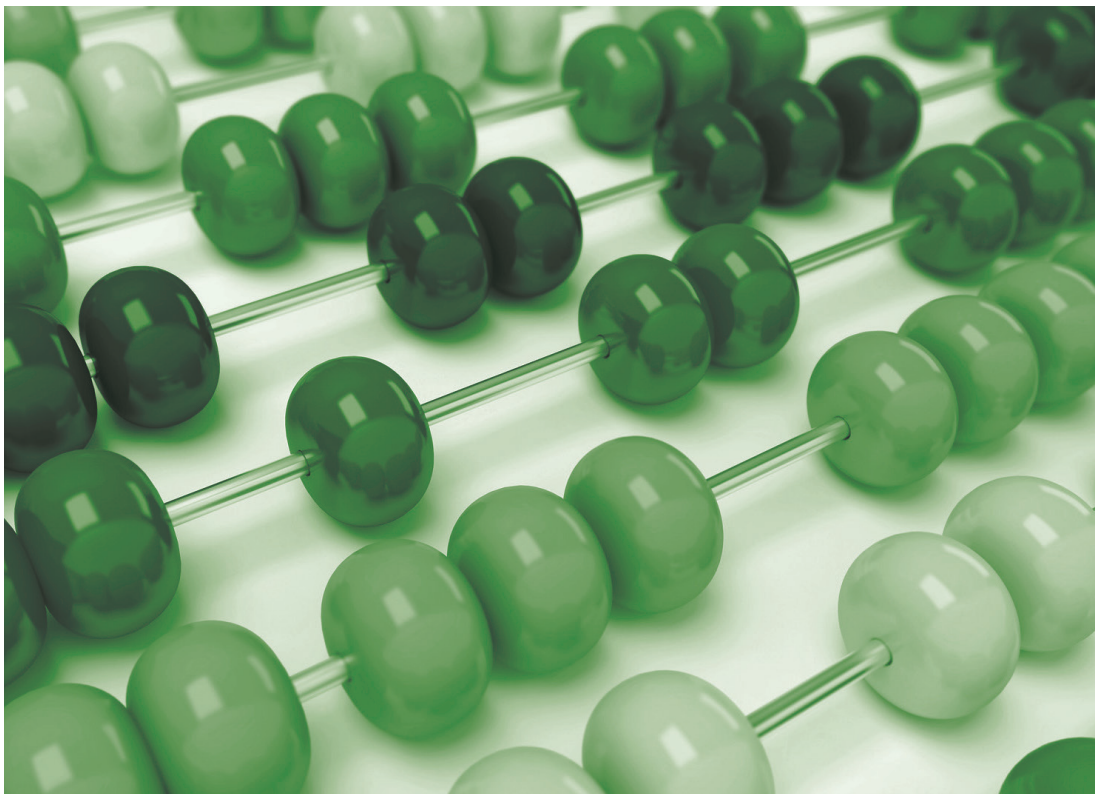


Enrico Zimuel

II EDIZIONE  
AGGIORNATA A PHP 7.4

# Sviluppare in PHP 7

Realizzare applicazioni web e API professionali



---

# **Sviluppare in PHP 7**

II EDIZIONE



---

# Sviluppare in PHP 7

II EDIZIONE

Enrico Zimuel

**tecniche nuove**

Copyright per l'edizione italiana:  
© 2019 Tecniche Nuove, via Eritrea 21, 20157 Milano  
Redazione: tel. 0239090258  
e-mail: [libri@tecnicheNuove.com](mailto:libri@tecnicheNuove.com)  
Vendite: tel. 0239090440, fax 0239090335  
e-mail: [vendite-libri@tecnicheNuove.com](mailto:vendite-libri@tecnicheNuove.com)  
<http://www.tecnicheNuove.com>

ISBN: 978-88-481-4031-7  
ISBN (PDF): 978-88-481-4032-4  
ISBN (E-PUB): 978-88-481-4033-1

Tutti i diritti sono riservati. Nessuna parte del libro può essere riprodotta o diffusa con un mezzo qualsiasi, fotocopie, microfilm o altro, senza il permesso dell'editore

All rights reserved. No part of this book shall be reproduced, stored in a retrieval system, or transmitted, by any means, electronic, mechanical photocopying, recording or otherwise without written permission from the publisher

Realizzare un libro è un'operazione complessa, che richiede numerosi controlli: sul testo, sulle immagini e sulle relazioni che si stabiliscono tra essi. L'esperienza suggerisce che è praticamente impossibile pubblicare un libro privo di errori. Saremo quindi grati ai lettori che vorranno segnalarceli.

Realizzazione editoriale: Mokarta sas, Gorgonzola (MI)  
Stampa: Logo, Borgoricco (PD)  
Finito di stampare nel mese di novembre 2019  
*Printed in Italy*

**Questo libro è disponibile e acquistabile in versione digitale su [www.tecnicheNuove.com](http://www.tecnicheNuove.com)**

# Sommario

<b>Prefazione.....</b>	<b>XIII</b>
<b>Introduzione .....</b>	<b>XVII</b>
<b>Capitolo 1 - Introduzione a PHP .....</b>	<b>1</b>
Storia ed evoluzione di PHP	1
Licenza d'utilizzo	3
Il logo e la mascotte di PHP	4
L'interprete del linguaggio	6
L'ecosistema PHP	10
La community PHP	11
La scena italiana	13
<b>Capitolo 2 - Il linguaggio.....</b>	<b>15</b>
Hello World	15
Tipi di dati e variabili	17
Integer	19
Double	20
String	21
Array	24
Stringhe come array	27
Istruzioni condizionali	28
Cicli iterativi	33
For	33
While	36
Do-while	36
Break e continue	37
Funzioni	38
La visibilità delle variabili	40

Passaggio per valore o per riferimento	41
Parametri opzionali e operatore variadic	42
Funzioni ricorsive	44
Funzioni anonime	46
Gestione dei tipi in ingresso e uscita	47
<b>Capitolo 3 - Programmazione a oggetti .....</b>	<b>53</b>
Le classi	54
Costruttore di classe	58
Namespace	59
Autoloading	60
Ereditarietà	61
Classi astratte e final	64
Interfacce	66
Entità statiche	69
Trait	71
Classi anonime	73
<b>Capitolo 4 - Gestione degli errori.....</b>	<b>75</b>
Tipi di errore	76
Log degli errori	79
Gestione delle eccezioni	83
<b>Capitolo 5 - Organizzare un progetto in PHP.....</b>	<b>89</b>
Composer	90
Aggiornare le dipendenze	92
Autoloading	94
Scelta del repository	96
Come scegliere un progetto open source	98
Versionamento dei sorgenti	101
Avviare un progetto con Git	102
Branch	104
Push nel repository github	105
Pull dal repository github	107
Gestione dei conflitti	108
Pull request	109
Test automatici con PHPUnit	110
Installazione di PHPUnit	112
Come scrivere un test automatico	112
Configurazione ed esecuzione di una suite di test	115
Code coverage	116
Organizzazione dei file	118
<b>Capitolo 6 - File e database.....</b>	<b>121</b>
Gestione dei file in PHP	122
Utilizzo di fwrite() e fread()	123
Lock dei file	127

File CSV	129
File JSON	131
File XML	135
Database	140
Utilizzo di PDO	145
ORM e Doctrine	150
Database NoSQL e MongoDB	163
<b>Capitolo 7 - Il web .....</b>	<b>169</b>
Il protocollo HTTP	169
Il web server	173
Apache	174
Nginx	176
PHP Web Server	177
Richieste HTTP in PHP	180
Risposte HTTP in PHP	187
Gestione di un template HTML	189
Invio di file	191
Cookie	194
Sessioni	196
Routing di una richiesta HTTP	203
Lo standard PSR-7	207
<b>Capitolo 8 - Struttura e gestione di un'applicazione web .....</b>	<b>215</b>
Architetture MVC	216
Architetture middleware	224
Infrastruttura di un'applicazione web	234
Deploy di un'applicazione PHP	236
Deployer	238
Ansible	241
Sistemi di virtualizzazione	242
<b>Capitolo 9 - Sviluppo di web API .....</b>	<b>245</b>
Le caratteristiche di una web API	246
Architetture REST	249
HAL	254
Sviluppo di API REST middleware	258
Autenticazione e autorizzazione	261
HTTP Basic Access Authentication	261
HTTP Digest Access Authentication	264
OAuth2	267
Documentazione	272
Apigility	275
<b>Capitolo 10 - Sicurezza in PHP .....</b>	<b>285</b>
Attaccare un'applicazione web	285
Injection	286



SQL Injection	286
Directory traversal	290
Cross-Site Scripting (XSS)	291
Proteggere dati sensibili	294
Memorizzare una password utente	294
Cifrare informazioni sensibili	300
Authenticated encryption	301
Crittografia a chiave pubblica	306
Autenticazione	312
Cross-Site Request Forgery (CSRF)	314
Utilizzo di librerie con vulnerabilità conosciute	320
Autorizzazioni	321
ACL, Access Control List	322
RBAC, Role-based Access Control	324
<b>Capitolo 11 - Design pattern in PHP.....</b>	<b>329</b>
Tipologie di design pattern	329
Dependency injection	330
Pattern creazionali	336
Abstract Factory	336
Builder	337
Prototype	340
Singleton	343
Pattern strutturali	344
Adapter	344
Bridge	345
Decorator	347
Facade	349
Proxy	350
Pattern comportamentali	352
Chain of Responsibility	352
Command	354
Iterator	356
Observer	358
Strategy	361
<b>Capitolo 12 - Le ultime novità di PHP.....</b>	<b>365</b>
PHP 7.3	365
ext_skel	366
net_get_interfaces()	366
Nuova sintassi heredoc e nowdoc	367
Valori per riferimento in list()	368
Deprecato l'utilizzo di costanti case-insensitive	369
Aggiunte costanti predefinite cURL	370
FILTER_SANITIZE_ADD_SLASHES	372
Aggiunto il supporto al formato WebP	372
Nuove funzioni matematiche in GMP	372

JSON_THROW_ON_ERROR	373
is_countable()	373
array_key_first() e array_key_last()	374
gc_status()	374
hrtime()	375
Supporto di valori float in SimpleXML	375
Aggiunta la versione Argon2id per le password	376
Utilizzo di PCRE2 per le espressioni regolari	376
Utilizzo della virgola finale nelle funzioni	377
PHP 7.4	378
Aggiunto il tipo sulle proprietà di classe	378
FFI (Foreign Function Interface)	382
Pre-caricamento (preloading)	385
Spread operator sugli array	387
Funzioni freccia (arrow functions)	388
Operatore di assegnazione null coalescing	389
Riferimenti deboli (Weak References)	390
Covarianza e controvarianza	391
Nuovo meccanismo di serializzazione degli oggetti	392
Performance di PHP 7.3 e 7.4	393
PHP 8.0	395

<b>Bibliografia.....</b>	<b>397</b>
--------------------------	------------

<b>Indice analitico .....</b>	<b>403</b>
-------------------------------	------------



*A Valentina e Alberto,  
i 10 amori della mia vita.*



# Prefazione

*Ovvero “come ho imparato a non preoccuparmi e ad amare lo sviluppo web”.*

Una ricerca veloce su Internet volta a conoscere nel più breve tempo possibile quale sia il linguaggio più importante o anche solo il più popolare nell’ambito dello sviluppo web non vi lascerà soddisfatti. Se sarete fortunati incapperete in una lista di almeno 10 linguaggi di programmazione che, vi verrà detto, dovrete imparare a usare a livello avanzato se volete spacciare sul vostro curriculum l’essere degli sviluppatori esperti di siti web. È probabile che se aveste effettuato questa ricerca qualche anno fa o se la rifarete tra qualche anno, la lista dei risultati vi avrà mostrato e vi mostrerà delle liste sostanzialmente diverse. Con qualche piccola eccezione. Ad esempio, un invariante dagli albori del Web a oggi (e, ci scommetto, anche per il futuro) è HTML. Non è un linguaggio di programmazione, anche se a volte nei curriculum si trova sotto questa voce (il che già la direbbe lunga sulla competenza di chi un tale cv lo ha compilato...), ma serve a comporre e a fornire una semantica minimale alla vostra pagina web. Il Web è praticamente nato con HTML, la cui prima versione è stata rilasciata nel 1993 e che oggi si è evoluto nella versione 5. È facile da imparare, ma parecchio limitato se la vostra intenzione è quella di avere delle pagine che abbiano stile e carattere.

Per aggiungere stile al vostro sito, già dal 1994 esiste un linguaggio di descrizione della presentazione del vostro documento scritto in HTML. Si chiama CSS, oggi è arrivato alla sua terza versione e, da quando è stato introdotto, ha fatto coppia fissa con HTML. Se fossi un informatico teorico vi direi che ci possiamo fermare qui: qualcuno si è divertito a dimostrare che HTML5 e CSS3, insieme, costituiscono un linguaggio “Turing complete”. In altre parole, se un problema è un problema computazionale e, pertanto, può essere risolto per mezzo di un algoritmo, allora il gioco è fatto: potete senz’altro trovare una se-

quenza di istruzioni HTML5/CSS3 che possa convertire quell'algoritmo in qualcosa che la macchina interpreta ed esegue automaticamente.

Faccio pubblica ammenda: non sono un informatico teorico. Per quanto nel mio lavoro di ricercatore mi nutra quotidianamente di formalizzazioni, strumenti matematici, modelli e sperimentazioni che devono seguire una metodologia rigorosa e solida dal punto di vista teorico, nella mia attività quotidiana e in quella dei miei collaboratori c'è sempre quel momento in cui è necessario scrivere del codice il più velocemente possibile, nel modo migliore possibile: è l'approccio che alcuni chiamano "quick and dirty" e alzi la mano chi non abbia mai avuto bisogno di ricorrervi e che spesso si sia trovato nelle condizioni di ammettere di non essere sufficientemente "quick" o che gli strumenti a disposizione non fossero abbastanza "dirty".

Abbandonata la risposta facile dal punto di vista teorico che cerca di convincervi che HTML5 e CSS3 contengono già tutto quello di cui avete bisogno per sviluppare siti web completi e dinamici, rimane allora il problema di quale linguaggio adottare per gestire l'interazione, la gestione di dati, la personalizzazione, le animazioni, etc. Insomma, tutto quello che oggi siamo abituati a pensare essere la chiave di un sito web.

Tornando agli albori del Web, la risposta era immediata: potevano essere creati dei programmi in linguaggio C da inserire nella cartella "cgi-bin" lato server. L'output di questi programmi viaggiava dritta dritta sulla connessione TCP stabilita dal protocollo HTTP, arrivava all'host che eseguiva il browser e che aveva effettuato la richiesta di accesso alla pagina, con il comando GET o POST che fosse. Questa sequenza di caratteri finalmente giunta in input al processo client veniva quindi passata al modulo di "render" del browser per essere fruita dall'utente. Il modo più pulito per fare questo era che l'output, pertanto, fosse formattato già direttamente nel linguaggio HTML grazie a una sequenza interminabile di istruzioni printf. Chiunque abbia sviluppato per il Web tra il 1993 e il 1996 sa benissimo di cosa sto parlando. Non deve sorprendere che la diffusione in termini di popolarità di linguaggi di alto livello molto potenti (e tra di loro molto diversi) quali PERL e JAVA abbia tolto il linguaggio C dal trono del controllo server-side. Da un punto di vista tecnologico quello sembrava un punto di arrivo, ma era solo l'inizio di una rivoluzione.

Come spesso accade, le cose si complicarono velocemente: la qualità dei siti web raggiungibili da un comune browser diveniva sempre più professionale, l'aspettativa da parte dei clienti e degli operatori del settore saliva di conseguenza, formando un circolo al contempo virtuoso e vizioso che pretendeva lo sviluppo di siti sempre più dinamici, interattivi e personalizzabili in tempi decisamente stretti. JAVA non consentiva, oggi come allora, di fare le cose velocemente, "quick", appunto, mentre dall'altra parte dello spettro, PERL, forse "dirty" lo era fin troppo. Inoltre, come già accadeva per C, l'integrazione con HTML appariva forzata.

Non mi dilungherò qui sulla tecnologia su cui nessuno avrebbe davvero scommesso e che oggi è imprescindibile nel ciclo di vita di un serio sviluppo "full-stack": JavaScript. Il mondo dello sviluppo client-side è ancora oggi in piena evoluzione e difficilmente oggi possiamo pensare a un sito web efficiente e dinamico che non utilizzi JavaScript. Glisserò anche sulla rivoluzione apportata da HTTP 2.0 al protocollo che realizza il Web su reti TCP/IP e che consentiva il superamento del principio "state-less" introducendo

le gioie e i dolori dei “cookie”. Ci sarebbero tante cose da dire, ma ci vorrebbe un'altra prefazione per un altro libro. I paragrafi precedenti, invece, erano funzionali a dare un'idea del carattere di novità apportato da PHP al suo ingresso nel mondo degli sviluppatori: finalmente un linguaggio che permetteva di realizzare un sito dinamico in modalità “quick e dirty”, veloce come PERL, ma totalmente a suo agio all'interno di pagine web che potevano alternare righe scritte in normalissimo codice HTML e altre che contenevano sequenze di istruzioni PHP. Veloce, sporco, ma decisamente più leggibile nell'ambito dello sviluppo web rispetto ai suoi diretti concorrenti C e PERL.

Le innovazioni hanno bisogno di tempo prima di affermarsi: personalmente ho ignorato l'esistenza di PHP fino a quando non è stata rilasciata la versione 3 ed ho iniziato a usarlo costantemente dalla versione 4 in poi. Fino a quel momento perseverai in quella che oggi mi sembra sia stata una follia: alla fine degli anni '90, se si trattava di sviluppare del codice da eseguire lato server, preferivo rimanere all'interno della mia “comfort-zone”, continuando a scrivere istruzioni in linguaggio C in file che potevo comodamente modificare con l'editor VI. Preistoria: la mia conversione personale risale al 2000, proprio nel bel mezzo della prima bolla .com. Eppure proprio grazie a PHP, in quegli anni, ho iniziato a pensare seriamente che sviluppare per il Web fosse una cosa bella.

Da allora ho osservato, da utilizzatore, come questo linguaggio si sia trasformato in qualcosa sempre meno “dirt”, pur mantenendo la sua caratteristica “quick”. PHP 5 ha “osato” introdurre il paradigma della programmazione orientata agli oggetti, rendendo possibile un'ingegnerizzazione professionale e industriale dell'applicazione web finalmente concorrenziale al mondo JAVA. Le connessioni PDO hanno reso il passaggio dalle fasi di produzione a quelle di installazione e messa in esercizio decisamente meno doloroso. Le metodologie di testing si sono evolute (e questo libro che avete tra le mani vi fornisce una sezione chiarissima ed esauriente per sviluppare delle unità di test in modo professionale). Infine, ma sono sicuro che la storia di questo linguaggio andrà ben oltre nel futuro, siamo arrivati alla versione 7, che l'autore, Enrico Zimuel, vi illustra in questo volume interamente in italiano in modo tecnicamente preciso e didatticamente impeccabile. Studiando PHP 7 avrete difficoltà a immaginare che questo linguaggio sia stato salutato come la doverosa soluzione “sporca” a un problema che fino a quel momento veniva visto come il refugium peccatorum degli sviluppatori che avrebbero voluto scrivere codice e invece erano finiti a “fare siti web”. PHP 7 oggi è un linguaggio completo, allo stato dell'arte per quanto riguarda i paradigmi di programmazione e consente l'adozione di pattern di sviluppo moderni ed efficaci. Per non parlare della sua efficienza, che nulla ha a che vedere con la fama che si è guadagnato con le versioni precedenti e che oggi ha davvero poco da invidiare rispetto alle sue alternative. Non ha perso la sua vocazione principale e mentre C, JAVA e Python li trovate impiegati negli ambiti più disparati, PHP rimane il linguaggio per sviluppare soluzioni web lato server: ma è quello che vuole essere ed è per questo che lo fa bene.

Certamente le sfide del programmatore esperto di PHP oggi sono numerose: dagli albori del Web molti altri linguaggi e strumenti sono stati introdotti che hanno vissuto alcune stagioni di autentico successo. In alcuni casi, si tratta di tecnologie che rimarranno nel tempo, anche se nessuno ha la sfera di cristallo per poter dire quale tra queste sopravvivrà. La dannazione dell'informatico è quella di un professionista che non potrà mai permettersi di rimanere al calduccio della sua “comfort zone”. Ci sarà un momento in



cui si dovrà passare ad altro, ma decisamente quel momento, per quanto riguarda PHP, non è ancora arrivato.

Per quanto mi riguarda la lettura di questo libro è stata per me l'occasione di fare una passeggiata in un'automobile nuova di zecca con un caro amico, con una differenza fondamentale: ancora una volta io ero uno di quelli che doveva imparare i nuovi optional di quella macchina per usarli al meglio, ma anche per insegnarli. Enrico Zimuel, invece, molti di quegli optional ha contribuito a progettargli e a inserirli nell'ultimo modello, affinché quel veicolo rimanesse sempre all'avanguardia. Imparare a guidare un'auto in tale compagnia è un privilegio che non ha prezzo.

**Giancarlo Ruffo**

**Giancarlo Ruffo** è Professore Associato presso il Dipartimento di Informatica dell'Università degli Studi di Torino e insegna PHP dal 2011 all'interno del corso di "Tecnologie web". Si occupa di analisi di dati, reti complesse e scienze sociali computazionali e dirige il gruppo di ricerca ARC2S <http://arcs.di.unito.it> presso lo stesso dipartimento. Ha scritto più di 70 articoli scientifici pubblicati in riviste e atti di conferenze prevalentemente internazionali. Per analizzare dati privilegia l'uso di Python e per divertirsi e ricordare i bei tempi andati sogna di tornare prima o poi a programmare in C, ma ammira la professionalità degli esperti programmatori JAVA. Se deve scrivere del codice server side per un'applicazione web non ha dubbi e si concede al suo "dirty pleasure": il PHP.

# Introduzione

Quella che state leggendo è la seconda edizione del libro. Dopo il successo della prima e grazie ai numerosi feedback che ho ricevuto ho pensato di aggiornare il contenuto del libro, fermo alla versione 7.2 di PHP.

Questa nuova edizione contiene due capitoli in più. Il Capitolo 11 sui design pattern in PHP e il Capitolo 12 con le novità di PHP 7.3 e le ultimissime novità di PHP 7.4, che mentre scrivo non è ancora disponibile, la data prevista è il 28 novembre 2019.

PHP è un linguaggio di programmazione per lo sviluppo di applicazioni web nato nel lontano 1995. È utilizzato da circa l'80% di tutti i siti internet del mondo. Queste cifre parlano da sole, una tecnologia consolidata che si è diffusa negli anni a macchia d'olio.

Quali sono le caratteristiche che hanno reso PHP uno dei linguaggi più utilizzati sul web?

Sicuramente, la semplicità d'utilizzo di questa tecnologia. La curva di apprendimento di questo linguaggio è molto bassa, chiunque può imparare le basi del linguaggio in pochissimo tempo. Un'altra caratteristica vincente è l'infrastruttura snella, di facile installazione e onnipresente su tutte le piattaforme. PHP è un linguaggio interpretato che gira praticamente su tutti i sistemi operativi del mondo, compresi i sistemi IBM.

Un altro punto di forza è la sua *community*. Migliaia di software open source sono scritti in PHP, tra i tanti basti citare *Wordpress*, *Drupal*, *Magento*, *WikiMedia*. Inoltre, è possibile trovare una libreria o un framework PHP per qualsiasi esigenza.

La versione 7 di PHP ha segnato una nuova rinascita del linguaggio. Grazie alla riscrittura completa del motore del linguaggio, le performance di esecuzione di PHP 7 sono da record: è tra i più veloci linguaggi interpretati del mondo! Questa nuova versione ha introdotto numerose funzionalità interessanti, utilizzabili anche per applicazioni *enterprise*.

In questo libro si introduce l'utilizzo del linguaggio PHP 7 per lo sviluppo di applicazioni web professionali. Vengono introdotte le basi del linguaggio per arrivare, passo dopo passo, a sviluppare un'applicazione PHP a livello professionale. Per professionale intendo un'applicazione che consideri l'infrastruttura da utilizzare, l'utilizzo di basi di dati, i sistemi di monitor e logging, la scalabilità, la sicurezza, etc. Ho cercato di affrontare tutti questi aspetti nelle pagine di questo libro, offrendo qualche consiglio e spunto di riflessione ove possibile. Sono ormai tanti anni che utilizzo PHP a livello professionale. Ho avuto il piacere e la fortuna di lavorare a stretto contatto con gli autori del linguaggio a partire dalla versione 3 ossia Andi Gutmans e Zeev Suraski e negli anni, viaggiando per conferenze, ho conosciuto la maggior parte dei *core contributor* attuali. Ho potuto sporcarmi le mani con il codice C dello Zend Engine, per capire come funzionasse PHP *under the hood*.

Inoltre, ho macinato codice PHP insieme a numerosi programmatori di talento in tutto il mondo, sviluppando diverse librerie open source in PHP (Apigility, Expressive, Zend Framework, etc.), imparando da loro diversi trucchi e best practice. Con questo libro ho cercato di condividere queste mie esperienze anche legate alla gestione di un progetto open source.

## Com'è organizzato questo libro

---

I capitoli che troverete in questa seconda edizione del libro sono così suddivisi.

Nel **Capitolo 1** viene introdotto PHP, evidenziando il percorso storico e l'evoluzione del linguaggio. Si introduce anche il motore del linguaggio, lo *Zend Engine*, illustrando l'architettura interna del linguaggio. Oltre al linguaggio, sono messi in evidenza l'ecosistema di PHP e la sua community, in forte crescita.

Nel **Capitolo 2** si introducono le basi del linguaggio. Vengono presentate le variabili, le condizioni, i cicli iterativi, gli array, le funzioni, la ricorsione, etc. Particolare rilievo è dato alle nuove funzionalità della versione 7, come la dichiarazione di tipi scalari (*scalar type declaration*), l'utilizzo dei tipi *void* e *nullable*, l'operatore *variadic*, *spaceship*, *null coalesce*, etc.

Nel **Capitolo 3** viene introdotta la Programmazione Orientata agli Oggetti (OOP). Vengono illustrate le classi, i *namespace*, le funzioni di *autoload*, l'ereditarietà, le classi astratte e *final*, le interfacce, i *traits*, le classi anonime, etc.

Nel **Capitolo 4** viene affrontato l'argomento della gestione degli errori. Spesso questo argomento è sottovalutato ma è una caratteristica fondamentale di un'applicazione professionale. In questo capitolo vengono presentate le varie tipologie di errore, la gestione dei *log* e la nuova gerarchia delle eccezioni di PHP 7.

Nel **Capitolo 5** vengono illustrate le metodologie e le tecniche per la gestione di un progetto PHP. In particolare, viene introdotto l'utilizzo di *composer*, del versionamento del codice tramite *git*, dell'utilizzo dei test automatici con *PHPUnit*, dell'organizzazione dei

file in un progetto, etc. Inoltre viene affrontato il tema della scelta di un progetto *open source*, di quali caratteristiche tener presente e degli indicatori da utilizzare.

Nel **Capitolo 6** viene affrontato il tema della persistenza dei dati tramite file e database. Per i file sono illustrate le tecniche di base e alcune delle funzionalità avanzate, come le operazioni di *lock*. Inoltre, sono presi in esame i formati CSV, JSON e XML. Per i database relazionali vengono forniti esempi d'utilizzo con PDO e tramite ORM come *Doctrine*. Per i database *NoSQL* viene introdotto l'utilizzo di *MongoDB*.

Nel **Capitolo 7** si introduce l'utilizzo di PHP come linguaggio per il web. Viene illustrato il protocollo HTTP e le funzioni PHP per la gestione del flusso in ingresso e in uscita in un'applicazione web. Viene introdotto l'utilizzo dei web server come *Apache*, *nginx* e il *PHP web server*. Sono anche forniti esempi sull'invio di dati, l'upload dei file, la gestione dei *template* HTML, le sessioni, etc. Particolare risalto è dato all'utilizzo dello standard PSR-7 per la gestione dei messaggi HTTP.

Nel **Capitolo 8** vengono introdotte le architetture MVC e *middleware* con l'utilizzo dei progetti open source *Zend Framework* e *Expressive*. Inoltre, viene affrontato il tema del deploy di un'applicazione PHP con l'utilizzo dei progetti *Deployer* e *Ansible* e della virtualizzazione di un'applicazione web tramite *Vagrant* e *Docker*.

Nel **Capitolo 9** si affronta il tema dello sviluppo di web API in PHP. Vengono illustrate le architetture REST e l'utilizzo degli *hypermedia*. Sono affrontati i temi della *content negotiation*, del versionamento, dell'utilizzo dei formati JSON e HAL, dei sistemi di autenticazione HTTP Basic, Digest e OAuth2, della documentazione tramite *Swagger*, etc. Nel capitolo è incluso un esempio dell'architettura *middleware* per lo sviluppo di API REST. Infine, viene presentato il progetto *Apigility* per lo sviluppo di API HTTP professionali in PHP.

Nel **Capitolo 10** viene affrontato il tema della sicurezza. Sono presentati alcuni degli attacchi più famosi, riportati nella *Top Ten 2017* del progetto OWASP. Vengono presi in esame attacchi *SQL Injection*, *directory traversal*, *XSS*, *session fixation*, *CSRF*, etc. Per ogni attacco è illustrato un possibile rimedio, come ad esempio l'utilizzo del *Content Security Policy (CSP)*. Sono anche illustrate le tecniche per la protezione delle password, tramite gli algoritmi *bcrypt*, *scrypt* e *Argon2*. Viene inoltre illustrata la modalità di cifratura *Authenticated Encryption (AE)* e le funzioni di crittografia a chiave pubblica per autenticare e firmare digitalmente un documento.

Nel **Capitolo 11** vengono introdotti alcuni dei design pattern più utilizzati nel mondo della programmazione a oggetti in PHP. Questo capitolo contiene anche un'introduzione alla *Dependency Injection*, argomento fondamentale per poter gestire la complessità delle moderne applicazioni web.

Nel **Capitolo 12**, l'ultimo del libro, vengono presentate le numerose novità della versione 7.3 e 7.4 di PHP. In particolare, con la versione 7.4 sono state aggiunte alcune funzionalità molto importanti come l'aggiunta dei tipi sulle proprietà di classe e la funzionalità di pre-caricamento (*preloading*). Inoltre, in questo capitolo ho presentato un benchmark per mettere a confronto le velocità di esecuzione di PHP a partire dalla versione 7.1 fino alla 7.4, non vi anticipo il risultato.