Google Developer Student Clubs
Techno India University

Presents

WEB DASH

WEB DEVELOPMENT
BOOTCAMP

```
ild: Column(
 crossAxisAlignment: CrossAxis
 children: [
   /*2*/
   Conta
     pad
     chi
       '
       s

       )
     ),
   ),
   Text(
     'Ka
     sty
       c
     ),
   ),
```

Version Control and Git

Google Developer Student Clubs

Version control systems provide a systematic way to manage changes to the source code. It enables collaboration, facilitates tracking of modifications, and ensures a structured approach to development.

Benefits of Version Control

Collaboration	History	Branching	Conflict Resolution

Types of Version Control Systems

Centralized Version Control

Distributed Version Control

Google Developer Student Clubs
Techno India University

Git Fundamentals

```
lookup.KeyValue
f.constant(['em
=tf.constant([0
.lookup.StaticV

_buckets=5)
```

Key Concepts

Repository

A **repository** in the context of version control is a data structure that stores metadata for a set of files and directories. It serves as the central hub for a project.

Commit

A **commit** is a fundamental concept in version control systems, representing a snapshot of changes made to the files in a repository at a specific point in time.

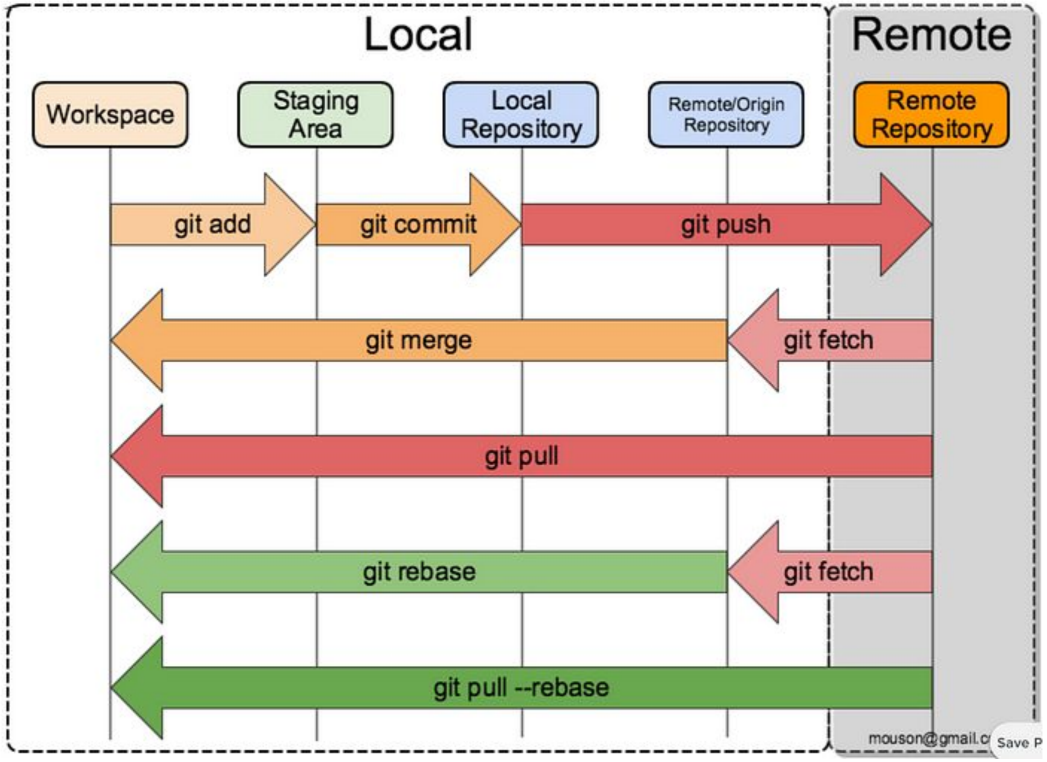
Branch

A **branch** is a parallel line of development within a repository. Think of it as an independent line of code evolution.

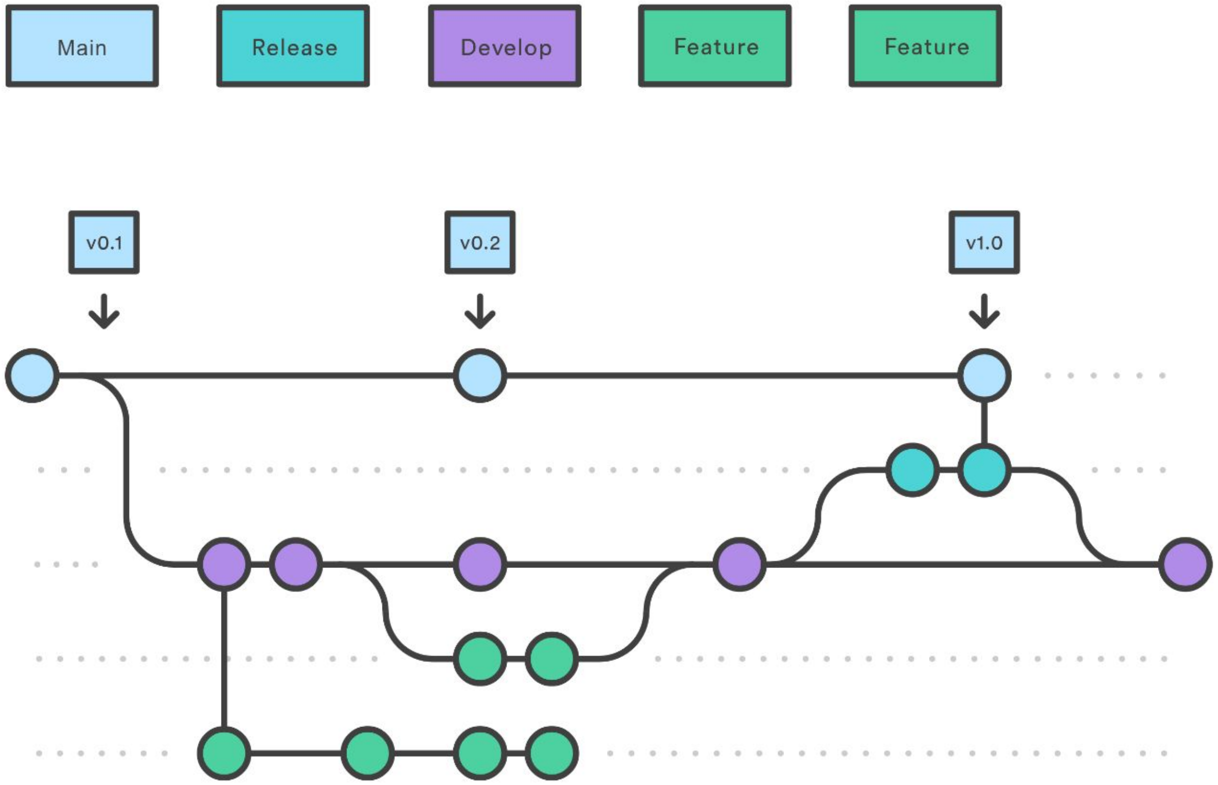
Merge

Merging is the process of combining changes from one branch into another. It brings together the independent lines of development, allowing the integration of new features or bug fixes.

Stages of Git Workflow



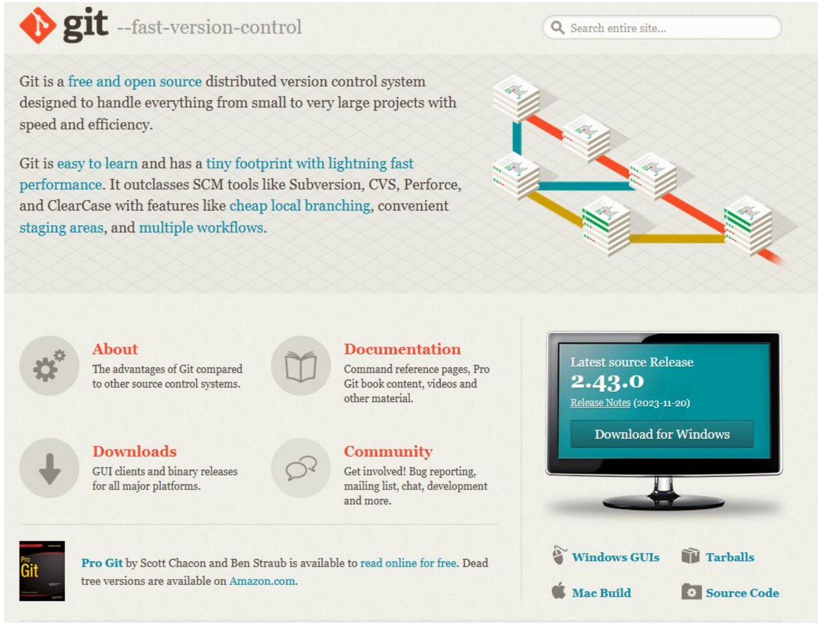
Standard Visualization of a Gitflow



Google Developer Student Clubs
Techno India University

Setting Up Git

https://git-scm.com/



git --fast-version-control

Search entire site...

Git is a free and open source distributed version control system designed to handle everything from small to very large projects with speed and efficiency.

Git is easy to learn and has a tiny footprint with lightning fast performance. It outclasses SCM tools like Subversion, CVS, Perforce, and ClearCase with features like cheap local branching, convenient staging areas, and multiple workflows.

About
The advantages of Git compared to other source control systems.

Documentation
Command reference pages, Pro Git book content, videos and other material.

Downloads
GUI clients and binary releases for all major platforms.

Community
Get involved! Bug reporting, mailing list, chat, development and more.

Latest source Release
2.43.0
Release Notes (2023-11-20)
Download for Windows

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

Windows GUIs Tarballs
Mac Build Source Code

Setting Username and Email

```
git config --global user.name "Your Name"
```

```
git config --global user.email "your.email@example.com"
```

Google Developer Student Clubs
Techno India University

Basic Git Commands

```
lookup.KeyValue
f.constant(['em
=tf.constant([0
.lookup.StaticV

_buckets=5)
```

git init - Initializing a Repository

git add - Staging Changes

git commit - Committing Changes

git log - Viewing Commit History

git diff - Viewing Changes

Google Developer Student Clubs
Techno India University

Branching and Merging

```
lookup.KeyValue
f.constant(['em
=tf.constant([0
.lookup.StaticV

_buckets=5)
```

Branch

Branching in Git is like creating parallel universes within your project. It allows you to work on new features or bug fixes without affecting the main codebase.

Creating a New Branch

```
git branch <branch-name>
```

$\downarrow$

```
git checkout <branch-name>
```

Deleting Branches

```
git branch -d <branch-name>
```

Merging

While branching allows for independent work, merging brings these independent lines of development back together. It's the process of combining changes from one branch into another.

Merging Branches

To merge changes from one branch into another, you typically switch to the branch you want to merge changes into and use:

```
git merge <source-branch>
```

Something for you to research

Merge Conflicts

Google Developer Student Clubs
Techno India University

Remote Repositories and Collaboration

```
lookup.KeyValue
f.constant(['en
=tf.constant([0
.lookup.StaticV

_buckets=5)
```

Introduction to Remote Repositories

A **remote repository** is a version-controlled repository that is not on your local machine but hosted on a **server** or a **cloud platform**.

It serves as a centralized hub where multiple developers can collaborate on a project. Each developer has their local copy of the repository, and changes can be synchronized between the local and remote repositories.

Cloning Remote Repositories

To clone a repository, you use the command:

```
git clone <repository-url>
```

This creates a local copy of the entire remote repository on your machine.

Fetching and Pulling Changes

Fetching Changes

```
git fetch
```

The git fetch command retrieves changes from the remote repository, updating your local copy's knowledge of the remote branches.

It doesn't automatically merge these changes into your working directory, providing you control over when and how to integrate the updates.

Pulling Changes

```
git pull
```

The git pull command, on the other hand, fetches changes and automatically merges them into your local working directory.

It's a convenient way to synchronize your local copy with the remote repository in one step.

Pushing Changes

```
git push
```

This command sends your commits to the remote repository, updating its history.

It's crucial to ensure you have the latest changes from the remote repository before pushing to avoid conflicts.

Force Pushing

```
git push --force
```

In certain situations, you might need to force push changes. This overwrites the remote branch with your local branch, potentially discarding other developers' changes.

Use force push cautiously and only when necessary

Project Time!

Your Own Github Repo

Each given project will be in a **different branch**.
With its included files and folders.

The main branch will contain a **readme.md** file containing :

- Your Name
- Batch
- Email

In total you will have **3 Projects** - 2 Small Project, 1 Frontend Project

```
1*/
ild: Column(
crossAxisAlignment: CrossAxis
children: [
  /*2*/
  Conta
    pad
    chi
      '
      s

      )
    ),
  ),
  Text(
    'Ka
    sty
      c
    ),
  ),
```

Some things for you to research

Google Developer Student Clubs

Forking

Git Reflog

Rebasing

Reverting

Merge Conflicts

See You Tomorrow!