1. Record Nr. UNINA9910254759603321

Autore	Pitt Christopher
Titolo	Typed PHP [[electronic resource]] : Stronger Types For Cleaner Code / / by Christopher Pitt
Pubbl/distr/stampa	Berkeley, CA : , : Apress : , : Imprint : Apress, , 2016
ISBN	1-4842-2114-1
Edizione	[1st ed. 2016.]
Descrizione fisica	1 online resource (XVIII, 76 p. 8 illus., 7 illus. in color.)
Collana	Expert's Voice
Disciplina	005.11
Soggetti	Computer programming Programming languages (Electronic computers) Web Development Programming Techniques Programming Languages, Compilers, Interpreters
Lingua di pubblicazione	Inglese
Formato	Materiale a stampa
Livello bibliografico	Monografia
Note generali	Includes index.
Nota di contenuto	1. The State of PHP -- 2. Structure -- 3. Extensions -- 4. Design -- 5. Implementation.
Sommario/riassunto	Discover how stronger types mean cleaner, more efficient, and optimized PHP applications. This unique book looks at typed PHP: PHP types, strings, regular expressions, and more from PHP 7 as found in standard PHP libraries, user libraries, extensions, and cross-compilers. You'll see how to create a set of reusable tools that unify and ease the scalar types of PHP. PHP has a rich history and a dominant place on the web. It has achieved much despite language inconsistencies and difficulties. Bjarne Stroustrup once said: "There are only two kinds of languages: the ones people complain about and the ones nobody uses". PHP is one of those languages that everybody uses, yet that's often seen as a good reason to ignore the bad parts and just get stuff done. We're all for getting stuff done, and to that end, the author has used Plain Ol' PHP for many years. It's always bugged him how procedural PHP is, in an ecosystem of OOP libraries and frameworks. So he decided to take a deeper look at building a stronger type system on top of PHP. That's the goal of this book. What You'll Learn Discover the fundamentals of PHP strings, regex, underscores, native function

inconsistencies, and more Examine the structure of PHP types including boxing, regex, namespace functions, composer autoload, null problem, optional values, and more Work with extensions like vagrant + phansible, provisioning, vagrant commands, SPL types, scalar objects, zephir, and more Design using scalar, SPL, zephir, structure types, resolving types, chaining, combining number types, PHPUnit, packaging, and more Plan for the future using a case study example.