in Italy
123456789CROCRO54321

Marco Avvenuti
Mario G.C.A. Cimino

Laboratori di programmazione Web

HTML, CSS, Javascript e PHP

McGraw-Hill

Milano • New York • San Francisco • Washington D.C. • Auckland
Bogotá • Lisboa • London • Madrid • Mexico City • Montreal
New Delhi • San Juan • Singapore • Sydney • Tokyo • Toronto

Prefazione

Il successo del Web segna una pietra miliare nella storia delle comunicazioni e contribuisce alla diffusione capillare di Internet. Nato ufficialmente il 6 agosto 1991, quando nella rete viene pubblicato il primo sito, conta a oggi centinaia di milioni di siti attivi, decine di miliardi di pagine pubblicamente accessibili e più di un miliardo di utenti. L'approccio aperto e neutrale rende l'impatto sociale del Web più pervasivo e dinamico, e per certi versi antagonista, dei media tradizionali: una fonte di documenti collegati tra loro, percepita come inesauribile, è ora facilmente accessibile tramite una rete globale.

Gli "ingredienti" che hanno decretato questo risultato sono: un protocollo di pubblico dominio (HTTP), che permette di "navigare" e "scaricare" le pagine indipendentemente dal loro contenuto o locazione, così come di aggiungere nuove "risorse" informative senza chiedere il permesso ad alcuna autorità centrale; uno schema uniforme per identificare e localizzare le risorse (URI o URL); il linguaggio HTML, con il quale è possibile scrivere le pagine come testo in chiaro, accessibili nella loro struttura oltre che nel contenuto, usando un qualunque editor di testo.

Il linguaggio HTML è oggi concepito per definire il contenuto logico e non l'aspetto del documento. Questa distinzione è importante perché, per garantire la massima diffusione e accessibilità dei contenuti, è opportuno che i documenti siano destinati a diversi tipi di terminale. Non sempre questi sono dotati di adeguate capacità di presentazione grafica; possono anzi averne di limitate o non averne affatto (se per esempio sono obsoleti o sono terminali mobili o sono adoperati da persone diversamente abili), o magari sono dispositivi che hanno necessità di estrarre informazioni dal documento senza visualizzarlo (per esempio, per creare indici nei motori di ricerca o per importare documenti HTML in un formato proprietario). Questo significa che, a seconda del tipo di terminale, lo stesso documento deve avere diverse modalità di presentazione, definite in un documento complementare scritto in un linguaggio specifico (CSS).

Inizialmente le pagine HTML erano solo statiche, vale a dire file con un determinato contenuto informativo arricchito da informazioni di struttura. Nonostante le limitazioni delle pagine statiche, la semplicità della tecnologia ne decretò un immediato successo, facendo nascere la cosiddetta era del Web. Successivamente, le pagine furono rese dinamiche nei contenuti introducendo meccanismi, come l'interfaccia CGI, in grado di consentire al server di invocare specifiche applicazioni per la produzione "on-demand" di contenuti. Per esempio, con i dati provenienti da un database interrogato in accordo a modalità definite dall'utente.

Per fornire maggiore interattività e dinamicità alle pagine, da un lato i browser furono potenziati tramite linguaggi di scripting (Javascript) in grado di modificare le pagine direttamente dal lato client, dall'altro nacquero linguaggi di programmazione specifici per la produzione dinamica di pagine dal lato server (tra i più diffusi abbiamo oggi JSP, ASP.NET e PHP), trasformando i Web server da semplici archivi di pagine a motori di produzione delle medesime (application server).

La complessa evoluzione degli standard del Web è stata da sempre guidata dal W3C, un organismo composto dalle principali aziende informatiche, società, associazioni, università e istituti di ricerca al mondo. Sin dagli inizi il W3C ha cercato di contrastare la progettazione di siti ad-hoc per terminali con specifiche risoluzioni (le etichette "best viewed with 800×600 screen resolution") o per specifici browser ("Best viewed by ? browser") e qualsiasi schema di progetto che privilegiasse alcune tipologie di terminale. Un'altra questione che da sempre ha costituito un ostacolo alle potenzialità del Web è la cosiddetta "guerra dei browser", ossia il conflitto commerciale e di immagine tra diversi produttori, che tentano di imporsi sul mercato differenziandosi e implementando diverse concezioni degli standard. Ancora oggi vi sono dei costrutti che occorre scrivere in più versioni, perché non esiste una sola forma sintattica valida per tutti i principali browser.

In un contesto dalle esigenze sempre crescenti, l'attività del W3C è stata caratterizzata da continue evoluzioni della struttura dei linguaggi. Per garantire compatibilità all'indietro, le successive versioni dei linguaggi mantengono molti dei costrutti precedenti etichettandoli come "disapprovati" ma al contempo validi all'interno di uno schema "di transizione". Di conseguenza, i browser sono diventati sempre più "toleranti", al fine di poter visualizzare pagine scritte in versioni diverse e contemporaneamente di comprendere estensioni, specifiche per ogni browser e anticipatrici rispetto agli standard. Questo ha comportato problemi di sicurezza, che si sono protratti anche dopo la formulazione di un modello per le restrizioni dello scripting lato client. L'evoluzione dell'HTML culminò nel dicembre 1999 con una versione "stabile" e ancora oggi di riferimento, la 4.01. L'evoluzione parallela del CSS portò nel 2004 a emanare la specifica CSS 2.1, ancora oggi lo standard di riferimento.

Tutti i principali browser sanno interpretare oggi l'HTML 4.01 e il CSS 2.1 in modo abbastanza uniforme. Ma per poter visualizzare anche i contenuti pubblicati in passato, essi sono in grado di visualizzare documenti non corretti secondo tali standard, e addirittura documenti non corretti secondo nessuno standard. Tuttavia, poiché non tutti i browser riparano i documenti nello stesso modo, un documento non valido può mostrarsi e avere un comportamento diverso su browser differenti. Il W3C conduce da tempo un'eccellente campagna di "bonifica" volta alla produzione di pagine formalmente corrette, attraverso la diffusione di servizi di validazione e di targhette di riconoscimento delle pagine valide.

Sul piano delle piattaforme di scripting è ancora in atto una competizione sui diversi linguaggi e sui relativi ambienti di sviluppo e di esecuzione, per cui oggi una qualsiasi soluzione web-based di una certa dimensione coinvolge l'accesso a piattaforme ibride. Questo si traduce nella necessità di dover sviluppare adoperando diverse sintassi, di sapersi orientare nella consultazione di diverse specifiche e di saper programmare secondo uno stile facilmente portabile da un linguaggio all'altro. È quindi

di importanza fondamentale per il progettista Web conoscere paradigmi quali l'Object-Oriented (OO), il Document Object Model (DOM), l'architettura three-tier, il pattern model-view-controller (MVC), che consentono di vedere i sistemi a un livello di astrazione superiore, ponendo in secondo piano la sintassi e mettendo in condizioni di identificare le analogie tra i diversi linguaggi (per esempio, JavaScript e VBScript; ASP.NET, JSP e PHP).

Metodologia

L'obiettivo di quest'opera è quello di fornire una corretta metodologia e un'ideale collezione di esercitazioni di laboratorio per insegnare a progettare e sviluppare un'applicazione Web completa secondo gli standard HTML, CSS, Javascript e PHP. Attraverso la specifica e la risoluzione di nove progetti didattici, lo studente viene progressivamente guidato a comprendere, modificare ed estendere un'applicazione completa, dal lato client al lato server. Il volume contiene anche un'ampia raccolta di FAQ sulla risoluzione dei problemi di progettazione più ricorrenti, e in capitolo con le linee guida per aiutare sia lo studente sia il docente nella definizione e valutazione di progetti didattici.

Il metodo prevede che il lettore conosca già gli aspetti fondamentali dei linguaggi. Si è ritenuto opportuno non appesantire l'opera con la descrizione e la specifica delle varie tecnologie, in quanto queste possono essere facilmente reperite nella copiosa documentazione esistente in rete. Viene comunque fornita una guida ai siti più significativi e alle specifiche del consorzio W3C. Il percorso pedagogico si articola in nove progetti da realizzare in laboratorio, per ciascuno dei quali viene definita una specifica dei contenuti e delle funzioni che si vogliono ottenere, seguita dal codice che implementa una possibile soluzione. La specifica, pur espressa in linguaggio naturale, segue uno stile e un formato tipici del mondo professionale. In più, contiene suggerimenti e brevi approfondimenti degli aspetti tecnologici utili per la realizzazione del progetto. Nelle soluzioni proposte vengono giustificate le scelte implementative più critiche. Nell'ottica di fornire una raccolta di soluzioni ai problemi più ricorrenti, ma soprattutto per abituare lo studente all'auto-apprendimento continuo nel campo delle tecnologie trattate, l'opera si conclude con un capitolo organizzato in FAQ, nel quale vengono presi in considerazione singoli aspetti realizzativi e ne vengono proposte le soluzioni secondo il rigore richiesto dagli standard professionali.

Destinatari

Il materiale di questo testo nasce dall'esperienza maturata in diversi anni di insegnamento delle tecnologie Web presso la facoltà di Ingegneria dell'Università di Pisa. Trattando le tecnologie di base, è principalmente destinato a studenti e docenti di un corso di laurea triennale, sia nelle discipline scientifiche sia in quelle umanistiche, dove sia previsto un insegnamento sulla progettazione Web (Informatica, Ingegneria Informatica, Informatica Umanistica ecc.). Ma si rivolge anche a tutte quelle figure professionali che richiedono la capacità di produrre e gestire contenuti Web.

*Marco Avvenuti
Mario G.C.A. Cimino*

Indice generale

Parte 1 Laboratori		1
Capitolo 1	Il linguaggio HTML	3
1.1	Obiettivo	3
1.2	Prerequisiti	3
1.3	Strumenti e specifiche	3
1.4	Traccia	4
1.5	Soluzione proposta	8
Capitolo 2	Cascading Style Sheets (CSS)	15
2.1	Obiettivo	15
2.2	Prerequisiti	15
2.3	Strumenti e specifiche	15
2.4	Traccia	15
2.5	Soluzione proposta	20
Capitolo 3	Javascript Core (ECMA-262)	25
3.1	Obiettivo	25
3.2	Prerequisiti	25
3.3	Strumenti e specifiche	25
3.4	Traccia	25
3.5	Suggerimenti	28
3.6	Soluzione proposta	28
3.7	Espressioni regolari	32
Capitolo 4	Client-Side Javascript	33
4.1	Obiettivo	33
4.2	Prerequisiti	33
4.3	Traccia	33
4.4	Suggerimenti	36
4.5	Esecuzione dei programmi Javascript	36
4.6	Soluzione proposta	37
4.7	Cenni sulla sicurezza in Javascript	42
Capitolo 5	Il Document Object Model (DOM) in Javascript	45
5.1	Obiettivo	45
5.2	Prerequisiti	45
5.3	Strumenti e specifiche	45

5.4	Traccia	46
5.5	Suggerimenti	48
5.6	Organizzazione delle API DOM e cenni di DHTML	49
5.7	Soluzione proposta	51
5.8	Bookmarklet	55
Capitolo 6	CSS e Dynamic HTML (DHTML)	57
6.1	Obiettivo	57
6.2	Prerequisiti	57
6.3	Strumenti e specifiche	57
6.4	Traccia	57
6.5	Suggerimenti	59
6.6	Cenni di programmazione a eventi	60
6.7	Cenni di tecniche di compatibilità	61
6.8	Soluzione proposta	63
Capitolo 7	Architetture web a tre strati	69
7.1	Obiettivo	69
7.2	Prerequisiti	69
7.3	Strumenti e specifiche	69
7.4	Traccia	69
7.5	Soluzione proposta	71
Capitolo 8	PHP Core	81
8.1	Obiettivo	81
8.2	Prerequisiti	81
8.3	Strumenti e specifiche	81
8.4	Traccia	81
8.5	Suggerimenti	84
8.6	Soluzione proposta	84
Capitolo 9	Database-Driven Web Systems	91
9.1	Obiettivo	91
9.2	Prerequisiti	91
9.3	Strumenti e specifiche	91
9.4	Traccia	91
9.5	Soluzione proposta	93
9.6	Iniezione di codice SQL	100
Parte 2 FAQ		103
Capitolo 10	HTML	105
10.1	Entità HTML e Charset ISO 8859-1	105
10.2	Link a file di vario tipo, MIME type	108
10.3	Inclusione strict di componenti Applet e Flash	111
10.4	Formati immagine: GIF, JPEG e PNG	112
10.5	Formati documentali: DOC, TEX, PS, PDF	113
10.6	Formati multimediali: WAV, MIDI, MP3, MPEG e DIVX	114

10.7	Copia di un sito	115
10.8	Interlinea, spaziatura, non breaking space	115
10.9	Frame noborder, noscroll, noresize	117
Capitolo 11	CSS	119
11.1	Disposizione campi inserimento senza tabelle	119
11.2	Layout “a tabella”, ma senza tabelle	121
11.3	Layout frame, ma senza frame. Il posizionamento in CSS	123
11.4	Text area con contenuto indentato	128
11.5	Tabella centrata	129
11.6	Simulatore palmare	130
11.7	Cambiare l’immagine del cursore	131
11.8	Scacchiera di elementi congiunti	132
11.9	Box Model	133
11.10	Tabella di elementi congiunti	135
Capitolo 12	Javascript	137
12.1	Passaggio per valore e riferimento	137
12.2	Proteggere indirizzi email da spider	138
12.3	Espressioni regolari	140
12.4	Valori iniziali e correnti in form	142
12.5	Invocare funzioni altro frame	145
12.6	Eventi del mouse	147
12.7	Tabella con righe dinamicamente modificabili	148
12.8	Individuare un aggregato di oggetti eterogenei tramite API DOM	150
12.9	Muovere immagini generate dinamicamente	152
12.10	Estrazione del codice generato da Javascript	154
12.11	API specifiche di Internet Explorer e Mozilla Firefox	155
12.12	Coordinate mouse	156
12.13	Menu creato dinamicamente	157
12.14	Drag&Drop	159
12.15	Accesso a file XML	161
12.16	Ordinamento di oggetti Javascript	164
12.17	Controllare midi da Javascript	167
12.18	Timer multipli e Mozilla Firefox	169
12.19	Proprietà di un elemento immagine	171
12.20	Manipolare una pagina proveniente da un dominio esterno	172
12.21	Contenuto scorrevole attraverso una finestra	174
12.22	Finestra popup	175
12.23	Accedere a finestra principale da una finestra popup	176
12.24	Scorrere la window tramite Javascript	177
12.25	Gestione di un array di oggetti	179
12.26	Gestione delle date	180
12.27	Composizione automatica di un messaggio di posta	182
12.28	Passaggio di parametri con gestore di evento dinamico	184
12.29	Controllore generico per la verifica di campi in moduli	185

Capitolo 13	PHP	189
13.1	Sito oscurato da hacker	189
13.2	Iniezione html	190
13.3	Passaggio variabili in PHP	191
13.4	Gestione file	193
13.5	Includere frammenti esterni	194
13.6	HTTP GET	195
13.7	HTTP POST	196
13.8	File upload	198
13.9	Invio di dati intermedi tramite tecnologia Ajax	199
13.10	Elaborazione in Javascript di dati generati tramite PHP	201
13.11	API DOM in PHP	203
13.12	Funzioni PHP per richieste HTTP POST	204
Appendice	Metodologia di sviluppo di progetti didattici	207
	Indice dei siti Web	213
	Indice analitico	215