React

Complete

हमेशा की छुट्टी

Learning Roadmap

1. Introduction to Version Control
2. Getting Started with GitHub
3. Getting Started with Git
4. Working with GitHub UI
5. Essential Git Commands
6. Branching and Merging
7. Advance GitHub Features
8. Git Best Practices
9. Advanced Git Operations

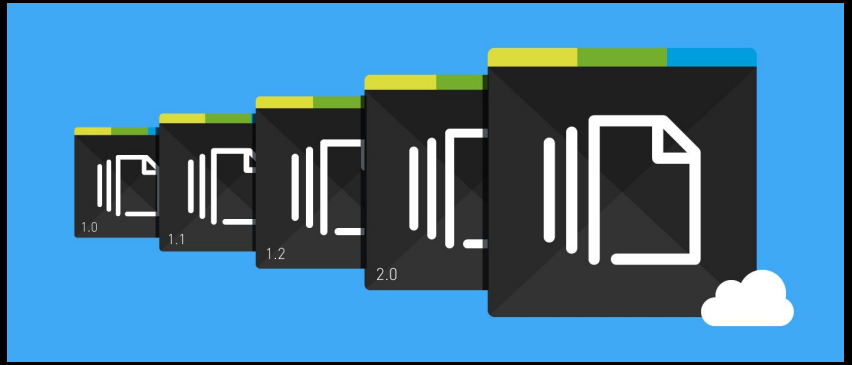
1. Introduction to Version Control

1. What is Version Control?
2. Need of Version Control
3. Benefits of Version Control
4. Principle of Version Control
5. Evolution of Version Control
6. Diff Based vs Copy Based
7. Overview of Git

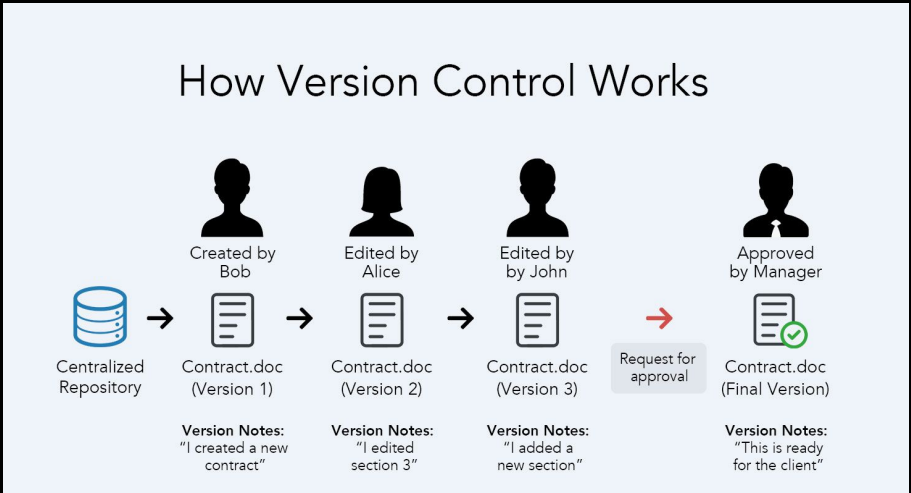
React

1.1 What is Version Control?

1. A system that tracks changes to files over time
2. Manages different versions of source code and documents.
3. Tracks modifications and maintains a history of file versions.
4. Allows multiple people to work on the same project
5. Records who made what changes and when
6. Facilitates collaboration by managing contributions and changes from different team members.



1.2 Need of Version Control



1. Helps manage and track changes in codebases or documents efficiently.
2. Prevents loss of work and supports rollback to previous versions.
3. Enables collaboration among team members without overwriting each other's work.
4. Provides accountability by showing who made what changes and when.

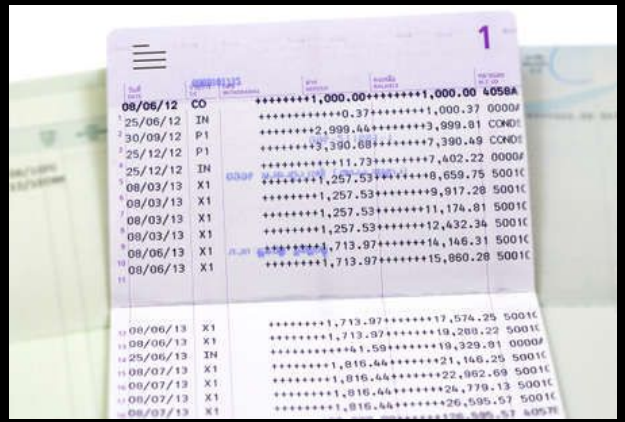
1.3 Benefits of Version Control

1. Collaboration: Multiple people can work on the same project without conflict.
2. Backup: Every change is saved, allowing retrieval of previous versions.
3. Accountability: Clearly shows who made changes and when.
4. Conflict Resolution: Handles merging and resolving conflicting changes in collaborative environments.

1.4 Principle of Version Control

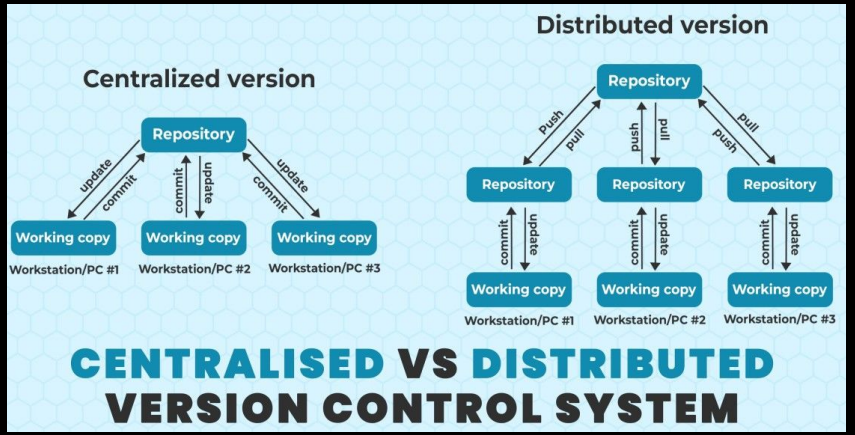
Result = Start + Sum(Changes)

Bank Account



1.5 Evolution of Version Control

1. Manual Versioning: Early developers used to manually name file versions.
2. Local Version Control Systems: Initial tools like RCS stored versions on individual machines.
3. Centralized Version Control: Systems like CVS and SVN introduced a single server to store files.
4. Distributed Version Control: Tools like Git and Mercurial allow users to have full project history on their local machines.
5.



1.6 Diff Based vs Copy Based

Feature	Diff-Based Version Control	Copy-Based Version Control
Storage Efficiency	Stores only differences (changes) between versions	Stores a full copy of the file for each version
Storage Usage	More efficient, uses less storage	Less efficient, uses more storage
Speed	Faster for tracking and committing small changes	Slower as every file is saved in its entirety
Tracking Mechanism	Tracks changes at a granular level (lines of code)	Saves entire file versions for each change
Examples	Git, SVN, Mercurial	Early manual version control, simple backup systems
Use Case	Ideal for large projects with frequent small changes	Useful in basic systems with fewer changes

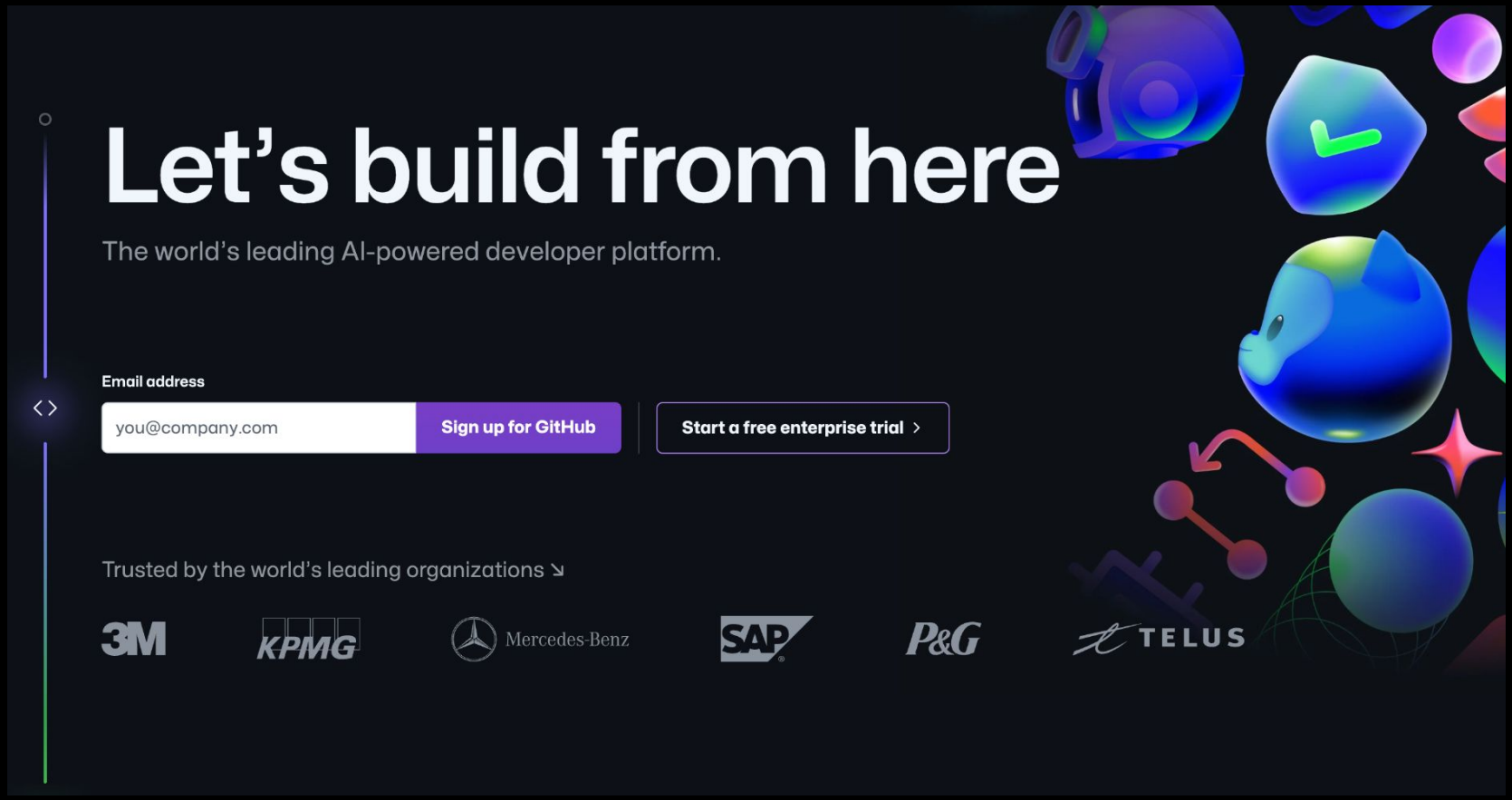
1.7 Overview of git

1. **Created by Linus Torvalds**: Developed in 2005 by the creator of Linux to manage the Linux kernel.
2. **Humorous Name**: "Git" is British slang for an unpleasant person, chosen humorously by Torvalds.
3. **Inspired by BitKeeper**: Git was created after Linux developers lost access to the proprietary BitKeeper system.
4. **Fully Distributed**: Every developer has a complete copy of the project history, allowing offline work and safer collaboration.
5. **Snapshot System**: Git stores snapshots of the project, not just differences, making it fast and efficient.
6. **Widely Used**: Git is the most popular version control system, powering platforms like GitHub and GitLab.
7. **Fast and Efficient**: Designed to handle large projects with speed.
8. **Branching Model**: Git makes branching and merging easy and efficient.

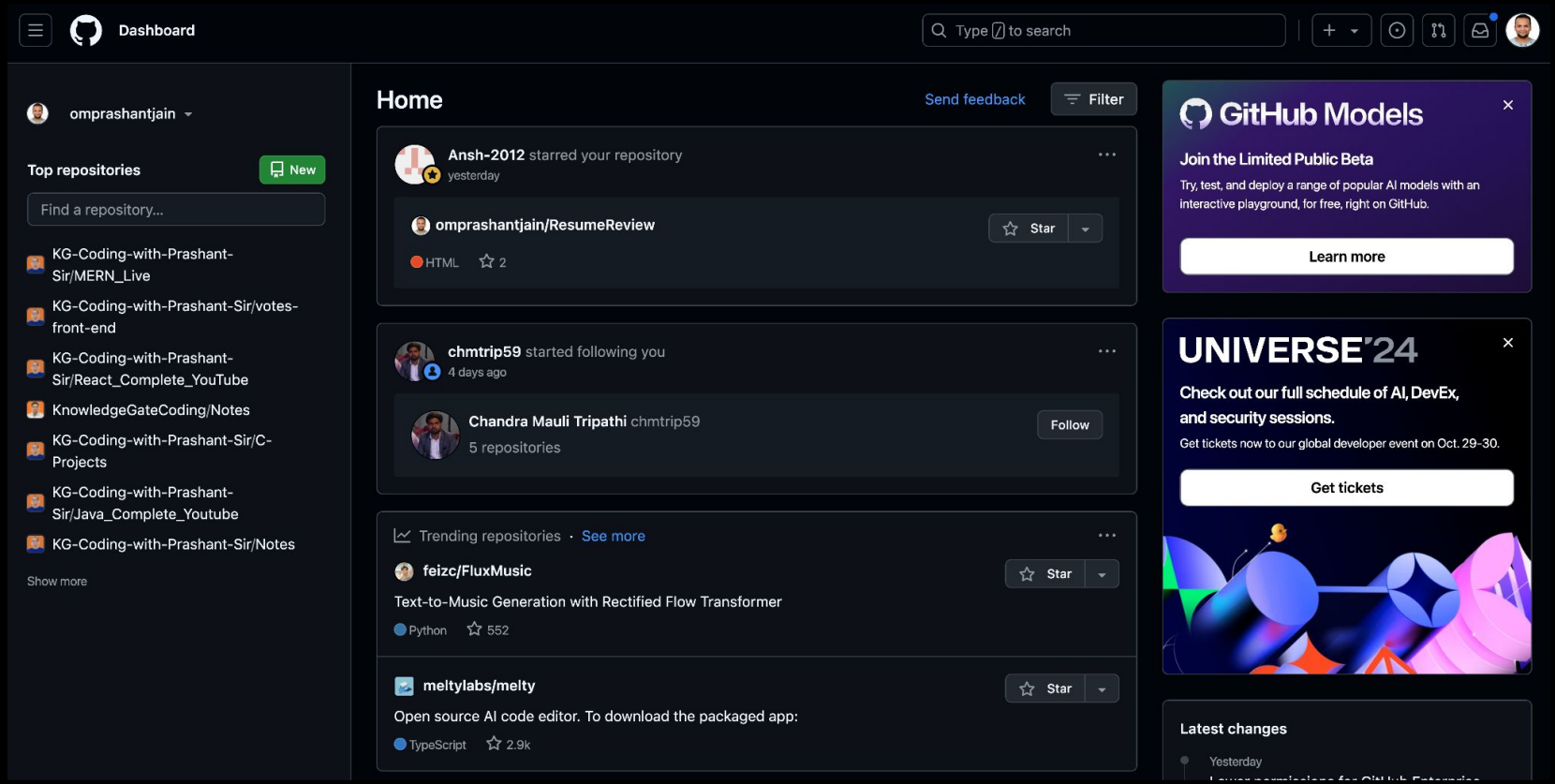
2. Getting Started with GitHub

1. Creating a GitHub account
2. Tour of the GitHub interface
3. Creating a repository on GitHub
4. Exploring Popular Repository
5. GitHub README files

2.1 Creating a GitHub account



2.2 Tour of the GitHub interface



2.3 Creating a repository on GitHub

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? Import a repository.

Required fields are marked with an asterisk (*).

Owner * KG-Coding-with-Prashant-Sir /

Repository name *

Great repository names are short and memorable. Need inspiration? How about psychic-guide ?

Description (optional)

- **Public**
  Anyone on the internet can see this repository. You choose who can commit.
- **Private**
  You choose who can see and commit to this repository.

Initialize this repository with:

- **Add a README file**
  This is where you can write a long description for your project. Learn more about READMEs.

Add .gitignore

.gitignore template: None

Choose which files not to track from a list of templates. Learn more about ignoring files.

Choose a license

License: None

A license tells others what they can and can't do with your code. Learn more about licenses.

You are creating a private repository in the KG-Coding-with-Prashant-Sir organization.

Create repository

2.4 Exploring Popular Repos

facebook / react

Type / to search

Code | Issues 664 | Pull requests 171 | Actions | Projects | Wiki | Security | Insights

react Public

Watch 6630 | Fork 46.3k | Star 227k

main | 268 Branches | 143 Tags | Go to file t | Add file | Code

sebmarkbage [DevTools] Build Updater List from the Commit instead of ... 99cba2b · 10 hours ago 19,396 Commits

.codesandbox	Codesandbox: upgrade to Node.js 18 (#26330)	last year
.github	[ez] Remove trailing space from babel-refresh header	2 weeks ago
compiler	[compiler] Add enablePropagateDepsInHIR flag	19 hours ago
fixtures	[Flight] Enable Server Action Source Maps in flight-esm Fi...	2 weeks ago
packages	[DevTools] Build Updater List from the Commit instead of ...	10 hours ago
scripts	Temporarily disable suspending during work loop (#30762)	3 days ago
.editorconfig	https link to editorconfig.org (#18421)	4 years ago
.eslintignore	Fix ESLint and Prettier configs for React Compiler (#29073)	4 months ago
.eslintrc.js	chore[react-devtools]: add global for native and use it to f...	last month
.git-blame-ignore-revs	Add run prettier commit to .git-blame-ignore-revs	2 months ago
.gitattributes	.gitattributes to ensure LF line endings when we should	11 years ago

About

The library for web and native user interfaces.

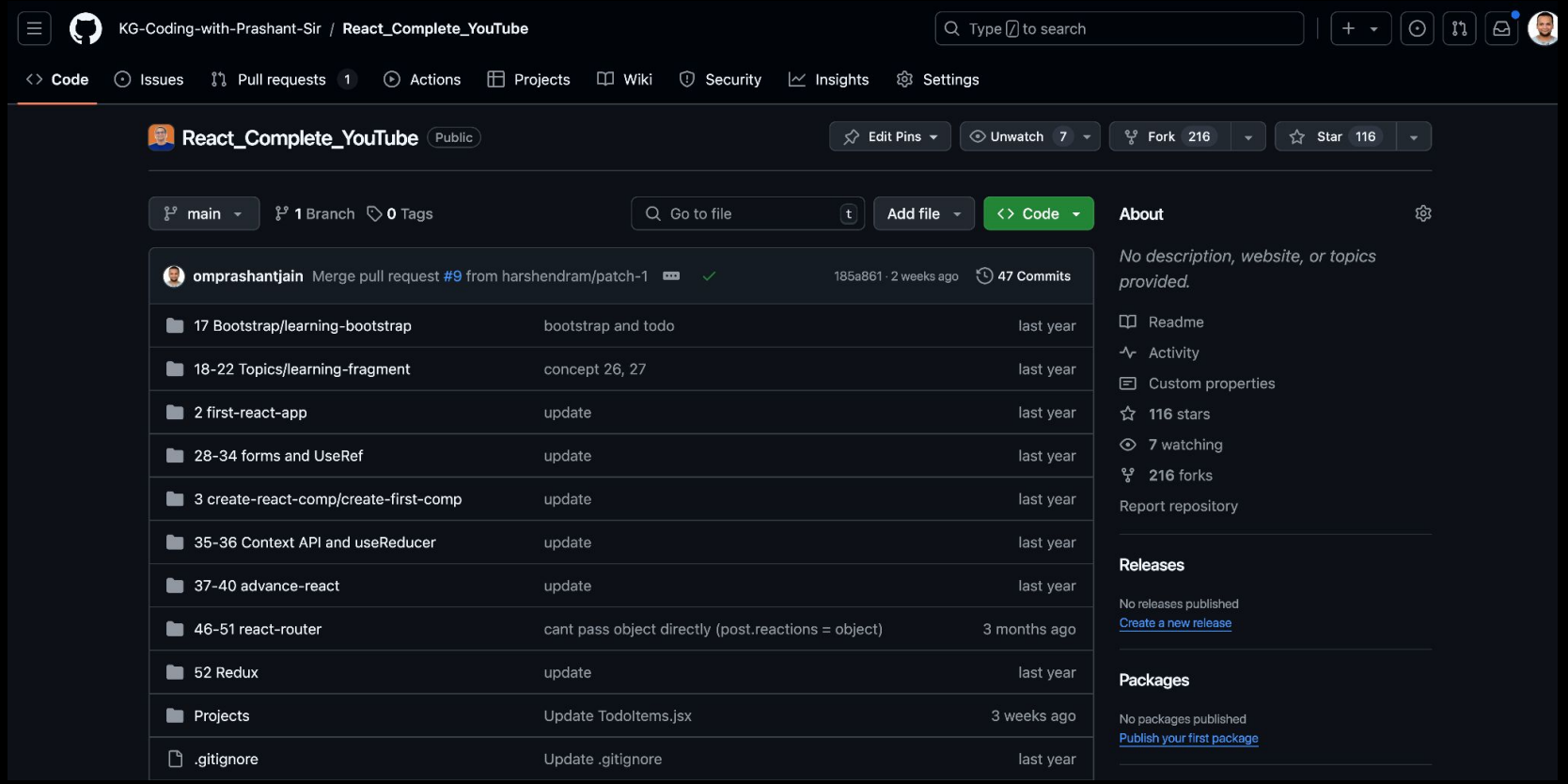
react.dev

react | javascript | library | ui | frontend | declarative

Readme
MIT license
Code of conduct
Security policy
Activity
Custom properties
227k stars
6.6k watching
46.3k forks
Report repository

Releases 101

2.4 Exploring Popular Repos



2.5 GitHub README files

README | Code of conduct | MIT license | Security

React

license MIT | npm v18.3.1 | (Runtime) Build and Test passing | (Compiler) TypeScript passing | PRs welcome

React is a JavaScript library for building user interfaces.

- **Declarative:** React makes it painless to create interactive UIs. Design simple views for each state in your application, and React will efficiently update and render just the right components when your data changes. Declarative views make your code more predictable, simpler to understand, and easier to debug.
- **Component-Based:** Build encapsulated components that manage their own state, then compose them to make complex UIs. Since component logic is written in JavaScript instead of templates, you can easily pass rich data through your app and keep the state out of the DOM.
- **Learn Once, Write Anywhere:** We don't make assumptions about the rest of your technology stack, so you can develop new features in React without rewriting existing code. React can also render on the server using Node and power mobile apps using React Native.

Learn how to use React in your project.

Installation

React has been designed for gradual adoption from the start, and **you can use as little or as much React as you need**:

- Use Quick Start to get a taste of React.
- Add React to an Existing Project to use as little or as much React as you need.
- Create a New React App if you're looking for a powerful JavaScript toolchain.

3. Getting Started with Git

1. Installing Git
2. Configuring Git (user name, email)
3. Git help and documentation
4. What is a Repository
5. What is a Commit

3.1 Installing Git

git --local-branching-on-the-cheap

Type / to search entire site...

About
Documentation
Downloads
GUI Clients
Logos
Community

The entire **Pro Git book** written by Scott Chacon and Ben Straub is available to read online for free. Dead tree versions are available on Amazon.com.

Downloads

macOS Windows Linux/Unix

Older releases are available and the Git source repository is on GitHub.

GUI Clients

Git comes with built-in GUI tools (**git-gui**, **gitk**), but there are several third-party tools for users looking for a platform-specific experience.

View GUI Clients →

Logos

Various Git logos in PNG (bitmap) and EPS (vector) formats are available for use in online and print projects.

View Logos →

Git via Git

If you already have Git installed, you can get the latest development version via Git itself:

```
git clone https://github.com/git/git
```

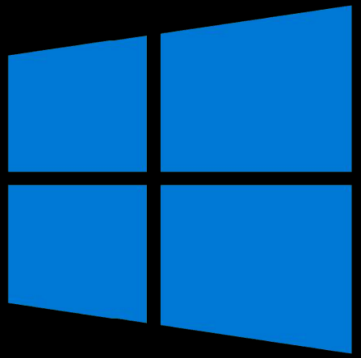
You can also always browse the current contents of the git repository using the web interface.

About this site
Patches, suggestions, and comments are welcome.

Git is a member of Software Freedom Conservancy

React

3.1 Windows Setup



Windows

3.1 Windows Setup

Download for Windows

Click here to download the latest (**2.46.0**) **32-bit** version of **Git for Windows**. This is the most recent maintained build. It was released **about 1 month ago**, on 2024-07-29.

Other Git for Windows downloads

Standalone Installer
32-bit Git for Windows Setup.

64-bit Git for Windows Setup.

Portable ("thumbdrive edition")
32-bit Git for Windows Portable.

64-bit Git for Windows Portable.

Using winget tool
Install winget tool if you don't already have it, then type this command in command prompt or Powershell.

```
winget install --id Git.Git -e --source winget
```

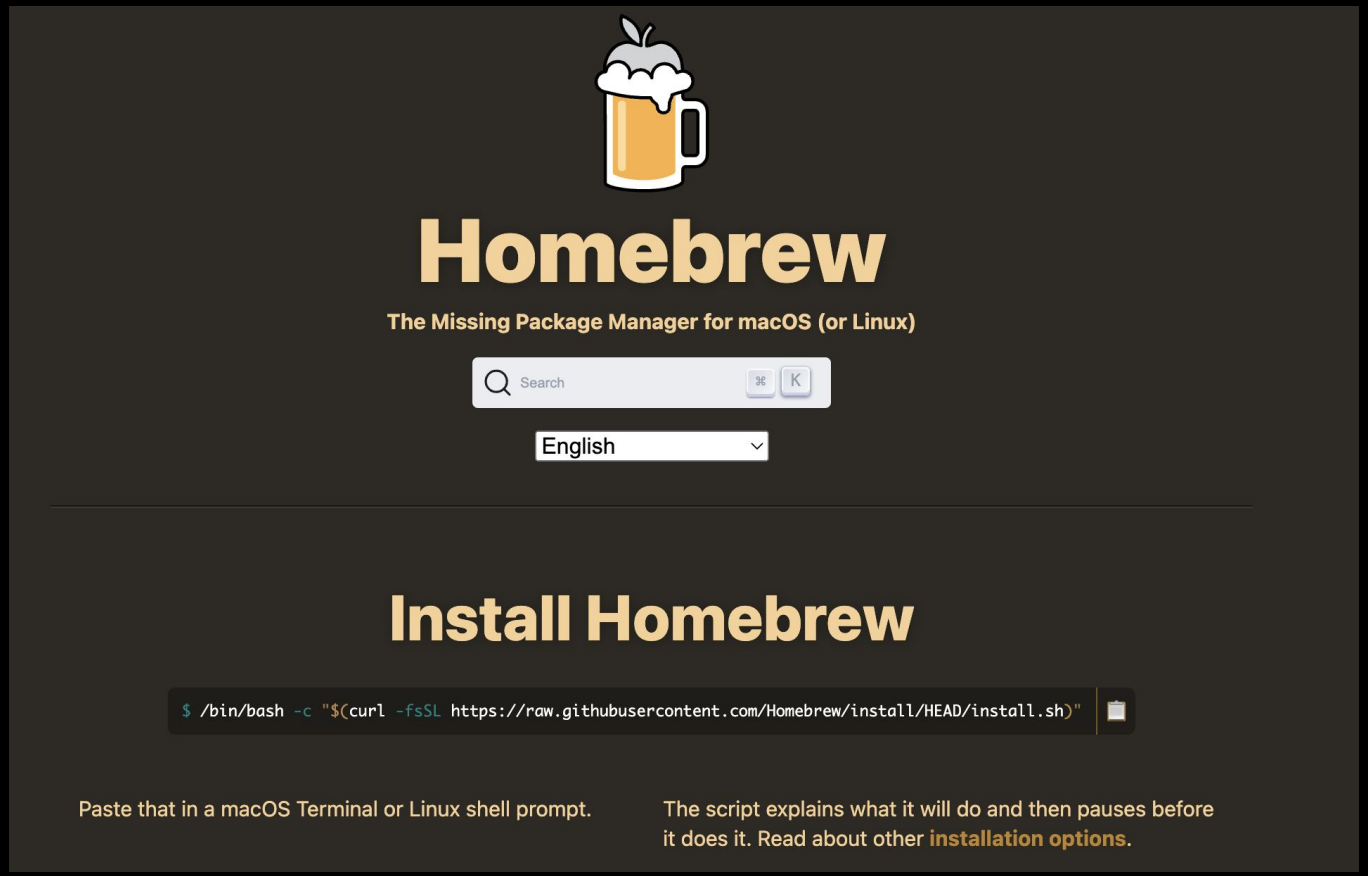
The current source code release is version **2.46.0**. If you want the newer version, you can build it from the source code.

React

3.1 MAC Setup

mac OS

3.1 MAC Setup


Homebrew
The Missing Package Manager for macOS (or Linux)
Search
⌘ K
English
Install Homebrew
$ /bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"
Paste that in a macOS Terminal or Linux shell prompt.
The script explains what it will do and then pauses before it does it. Read about other installation options.

React

3.1 MAC Setup

Download for macOS

There are several options for installing Git on macOS. Note that any non-source distributions are provided by third parties, and may not be up to date with the latest source release.

Choose one of the following options for installing Git on macOS:

Homebrew

Install homebrew if you don't already have it, then:

```
$ brew install git
```

MacPorts

Install MacPorts if you don't already have it, then:

```
$ sudo port install git
```

3.2 Configuring Git (user name)

Setting your Git username for *every* repository on your computer

1. Open Terminal.
2. Set a Git username:

```
git config --global user.name "Mona Lisa"
```

3. Confirm that you have set the Git username correctly:

```
$ git config --global user.name
> Mona Lisa
```

3.2 Configuring Git (email)

Setting your email address for every repository on your computer

1. Open Terminal.
2. Set an email address in Git. You can use your GitHub-provided `noreply` email address or any email address.

```
git config --global user.email "YOUR_EMAIL"
```

3. Confirm that you have set the email address correctly in Git:

```
$ git config --global user.email
email@example.com
```

4. Add the email address to your account on GitHub, so that your commits are attributed to you and appear in your contributions graph. For more information, see "Adding an email address to your GitHub account."

3.3 Git help and documentation

git --fast-version-control

Type / to search entire site...

About

Documentation

- Reference
- Book
- Videos
- External Links

Downloads

Community

Reference

Quick reference guides: **GitHub Cheat Sheet** | **Visual Git Cheat Sheet**

Complete list of all commands

Setup and Config

- git
- config
- help
- bugreport
- Credential helpers

Getting and Creating Projects

- init
- clone

Basic Snapshotting

- add
- status

Guides

- gitattributes
- Command-line interface conventions
- Everyday Git
- Frequently Asked Questions (FAQ)
- Glossary
- Hooks
- gitignore
- gitmodules
- Revisions
- Submodules
- Tutorial
- Workflows
- All guides...

3.3 Git help and documentation

git --distributed-even-if-your-workflow-isnt

Type / to search entire site...

About

Documentation

- Reference
- Book
- Videos
- External Links

Downloads

Community

This book is available in English.

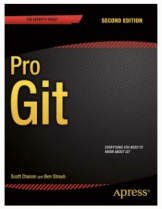
Full translation available in

azərbaycan dili,
български език,
Deutsch,
Español,
Français,
Ελληνικά,
日本語,
한국어,

Book

The entire Pro Git book, written by Scott Chacon and Ben Straub and published by Apress, is available here. All content is licensed under the Creative Commons Attribution Non Commercial Share Alike 3.0 license. Print versions of the book are available on Amazon.com.

The version found here has been updated with corrections and additions from hundreds of contributors. If you see an error or have a suggestion, patches and issues are welcome in its GitHub repository.



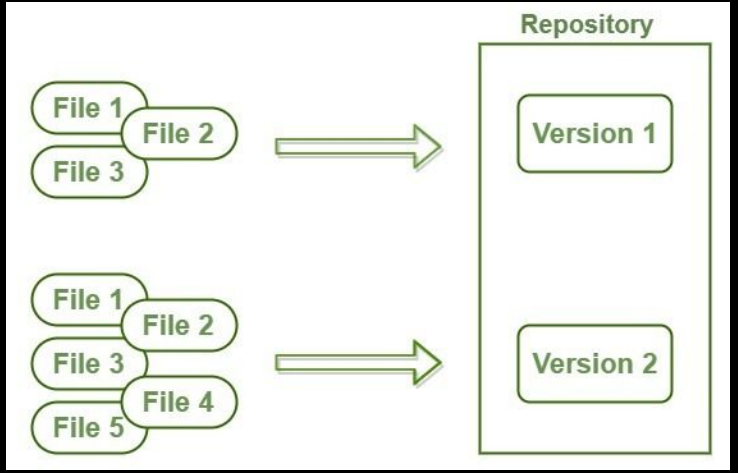
2nd Edition (2014)

Download Ebook

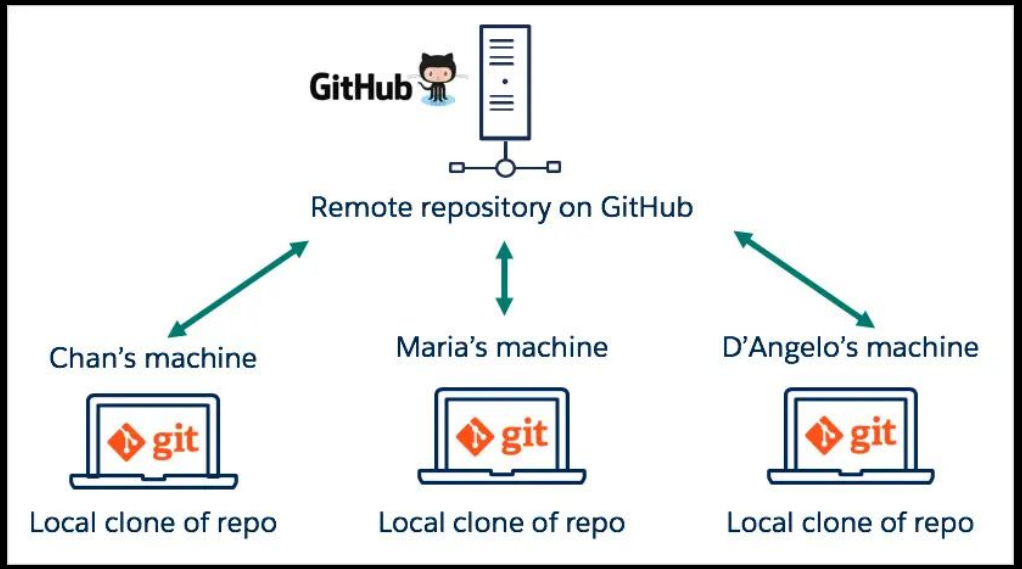
pdf epub

1. **Getting Started**
   - 1.1 About Version Control
   - 1.2 A Short History of Git
   - 1.3 What is Git?
   - 1.4 The Command Line
   - 1.5 Installing Git
   - 1.6 First-Time Git Setup
   - 1.7 Getting Help
   - 1.8 Summary

3.4 What is a Repository



3.4 What is a Repository



- A repository (repo) is a storage space for all project files and history.
- It tracks changes over time, enabling collaboration and version control.
- Repos can be local (on your machine) or remote (like GitHub).
- Contains subdirectories like the `.git` folder for managing versions.
- Git commands operate within a repository.

3.5 What is a Commit

- Snapshots of your project at a specific point in time
- Each commit has a unique identifier (SHA-1 hash)
- Contain metadata such as author, date, and commit message
- Form a chain, with each commit pointing to its parent(s)
- Allow you to revert to previous states of your project

3.5 What is a Commit