Test Driven Development With Python Obey The Testing Goat Using Django Selenium And Javascript

Your code is a testament to your skills as a developer. No matter what language you use, code should be clean, elegant, and uncluttered. By using test-driven development (TDD), you'll write code that's easy to understand, retains its elegance, and works for months, even years, to come. With this indispensable guide, you'll learn how to use TDD with three different languages: Go, JavaScript, and Python. Author Saleem Siddiqui shows you how to tackle domain complexity using a unit test-driven approach. TDD partitions requirements into small, implementable features, enabling you to solve problems irrespective of the languages and frameworks you use. With Learning Test-Driven Development at your side, you'll learn how to incorporate TDD into your regular coding practice. This book helps you: Use TDD's divide-and-conquer approach to tame domain complexity Understand how TDD works across languages, testing frameworks, and domain concepts Learn how TDD enables continuous integration Support refactoring and redesign with TDD Learn how to write a simple and effective unit test harness in JavaScript Set up a continuous integration environment with the unit tests produced during TDD Write clean, uncluttered code using TDD in Go, JavaScript, and Python Learn the fundamentals of Python (3.7) and how to apply it to data science, programming, and web development. Fully updated to include hands-on tutorials and projects. Key Features Learn the fundamentals of Python programming with interactive projects Apply Python to data science with tools such as IPython and Jupyter Utilize Python for web development and build a

real-world app using Django Book Description Learn Python Programming is a quick, thorough, and practical introduction to Python - an extremely flexible and powerful programming language that can be applied to many disciplines. Unlike other books, it doesn't bore you with elaborate explanations of the basics but gets you up-and-running, using the language. You will begin by learning the fundamentals of Python so that you have a rock-solid foundation to build upon. You will explore the foundations of Python programming and learn how Python can be manipulated to achieve results. Explore different programming paradigms and find the best approach to a situation; understand how to carry out performance optimization and effective debugging; control the flow of a program; and utilize an interchange format to exchange data. You'll also walk through cryptographic services in Python and understand secure tokens. Learn Python Programming will give you a thorough understanding of the Python language. You'll learn how to write programs, build websites, and work with data by harnessing Python's renowned data science libraries. Filled with real-world examples and projects, the book covers various types of applications, and concludes by building real-world projects based on the concepts you have learned. What you will learn Get Python up and running on Windows, Mac, and Linux Explore fundamental concepts of coding using data structures and control flow Write elegant, reusable, and efficient code in any situation Understand when to use the functional or OOP approach Cover the basics of security and concurrent/asynchronous programming Create bulletproof, reliable software by writing tests Build a simple website in Django Fetch, clean, and manipulate data Who this book is for Learn Python Programming is for individuals with relatively little experience in coding or Python. It's also ideal for aspiring programmers who need to write scripts or programs to accomplish tasks. The book shows you how to create a full-

fledged application.

This book is intended for Python developers who want to use the principles of test-driven development (TDD) to create efficient and robust applications. In order to get the best out of this book, you should have development experience with Python.

Write clean code that works with the help of this groundbreaking software method. Example-driven teaching is the basis of Beck's step-by-step instruction that will have readers using TDD to further their projects.

As Python continues to grow in popularity, projects are becoming larger and more complex. Many Python developers are now taking an interest in high-level software architecture patterns such as hexagonal/clean architecture, event-driven architecture, and strategic patterns prescribed by domain-driven design (DDD). But translating those patterns into Python isn't always straightforward. With this practical guide, Harry Percival and Bob Gregory from MADE.com introduce proven architectural design patterns to help Python developers manage application complexity. Each pattern is illustrated with concrete examples in idiomatic Python that explain how to avoid some of the unnecessary verbosity of Java and C# syntax. You'll learn how to implement each of these patterns in a Pythonic way. Architectural design patterns include: Dependency inversion, and its links to ports and adapters (hexagonal/clean architecture) Domain-driven design's distinction between entities, value objects, and aggregates Repository and Unit of Work patterns for persistent storage Events, commands, and the message bus Command Query Responsibility Segregation (CQRS) Event-driven architecture and reactive microservices

Bewährte Patterns für komplexe Python-Projekte bekannte Architekturpatterns - endlich in

idiomatischem Python die Komplexität anspruchsvoller Projekte erfolgreich managen den größten Nutzen aus den Testsuiten herausholen Pythons Popularität wächst weiterhin und mit Python werden inzwischen komplexe Projekte realisiert. Viele Python-Entwicklerinnen und -Entwickler interessieren sich deshalb für High-Level-Design-Patterns wie hexagonale Architektur, ereignisgesteuerte Architektur und die strategischen Patterns, die durch das Domain-Driven Design vorgegeben sind. Das Übertragen dieser Patterns nach Python ist allerdings nicht immer einfach. In diesem Praxisbuch stellen Harry Percival und Bob Gregory von MADE.com erprobte Architekturpatterns vor, die Python-Entwickler dabei unterstützen, die Komplexität von Anwendungen im Griff zu behalten – und den größtmöglichen Nutzen aus den Testsuiten zu ziehen. Jedes Pattern wird durch Beispiele in schönem, idiomatischem Python illustriert; dabei wird die Weitschweifigkeit der Java- oder C#-Syntax vermieden. Ao conduzir você pelo desenvolvimento de uma verdadeira aplicação web do início ao fim, este guia "mão na massa" mostra as vantagens práticas do TDD (Test-Driven Development, ou Desenvolvimento Orientado a Testes) com Python. Você aprenderá a escrever e a executar testes antes de construir cada parte de sua aplicação, e então desenvolverá a quantidade mínima de código necessária para que os testes passem. O resultado? Um código limpo, que funcionará. Durante o processo, você conhecerá o básico sobre Django, Selenium, Git, jQuery e Mock, junto com as técnicas atuais para desenvolvimento web. Se estiver pronto para levar suas habilidades com Python para o próximo patamar, este livro – atualizado com Python 3.6 – mostrará claramente como o TDD incentiva você a criar designs simples e pode lhe inspirar confiança. ? mergulhe no fluxo de trabalho de TDD, incluindo o ciclo de testes de unidade/código e refatoração; ? utilize testes de unidade para classes e funções, e testes

funcionais para interações com usuários no navegador; ? saiba quando e como usar objetos simulados e conheça os prós e contras dos testes isolados versus testes integrados; ? teste e automatize suas implantações com um servidor de staging; ? aplique testes nos plugins de terceiros que você integrar ao seu site; ? execute testes automaticamente usando um ambiente de Integração Contínua; ? utilize TDD para construir uma API REST com uma interface de frontend Ajax.

As Python continues to grow in popularity, projects are becoming larger and more complex. Many Python developers are now taking an interest in high-level software design patterns such as hexagonal/clean architecture, event-driven architecture, and the strategic patterns prescribed by domain-driven design (DDD). But translating those patterns into Python isn't always straightforward. With this hands-on guide, Harry Percival and Bob Gregory from MADE.com introduce proven architectural design patterns to help Python developers manage application complexity—and get the most value out of their test suites. Each pattern is illustrated with concrete examples in beautiful, idiomatic Python, avoiding some of the verbosity of Java and C# syntax. Patterns include: Dependency inversion and its links to ports and adapters (hexagonal/clean architecture) Domain-driven design's distinction between entities, value objects, and aggregates Repository and Unit of Work patterns for persistent storage Events, commands, and the message bus Command-query responsibility segregation (CQRS) Event-driven architecture and reactive microservices

Master the art of using Python for a diverse range of network engineering tasks Key Features Explore the power of Python libraries to tackle difficult network problems efficiently and effectively Use Python for network device automation, DevOps, and software-defined

networking Become an expert in implementing advanced network-related tasks with Python Book Description Networks in your infrastructure set the foundation for how your application can be deployed, maintained, and serviced. Python is the ideal language for network engineers to explore tools that were previously available to systems engineers and application developers. In this second edition of Mastering Python Networking, you'll embark on a Python-based journey to transition from traditional network engineers to network developers ready for the next-generation of networks. This book begins by reviewing the basics of Python and teaches you how Python can interact with both legacy and API-enabled network devices. As you make your way through the chapters, you will then learn to leverage high-level Python packages and frameworks to perform network engineering tasks for automation, monitoring, management, and enhanced security. In the concluding chapters, you will use Jenkins for continuous network integration as well as testing tools to verify your network. By the end of this book, you will be able to perform all networking tasks with ease using Python. What you will learn Use Python libraries to interact with your network Integrate Ansible 2.5 using Python to control Cisco, Juniper, and Arista eAPI network devices Leverage existing frameworks to construct high-level APIs Learn how to build virtual networks in the AWS Cloud Understand how Jenkins can be used to automatically deploy changes in your network Use PyTest and Unittest for Test-Driven Network Development Who this book is for Mastering Python Networking is for network engineers and programmers who want to use Python for networking. Basic familiarity with Python programming and networking-related concepts such as Transmission Control Protocol/Internet Protocol (TCP/IP) will be useful.
By taking you through the development of a real web application from beginning to end, this

hands-on guide demonstrates the practical advantages of test-driven development (TDD) with Python. You{u2019}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{u2019}ll learn the basics of Django, Selenium, Git, jQuery, and Mock, along with current web development techniques. If you{u2019}re ready to take your Python skills to the next level, this book 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 site Use a Continuous Integration environment to run your tests automatically.

Discover how to develop reliable, high-quality Python code with unit testing and test-driven development.

Learn how to automate unit tests of Python 3 with automation libraries, such as doctest, unittest, nose, nose2, pytest, and selenium. This book explores important concepts in software test automation and demonstrates how to automate, organize, and execute unit tests with Python. It also introduces readers to the concepts of web browser automation and logging. This new edition starts with an introduction to Python 3. Next, it covers doctest and pydoc. This is followed by a discussion on unittest, a framework that comes packaged with Python 3 itself. There is a dedicated section on creating test suites, followed by an explanation of how nose2 provides automatic test module discovery. Moving forward, you will learn about pytest, the

most popular third-party library and testrunner for Python. You will see how to write and execute tests with pytest. You'll also learn to discover tests automatically with pytest. This edition features two brand new chapters, the first of which focuses on the basics of web browser automation with Selenium. You'll learn how to use Selenium with unittest to write test cases for browser automation and use the Selenium IDE with web browsers such as Chrome and Firefox. You'll then explore logging frameworks such as Python's built-in logger and the third-party framework loguru. The book concludes with an exploration of test-driven development with pytest, during which you will execute a small project using TDD methodology. What You Will Learn Start testing with doctest and unittest Understand the idea of unit testing Get started with nose 2 and pytest Learn how to use logger and loguru Work with Selenium and test driven development Who This Book Is ForPython developers, software testers, open source enthusiasts, and contributors to the Python community.

This practical guide to test-driven development will help developers working with Python put their knowledge to work. You'll learn how to adopt an effective test-driven approach to Python software development with WebTest and web frameworks. The book is filled with hands-on examples to enable you to write reliable test suites to ensure your ...

"In this video tutorial, you'll learn about the PyTest testing library and how it's used to write unit tests in Python. You'll also set up some common Python development environments to use PyTest. You'll create isolated test environments with Test Doubles and learn how to implement and use them with unittest.mock. Moving on, you'll get to know some of the best practices in Unit Testing and TDD and get some hands-on experience with programming by implementing unit tests using TDD in Python. By the end of this course, you'll be able to apply the practices

of Unit Testing and TDD on a daily basis to radically increase the quality of your code and help you and your company achieve your goals faster than ever before."--Resource description page.

Fix everyday testing problems in Python with the help of this solution-based guide Key Features Use powerful tools such as doctest and unittest to make testing convenient Apply automation testing to an existing legacy system that isn't test oriented A practical guide to ease testing in Python using real-world examples Book Description Automated testing is the best way to increase efficiency while reducing the defects of software testing. It helps find bugs in code easily and at an early stage so that they can be tackled efficiently. This book delves into essential testing concepts used in Python to help you build robust and maintainable code. Python Testing Cookbook begins with a brief introduction to Python's unit testing framework to help you write automated test cases. You will learn how to write suitable test sets for your software and run automated test suites with Nose. You will then work with the unittest.mock library, which allows you to replace the parts of your system that are being tested with mock objects and make assertions about how they have been used. You will also see how to apply Test-driven Development (TDD) and Behavior-driven Development (BDD) and how to eliminate issues caused by TDD. The book explains how to integrate

automated tests using Continuous Integration and perform smoke/load testing. It also covers best practices and will help you solve persistent testing issues in Python. The book concludes by helping you understand how doctest works and how Selenium can be used to test code efficiently. What you will learn Run test cases from the command line with increased verbosity Write a Nose extension to pick tests based on regular expressions Create testable documentation using doctest Use Selenium to test the Web User Interface Write a testable story with Voidspace Mock and Nose Configure TeamCity to run Python tests on commit Update project-level scripts to provide coverage reports Who this book is for If you're a Python developer who wants to take testing to the next level and would like to expand your testing skills, this book is for you. It is assumed that you have some Python programming knowledge.

Your code is a testament to your skills as a developer. No matter what language you use, your code should be clean, elegant, and uncluttered? With test-driven development (TDD), you'll write better code--code that's easy to understand, retains its elegance, and works for years to come. This indispensable guide will show you how TDD works in three different languages: Go, JavaScript, and Python. With Learning Test-Driven Development at your side, you'll be able to: Tame domain complexity using a divide-and-conquer approach Understand how

TDD works across languages, testing frameworks, and domain concepts See how TDD enables continuous integration and continuous delivery Support refactoring and redesign with TDD Set up a continuous integration environment with the unit tests produced during TDD Write clean, uncluttered code using TDD in Go, JavaScript, and Python

Explore MicroPython through a series of hands-on projects and learn to design and build your own embedded systems using the MicroPython Pyboard, ESP32, the STM32 IoT Discovery kit, and the OpenMV camera module. Key Features Delve into MicroPython Kernel and learn to make modifications that will enhance your embedded applications Design and implement drivers to interact with a variety of sensors and devices Build low-cost projects such as DIY automation and object detection with machine learning Book Description With the increasing complexity of embedded systems seen over the past few years, developers are looking for ways to manage them easily by solving problems without spending a lot of time on finding supported peripherals. MicroPython is an efficient and lean implementation of the Python 3 programming language, which is optimized to run on microcontrollers. MicroPython Projects will guide you in building and managing your embedded systems with ease. This book is a comprehensive project-based guide that will help you build a wide range of projects and give you

the confidence to design complex projects spanning new areas of technology such as electronic applications, automation devices, and IoT applications. While building seven engaging projects, you'll learn how to enable devices to communicate with each other, access and control devices over a TCP/IP socket, and store and retrieve data. The complexity will increase progressively as you work on different projects, covering areas such as driver design, sensor interfacing, and MicroPython kernel customization. By the end of this MicroPython book, you'll be able to develop industry-standard embedded systems and keep up with the evolution of the Internet of Things. What you will learn Develop embedded systems using MicroPython Build a custom debugging tool to visualize sensor data in real-time Detect objects using machine learning and MicroPython Discover how to minimize project costs and reduce development time Get to grips with gesture operations and parsing gesture data Learn how to customize and deploy the MicroPython kernel Explore the techniques for scheduling application tasks and activities Who this book is for If you are an embedded developer or hobbyist looking to build interesting projects using MicroPython, this book is for you. A basic understanding of electronics and Python is required while some MicroPython experience will be helpful. 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!

This guide for programmers teaches how to practice Test Driven Development (TDD), also called Test First Development. Contrary to the accepted approach to testing, when you practice TDD you write tests for code before you write the code being tested. This text provides examples in Java.

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 site Run tests automatically by using a Continuous Integration environment Use TDD to build a REST API with a front-end Ajax interface This book is for Django developers with little or no knowledge of test-driven development or testing in general. Familiarity with the command line, setting up a Python virtual environment, and starting a Django project are assumed. ***** WAGmob: Over One million Paying Customers ***** WAGmob brings you, simpleNeasy, on-the-go learning ebook for "Test Driven Development". The ebook provides: Snack sized chapters for easy learning. Designed for both students and adults. This ebook provides a quick summary of essential concepts in Test Driven Development by following snack sized chapters: Introduction: • Introduction • Test First Development (TFD) • Benefits of Test-Driven Development • Process Example to TDD Approach Introduction to Unit Testing: • What is Unit Testing? • Method • When is it Performed? • Who Performs it? • Benefits of Unit Testing • Mock Objects • Why Mocking is Important? • Test Double • Types of Test Doubles A Quick Review of Refactoring: • What is Code Refactoring? • Overview of Refactoring • Why do You Refactor? • When do You Refactor? • Steps for Refactoring • Two Categories of Benefits to the Activity of Refactoring Refactoring Examples: • Refactoring Examples • Rename Class/

Method/ Variables • Method Slicing/Extraction • Architecture Driven Refactoring – Modularity • Movement of Methods or Class • Code to Interface • Constructors Chaining Phases of Test Driven Development: • Steps to be followed in Test Driven Development • Test Structure • Shortcomings Software of Test Driven Development: • Software for Test Driven Development • CppUTest • csUnit • DbUnit • jMock • JUnit • NUnit • PHPUnit Integration Testing: • Integration Testing • Why is Integration Testing Required? • Big Bang • Top Down • Bottom Up • Limitations GUI Testing: • GUI Testing • Text Based GUI Testing Framework • Introducing Bailey Testing Framework (Graphic based GUI Testing Framework) • How it Works? • Pseudo Code .NET TDD Iteration I: • .NET TDD (Test Driven Development) by Example • Introduction • Development Costs • Sample Code • The Tools • Iteration I • Creating the Libraries • Going Back to the Requirements • First Two Tests – RED • Get the Tests Failing with the Minimal Amount of Code • Using the Test Explorer to View and Run the Tests • Make the Test Pass (Green) • Make Some Changes .NET TDD Iteration II: • Iteration II • Introduce More Tests (Red) • Make the Test Pass (a second time; Green) • Debugging Tests About WAGmob ebooks: 1) A companion ebook for on-the-go, bite-sized learning. 2) Over One million paying customers from 175+ countries. Why WAGmob ebooks: 1) Beautifully simple, Amazingly easy, Massive

selection of ebooks. 2) Effective, Engaging and Entertaining ebooks. 3) An incredible value for money. Lifetime of free updates! WAGmob Vision : simpleNeasy ebooks for a lifetime of on-the-go learning WAGmob Mission : A simpleNeasy WAGmob ebook in every hand. Visit us : www.SimpleNEasyBook.Com Please write to us at Team (at)simpleNeasyBook.Com. We would love to improve this Book. Fundamental testing methodologies applied to the popular Pythonlanguage Testing Python; Applying Unit Testing, TDD, BDD andAcceptance Testing is the most comprehensive book available ontesting for one of the top software programming languages in theworld. Python is a natural choice for new and experienceddevelopers, and this hands-on resource is a much needed guide toenterprise-level testing development methodologies. The book willshow you why Unit Testing and TDD can lead to cleaner, moreflexible programs. Unit Testing and Test-Driven Development (TDD) are increasinglymust-have skills for software developers, no matter what languagethey work in. In enterprise settings, it's critical for developersto ensure they always have working code, and that's what makestesting methodologies so attractive. This book will teach you themost widely used testing strategies and will introduce to you tostill others, covering performance testing, continuous testing, andmore. Learn Unit Testing and

TDD—important developmentmethodologies that lie at the heart of Agile development Enhance your ability to work with Python to develop powerful,flexible applications with clean code Draw on the expertise of author David Sale, a leading UKdeveloper and tech commentator Get ahead of the crowd by mastering the underappreciated worldof Python testing Knowledge of software testing in Python could set you apart fromPython developers using outmoded methodologies. Python is a naturalfit for TDD and Testing Python is a must-read text foranyone who wants to develop expertise in Python programming.

The agile development movement represents the latest advances in tools and techniques intended to boost developer productivity. This is the first book to apply these sought after principles to Python developers, introducing both the tools and techniques built and supported by the Python community. Authored by Jeff Younker, who is perhaps best known for his creation of a popular Python testing framework, this book is sure to be a hit among readers who may have reached their limits of knowledge regarding the Python language, yet are seeking to improve their understanding of how sound processes can boost productivity to unparalleled heights.

By taking you through the development of a real web application from beginning

to end, 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 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 site Use a Continuous Integration environment to run your tests automatically

If you program in C++ you've been neglected. Test-driven development (TDD) is a modern software development practice that can dramatically reduce the number of defects in systems, produce more maintainable code, and give you the confidence to change your software to meet changing needs. But C++ programmers have been ignored by those promoting TDD--until now. In this

book, Jeff Langr gives you hands-on lessons in the challenges and rewards of doing TDD in C++. Modern C++ Programming With Test-Driven Development, the only comprehensive treatment on TDD in C++ provides you with everything you need to know about TDD, and the challenges and benefits of implementing it in your C++ systems. Its many detailed code examples take you step-by-step from TDD basics to advanced concepts. As a veteran C++ programmer, you're already writing high-quality code, and you work hard to maintain code quality. It doesn't have to be that hard. In this book, you'll learn: how to use TDD to improve legacy C++ systems how to identify and deal with troublesome system dependencies how to do dependency injection, which is particularly tricky in C++ how to use testing tools for C++ that aid TDD new C++11 features that facilitate TDD As you grow in TDD mastery, you'll discover how to keep a massive C++ system from becoming a design mess over time, as well as particular C++ trouble spots to avoid. You'll find out how to prevent your tests from being a maintenance burden and how to think in TDD without giving up your hard-won C++ skills. Finally, you'll see how to grow and sustain TDD in your team. Whether you're a complete unit-testing novice or an experienced tester, this book will lead you to mastery of test-driven development in C++. What You Need A C++ compiler running under Windows or Linux, preferably one that supports C++11. Examples

presented in the book were built under gcc 4.7.2. Google Mock 1.6 (downloadable for free; it contains Google Test as well) or an alternate C++ unit testing tool. Most examples in the book are written for Google Mock, but it isn't difficult to translate them to your tool of choice. A good programmer's editor or IDE. cmake, preferably. Of course, you can use your own preferred make too. CMakeLists.txt files are provided for each project. Examples provided were built using cmake version 2.8.9. Various freely-available third-party libraries are used as the basis for examples in the book. These include:- cURL- JsonCpp- Boost (filesystem, date_time/gregorian, algorithm, assign)Several examples use the boost headers/libraries. Only one example uses cURL and JsonCpp. ALLES STAAT OP HET SPEL DE JACHT OP DE ULTIEME PRIJS ARE YOU READY? In 2045 leeft een aanzienlijk deel van de wereldbevolking in diepe armoede. De enige manier waarop Wade Watts aan alle misère kan ontsnappen, is de OASIS: een online virtuele wereld waar miljarden mensen tegelijk in kunnen vertoeven. Wanneer James Halliday, de excentrieke uitvinder van de OASIS, overlijdt, laat hij een zoektocht achter vol puzzels die te maken hebben met zijn obsessie met de popcultuur van de jaren tachtig. Degene die de jacht als eerste voltooit, zal het vermogen van Halliday erven én de volledige controle krijgen over de OASIS. Wanneer Wade als allereerste een van de puzzels oplost, wordt

hij van alle kanten opgejaagd door nietsontziende rivalen. Vanaf dan is er voor Wade nog maar één manier om te overleven: winnen.

Roman over de levenslange vriendschap tussen twee echtparen Nawoord Jane Smiley TEN TIJDE VAN DE GROTE DEPRESSIE ontmoeten twee jonge stellen elkaar voor het eerst. Charity en Sally zijn in verwachting, Sid en Larry werken beiden aan de faculteit Engels van de Universiteit van Wisconsin. Een levenslange vriendschap is geboren. Bijna vier decennia lang delen de echtparen lief en leed. Wat behouden blijft, geschreven vanuit het perspectief van de oude Larry, is een roman die uitblinkt in stil realisme en een diep mededogen, waaruit de levenswijsheid blijkt van de auteur, die het boek - zijn laatste roman - op 78-jarige leeftijd schreef. WALLACE STEGNER (1909-1993) was historicus, milieuactivist en auteur van een bijzonder omvangrijk oeuvre, waaronder klassieke romans als Angle of Repose (Pulitzer Prize 1972), The Spectator Bird (National Book Award 1977) en Recapitulation (1979). In 1980 ontving hij de eerste Robert Kirsch Award, voor zijn bijdrage aan de Amerikaanse letteren. 'Een wonderbaarlijk boek, geschreven met de wijsheid van ouderen, maar zonder oud te lijken. Een roman met een immens narratieve kracht' THE INDEPENDENT 'Een magnifiek gecomponeerde roman die bruist van levenswijsheid' THE WASHINGTON POST

Handboek voor de uitvoering van ICT-projecten volgens een internationale, gezaghebbende standaard.

Unleash the power of Python 3 objects About This Book Stop writing scripts and start architecting programs Learn the latest Python syntax and libraries A practical, hands-on tutorial that teaches you all about abstract design patterns and how to implement them in Python 3 Who This Book Is For If you're new to object-oriented programming techniques, or if you have basic Python skills and wish to learn in depth how and when to correctly apply object-oriented programming in Python to design software, this is the book for you. What You Will Learn Implement objects in Python by creating classes and defining methods Separate related objects into a taxonomy of classes and describe the properties and behaviors of those objects via the class interface Extend class functionality using inheritance Understand when to use object-oriented features, and more importantly when not to use them Discover what design patterns are and why they are different in Python Uncover the simplicity of unit testing and why it's so important in Python Grasp common concurrency techniques and pitfalls in Python 3 Exploit object-oriented programming in key Python technologies such as Kivy and Django. Object-oriented programming concurrently with asyncio In Detail Python 3 is more versatile and easier to use than ever. It runs on all major

platforms in a huge array of use cases. Coding in Python minimizes development time and increases productivity in comparison to other languages. Clean, maintainable code is easy to both read and write using Python's clear, concise syntax. Object-oriented programming is a popular design paradigm in which data and behaviors are encapsulated in such a way that they can be manipulated together. Many modern programming languages utilize the powerful concepts behind object-oriented programming and Python is no exception. Starting with a detailed analysis of object-oriented analysis and design, you will use the Python programming language to clearly grasp key concepts from the object-oriented paradigm. This book fully explains classes, data encapsulation, inheritance, polymorphism, abstraction, and exceptions with an emphasis on when you can use each principle to develop well-designed software. You'll get an in-depth analysis of many common object-oriented design patterns that are more suitable to Python's unique style. This book will not just teach Python syntax, but will also build your confidence in how to program. You will also learn how to create maintainable applications by studying higher level design patterns. Following this, you'll learn the complexities of string and file manipulation, and how Python distinguishes between binary and textual data. Not one, but two very powerful automated testing systems will be introduced in the book. After you discover the

joy of unit testing and just how easy it can be, you'll study higher level libraries such as database connectors and GUI toolkits and learn how they uniquely apply object-oriented principles. You'll learn how these principles will allow you to make greater use of key members of the Python eco-system such as Django and Kivy. This new edition includes all the topics that made Python 3 Object-oriented Programming an instant Packt classic. It's also packed with updated content to reflect recent changes in the core Python library and covers modern third-party packages that were not available on the Python 3 platform when the book was first published. Style and approach Throughout the book you will learn key object-oriented programming techniques demonstrated by comprehensive case studies in the context of a larger project.

Quickly learn how to automate unit testing of Python 3 code with Python 3 automation libraries, such as doctest, unittest, nose, nose2, and pytest. This book explores the important concepts in software testing and their implementation in Python 3 and shows you how to automate, organize, and execute unit tests for this language. This knowledge is often acquired by reading source code, manuals, and posting questions on community forums, which tends to be a slow and painful process. Python Unit Test Automation will allow you to quickly ramp up your understanding of unit test libraries for Python 3 through the

practical use of code examples and exercises. All of which makes this book a great resource for software developers and testers who want to get started with unit test automation in Python 3 and compare the differences with Python 2. This short work is your must-have quick start guide to mastering the essential concepts of software testing in Python. What You'll Learn: Essential concepts in software testing Various test automation libraries for Python, such as doctest, unittest, nose, nose2, and pytest Test-driven development and best practices for test automation in Python Code examples and exercises Who This Book Is For: Python developers, software testers, open source enthusiasts, and contributors to the Python community

The book begins with the very foundations of automated testing, and expands on them until the best-practice tools and techniques are fully covered. New concepts are illustrated with step-by-step hands-on exercises. Testing will be easier and more enjoyable with this beginner's guide. If you are a Python developer and want to write tests for your applications, this book will get you started and show you the easiest way to learn testing. You need to have sound Python programming knowledge to follow along. An awareness of software testing would be good, but no formal knowledge of testing is expected nor do you need to have any knowledge of the libraries discussed in the book.

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 site Run tests automatically by using a Continuous Integration environment Use TDD to build a REST API with a front-end Ajax interface You've learned the basics of Python, but how do you take your skills to the next stage? Even if you know enough to be productive, there are a number of features that can take you to the next level in Python. Pro Python, Second Edition

explores concepts and features normally left to experimentation, allowing you to be even more productive and creative. In addition to pure code concerns, Pro Python develops your programming techniques and approaches, which will help make you a better Python programmer. This book will improve not only your code but also your understanding and interaction with the many established Python communities. This book takes your Python knowledge and coding skills to the next level. It shows you how to write clean, innovative code that will be respected by your peers. With this book, make your code do more with introspection and meta-programming. And learn and later use the nuts and bolts of an application, tier-by-tier as a complex case study along the way. For more information, including a link to the source code referenced in the book, please visit http://propython.com/.

Harness the power of Python objects and data structures to implement algorithms for analyzing your data and efficiently extracting information Key Features Turn your designs into working software by learning the Python syntax Write robust code with a solid understanding of Python data structures Understand when to use the functional or the OOP approach Book Description This Learning Path helps you get comfortable with the world of Python. It starts with a thorough and practical introduction to Python. You'll quickly start writing

programs, building websites, and working with data by harnessing Python's renowned data science libraries. With the power of linked lists, binary searches, and sorting algorithms, you'll easily create complex data structures, such as graphs, stacks, and queues. After understanding cooperative inheritance, you'll expertly raise, handle, and manipulate exceptions. You will effortlessly integrate the object-oriented and not-so-object-oriented aspects of Python, and create maintainable applications using higher level design patterns. Once you've covered core topics, you'll understand the joy of unit testing and just how easy it is to create unit tests. By the end of this Learning Path, you will have built components that are easy to understand, debug, and can be used across different applications. This Learning Path includes content from the following Packt products: Learn Python Programming - Second Edition by Fabrizio Romano Python Data Structures and Algorithms by Benjamin Baka Python 3 Object-Oriented Programming by Dusty Phillips What you will learn Use data structures and control flow to write code Use functions to bundle together a sequence of instructions Implement objects in Python by creating classes and defining methods Design public interfaces using abstraction, encapsulation and information hiding Raise, define, and manipulate exceptions using special error objects Create bulletproof and reliable software by writing unit tests Learn the

common programming patterns and algorithms used in Python Who this book is for If you are relatively new to coding and want to write scripts or programs to accomplish tasks using Python, or if you are an object-oriented programmer for other languages and seeking a leg up in the world of Python, then this Learning Path is for you. Though not essential, it will help you to have basic knowledge of programming and OOP.

Copyright: 21d085d6ab8c0d5364e7e8ec0aee5a7a