

GIT AND GITHUB

Introduction to Git

A Version Control System for Developers



Git and Git-hub

1. **Introduction to Git**
2. **Key Concepts in Git**
3. **Getting started with Git**
4. **Basic Commands**
5. **Best Practices for Git**
6. **Git Workflows**

- Git is a popular version control system that allows developers to manage and track changes to code and files over time.
- It is widely used in software development to collaborate with other developers, track changes, and deploy updates to production.
- It allows developers to track changes made to their code, collaborate with others, and manage multiple versions of a project.



Key Concepts in Git

Repository:

Git and Git-hub

1. **Introduction to Git**
2. **Key Concepts in Git**
3. **Getting started with Git**
4. **Basic Commands**
5. **Best Practices for Git**
6. **Git Workflows**

A Git repository is a collection of files and directories that make up a project, along with the history of changes to those files over time.

Each repository has a unique URL or address that allows developers to access and collaborate on the code.



Key Concepts in Git

Commit:

Git and Git-hub

1. **Introduction to Git**
2. **Key Concepts in Git**
3. **Getting started with Git**
4. **Basic Commands**
5. **Best Practices for Git**
6. **Git Workflows**

A commit is a snapshot of the code at a particular point in time.

It represents a set of changes made to the files in the repository since the last commit, along with a commit message describing the changes.



Key Concepts in Git

Branch:

Git and Git-hub

1. **Introduction to Git**
2. **Key Concepts in Git**
3. **Getting started with Git**
4. **Basic Commands**
5. **Best Practices for Git**
6. **Git Workflows**

A branch is a parallel version of the repository's code that allows developers to work on changes without affecting the main branch, known as the "master" branch.

Each branch has its own commit history, allowing developers to experiment with new features or fixes without affecting the main codebase.



Key Concepts in Git

Merge:

Git and Git-hub

1. **Introduction to Git**
2. **Key Concepts in Git**
3. **Getting started with Git**
4. **Basic Commands**
5. **Best Practices for Git**
6. **Git Workflows**

Merging is the process of combining changes from one branch into another.

This is typically done when a feature or fix is complete and ready to be integrated into the main codebase.



Getting Started with Git

Installing Git:

Git and Git-hub

1. **Introduction to Git**
2. **Key Concepts in Git**
3. **Getting started with Git**
4. **Basic Commands**
5. **Best Practices for Git**
6. **Git Workflows**

Git can be installed on different platforms, including Windows, macOS, and Linux.

The easiest way to install Git is to download and run the installer from the Git website.

Alternatively, Git can be installed using a package manager on Linux or through a third-party tool on macOS.

[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 →](#)

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](#).



Logos

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

[View Logos →](#)





Getting Started with Git

Configuring Git:

Git and Git-hub

1. **Introduction to Git**
2. **Key Concepts in Git**
3. **Getting started with Git**
4. **Basic Commands**
5. **Best Practices for Git**
6. **Git Workflows**

Once Git is installed, it needs to be configured with user information, such as the user's name and email address.

Configuration can be done through the Git command line using the "git config" command, or by editing the Git configuration file directly.



Getting Started with Git

Creating a Git Repository:

Git and Git-hub

1. [**Introduction to Git**](#)
2. [**Key Concepts in Git**](#)
3. [**Getting started with Git**](#)
4. [**Basic Commands**](#)
5. [**Best Practices for Git**](#)
6. [**Git Workflows**](#)

A Git repository is a folder or directory where Git tracks changes to files and directories. To create a Git repository, navigate to the desired folder and use the "git init" command.

This command initializes an empty Git repository in the folder and creates a hidden ".git" folder to store Git's internal files.



Basic Commands

Clone:

Git and Git-hub

1. **Introduction to Git**
2. **Key Concepts in Git**
3. **Getting started with Git**
4. **Basic Commands**
5. **Best Practices for Git**
6. **Git Workflows**

To create a local copy of a remote repository, use the git clone command followed by the repository's URL.

For example:

git clone

<https://github.com/example/repo.git>



Basic Commands

Add:

Git and Git-hub

1. **Introduction to Git**
2. **Key Concepts in Git**
3. **Getting started with Git**
4. **Basic Commands**
5. **Best Practices for Git**
6. **Git Workflows**

To add changes to the staging area, use the `git add` command followed by the file or directory you want to stage.

For example:

`git add index.html`



Basic Commands

Commit:

Git and Git-hub

1. **Introduction to Git**
2. **Key Concepts in Git**
3. **Getting started with Git**
4. **Basic Commands**
5. **Best Practices for Git**
6. **Git Workflows**

To create a new commit with the staged changes, use the git commit command followed by a commit message describing the changes.

For example:

git commit -m "Add new feature"



Basic Commands

Branch:

Git and Git-hub

1. **Introduction to Git**
2. **Key Concepts in Git**
3. **Getting started with Git**
4. **Basic Commands**
5. **Best Practices for Git**
6. **Git Workflows**

To create a new branch, use the git branch command followed by the name of the new branch.

For example:

git branch new-feature



Basic Commands

Branch:

Git and Git-hub

1. **Introduction to Git**
2. **Key Concepts in Git**
3. **Getting started with Git**
4. **Basic Commands**
5. **Best Practices for Git**
6. **Git Workflows**

To switch to a different branch, use the git checkout command followed by the name of the branch.

For example:

git checkout new-feature



Basic Commands

Merge:

Git and Git-hub

1. **Introduction to Git**
2. **Key Concepts in Git**
3. **Getting started with Git**
4. **Basic Commands**
5. **Best Practices for Git**
6. **Git Workflows**

To merge changes from one branch into another, first switch to the target branch using `git checkout`, then use the `git merge` command followed by the name of the source branch. For example:

`git checkout master git merge new-feature`



Basic Commands

Push:

Git and Git-hub

1. **Introduction to Git**
2. **Key Concepts in Git**
3. **Getting started with Git**
4. **Basic Commands**
5. **Best Practices for Git**
6. **Git Workflows**

To push your changes to a remote repository, use the git push command followed by the name of the remote repository and branch.

For example:

git push origin master



Basic Commands

Pull:

Git and Git-hub

1. **Introduction to Git**
2. **Key Concepts in Git**
3. **Getting started with Git**
4. **Basic Commands**
5. **Best Practices for Git**
6. **Git Workflows**

To fetch and merge changes from a remote repository, use the `git pull` command followed by the name of the remote repository and branch.

For example:

`git pull origin master`



Best Practices for Git

Git and Git-hub

1. **Introduction to Git**
2. **Key Concepts in Git**
3. **Getting started with Git**
4. **Basic Commands**
5. **Best Practices for Git**
6. **Git Workflows**

- Commit early and often to keep track of changes and make it easier to revert if needed.
- Write descriptive commit messages that explain the changes made in the commit.
- Use branches to work on new features or bug fixes without affecting the main codebase.
- Pull changes frequently to keep your local repository up to date with the remote repository.
- Use Git to collaborate with other developers and resolve conflicts when merging changes.



Git Workflows

Git and Git-hub

1. **Introduction to Git**
2. **Key Concepts in Git**
3. **Getting started with Git**
4. **Basic Commands**
5. **Best Practices for Git**
6. **Git Workflows**

Git provides different workflows to manage changes to a codebase.

The three most common workflows are :

1. centralized
2. feature branch
3. Gitflow.



Git Workflows

Centralized Workflow:

Git and Git-hub

1. **Introduction to Git**
2. **Key Concepts in Git**
3. **Getting started with Git**
4. **Basic Commands**
5. **Best Practices for Git**
6. **Git Workflows**

In a centralized workflow, all changes are made directly to the main branch of the repository.

This workflow is simple and straightforward, but it can lead to conflicts if multiple developers are working on the same files at the same time.



Git Workflows

Feature Branch Workflow:

Git and Git-hub

1. **Introduction to Git**
2. **Key Concepts in Git**
3. **Getting started with Git**
4. **Basic Commands**
5. **Best Practices for Git**
6. **Git Workflows**

In a feature branch workflow, each feature or change is developed in a separate branch.

This allows developers to work independently and avoid conflicts. Once a feature is complete, it can be merged back into the main branch.



Git Workflows

Gitflow Workflow:

Git and Git-hub

1. **Introduction to Git**
2. **Key Concepts in Git**
3. **Getting started with Git**
4. **Basic Commands**
5. **Best Practices for Git**
6. **Git Workflows**

The Gitflow workflow is similar to the feature branch workflow, but it adds additional branches for release management.

In this workflow, there are separate branches for feature development, release preparation, and bug fixes.



THANK YOU