

1. Record Nr.	UNISA996207437203316
Titolo	Cell division
Pubbl/distr/stampa	[London], : BioMed Central, 2006-
ISSN	1747-1028
Soggetti	Cell division Biology Cell Division Fulltext Internet Resources. Periodicals. Periodical
Lingua di pubblicazione	Inglese
Formato	Materiale a stampa
Livello bibliografico	Periodico
Note generali	Refereed/Peer-reviewed Title from PDF caption (publisher's Web site, viewed Apr. 11, 2006).

2. Record Nr.	UNINA9911046596203321
Autore	Peacock Ashley
Titolo	Creating software with modern diagramming techniques : build better software with Mermaid // Ashley Peacock
Pubbl/distr/stampa	Raleigh, North Carolina : , : The Pragmatic Bookshelf, , [2023] ©2023
ISBN	9798888650226 9781680509830
Edizione	[First edition.]
Descrizione fisica	1 online resource (149 pages)
Collana	Pragmatic exPress Pragmatic programmers
Disciplina	658.4
Soggetti	Flow charts JavaScript (Computer program language)
Lingua di pubblicazione	Inglese
Formato	Materiale a stampa
Livello bibliografico	Monografia
Nota di bibliografia	Includes bibliographical references.
Nota di contenuto	Cover -- Table of Contents -- Acknowledgments -- Preface -- Who Should Read This Book? -- What's in This Book? -- What's Not in This Book? -- How to Read This Book -- Online Resources -- Introduction -- Diagramming Techniques -- Diagramming Tools -- Creating Diagrams -- Using the Command Line -- 1. Document Your Domain -- Determine the Important Entities -- Document Our First Relationship -- Define Associations -- Define Composite Relationships -- Define Aggregate Relationships -- Decide Between Association, Aggregation, and Composition -- Document Your Own Domain -- What You've Learned -- 2. Enhance Your Domain Model -- Define Inheritance -- Describe Relationships -- Add Multiplicity -- Add a Title -- Improve Readability -- Enrich Nodes with Links -- Enhance Your Domain Model -- What You've Learned -- 3. Visualize Application and User Flows -- Define Actors and Participants -- Add Our First Interaction -- Show Branching Logic -- Display Asynchronous Messages -- Display Length of Interactions with Activations -- Add Additional Information with Notes -- Annotate Your Diagram with Sequence Numbers -- Create Dropdown Menus -- Visualize Your Own Application Flow -- What You've Learned -- 4. Model Your Architecture -- Using the C4 Model --

Creating a System Context Diagram -- Add Nodes -- Connect Nodes -- Add Some Style -- Create Your Own System Context Diagram -- What You've Learned -- 5. Detail Your System's Containers -- Define the First Two Containers -- Create Clear Boundaries with Subgraphs -- Add Supporting Systems -- Improve Readability with Link Lengths -- Display Asynchronous Interactions -- Additional Arrow Types -- Create Your Own Container Diagram -- What You've Learned -- 6. Structure Your Components and Code -- Code Diagram -- Leverage Flowcharts for Complex Flows -- What You've Learned -- 7. Design Database Schemas.

Use Entity Relationship Diagrams -- Define Our First Entity -- Relate Entities -- Add Zero-to-Many Relationships -- Enrich Schemas with Keys -- Comment Your Columns -- Define Zero-or-One Relationships -- Describe Non-identifying Relationships -- Finalize Streamy's ERD -- Design Your Database Schema -- What You've Learned -- 8. Visualize Code Flows -- Use Sequence Diagrams to Understand Class Interactions -- Define Loops -- Show Parallel Processes -- What You've Learned -- 9. Design and Refactor Your Applications -- Define Classes -- Show Dependencies with Relationships -- Refactor the Classes -- Introduce a Request Class -- Define Interfaces -- Create a Class Diagram -- What You've Learned -- 10. Render Diagrams Using Native Support -- Leverage Native Mermaid Integrations -- Render Mermaid Within Markdown Files -- Where Should You Include Diagrams? -- What About Websites Without Native Support? -- Keep Diagrams Up-to-Date -- Render a Diagram on GitHub -- What You've Learned -- 11. Create a Static Site with Mermaid Diagrams -- Devise a Plan of Action -- Learn the Basics of a GitHub Action -- Start by Defining the Events That Trigger the Action -- Check Out the Repository's Code -- Convert Mermaid Markup to SVGs in Markdown -- Build Jekyll Artifacts -- Deploy to GitHub Pages -- Run the Action -- Render Diagrams on Page Load -- What You've Learned -- 12. What You've Learned -- Bibliography.

---

#### Sommario/riassunto

Diagrams communicate relationships more directly and clearly than words ever can. Using only text-based markup, create meaningful and attractive diagrams to document your domain, visualize user flows, reveal system architecture at any desired level, or refactor your code. With the tools and techniques this book will give you, you'll create a wide variety of diagrams in minutes, share them with others, and revise and update them immediately on the basis of feedback. Adding diagrams to your professional vocabulary will enable you to work through your ideas quickly when working on your own code or discussing a proposal with colleagues. Expand your professional vocabulary by learning to communicate with diagrams as easily and naturally as speaking or writing. This book will provide you with the skills and tools to turn ideas into clear, meaningful, and attractive diagrams in mere minutes, using nothing more complicated than text-based markup. You'll learn what kinds of diagrams are suited to each of a variety of use cases, from documenting your domain to understanding how complex code pieces together. Model your software's architecture, creating diagrams focused broadly or narrowly, depending on the audience. Visualize application and user flows, design database schemas, and use diagrams iteratively to design and refactor your application. You'll be able to use technical diagramming to improve your day-to-day workflow. You will better understand the codebase you work in, communicate ideas more effectively and immediately with others, and more clearly document the architecture with C4 diagrams. Manually creating diagrams is cumbersome and time-consuming. You'll learn how to use text-based tools like Mermaid

to rapidly turn ideas into diagrams. And You'll learn how to keep your diagrams up to date and seamlessly integrated into your engineering workflow. You'll be better at visualizing and communicating when you add diagrams to your standard vocabulary.

---