

building APIs with Python 3.6 and its later versions based on standard Python-type hints. With this book, you'll be able to create fast and reliable data science API backends using practical examples. This book starts with the basics of the FastAPI framework and associated modern Python programming language concepts. You'll then be taken through all the aspects of the framework, including its powerful dependency injection system and how you can use it to communicate with databases, implement authentication and integrate machine learning models. Later, you'll cover best practices relating to testing and deployment to run a high-quality and robust application. You'll also be introduced to the extensive ecosystem of Python data science packages. As you progress, you'll learn how to build data science applications in Python using FastAPI. The book also demonstrates how to develop fast and efficient machine learning prediction backends and test them to achieve the best performance. Finally, you'll see how to implement a real-time face detection system using WebSockets and a web browser as a client. By the end of this FastAPI book, you'll have not only learned how to implement Python in data science projects but also how to maintain and design them to meet high programming standards with the help of FastAPI. What You Will Learn: Explore the basics of modern Python and async I/O programming Get to grips with basic and advanced concepts of the FastAPI framework Implement a FastAPI dependency to efficiently run a machine learning model Integrate a simple face detection algorithm in a FastAPI backend Integrate common Python data science libraries in a web backend Deploy a performant and reliable web backend for a data science application Who this book is for: This Python data science book is for data scientists and software developers interested in gaining knowledge of FastAPI and its ecosystem to build data science applications. Basic knowledge of data science and machine learning concepts and how to apply them in Python is recommended.

Your Python code may run correctly, but you need it to run faster. By exploring the fundamental theory behind design choices, the updated edition of this practical guide, expanded and enhanced for Python 3, helps you gain a deeper understanding of Python's implementation. You'll learn how to locate performance bottlenecks and significantly speed up your code in high-data-volume programs. How can you take advantage of multicore architectures or clusters? Or build a system that can scale up and down without losing reliability? Experienced Python programmers and students alike will learn concrete solutions to these and other issues, along with war stories from companies that use high-performance Python for social media analytics, productionized machine learning, and other situations. Get a better grasp of NumPy, Cython, and profilers Learn how Python abstracts the underlying computer architecture Understand performant pandas Use profiling to find bottlenecks in CPU time and memory usage Write efficient programs by choosing appropriate data structures Speed up matrix, vector, and even tensor computations Use tools to compile Python down to machine code, on CPUs and GPUs Manage multiple I/O and computational operations concurrently Convert multiprocessing code to run on a local or remote cluster Deploy code faster using tools like Docker Solve large problems while using less RAM Get real-life stories and lessons from Python programmers

????????????????,“??”????????????????????????????/????

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

????????????????????,????:????????;????????;????????;????????;????????;????????;????????;????????

Apache Spark is amazing when everything clicks. But if you haven't seen the performance improvements you expected, or still don't feel confident enough to use Spark in production, this practical book is for you. Authors Holden Karau and Rachel Warren demonstrate performance optimizations to help your Spark queries run faster and handle larger data sizes, while using fewer resources. Ideal for software engineers, data engineers, developers, and system administrators working with large-scale data applications, this book describes techniques that can reduce data infrastructure costs and developer hours. Not only will you gain a more comprehensive understanding of Spark, you'll also learn how to make it sing. With this book, you'll explore: How Spark SQL's new interfaces improve performance over SQL's RDD data structure The choice between data joins in Core Spark and Spark SQL Techniques for getting the most out of standard RDD transformations How to work around performance issues in Spark's key/value pair paradigm Writing high-performance Spark code without Scala or the JVM How to test for functionality and performance when applying suggested improvements Using Spark MLlib and Spark ML machine learning libraries Spark's Streaming components and external community packages

????????????????????????????????,???C++??

React 16 Tooling covers the most important tools, utilities, and libraries that every React developer needs to know -- in detail. Key Features Each chapter presents meta-development solutions to help React developers The tools used are presented in a practical, solution-oriented approach with no fluff The chapters are arranged in a logical order

Get Free High Performance Python Practical Performant Programming For Humans

The Go Workshop takes you from being a novice Go programmer to a confident developer who can leverage the key features of the language to build real-world applications. This book helps you cut through excessive theory and delve into the practical features and techniques that are commonly applied to design performant, scalable applications.

[Copyright: 8c24fc30d734a7787518c6212bd8259b](#)