



**WEB AGE SOLUTIONS**

Powered by **Axcel** Learning

# Lecture Book

WA3663 GitHub and GitHub Codebase  
Management

Version 1.0.3

© 2025 Web Age Solutions, LLC

Revision 1.0.3 published on 2025-01-30.

All rights reserved. No part of this book may be reproduced or used in any form or by any electronic, mechanical, or other means, currently available or developed in the future, including photocopying, recording, digital scanning, or sharing, or in any information storage or retrieval system, without permission in writing from the publisher.

**Trademark Notice:** Product or corporate names may be trademarks or registered trademarks, and are used only for identification and explanation without intent to infringe.

To obtain authorization for any such activities (e.g., reprint rights, translation rights), to customize this book, or for other sales inquiries, please contact:

Web Age Solutions, LLC  
1 California Street Suite 2900  
San Francisco, CA 94111  
<https://www.webagesolutions.com>

USA: 1-877-517-6540, email: [getinfousa@webagesolutions.com](mailto:getinfousa@webagesolutions.com)  
Canada: 1-877-812-8887 toll free, email: [getinfo@webagesolutions.com](mailto:getinfo@webagesolutions.com)



# GitHub and GitHub Codebase Management

Chapter 1 – GitHub

# 1.1 What is GitHub?

## GitHub is:

- A web site for hosting and managing Git repositories.
- A tool for storing, tracking and collaborating on code.
- A shared workspace for developers.
- An environment that supports open-source projects.
- A platform for building, testing and deploying software.



## 1.2 GitHub – Features

- **Version Control:** Tracking changes to code over time
- **Collaboration:** Enables multiple developers to work on the same codebase
- **Code Reviews:** Facilitates peer feedback and improved code quality
- **Issue Tracking:** Helping manage bugs, feature requests, and other tasks
- **Pull Requests:** Proposing and reviewing code changes before merging them into the codebase
- **Branching:** Creating separate lines of development
- **Cloning:** Creating a local copy of a repository that can be synced with the shared repo
- **Project Management:** Tools for planning, organizing, and tracking a project's progress
- **Integrations:** Connecting with other tools and services in the development workflow

# 1.3 GitHub – Repository Example

The screenshot shows the GitHub interface for the 'facebook/react' repository. At the top, the repository name 'facebook / react' is displayed. Below it, navigation tabs include 'Code', 'Issues (780)', 'Pull requests (188)', 'Actions', 'Projects', 'Wiki', 'Security', and 'Insights'. The repository is marked as 'Public'. Action buttons for 'Watch (6631)', 'Fork (47.4k)', and 'Star (232k)' are visible. The 'main' branch is selected. A 'Code' button is prominent. Below the branch selector, a list of recent commits is shown, including one by 'Biki-das' titled 'Fix:- Improve HOC support and state pr...'. The right sidebar contains an 'About' section describing React as 'The library for web and native user interfaces.', a link to 'react.dev', and tags for 'react', 'javascript', 'library', 'ui', 'frontend', and 'declarative'. It also lists links for 'Readme', 'MIT license', 'Code of conduct', and 'Security policy'.

facebook / react

Code Issues (780) Pull requests (188) Actions Projects Wiki Security Insights

react Public

Watch (6631) Fork (47.4k) Star (232k)

main

Go to file + Code

**Biki-das** Fix:- Improve HOC support and state pr... de1eaa2 · 1 hour ago

.codesandbox	Codesandbox: upgrade to Node....	2 years ago
.github	[ci] Use correct actor when check...	last week
compiler	Move effect dep inference tests t...	2 days ago
fixtures	Add Transition Types (#32105)	3 days ago
packages	Fix:- Improve HOC support and s...	1 hour ago
scripts	[Fizz] Support Suspense boundar...	last week

**About**

The library for web and native user interfaces.

react.dev

react javascript library ui frontend declarative

Readme MIT license Code of conduct Security policy

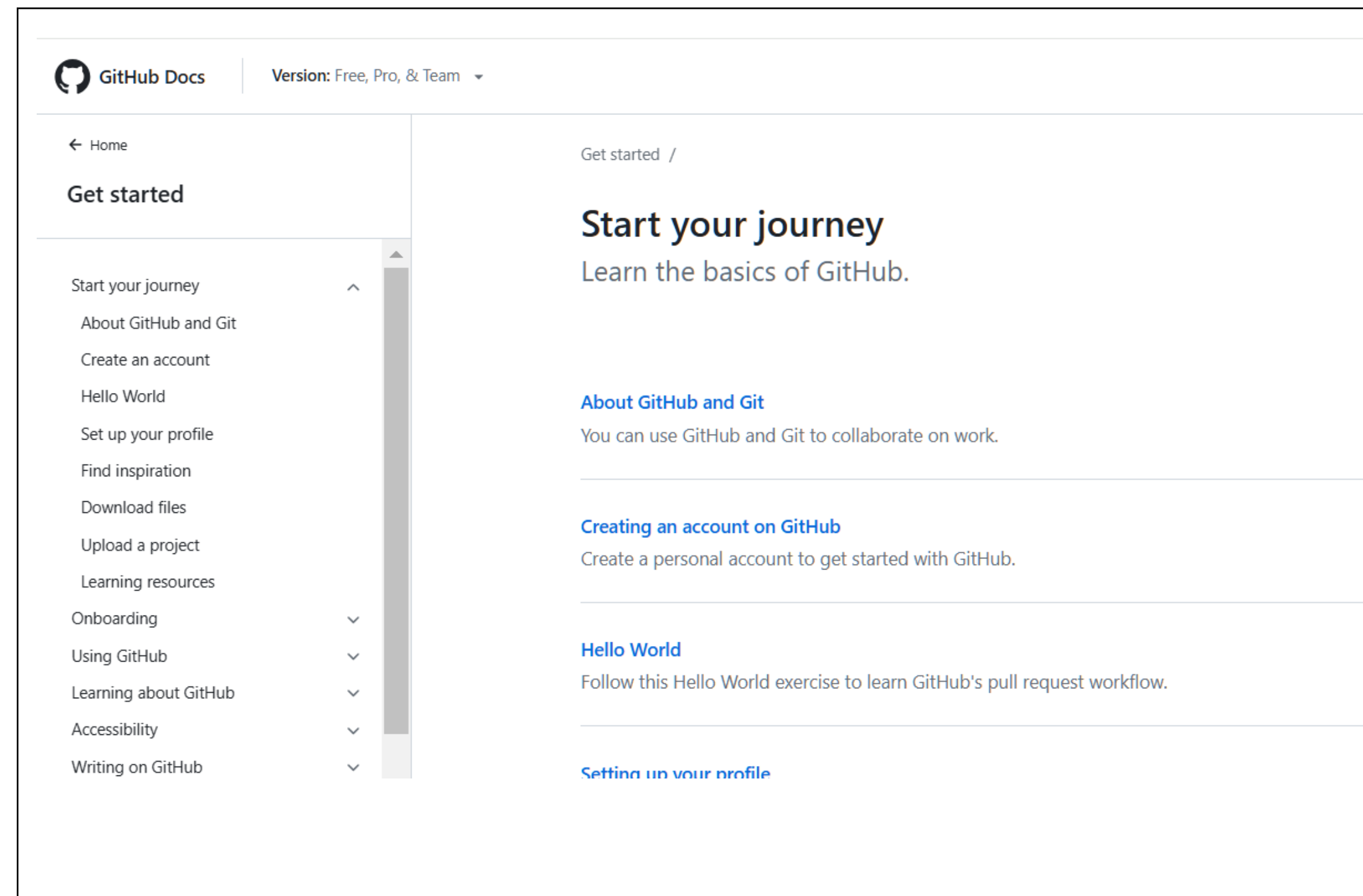
## 1.4 GitHub –User Interface Tabs

- **Code:** The heart of the GitHub, where you store and manage all your project files
- **Issues:** A to-do list for your project, for tracking bugs, feature requests, and other tasks
- **Pull Requests:** A system to propose changes to the codebase. You create a pull request, other developers review it, and then they can be merged into the main project
- **Actions:** Automate tasks like running tests or deploying your code
- **Projects:** Organize your work using kanban boards, to-do lists, and other tools
- **Wiki:** A knowledge base for storing important project information and documentation
- **Security:** Features to help you identify and fix vulnerabilities in your code
- **Insights:** Visualizations that show how your project is being developed and used

# 1.5 Start your Journey

In-depth articles covering many GitHub related topics

<https://docs.github.com/en/get-started/start-your-journey>

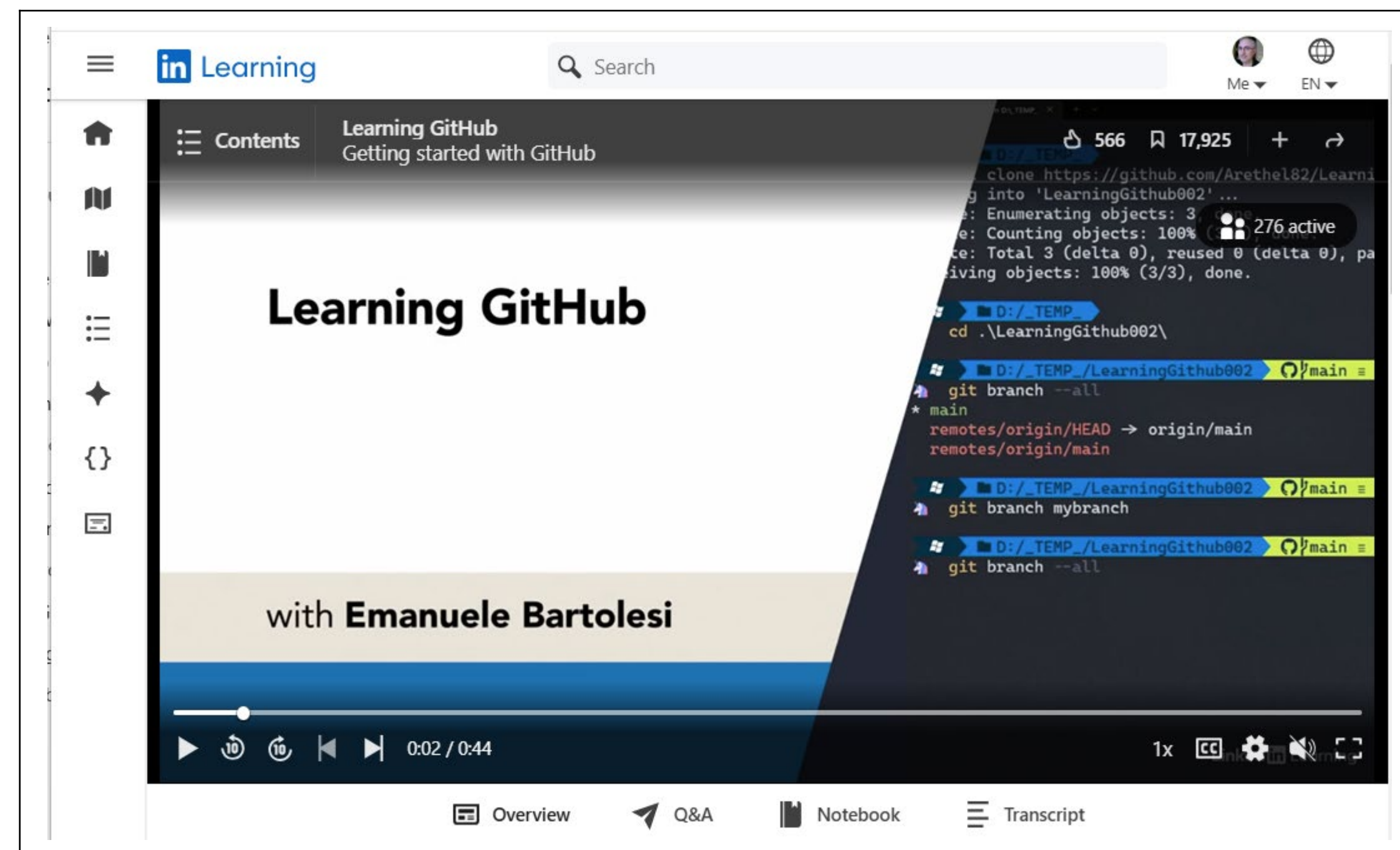




# 1.6 Learning GitHub

Video course. Review of GitHub features and functions

<https://www.linkedin.com/learning/learning-github-18719601>



# 1.7 GitHub: The Big Picture

Pluralsight video course. Provides an overview of GitHub.

<https://app.pluralsight.com/library/courses/github-big-picture/table-of-contents>

The screenshot shows the Pluralsight interface for the course 'GitHub: The Big Picture' by Aaron Stewart. The left sidebar contains navigation links: Home, Browse all, Iris, Hands-on, Channels, Role IQ, Certifications, Paths, Skill IQ, Help & support, and Send Feedback. The main content area features a hero section with the course title, author name, and a description: 'This course is a big picture introduction and overview of GitHub — an online code hosting platform used for collaborating, building, shipping, and maintaining software.' Below this are buttons for 'Resume Course', 'Bookmark', 'Add to Channel', and 'Download Course'. A 'Table of contents' tab is active, showing a list of course sections: 'Course Overview' (1m 50s) and 'What Is GitHub? (And What Is It Used For?)' (14m 30s). The right sidebar provides course details: 'Course author' (Aaron Stewart), 'Course info' (Level: Beginner, Rating: 5 stars (188), My rating: 5 stars, Duration: 1h 6m, Updated: Sep 2022, Language: English).





# GitHub and GitHub Codebase Management

Chapter 2 – GitHub Codebase Management

## 2.1 Codebase Management – Definition

"Codebase management refers to the practices, processes, and tools used to control, maintain, and organize the source code of a software project. It can work on the codebase simultaneously with involves tracking the changes made to the code, ensuring that different developers out conflicts, and maintaining the quality and integrity of the code over time."



## 2.2 Key Aspects of Codebase Management – List

1. Version Control
2. Branching and Merging
3. Code Review
4. Dependency Management
5. Continuous Integration and Continuous Deployment (CI/CD)
6. Issue Tracking
7. Documentation
8. Code Quality and Testing
9. Release Management

## 2.2 Key Aspects – Details 1/3

### Version Control:

- Version control systems (VCS) like Git are central to managing a codebase. These tools track changes made to the code over time, allowing developers to revert to earlier versions, branch out to work on different features, and merge changes when necessary.

### Branching and Merging:

- Each developer works on their own branch and makes changes without affecting the main codebase.
- Merging allows changes from different branches to be integrated back into the main.
- Pull requests are often used for reviewing and integrating changes.

### Code Review:

- Codebase management encourages thorough code reviews before changes are merged. This ensures that new code is high quality, follows project conventions, and does not introduce bugs or vulnerabilities.

## 2.2 Key Aspects – Details 2/3

### **Dependency Management:**

Most software projects depend on external libraries or packages. Codebase management includes managing these dependencies, ensuring that the correct versions are used, and avoiding conflicts or compatibility issues.

### **Continuous Integration and Continuous Deployment (CI/CD):**

In modern codebase management, CI/CD pipelines are often implemented to automatically test, build, and deploy the code whenever changes are made. This ensures that the codebase remains stable, functional, and up-to-date.

### **Issue Tracking:**

Codebase management is closely integrated with issue tracking systems (like Jira or GitHub Issues). These systems allow developers to associate tasks or bugs with specific code changes, ensuring that the project is progressing smoothly.

## 2.2 Key Aspects – Details 3/3

### **Documentation:**

Proper documentation is part of good codebase management. This includes inline comments, external documentation (like READMEs or wikis), and other supporting materials to help developers understand how to work with the codebase.

### **Code Quality and Testing:**

Ensuring high code quality through practices like automated testing (unit tests, integration tests) is a key aspect of managing a codebase. This helps prevent bugs and regressions and ensures that the code remains maintainable.

### **Release Management:**

Codebase management also includes planning and controlling the release cycle, managing versioning, and handling deployment processes.



## 2.3 GitHub Codebase Management

"The use of GitHub to manage a project's codebase"

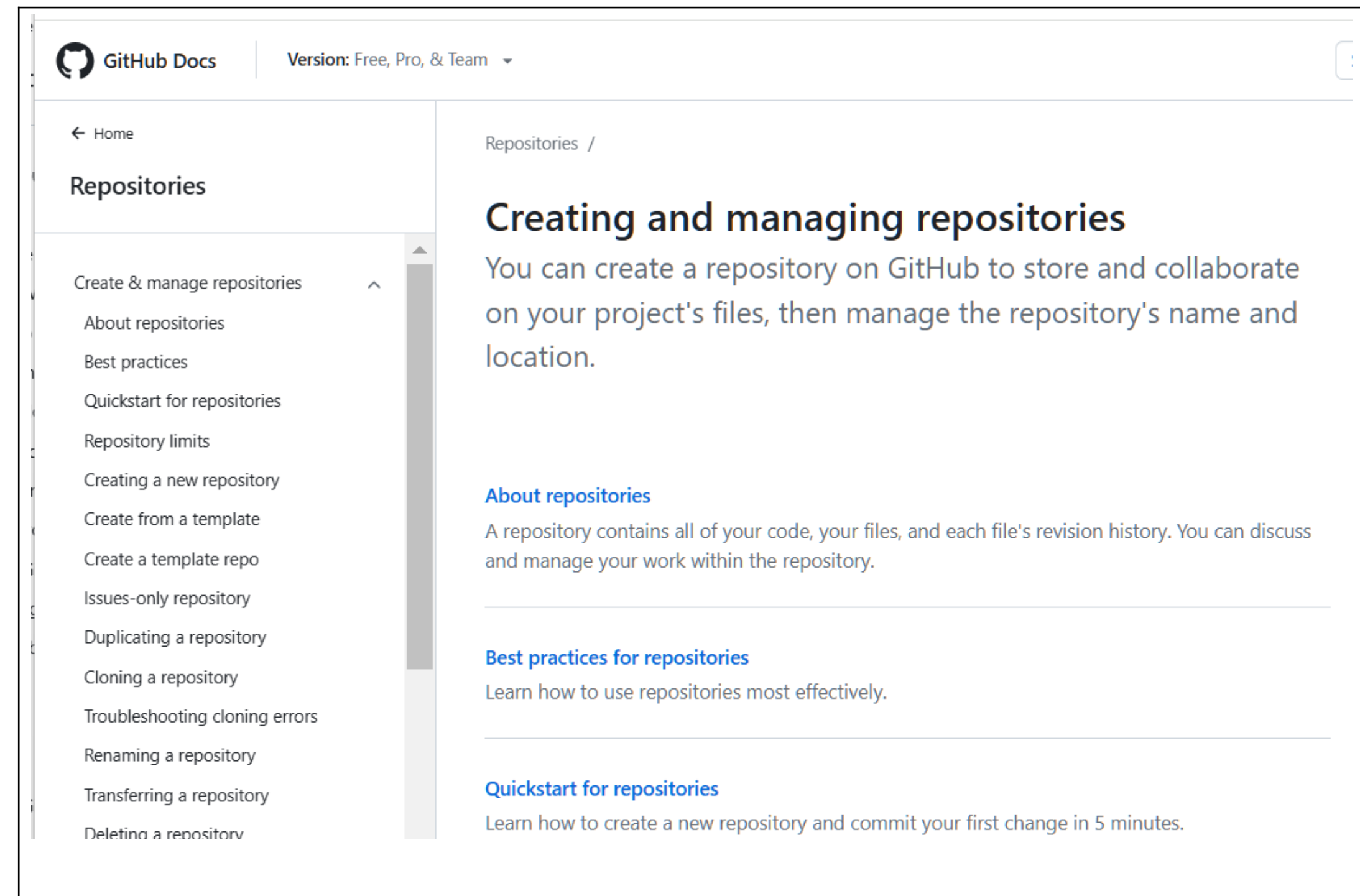
Aspects Manageable through GitHub:

- **Repository Organization:** Structuring of repositories for different projects and teams.
- **Workflows:** Clear processes for code development, review, and deployment.
- **Security:** Protecting the codebase from vulnerabilities and unauthorized access.
- **Automation:** Automating tasks like building, testing, and deploying code.
- **Integration:** Integrating of SCM with other development tools (e.g., CI/CD pipelines).

## 2.4 Creating and Managing Repositories

In-depth information about creating and managing repositories on GitHub.

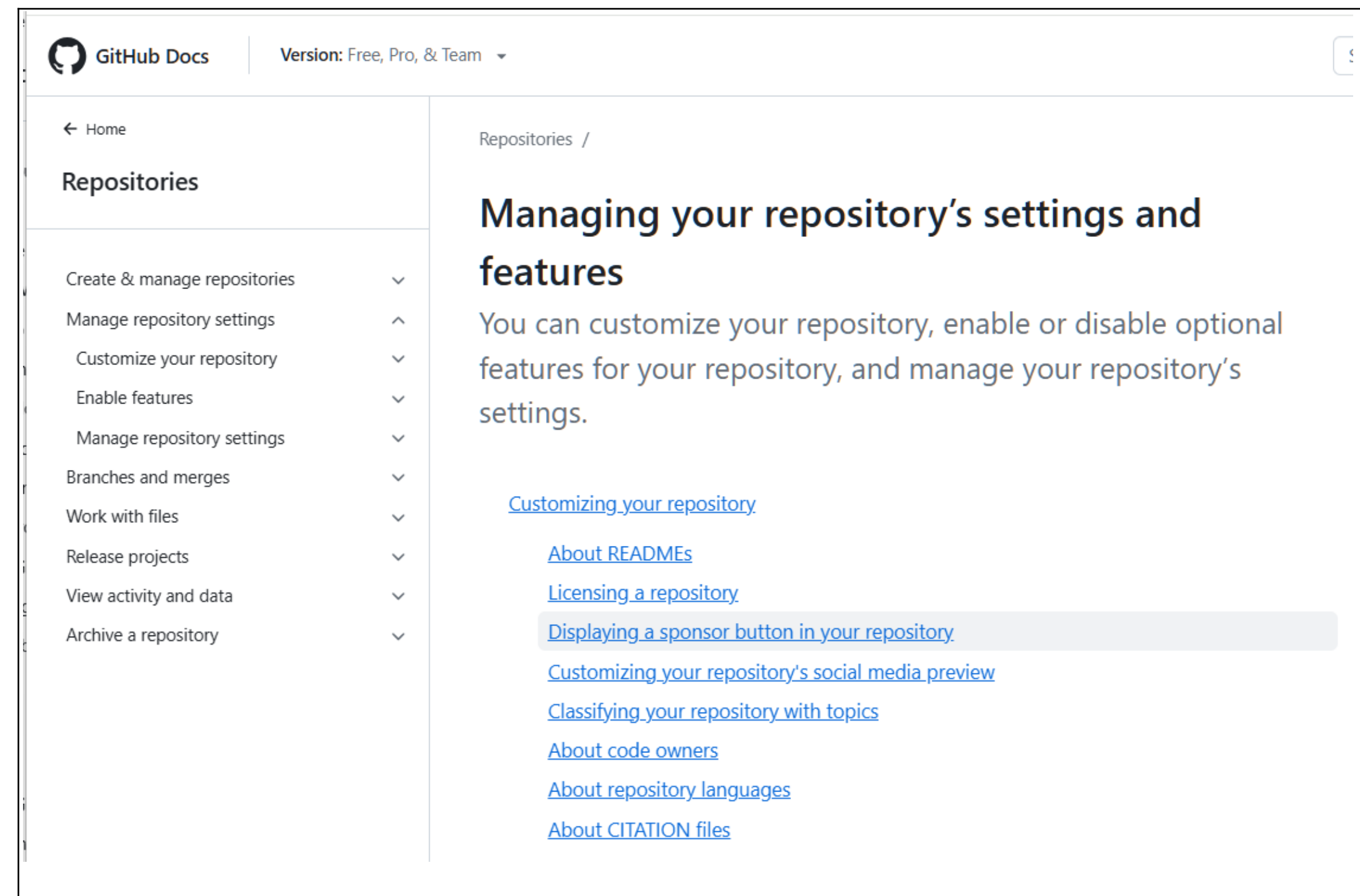
<https://docs.github.com/en/repositories/creating-and-managing-repositories>



# 2.5 Managing your Repository's Settings & Features

Covers setup of settings for repositories

<https://docs.github.com/en/repositories/managing-your-repositorys-settings-and-features>



## 2.6 Ten GitHub Best Practices for Repository Management

Video with practical advice for repository management.

<https://www.youtube.com/watch?v=VvgoOgWFiZY>

