

1. Record Nr.	UNINA9910785226503321
Autore	Phillips Dusty
Titolo	Python 3 object oriented programming [[electronic resource]] : harness the power of Python 3 objects // Dusty Phillips
Pubbl/distr/stampa	Birmingham, U.K., : Packt Open Source, 2010
ISBN	1-282-71248-9 9786612712487 1-84951-127-6
Edizione	[1st edition]
Descrizione fisica	1 online resource (405 p.)
Collana	Community experience distilled
Disciplina	005.13 006.76
Soggetti	Python (Computer program language) Object-oriented programming (Computer science) Object-oriented programming languages
Lingua di pubblicazione	Inglese
Formato	Materiale a stampa
Livello bibliografico	Monografia
Note generali	Includes index.
Nota di contenuto	Cover; Copyright; Credits; About the Author; About the Reviewers; Table of Contents; Preface; Chapter 1: Object-oriented Design; Object-oriented?; Objects and classes; Specifying attributes and behaviors; Data describes objects; Behaviors are actions; Hiding details and creating the public interface; Composition and inheritance; Inheritance; Inheritance provides abstraction; Multiple inheritance; Case study; Exercises; Summary; Chapter 2: Objects in Python; Creating Python classes; Adding attributes; Making it do something; Initializing the object; Explaining yourself; Modules and packages Organizing the modules Absolute imports; Relative imports; Who can access my data?; Case study; Exercises; Summary; Chapter 3: When Objects are Alike; Basic inheritance; Extending built-ins; Overriding and super; Multiple inheritance; The diamond problem; Different sets of arguments; Polymorphism; Case study; Exercises; Summary; Chapter 4: Expecting the Unexpected; Raising exceptions; Raising an exception; What happens when an exception occurs?; Handling exceptions; Exception hierarchy; Defining our own exceptions; Exceptions aren't exceptional; Case study; Exercises; Summary

Chapter 5: When to Use Object-oriented Programming
Treat objects as objects; Using properties to add behavior to class data; How it works; Decorators: another way to create properties; When should we use properties?; Managing objects; Removing duplicate code; In practice; Or we can use composition; Case study; Exercises; Summary; Chapter 6: Python Data Structures; Empty objects; Tuples and named tuples; Named tuples; Dictionaries; When should we use dictionaries?; Using defaultdict; Lists; Sorting lists; Sets; Extending built-ins; Case study; Exercises; Summary
Chapter 7: Python Object-oriented Shortcuts
Python built-in functions; Len; Reversed; Enumerate; Zip; Other functions; Comprehensions; List comprehensions; Set and dictionary comprehensions; Generator expressions; Generators; An alternative to method overloading; Default arguments; Variable argument lists; Unpacking arguments; Functions are objects too; Using functions as attributes; Callable objects; Case study; Exercises; Summary; Chapter 8: Python Design Patterns I; Design patterns; Decorator pattern; Decorator example; Decorators in Python; Observer pattern; Observer example
Strategy pattern
Strategy example; Strategy in Python; State pattern; State example; State versus strategy; Singleton pattern; Singleton implementation; Module variables can mimic singletons; Template pattern; Template example; Exercises; Summary; Chapter 9: Python Design Patterns II; Adapter pattern; Facade pattern; Flyweight pattern; Command pattern; Abstract factory pattern; Composite pattern; Exercise; Summary; Chapter 10: Files and Strings; Strings; String manipulation; String formatting; Escaping braces; Keyword arguments; Container lookups; Object lookups; Making it look right
Strings are Unicode

Sommario/riassunto

Harness the power of Python 3 objects
