

The Python Standard Library By Example

From the duo behind the massively successful and award-winning podcast *Stuff You Should Know* comes an unexpected look at things you thought you knew. Josh Clark and Chuck Bryant started the podcast *Stuff You Should Know* back in 2008 because they were curious—curious about the world around them, curious about what they might have missed in their formal educations, and curious to dig deeper on stuff they thought they understood. As it turns out, they aren't the only curious ones. They've since amassed a rabid fan base, making *Stuff You Should Know* one of the most popular podcasts in the world. Armed with their inquisitive natures and a passion for sharing, they uncover the weird, fascinating, delightful, or unexpected elements of a wide variety of topics. The pair have now taken their near-boundless "whys" and "hows" from your earbuds to the pages of a book for the first time—featuring a completely new array of subjects that they've long wondered about and wanted to explore. Each chapter is further embellished with snappy visual material to allow for rabbit-hole tangents and digressions—including charts, illustrations, sidebars, and footnotes. Follow along as the two dig into the underlying stories of everything from the origin of Murphy beds, to the history of facial hair, to the psychology of being lost. Have you ever wondered about the world around you, and wished to see the magic in everyday things? Come get curious with *Stuff You Should Know*. With Josh and Chuck as your guide, there's something interesting about everything (...except maybe jackhammers).

Create Python packages to share your code in a scalable and maintainable way. Improve team productivity, publish helpful libraries, or even start your own open source project following the latest Python packaging standards. *Publishing Python Packages* teaches you how to easily share your Python code with your team and the outside world. Learn a repeatable and highly automated process for package maintenance that's based on the best practices, tools, and standards of Python packaging. *Publishing Python Packages* book walks you through creating a complete package, including a C extension, and guides you all the way to publishing on the Python Package Index. You'll get hands-on experience with the latest packaging tools, and learn the ins-and-outs of package testing and continuous integration. Whether you're entirely new to Python packaging or looking for optimal ways to maintain and scale your packages, this fast-paced and engaging guide is for you. Purchase of the print book includes a free eBook in PDF, Kindle, and ePub formats from Manning Publications.

Introducing the Boost libraries: the next breakthrough in C++ programming Boost takes you far beyond the C++ Standard Library, making C++ programming more elegant, robust, and productive. Now, for the first time, a leading Boost expert systematically introduces the broad set of Boost libraries and teaches best practices for their use. Writing for intermediate-to-advanced C++ developers, Björn Karlsson briefly outlines all 58 Boost libraries, and then presents comprehensive coverage of 12 libraries you're likely to find especially useful. Karlsson's topics range from smart pointers and conversions to containers and data structures, explaining exactly how using each library can improve your code. He offers detailed coverage of higher-order function objects that enable you to write code that is more concise, expressive, and readable. He even takes you "behind the scenes" with Boost, revealing tools and techniques for creating your own generic libraries. Coverage includes Smart pointers that provide automatic lifetime management of objects and simplify resource sharing Consistent, best-practice solutions for performing type conversions and lexical conversions Utility classes that make programming simpler and clearer Flexible container libraries that solve common problems not covered by the C++ Standard Library Powerful support for regular expressions with Boost.Regex Function objects defined at the call site with Boost.Bind and Boost.Lambda More flexible callbacks with Boost.Function Managed signals and slots (a.k.a. the Observer pattern) with Boost.Signals The Boost libraries are proving so useful that many of them are

Read Book The Python Standard Library By Example

planned for inclusion in the next version of the C++ Standard Library. Get your head start now, with *Beyond the C++ Standard Library*.

This book offers a highly accessible introduction to natural language processing, the field that supports a variety of language technologies, from predictive text and email filtering to automatic summarization and translation. With it, you'll learn how to write Python programs that work with large collections of unstructured text. You'll access richly annotated datasets using a comprehensive range of linguistic data structures, and you'll understand the main algorithms for analyzing the content and structure of written communication. Packed with examples and exercises, *Natural Language Processing with Python* will help you: Extract information from unstructured text, either to guess the topic or identify "named entities" Analyze linguistic structure in text, including parsing and semantic analysis Access popular linguistic databases, including WordNet and treebanks Integrate techniques drawn from fields as diverse as linguistics and artificial intelligence This book will help you gain practical skills in natural language processing using the Python programming language and the Natural Language Toolkit (NLTK) open source library. If you're interested in developing web applications, analyzing multilingual news sources, or documenting endangered languages -- or if you're simply curious to have a programmer's perspective on how human language works -- you'll find *Natural Language Processing with Python* both fascinating and immensely useful.

Your Python code may run correctly, but you need it to run faster. Updated for Python 3, this expanded edition shows you how to locate performance bottlenecks and significantly speed up your code in high-data-volume programs. By exploring the fundamental theory behind design choices, *High Performance Python* helps you gain a deeper understanding of Python's implementation. How do you take advantage of multicore architectures or clusters? Or build a system that scales up and down without losing reliability? Experienced Python programmers will learn concrete solutions to many issues, along with war stories from companies that use high-performance Python for social media analytics, productionized machine learning, and more. Get a better grasp of NumPy, Cython, and profilers Learn how Python abstracts the underlying computer architecture Use profiling to find bottlenecks in CPU time and memory usage Write efficient programs by choosing appropriate data structures Speed up matrix and vector computations Use tools to compile Python down to machine code Manage multiple I/O and computational operations concurrently Convert multiprocessing code to run on local or remote clusters Deploy code faster using tools like Docker

The programming language Python was conceived in the late 1980s, [1] and its implementation was started in December 1989[2] by Guido van Rossum at CWI in the Netherlands as a successor to the ABC (programming language) capable of exception handling and interfacing with the Amoeba operating system.[3] Van Rossum is Python's principal author, and his continuing central role in deciding the direction of Python is reflected in the title given to him by the Python community, Benevolent Dictator for Life (BDFL).[4][5] Python was named for the BBC TV show Monty Python's Flying Circus.[6] Python 2.0 was released on October 16, 2000, with many major new features, including a cycle-detecting garbage collector (in addition to reference counting) for memory management and support for Unicode. However, the most important change was to the development process itself, with a shift to a more transparent and community-backed process.[7] Python 3.0, a major, backwards-incompatible release, was released on December 3, 2008[8] after a long period of testing. Many of its major features have also been backported to the backwards-compatible Python 2.6 and 2.7.[9] In February 1991, van Rossum published the code (labeled version 0.9.0) to alt.sources.[10] Already present at this stage in development were classes with inheritance, exception handling, functions, and the core datatypes of list, dict, str and so on. Also in this initial release was a module system borrowed from Modula-3; Van Rossum describes the module as "one of Python's major programming units." [1] Python's exception model also

Read Book The Python Standard Library By Example

resembles Modula-3's, with the addition of an else clause.[3] In 1994 comp.lang.python, the primary discussion forum for Python, was formed, marking a milestone in the growth of Python's userbase.[1] Python reached version 1.0 in January 1994. The major new features included in this release were the functional programming tools lambda, map, filter and reduce. Van Rossum stated that "Python acquired lambda, reduce(), filter() and map(), courtesy of a Lisp hacker who missed them and submitted working patches." [11] The last version released while Van Rossum was at CWI was Python 1.2. In 1995, Van Rossum continued his work on Python at the Corporation for National Research Initiatives (CNRI) in Reston, Virginia whence he released several versions. By version 1.4, Python had acquired several new features. Notable among these are the Modula-3 inspired keyword arguments (which are also similar to Common Lisp's keyword arguments) and built-in support for complex numbers. Also included is a basic form of data hiding by name mangling, though this is easily bypassed.[12] During Van Rossum's stay at CNRI, he launched the Computer Programming for Everybody (CP4E) initiative, intending to make programming more accessible to more people, with a basic "literacy" in programming languages, similar to the basic English literacy and mathematics skills required by most employers. Python served a central role in this: because of its focus on clean syntax, it was already suitable, and CP4E's goals bore similarities to its predecessor, ABC. The project was funded by DARPA.[13] As of 2007, the CP4E project is inactive, and while Python attempts to be easily learnable and not too arcane in its syntax and semantics, reaching out to non-programmers is not an active concern.[14] Here are what people are saying about the book: This is the best beginner's tutorial I've ever seen! Thank you for your effort. -- Walt Michalik The best thing i found was "A Byte of Python," which is simply a brilliant book for a beginner. It's well written, the concepts are well explained with self evident examples. -- Joshua Robin Excellent gentle introduction to programming #Python for beginners -- Shan Rajasekaran Best newbie guide to python -- Nickson Kaigi start to love python with every single page read -- Herbert Feutl perfect beginners guide for python, will give u key to unlock magical world of python

If you need help writing programs in Python 3, or want to update older Python 2 code, this book is just the ticket. Packed with practical recipes written and tested with Python 3.3, this unique cookbook is for experienced Python programmers who want to focus on modern tools and idioms. Inside, you'll find complete recipes for more than a dozen topics, covering the core Python language as well as tasks common to a wide variety of application domains. Each recipe contains code samples you can use in your projects right away, along with a discussion about how and why the solution works. Topics include: Data Structures and Algorithms Strings and Text Numbers, Dates, and Times Iterators and Generators Files and I/O Data Encoding and Processing Functions Classes and Objects Metaprogramming Modules and Packages Network and Web Programming Concurrency Utility Scripting and System Administration Testing, Debugging, and Exceptions C Extensions

By taking you through the development of a real web application from beginning to end, the second edition of this hands-on guide demonstrates the practical advantages of test-driven development (TDD) with Python. You'll learn how to write and run tests before building each part of your app, and then develop the minimum amount of code required to pass those tests. The result? Clean code that works. In the process, you'll learn the basics of Django, Selenium, Git, jQuery, and Mock, along with current web development techniques. If you're ready to take your Python skills to the next level, this book—updated for Python 3.6—clearly demonstrates how TDD encourages simple designs and inspires confidence. Dive into the TDD workflow, including the unit test/code cycle and refactoring Use unit tests for classes and functions, and functional tests for user interactions within the browser Learn when and how to use mock objects, and the pros and cons of isolated vs. integrated tests Test and automate your deployments with a staging server Apply tests to the third-party plugins you integrate into your

Read Book The Python Standard Library By Example

site Run tests automatically by using a Continuous Integration environment Use TDD to build a REST API with a front-end Ajax interface

Provides information on the Python 2.7 library offering code and output examples for working with such tasks as text, data types, algorithms, math, file systems, networking, XML, email, and runtime.

Create succinct and expressive implementations with functional programming in Python Key Features Learn how to choose between imperative and functional approaches based on expressiveness, clarity, and performance Get familiar with complex concepts such as monads, concurrency, and immutability Apply functional Python to common Exploratory Data Analysis (EDA) programming problems Book Description If you're a Python developer who wants to discover how to take the power of functional programming (FP) and bring it into your own programs, then this book is essential for you, even if you know next to nothing about the paradigm. Starting with a general overview of functional concepts, you'll explore common functional features such as first-class and higher-order functions, pure functions, and more.

You'll see how these are accomplished in Python 3.6 to give you the core foundations you'll build upon. After that, you'll discover common functional optimizations for Python to help your apps reach even higher speeds. You'll learn FP concepts such as lazy evaluation using Python's generator functions and expressions. Moving forward, you'll learn to design and implement decorators to create composite functions. You'll also explore data preparation techniques and data exploration in depth, and see how the Python standard library fits the functional programming model. Finally, to top off your journey into the world of functional Python, you'll at look at the PyMonad project and some larger examples to put everything into perspective.

What you will learn Use Python's generator functions and generator expressions to work with collections in a non-strict (or lazy) manner Utilize Python library modules including itertools, functools, multiprocessing, and concurrent features to ensure efficient functional programs Use Python strings with object-oriented suffix notation and prefix notation Avoid stateful classes with families of tuples Design and implement decorators to create composite functions Use functions such as max(), min(), map(), filter(), and sorted() Write higher-order functions Who this book is for This book is for Python developers who would like to perform Functional programming with Python. Python Programming knowledge is assumed.

Complete recipes spread across 15 chapters to help you overcome commonly faced issues by Python for everybody across the globe. Each recipe takes a problem-solution approach to resolve for effective Python. Key Features

Develop expressive and effective Python programs Best practices and common idioms through carefully explained recipes Discover new ways to apply Python for data-focused development Make use of Python's optional type annotations Book Description Python is the preferred choice of developers, engineers, data scientists, and hobbyists everywhere. It is a great language that can power your applications and provide great speed, safety, and scalability. It can be used for simple scripting or sophisticated web applications. By exposing Python as a series of simple recipes, this book gives you insight into specific language features in a particular context. Having a tangible context helps make the language or a given standard library feature easier to understand. This book comes with 133 recipes on the latest version of Python 3.8. The recipes will benefit everyone, from beginners just starting out with Python to experts. You'll not only learn Python programming concepts but also how to build complex applications. The recipes will touch upon all necessary Python concepts related to data structures, object oriented programming, functional programming, and statistical programming. You will get acquainted with the nuances of Python syntax and how to effectively take advantage of it. By the end of this Python book, you will be equipped with knowledge of testing, web services, configuration, and application integration tips and tricks. You will be armed with the knowledge of how to create applications with flexible logging, powerful configuration, command-line options, automated unit tests, and good documentation.

Read Book The Python Standard Library By Example

What you will learn See the intricate details of the Python syntax and how to use it to your advantage Improve your coding with Python readability through functions Manipulate data effectively using built-in data structures Get acquainted with advanced programming techniques in Python Equip yourself with functional and statistical programming features Write proper tests to be sure a program works as advertised Integrate application software using Python Who this book is for The Python book is for web developers, programmers, enterprise programmers, engineers, and big data scientists. If you are a beginner, this book will get you started. If you are experienced, it will expand your knowledge base. A basic knowledge of programming would help.

The ever-growing wealth of information has led to the emergence of a fourth paradigm of science. This new field of activity – data science – includes computer science, mathematics and a given specialist domain. This book focuses on chemistry, explaining how to use data science for deep insights and take chemical research and engineering to the next level. It covers modern aspects like Big Data, Artificial Intelligence and Quantum computing. Get your guided tour through the Python 3.9 interpreter: Unlock the inner workings of the Python language, compile the Python interpreter from source code, and participate in the development of CPython. Are there certain parts of Python that just seem like magic? This book explains the concepts, ideas, and technicalities of the Python interpreter in an approachable and hands-on fashion. Once you see how Python works at the interpreter level, you can optimize your applications and fully leverage the power of Python. By the End of the Book You'll Be Able To: Read and navigate the CPython 3.9 interpreter source code. You'll deeply comprehend and appreciate the inner workings of concepts like lists, dictionaries, and generators. Make changes to the Python syntax and compile your own version of CPython, from scratch. You'll customize the Python core data types with new functionality and run CPython's automated test suite. Master Python's memory management capabilities and scale your Python code with parallelism and concurrency. Debug C and Python code like a true professional. Profile and benchmark the performance of your Python code and the runtime. Participate in the development of CPython and know how to contribute to future versions of the Python interpreter and standard library. How great would it feel to give back to the community as a "Python Core Developer?" With this book you'll cover the critical concepts behind the internals of CPython and how they work with visual explanations as you go along. Each page in the book has been carefully laid out with beautiful typography, syntax highlighting for code examples. What Python Developers Say About The Book: "It's the book that I wish existed years ago when I started my Python journey. [...] After reading this book your skills will grow and you will be able solve even more complex problems that can improve our world." - Carol Willing, CPython Core Developer & Member of the CPython Steering Council "CPython Internals is a great (and unique) resource for anybody looking to take their knowledge of Python to a deeper level." - Dan Bader, Author of Python Tricks "There are a ton of books on Python which teach the language, but I haven't really come across anything that would go about explaining the internals to those curious minded." - Milan Patel, Vice President at (a major investment bank)

Demonstrates the programming language's strength as a Web development tool, covering syntax, data types, built-ins, the Python standard module library, and real world examples. An innovative reference reveals the many capabilities of the Python Standard Library, which is a compilation of commonly used procedures that can be pasted into a Python script, by providing over 300 real-world example scripts. Original. (Intermediate/Advanced)

The second edition of this best-selling Python book (over 500,000 copies sold!) uses Python 3 to teach even the technically uninclined how to write programs that do in minutes what would take hours to do by hand. There is no prior programming experience required and the book is loved by liberal arts majors and geeks alike. If you've ever spent hours renaming files or

Read Book The Python Standard Library By Example

updating hundreds of spreadsheet cells, you know how tedious tasks like these can be. But what if you could have your computer do them for you? In this fully revised second edition of the best-selling classic *Automate the Boring Stuff with Python*, you'll learn how to use Python to write programs that do in minutes what would take you hours to do by hand--no prior programming experience required. You'll learn the basics of Python and explore Python's rich library of modules for performing specific tasks, like scraping data off websites, reading PDF and Word documents, and automating clicking and typing tasks. The second edition of this international fan favorite includes a brand-new chapter on input validation, as well as tutorials on automating Gmail and Google Sheets, plus tips on automatically updating CSV files. You'll learn how to create programs that effortlessly perform useful feats of automation to:

- Search for text in a file or across multiple files
- Create, update, move, and rename files and folders
- Search the Web and download online content
- Update and format data in Excel spreadsheets of any size
- Split, merge, watermark, and encrypt PDFs
- Send email responses and text notifications
- Fill out online forms

Step-by-step instructions walk you through each program, and updated practice projects at the end of each chapter challenge you to improve those programs and use your newfound skills to automate similar tasks. Don't spend your time doing work a well-trained monkey could do. Even if you've never written a line of code, you can make your computer do the grunt work. Learn how in *Automate the Boring Stuff with Python, 2nd Edition*.

The one-stop resource for all your Python queries Powerful and flexible, Python is one of the most popular programming languages in the world. It's got all the right stuff for the software driving the cutting-edge of the development world—machine learning, robotics, artificial intelligence, data science, etc. The good news is that it's also pretty straightforward to learn, with a simplified syntax, natural-language flow, and an amazingly supportive user community. The latest edition of *Python All-in-One For Dummies* gives you an inside look at the exciting possibilities offered in the Python world and provides a springboard to launch yourself into wherever you want your coding career to take you. These 7 straightforward and friendly mini-books assume the reader is a beginning programmer, and cover everything from the basic elements of Python code to introductions to the specific applications where you'll use it. Intended as a hands-on reference, the focus is on practice over theory, providing you with examples to follow as well as code for you to copy and start modifying in the "real world"—helping you get up and running in your area of interest almost right away. This means you'll be finishing off your first app or building and remote-controlling your own robot much faster than you can believe. Get a thorough grounding in the language basics Learn how the syntax is applied in high-profile industries Apply Python to projects in enterprise Find out how Python can get you into hot careers in AI, big data, and more Whether you're a newbie coder or just want to add Python to your magic box of tricks, this is the perfect, practical introduction—and one you'll return to as you grow your career.

Build optimized applications in Python by smartly implementing the standard library Key Features Strategic recipes for effective application development in Python Techniques to create GUIs and implement security through cryptography Best practices for developing readily scalable, production-ready applications Book Description The Python 3 Standard Library is a vast array of modules that you can use for developing various kinds of applications. It contains an exhaustive list of libraries, and this book will help you choose the best one to address specific programming problems in Python. The *Modern Python Standard Library Cookbook* begins with recipes on containers and data structures and guides you in performing effective text management in Python. You

Read Book The Python Standard Library By Example

will find Python recipes for command-line operations, networking, filesystems and directories, and concurrent execution. You will learn about Python security essentials in Python and get to grips with various development tools for debugging, benchmarking, inspection, error reporting, and tracing. The book includes recipes to help you create graphical user interfaces for your application. You will learn to work with multimedia components and perform mathematical operations on date and time. The recipes will also show you how to deploy different searching and sorting algorithms on your data. By the end of the book, you will have acquired the skills needed to write clean code in Python and develop applications that meet your needs. What you will learn Store multiple values per key in associative containers Create interactive character-based user interfaces Work with native time and display data for your time zone Read/write SGML family languages, both as a SAX and DOM parser to meet file sizes and other requirements Group equivalent items using itertools and sorted features together Use partials to create unary functions out of multi-argument functions Implement hashing algorithms to store passwords in a safe way Who this book is for If you are a developer who wants to write highly responsive, manageable, scalable, and resilient code in Python, this book is for you. Prior programming knowledge in Python will help you make the most out of the book.

This book is the first half of The Python Library Reference for Release 3.6.4, and covers chapters 1-18. The second book may be found with ISBN 9781680921090. The original Python Library Reference book is 1920 pages long. This book contains the original page numbers and index, along with the back sections fully intact. While reference-index describes the exact syntax and semantics of the Python language, this library reference manual describes the standard library that is distributed with Python. It also describes some of the optional components that are commonly included in Python distributions. Python's standard library is very extensive, offering a wide range of facilities as indicated by the long table of contents listed below. The library contains built-in modules (written in C) that provide access to system functionality such as file I/O that would otherwise be inaccessible to Python programmers, as well as modules written in Python that provide standardized solutions for many problems that occur in everyday programming. Some of these modules are explicitly designed to encourage and enhance the portability of Python programs by abstracting away platform-specifics into platform-neutral APIs. This book is available for free as a PDF at python.org.

Python 3 is the best version of the language yet: It is more powerful, convenient, consistent, and expressive than ever before. Now, leading Python programmer Mark Summerfield demonstrates how to write code that takes full advantage of Python 3's features and idioms. The first book written from a completely "Python 3" viewpoint, Programming in Python 3 brings together all the knowledge you need to write any program, use any standard or third-party Python 3 library, and create new library modules of your own. Summerfield draws on his many years of Python experience to share deep insights into Python 3 development you won't find anywhere else. He begins by illuminating Python's "beautiful heart": the eight key elements of Python you need to write robust, high-performance programs. Building on these core elements, he introduces new topics designed to strengthen your practical expertise—one concept and hands-on example at a time. This book's coverage includes Developing in Python using procedural, object-oriented, and functional programming paradigms Creating

Read Book The Python Standard Library By Example

custom packages and modules Writing and reading binary, text, and XML files, including optional compression, random access, and text and XML parsing Leveraging advanced data types, collections, control structures, and functions Spreading program workloads across multiple processes and threads Programming SQL databases and key-value DBM files Utilizing Python's regular expression mini-language and module Building usable, efficient, GUI-based applications Advanced programming techniques, including generators, function and class decorators, context managers, descriptors, abstract base classes, metaclasses, and more Programming in Python 3 serves as both tutorial and language reference, and it is accompanied by extensive downloadable example code—all of it tested with the final version of Python 3 on Windows, Linux, and Mac OS X.

Millions of software developers use Python, and it is a powerful foundation for classical and quantum computing. Dancing with Python teaches you how to create elegant and efficient code using Pythonic techniques. Its integrated introduction to quantum computing development helps you extend your skills to the next major computing technology.

"I don't even feel like I've scratched the surface of what I can do with Python" With Python Tricks: The Book you'll discover Python's best practices and the power of beautiful & Pythonic code with simple examples and a step-by-step narrative. You'll get one step closer to mastering Python, so you can write beautiful and idiomatic code that comes to you naturally. Learning the ins and outs of Python is difficult-and with this book you'll be able to focus on the practical skills that really matter. Discover the "hidden gold" in Python's standard library and start writing clean and Pythonic code today. Who Should Read This Book: If you're wondering which lesser known parts in Python you should know about, you'll get a roadmap with this book. Discover cool (yet practical!) Python tricks and blow your coworkers' minds in your next code review. If you've got experience with legacy versions of Python, the book will get you up to speed with modern patterns and features introduced in Python 3 and backported to Python 2. If you've worked with other programming languages and you want to get up to speed with Python, you'll pick up the idioms and practical tips you need to become a confident and effective Pythonista. If you want to make Python your own and learn how to write clean and Pythonic code, you'll discover best practices and little-known tricks to round out your knowledge. What Python Developers Say About The Book: "I kept thinking that I wished I had access to a book like this when I started learning Python many years ago." - Mariatta Wijaya, Python Core Developer "This book makes you write better Python code!" - Bob Belderbos, Software Developer at Oracle "Far from being just a shallow collection of snippets, this book will leave the attentive reader with a deeper understanding of the inner workings of Python as well as an appreciation for its beauty." - Ben Felder, Pythonista "It's like having a seasoned tutor explaining, well, tricks!" - Daniel Meyer, Sr. Desktop Administrator at Tesla Inc.

Discover clean ways to write code that will run on both Python 2 and 3. This book is tutorial-oriented with detailed examples of how to convert existing Python 2-compatible code to code that will run reliably on both Python 2 and 3. Although Python 3 is considered the future of Python, Python 2.x will be maintained for several more years, alongside Python 3, which is not backwards compatible. For those who need to support both versions, this book guides you through the process. Python 2 and 3 Compatibility

Read Book The Python Standard Library By Example

explains the syntactical differences between Python 2 and 3, and how to use Python packages Python-Future and Six to implement neutral compatibility. Developers working on either small, medium, or large projects will appreciate the author's clear explanations, detailed examples, and clean techniques to help them extend support for both versions to their existing Python 2-compatible projects. What You'll Learn Understand the syntactical differences between Python 2 and 3 Use the Six and Future libraries Review the new features in Python 3 Choose which Python versions to support when doing neutral support Decide on whether to port or provide support for both versions Who This Book Is For Professional Python developers and enthusiasts that want to implement Python 3 support for their existing Python 2 compatible code. Python is an easy to learn, powerful programming language. It has efficient high-level data structures and a simple but effective approach to object-oriented programming. Python's elegant syntax and dynamic typing, together with its interpreted nature, make it an ideal language for scripting and rapid application development in many areas on most platforms. The Python interpreter and the extensive standard library are freely available in source or binary form for all major platforms from the Python Web site, <https://www.python.org/>, and may be freely distributed. The same site also contains distributions of and pointers to many free third party Python modules, programs and tools, and additional documentation. The Python interpreter is easily extended with new functions and data types implemented in C or C++ (or other languages callable from C). Python is also suitable as an extension language for customizable applications. This tutorial introduces the reader informally to the basic concepts and features of the python language and system. It helps to have a Python interpreter handy for hands-on experience, but all examples are self contained, so the tutorial can be read off-line as well. For a description of standard objects and modules, see [library-index](#). [reference-index](#) gives a more formal definition of the language. To write extensions in C or C++, read [extending-index](#) and [c-api-index](#). There are also several books covering Python in depth. This tutorial does not attempt to be comprehensive and cover every single feature, or even every commonly used feature. Instead, it introduces many of Python's most noteworthy features, and will give you a good idea of the language's flavor and style. After reading it, you will be able to read and write Python modules and programs, and you will be ready to learn more about the various Python library modules described in [library-index](#). The Glossary is also worth going through.

A guide to completing Python projects for those ready to take their skills to the next level Python Projects is the ultimate resource for the Python programmer with basic skills who is ready to move beyond tutorials and start building projects. The preeminent guide to bridge the gap between learning and doing, this book walks readers through the "where" and "how" of real-world Python programming with practical, actionable instruction. With a focus on real-world functionality, Python Projects details the ways that Python can be used to complete daily tasks and bring efficiency to businesses and individuals alike. Python Projects is written specifically for those who know the Python syntax and lay of the land, but may still be intimidated by larger, more complex projects. The book provides a walk-through of the basic set-up for an application and the building and packaging for a library, and explains in detail the functionalities related to the projects. Topics include: *How to maximize the power of the standard library modules *Where to get third party libraries, and the best practices for utilization *Creating, packaging, and reusing libraries within and across projects *Building multi-layered functionality including networks, data, and user interfaces *Setting up development

Read Book The Python Standard Library By Example

environments and using virtualenv, pip, and more. Written by veteran Python trainers, the book is structured for easy navigation and logical progression that makes it ideal for individual, classroom, or corporate training. For Python developers looking to apply their skills to real-world challenges, Python Projects is a goldmine of information and expert insight.

Winner of the 2014 Jolt Award for "Best Book" "Whether you are an experienced programmer or are starting your career, Python in Practice is full of valuable advice and example to help you improve your craft by thinking about problems from different perspectives, introducing tools, and detailing techniques to create more effective solutions." --Doug Hellmann, Senior Developer, DreamHost

If you're an experienced Python programmer, Python in Practice will help you improve the quality, reliability, speed, maintainability, and usability of all your Python programs. Mark Summerfield focuses on four key themes: design patterns for coding elegance, faster processing through concurrency and compiled Python (Cython), high-level networking, and graphics. He identifies well-proven design patterns that are useful in Python, illuminates them with expert-quality code, and explains why some object-oriented design patterns are irrelevant to Python. He also explodes several counterproductive myths about Python programming--showing, for example, how Python can take full advantage of multicore hardware. All examples, including three complete case studies, have been tested with Python 3.3 (and, where possible, Python 3.2 and 3.1) and crafted to maintain compatibility with future Python 3.x versions. All code has been tested on Linux, and most code has also been tested on OS X and Windows. All code may be downloaded at www.qtrac.eu/pipbook.html. Coverage includes Leveraging Python's most effective creational, structural, and behavioral design patterns Supporting concurrency with Python's multiprocessing, threading, and concurrent.futures modules Avoiding concurrency problems using thread-safe queues and futures rather than fragile locks Simplifying networking with high-level modules, including xmlrpclib and RPyC Accelerating Python code with Cython, C-based Python modules, profiling, and other techniques Creating modern-looking GUI applications with Tkinter Leveraging today's powerful graphics hardware via the OpenGL API using pyglet and PyOpenGL

Easy to understand and fun to read, this updated edition of Introducing Python is ideal for beginning programmers as well as those new to the language. Author Bill Lubanovic takes you from the basics to more involved and varied topics, mixing tutorials with cookbook-style code recipes to explain concepts in Python 3. End-of-chapter exercises help you practice what you've learned. You'll gain a strong foundation in the language, including best practices for testing, debugging, code reuse, and other development tips. This book also shows you how to use Python for applications in business, science, and the arts, using various Python tools and open source packages.

"It's easy to start writing code with Python: that's why the language is so immensely popular. However, Python has unique strengths, charms, and expressivity that can be hard to grasp at first -- as well as hidden pitfalls that can easily trip you up if you aren't aware of them. Effective Python will help you harness the full power of Python to write exceptionally robust, efficient, maintainable, and well-performing code. Utilizing the concise, scenario-driven style pioneered in Scott Meyers's best-selling Effective C++, Brett Slatkin brings together 53 Python best practices, tips, shortcuts, and realistic code examples from expert programmers. Through realistic examples, Slatkin uncovers little-known Python quirks, intricacies, and idioms that powerfully impact code behavior and performance. You'll learn how to choose the most efficient and effective way to accomplish key tasks when multiple options exist, and how to write code that's easier to understand, maintain, and improve. Drawing on his deep understanding of Python's capabilities, Slatkin offers practical advice for each major area of development with both Python 3.x and Python 2.x. Coverage includes: * Algorithms * Objects * Concurrency * Collaboration * Built-in modules * Production techniques * And more Each section contains specific, actionable guidelines organized into items, each with carefully

Read Book The Python Standard Library By Example

worded advice supported by detailed technical arguments and illuminating examples. Using *Effective Python*, you can systematically improve all the Python code you write: not by blindly following rules or mimicking incomprehensible idioms, but by gaining a deep understanding of the technical reasons why they make sense."--[Source inconnue].

Today, anyone in a scientific or technical discipline needs programming skills. Python is an ideal first programming language, and *Introduction to Programming in Python* is the best guide to learning it. Princeton University's Robert Sedgewick, Kevin Wayne, and Robert Dondero have crafted an accessible, interdisciplinary introduction to programming in Python that emphasizes important and engaging applications, not toy problems. The authors supply the tools needed for students to learn that programming is a natural, satisfying, and creative experience. This example-driven guide focuses on Python's most useful features and brings programming to life for every student in the sciences, engineering, and computer science. Coverage includes Basic elements of programming: variables, assignment statements, built-in data types, conditionals, loops, arrays, and I/O, including graphics and sound Functions, modules, and libraries: organizing programs into components that can be independently debugged, maintained, and reused Object-oriented programming and data abstraction: objects, modularity, encapsulation, and more Algorithms and data structures: sort/search algorithms, stacks, queues, and symbol tables Examples from applied math, physics, chemistry, biology, and computer science—all compatible with Python 2 and 3 Drawing on their extensive classroom experience, the authors provide Q&As, exercises, and opportunities for creative practice throughout. An extensive amount of supplementary information is available at introcs.cs.princeton.edu/python. With source code, I/O libraries, solutions to selected exercises, and much more, this companion website empowers people to use their own computers to teach and learn the material.

Python Essential Reference is the definitive reference guide to the Python programming language — the one authoritative handbook that reliably untangles and explains both the core Python language and the most essential parts of the Python library. Designed for the professional programmer, the book is concise, to the point, and highly accessible. It also includes detailed information on the Python library and many advanced subjects that is not available in either the official Python documentation or any other single reference source. Thoroughly updated to reflect the significant new programming language features and library modules that have been introduced in Python 2.6 and Python 3, the fourth edition of *Python Essential Reference* is the definitive guide for programmers who need to modernize existing Python code or who are planning an eventual migration to Python 3. Programmers starting a new Python project will find detailed coverage of contemporary Python programming idioms. This fourth edition of *Python Essential Reference* features numerous improvements, additions, and updates: Coverage of new language features, libraries, and modules Practical coverage of Python's more advanced features including generators, coroutines, closures, metaclasses, and decorators Expanded coverage of library modules related to concurrent programming including threads, subprocesses, and the new multiprocessing module Up-to-the-minute coverage of how to use Python 2.6's forward compatibility mode to evaluate code for Python 3 compatibility Improved organization for even faster answers and better usability Updates to reflect modern Python programming style and idioms Updated and improved example code Deep coverage of low-level system and networking library modules — including options not covered in the standard documentation

Get complete instructions for manipulating, processing, cleaning, and crunching datasets in Python. Updated for Python 3.6, the second edition of this hands-on guide is packed with practical case studies that show you how to solve a broad set of data analysis problems effectively. You'll learn the latest versions of pandas, NumPy, IPython, and Jupyter in the process. Written by Wes McKinney, the creator of the Python pandas project, this book is a

Read Book The Python Standard Library By Example

practical, modern introduction to data science tools in Python. It's ideal for analysts new to Python and for Python programmers new to data science and scientific computing. Data files and related material are available on GitHub. Use the IPython shell and Jupyter notebook for exploratory computing Learn basic and advanced features in NumPy (Numerical Python) Get started with data analysis tools in the pandas library Use flexible tools to load, clean, transform, merge, and reshape data Create informative visualizations with matplotlib Apply the pandas groupby facility to slice, dice, and summarize datasets Analyze and manipulate regular and irregular time series data Learn how to solve real-world data analysis problems with thorough, detailed examples

Cut through the noise and get real results with a step-by-step approach to learning Python 3.X programming Key Features Ideal for the Python beginner who is getting started for the first time A step-by-step Python tutorial with exercises and activities that help build key skills

Structured to let you progress at your own pace, on your own terms Use your physical print copy to redeem free access to the online interactive edition Book Description You already know you want to learn Python, and a smarter way to learn Python 3 is to learn by doing. The Python Workshop focuses on building up your practical skills so that you can work towards building up your machine learning skills as a data scientist, write scripts that help automate your life and save you time, or even create your own games and desktop applications. You'll learn from real examples that lead to real results. Throughout The Python Workshop, you'll take an engaging step-by-step approach to understanding Python. You won't have to sit through any unnecessary theory. If you're short on time you can jump into a single exercise each day or spend an entire weekend learning about Python scripting. It's your choice.

Learning on your terms, you'll build up and reinforce key skills in a way that feels rewarding. Every physical print copy of The Python Workshop unlocks access to the interactive edition. With videos detailing all exercises and activities, you'll always have a guided solution. You can also benchmark yourself against assessments, track progress, and receive free content updates. You'll even earn a secure credential that you can share and verify online upon completion. It's a premium learning experience that's included with your printed copy. To redeem, follow the instructions located at the start of your Python book. Fast-paced and direct, The Python Workshop is the ideal companion for Python beginners. You'll build and iterate on your code like a software developer, learning along the way. This process means that you'll find that your new skills stick, embedded as best practice. A solid foundation for the years ahead. What you will learn Learn how to write clean and concise code with Python 3

Understand classes and object-oriented programming Tackle entry-level data science and create engaging visualizations Use Python to create responsive, modern web applications Automate essential day-to-day tasks with Python scripts Get started with predictive Python machine learning Who this book is for This book is designed for professionals, students, and hobbyists who want to learn Python and apply it to solve challenging real-world problems. Although this is a beginner's book, it will help if you already know standard programming topics, such as variables, if-else statements, and functions. Experience with another object-oriented program is beneficial, but not mandatory.

An authorised reissue of the long out of print classic textbook, Advanced Calculus by the late Dr Lynn Loomis and Dr Shlomo Sternberg both of Harvard University has been a revered but hard to find textbook for the advanced calculus course for decades. This book is based on an honors course in advanced calculus that the authors gave in the 1960's. The foundational material, presented in the unstarred sections of Chapters 1 through 11, was normally covered, but different applications of this basic material were stressed from year to year, and the book therefore contains more material than was covered in any

Read Book The Python Standard Library By Example

one year. It can accordingly be used (with omissions) as a text for a year's course in advanced calculus, or as a text for a three-semester introduction to analysis. The prerequisites are a good grounding in the calculus of one variable from a mathematically rigorous point of view, together with some acquaintance with linear algebra. The reader should be familiar with limit and continuity type arguments and have a certain amount of mathematical sophistication. As possible introductory texts, we mention Differential and Integral Calculus by R Courant, Calculus by T Apostol, Calculus by M Spivak, and Pure Mathematics by G Hardy. The reader should also have some experience with partial derivatives. In overall plan the book divides roughly into a first half which develops the calculus (principally the differential calculus) in the setting of normed vector spaces, and a second half which deals with the calculus of differentiable manifolds.

This is the eBook of the printed book and may not include any media, website access codes, or print supplements that may come packaged with the bound book. Master the Powerful Python 3 Standard Library through Real Code Examples “The genius of Doug’s approach is that with 15 minutes per week, any motivated programmer can learn the Python Standard Library. Doug’s guided tour will help you flip the switch to fully power-up Python’s batteries.” –Raymond Hettinger, Distinguished Python Core Developer The Python 3 Standard Library contains hundreds of modules for interacting with the operating system, interpreter, and Internet—all extensively tested and ready to jump-start application development. Now, Python expert Doug Hellmann introduces every major area of the Python 3.x library through concise source code and output examples. Hellmann’s examples fully demonstrate each feature and are designed for easy learning and reuse. You’ll find practical code for working with text, data structures, algorithms, dates/times, math, the file system, persistence, data exchange, compression, archiving, crypto, processes/threads, networking, Internet capabilities, email, developer and language tools, the runtime, packages, and more. Each section fully covers one module, with links to additional resources, making this book an ideal tutorial and reference. The Python 3 Standard Library by Example introduces Python 3.x’s new libraries, significant functionality changes, and new layout and naming conventions. Hellmann also provides expert porting guidance for moving code from 2.x Python standard library modules to their Python 3.x equivalents. Manipulate text with string, textwrap, re (regular expressions), and difflib Use data structures: enum, collections, array, heapq, queue, struct, copy, and more Implement algorithms elegantly and concisely with functools, itertools, and contextlib Handle dates/times and advanced mathematical tasks Archive and data compression Understand data exchange and persistence, including json, dbm, and sqlite Sign and verify messages cryptographically Manage concurrent operations with processes and threads Test, debug, compile, profile, language, import, and package tools Control interaction at runtime with interpreters or the environment

Read Book The Python Standard Library By Example

Build software that combines Python's expressivity with the performance and control of C (and C++). It's possible with Cython, the compiler and hybrid programming language used by foundational packages such as NumPy, and prominent in projects including Pandas, h5py, and scikits-learn. In this practical guide, you'll learn how to use Cython to improve Python's performance—up to 3000x— and to wrap C and C++ libraries in Python with ease. Author Kurt Smith takes you through Cython's capabilities, with sample code and in-depth practice exercises. If you're just starting with Cython, or want to go deeper, you'll learn how this language is an essential part of any performance-oriented Python programmer's arsenal. Use Cython's static typing to speed up Python code Gain hands-on experience using Cython features to boost your numeric-heavy Python Create new types with Cython—and see how fast object-oriented programming in Python can be Effectively organize Cython code into separate modules and packages without sacrificing performance Use Cython to give Pythonic interfaces to C and C++ libraries Optimize code with Cython's runtime and compile-time profiling tools Use Cython's prange function to parallelize loops transparently with OpenMP

Get a comprehensive, in-depth introduction to the core Python language with this hands-on book. Based on author Mark Lutz's popular training course, this updated fifth edition will help you quickly write efficient, high-quality code with Python. It's an ideal way to begin, whether you're new to programming or a professional developer versed in other languages. Complete with quizzes, exercises, and helpful illustrations, this easy-to-follow, self-paced tutorial gets you started with both Python 2.7 and 3.3— the latest releases in the 3.X and 2.X lines—plus all other releases in common use today. You'll also learn some advanced language features that recently have become more common in Python code. Explore Python's major built-in object types such as numbers, lists, and dictionaries Create and process objects with Python statements, and learn Python's general syntax model Use functions to avoid code redundancy and package code for reuse Organize statements, functions, and other tools into larger components with modules Dive into classes: Python's object-oriented programming tool for structuring code Write large programs with Python's exception-handling model and development tools Learn advanced Python tools, including decorators, descriptors, metaclasses, and Unicode processing Learn the Python skills and culture you need to become a productive member of any Python project. About This Book Taking a practical approach to studying Python A clear appreciation of the sequence-oriented parts of Python Emphasis on the way in which Python code is structured Learn how to produce bug-free code by using testing tools Who This Book Is For The Python Apprentice is for anyone who wants to start building, creating and contributing towards a Python project. No previous knowledge of Python is required, although at least some familiarity with programming in another language is helpful. What You Will Learn Learn the language of Python itself Get a start on the Python standard library

Read Book The Python Standard Library By Example

Learn how to integrate 3rd party libraries Develop libraries on your own Become familiar with the basics of Python testing In Detail Experienced programmers want to know how to enhance their craft and we want to help them start as apprentices with Python. We know that before mastering Python you need to learn the culture and the tools to become a productive member of any Python project. Our goal with this book is to give you a practical and thorough introduction to Python programming, providing you with the insight and technical craftsmanship you need to be a productive member of any Python project. Python is a big language, and it's not our intention with this book to cover everything there is to know. We just want to make sure that you, as the developer, know the tools, basic idioms and of course the ins and outs of the language, the standard library and other modules to be able to jump into most projects. Style and approach We introduce topics gently and then revisit them on multiple occasions to add the depth required to support your progression as a Python developer. We've worked hard to structure the syllabus to avoid forward references. On only a few occasions do we require you to accept techniques on trust, before explaining them later; where we do, it's to deliberately establish good habits.

The Hitchhiker's Guide to Python takes the journeyman Pythonista to true expertise. More than any other language, Python was created with the philosophy of simplicity and parsimony. Now 25 years old, Python has become the primary or secondary language (after SQL) for many business users. With popularity comes diversity—and possibly dilution. This guide, collaboratively written by over a hundred members of the Python community, describes best practices currently used by package and application developers. Unlike other books for this audience, The Hitchhiker's Guide is light on reusable code and heavier on design philosophy, directing the reader to excellent sources that already exist.

Learn how to program with Python from beginning to end. This book is for beginners who want to get up to speed quickly and become intermediate programmers fast!

[Copyright: f349010a788dce98a945c8280a787bde](https://www.python.org/psf/license/)