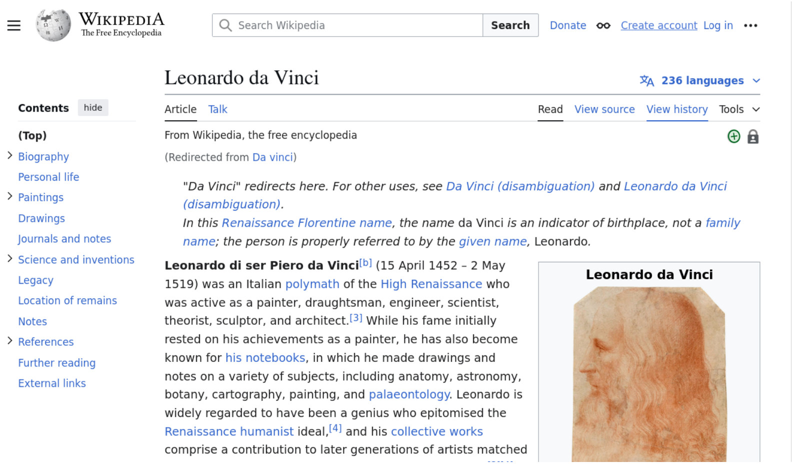
Git: A Developer’s Time Machine

Nandakumar Edamana

2025-01-16

Tracking Changes



Leonardo da Vinci

From Wikipedia, the free encyclopedia

(Redirected from Da vinci)

"Da Vinci" redirects here. For other uses, see Da Vinci (disambiguation) and Leonardo da Vinci (disambiguation).

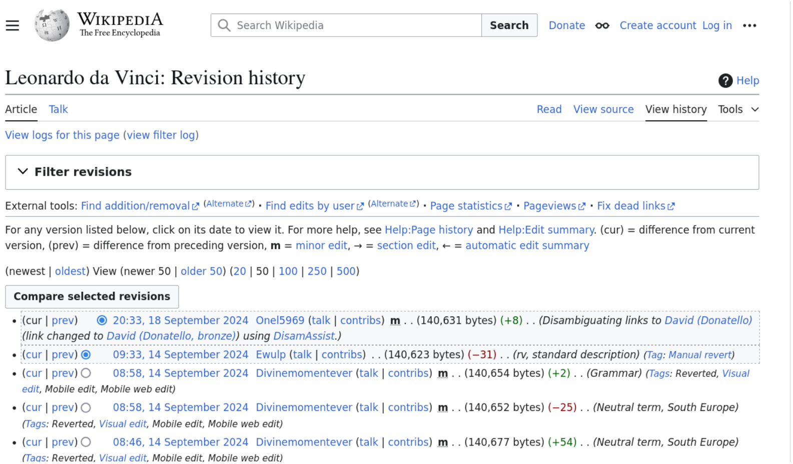
In this Renaissance Florentine name, the name da Vinci *is an indicator of birthplace, not a family name; the person is properly referred to by the given name,* Leonardo.

Leonardo di ser Piero da Vinci[b] (15 April 1452 – 2 May 1519) was an Italian polymath of the High Renaissance who was active as a painter, draughtsman, engineer, scientist, theorist, sculptor, and architect.[3] While his fame initially rested on his achievements as a painter, he has also become known for his notebooks, in which he made drawings and notes on a variety of subjects, including anatomy, astronomy, botany, cartography, painting, and palaeontology. Leonardo is widely regarded to have been a genius who epitomised the Renaissance humanist ideal,[4] and his collective works comprise a contribution to later generations of artists matched

Leonardo da Vinci

Figure 1: Wikipedia article about Da Vinci

Tracking Changes



Leonardo da Vinci: Revision history

View logs for this page (view filter log)

External tools: Find addition/removal (Alternate) • Find edits by user (Alternate) • Page statistics • Pageviews • Fix dead links

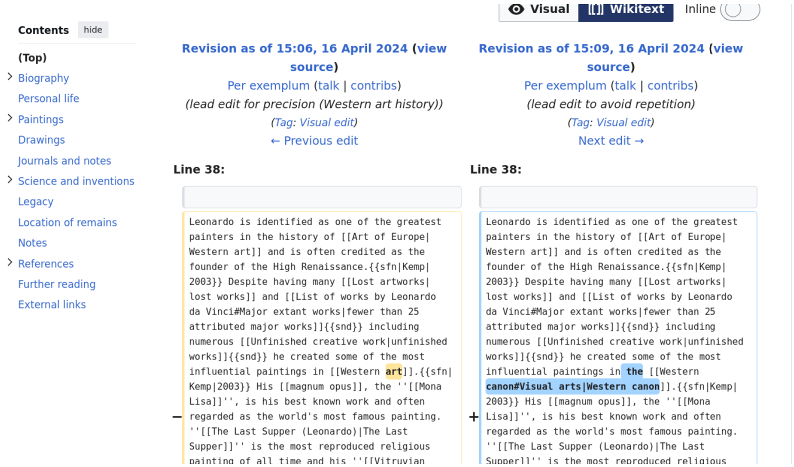
For any version listed below, click on its date to view it. For more help, see Help:Page history and Help:Edit summary. (cur) = difference from current version, (prev) = difference from preceding version, **m** = minor edit, → = section edit, ← = automatic edit summary

(newest | oldest) View (newer 50 | older 50) (20 | 50 | 100 | 250 | 500)

- (cur | prev) 20:33, 18 September 2024 Onel5969 (talk | contribs) **m** . . (140,631 bytes) (+8) . . *(Disambiguating links to David (Donatello) (link changed to David (Donatello, bronze)) using DisamAssist.)*
- (cur | prev) 09:33, 14 September 2024 Ewulp (talk | contribs) . . (140,623 bytes) (−31) . . *(rv, standard description) (Tag: Manual revert)*
- (cur | prev) 08:58, 14 September 2024 Divinemomentever (talk | contribs) **m** . . (140,654 bytes) (+2) . . *(Grammar) (Tags: Reverted, Visual edit, Mobile edit, Mobile web edit)*
- (cur | prev) 08:58, 14 September 2024 Divinemomentever (talk | contribs) **m** . . (140,652 bytes) (−25) . . *(Neutral term, South Europe) (Tags: Reverted, Visual edit, Mobile edit, Mobile web edit)*
- (cur | prev) 08:46, 14 September 2024 Divinemomentever (talk | contribs) **m** . . (140,677 bytes) (+54) . . *(Neutral term, South Europe) (Tags: Reverted, Visual edit, Mobile edit, Mobile web edit)*

Figure 2: Edit History of the Wikipedia article

Tracking Changes



Revision as of 15:06, 16 April 2024 (view source)
Per exemplum (talk | contribs)
(lead edit for precision (Western art history))
(Tag: Visual edit)
← Previous edit

Revision as of 15:09, 16 April 2024 (view source)
Per exemplum (talk | contribs)
(lead edit to avoid repetition)
(Tag: Visual edit)
Next edit →

Line 38:

```
−  Leonardo is identified as one of the greatest
   painters in the history of [[Art of Europe|
   Western art]] and is often credited as the
   founder of the High Renaissance.{{sfn|Kemp|
   2003}} Despite having many [[Lost artworks|
   lost works]] and [[List of works by Leonardo
   da Vinci#Major extant works|fewer than 25
   attributed major works]]{{snd}} including
   numerous [[Unfinished creative work|unfinished
   works]]{{snd}} he created some of the most
   influential paintings in [[Western art]].{{sfn|
   Kemp|2003}} His [[magnum opus]], the ''[[Mona
   Lisa]]'', is his best known work and often
   regarded as the world's most famous painting.
   ''[[The Last Supper (Leonardo)|The Last
   Supper]]'' is the most reproduced religious
   painting of all time and his ''[[Vitruvian
```

Line 38:

```
+  Leonardo is identified as one of the greatest
   painters in the history of [[Art of Europe|
   Western art]] and is often credited as the
   founder of the High Renaissance.{{sfn|Kemp|
   2003}} Despite having many [[Lost artworks|
   lost works]] and [[List of works by Leonardo
   da Vinci#Major extant works|fewer than 25
   attributed major works]]{{snd}} including
   numerous [[Unfinished creative work|unfinished
   works]]{{snd}} he created some of the most
   influential paintings in the [[Western
   canon#Visual arts|Western canon]].{{sfn|Kemp|
   2003}} His [[magnum opus]], the ''[[Mona
   Lisa]]'', is his best known work and often
   regarded as the world's most famous painting.
   ''[[The Last Supper (Leonardo)|The Last
   Supper]]'' is the most reproduced religious
```

Figure 3: Details of one edit of the Wikipedia article

Version Control

- Keep track of the changes you make

Version Control

- Keep track of the changes you make
- Locate and fix bugs easily

Version Control

- Keep track of the changes you make
- Locate and fix bugs easily
- Work independently on different features and merge them cleanly

Version Control

- Keep track of the changes you make
- Locate and fix bugs easily
- Work independently on different features and merge them cleanly
- Try out new features and rollback

Version Control

- Keep track of the changes you make
- Locate and fix bugs easily
- Work independently on different features and merge them cleanly
- Try out new features and rollback
- Protection from accidental data loss and corruption (to an extent)

Version Control

- Keep track of the changes you make
- Locate and fix bugs easily
- Work independently on different features and merge them cleanly
- Try out new features and rollback
- Protection from accidental data loss and corruption (to an extent)
- Not just Git; Mercurial, Fossil, etc. (out-of-fashion: SVN, CVS, Baazar)

Git vs GitHub

- Git can run totally offline, without even a local server

Git vs GitHub

- Git can run totally offline, without even a local server
- GitHub: an online platform to host and collaborate Git-based projects

Git vs GitHub

- Git can run totally offline, without even a local server
- GitHub: an online platform to host and collaborate Git-based projects
- Other: GitLab, BitBucket, etc.

Git vs GitHub

- Git can run totally offline, without even a local server
- GitHub: an online platform to host and collaborate Git-based projects
- Other: GitLab, BitBucket, etc.
- GitLab can be self-hosted

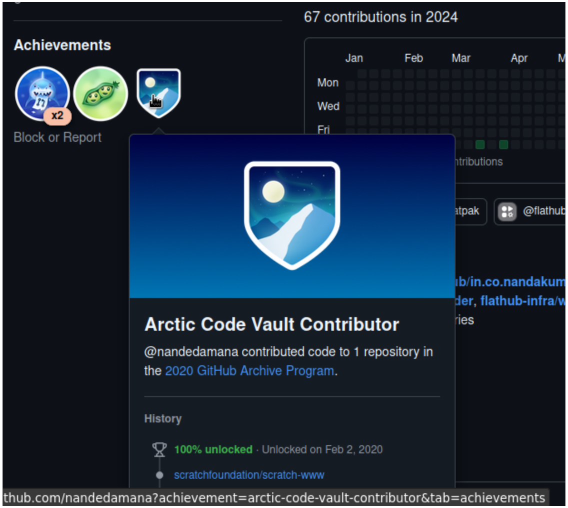
Git vs GitHub

- Git can run totally offline, without even a local server
- GitHub: an online platform to host and collaborate Git-based projects
- Other: GitLab, BitBucket, etc.
- GitLab can be self-hosted
- openforge.gov.in

Git vs GitHub

- Git can run totally offline, without even a local server
- GitHub: an online platform to host and collaborate Git-based projects
- Other: GitLab, BitBucket, etc.
- GitLab can be self-hosted
- openforge.gov.in

GitHub Profile


67 contributions in 2024

Achievements

Block or Report

Arctic Code Vault Contributor

@nandedamana contributed code to 1 repository in the 2020 GitHub Archive Program.

History

100% unlocked · Unlocked on Feb 2, 2020

scratchfoundation/scratch-www

thub.com/nandedamana?achievement=arctic-code-vault-contributor&tab=achievements

Figure 4: GitHub Badges

GitHub Contribution Types

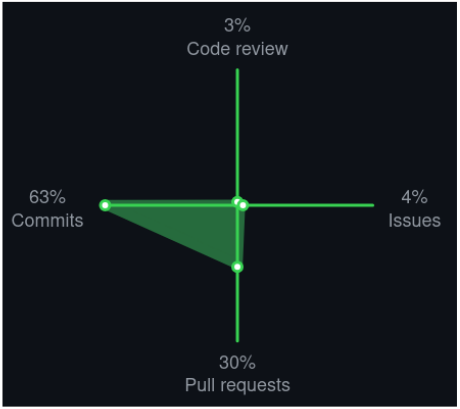
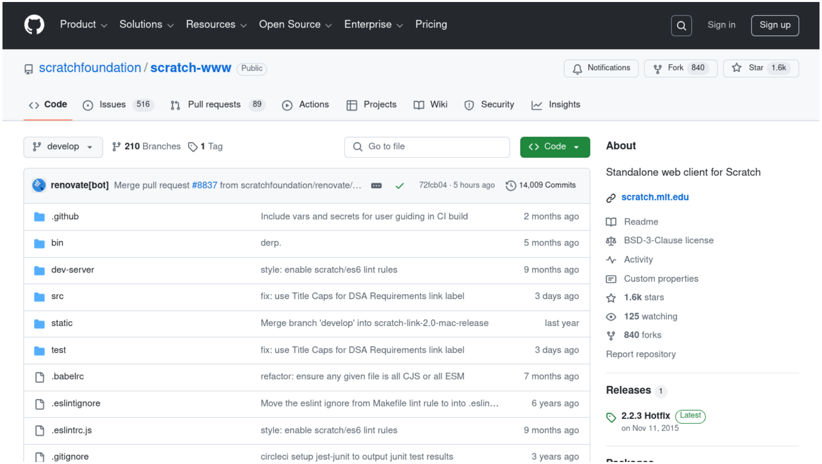


Figure 5: GitHub Contribution Types

A Project on GitHub



renovate[bot] Merge pull request #8837 from scratchfoundation/renovate/… 72fcb04 · 5 hours ago 14,009 Commits

.github	Include vars and secrets for user guiding in CI build	2 months ago
bin	derp.	5 months ago
dev-server	style: enable scratch/es6 lint rules	9 months ago
src	fix: use Title Caps for DSA Requirements link label	3 days ago
static	Merge branch 'develop' into scratch-link-2.0-mac-release	last year
test	fix: use Title Caps for DSA Requirements link label	3 days ago
.babelrc	refactor: ensure any given file is all CJS or all ESM	7 months ago
.eslintignore	Move the eslint ignore from Makefile lint rule to into .eslin…	6 years ago
.eslintrc.js	style: enable scratch/es6 lint rules	9 months ago
.gitignore	circleci setup jest-junit to output junit test results	3 years ago

About

Standalone web client for Scratch

scratch.mit.edu

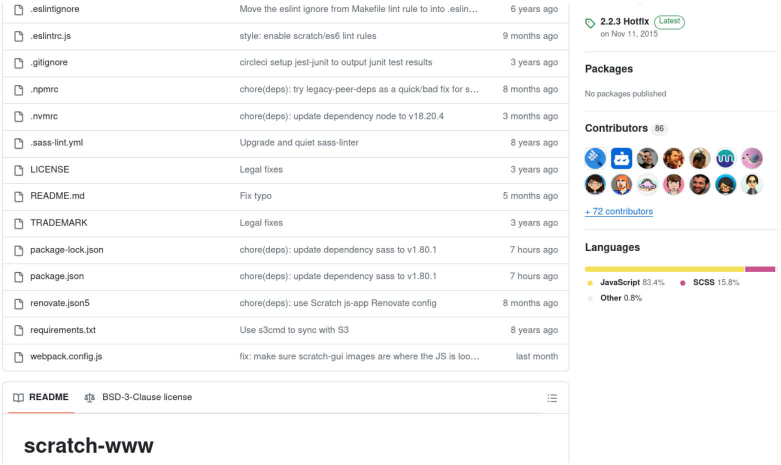
Readme
BSD-3-Clause license
Activity
Custom properties
1.6k stars
125 watching
840 forks
Report repository

Releases 1

2.2.3 Hotfix Latest
on Nov 11, 2015

Figure 6: scratch-www on GitHub (1)

A Project on GitHub



.eslintignore	Move the eslint ignore from Makefile lint rule to into .eslin…	6 years ago
.eslintrc.js	style: enable scratch/es6 lint rules	9 months ago
.gitignore	circleci setup jest-junit to output junit test results	3 years ago
.npmrc	chore(deps): try legacy-peer-deps as a quick/bad fix for s…	8 months ago
.nvmrc	chore(deps): update dependency node to v18.20.4	3 months ago
.sass-lint.yml	Upgrade and quiet sass-linter	8 years ago
LICENSE	Legal fixes	3 years ago
README.md	Fix typo	5 months ago
TRADEMARK	Legal fixes	3 years ago
package-lock.json	chore(deps): update dependency sass to v1.80.1	7 hours ago
package.json	chore(deps): update dependency sass to v1.80.1	7 hours ago
renovate.json5	chore(deps): use Scratch js-app Renovate config	8 months ago
requirements.txt	Use s3cmd to sync with S3	8 years ago
webpack.config.js	fix: make sure scratch-gui images are where the JS is loo…	last month

README BSD-3-Clause license

scratch-www

2.2.3 Hotfix Latest
on Nov 11, 2015

Packages

No packages published

Contributors 86

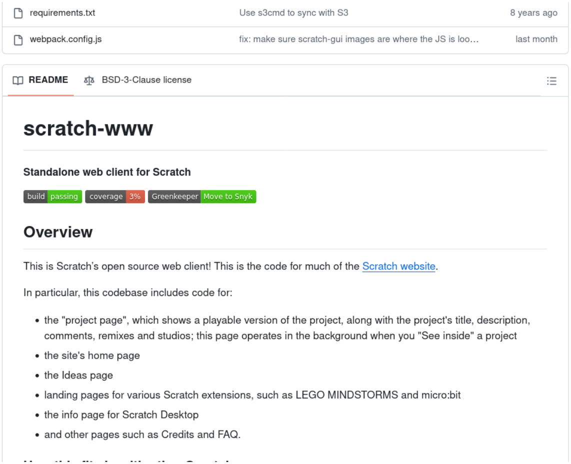
+ 72 contributors

Languages

● **JavaScript** 83.4% ● **SCSS** 15.8%
● **Other** 0.8%

Figure 7: scratch-www on GitHub (2)

A Project on GitHub



requirements.txt — Use s3cmd to sync with S3 — 8 years ago

webpack.config.js — fix: make sure scratch-gui images are where the JS is loo… — last month

scratch-www

Standalone web client for Scratch

Overview

This is Scratch's open source web client! This is the code for much of the Scratch website.

In particular, this codebase includes code for:

- the "project page", which shows a playable version of the project, along with the project's title, description, comments, remixes and studios; this page operates in the background when you "See inside" a project
- the site's home page
- the Ideas page
- landing pages for various Scratch extensions, such as LEGO MINDSTORMS and micro:bit
- the info page for Scratch Desktop
- and other pages such as Credits and FAQ.

Figure 8: scratch-www on GitHub (3)

Git in Action

`git log` output:

```
...
commit 668cf51dd378f815b1392bdbc0c08fb3bef53772
Author: Nandakumar Edamana <EMAIL>
Date:   Wed May 24 07:03:51 2023 +0530

    ui improvements including grid for viewport

commit 8048bab246e6d8e59a45cc7792bb93ee5e33c367
Author: Nandakumar Edamana <EMAIL>
Date:   Wed May 24 06:54:44 2023 +0530

    support hard brush strokes with anti-aliasing
...
```

Git in Action

How many commits did I create on Sundays?

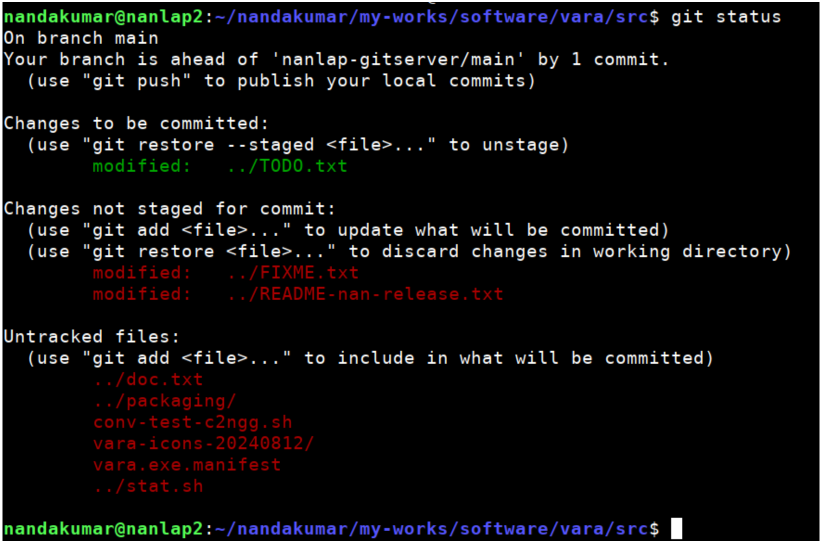
Git in Action

How many commits did I create on Sundays?

```
$ git log|grep 'Sun '|wc -l
78
```

Git Status

`git status` output:



```
nandakumar@nanlap2:~/nandakumar/my-works/software/vara/src$ git status
On branch main
Your branch is ahead of 'nanlap-gitserver/main' by 1 commit.
  (use "git push" to publish your local commits)

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   ../TODO.txt

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   ../FIXME.txt
        modified:   ../README-nan-release.txt

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        ../doc.txt
        ../packaging/
        conv-test-c2ngg.sh
        vara-icons-20240812/
        vara.exe.manifest
        ../stat.sh

nandakumar@nanlap2:~/nandakumar/my-works/software/vara/src$ █
```

Figure 9: git status

Terminology

States of a repo:

- **Working Tree** - files in your directory; sandbox; contains latest modifications
- **Index (Staging Area)** - changes added for committing
- **Branch History** - changes committed, last snapshot denoted by *HEAD*

One-time Setup for a User

```
git config --global user.name 'Your Name'
git config --global user.email ROLLNO@smail.iitpkd.ac.in
```

How to Start a Project

```
mkdir PROJECT-NAME
cd PROJECT-NAME
```

How to Start a Project

```
mkdir PROJECT-NAME
cd PROJECT-NAME

# initialize a repo in the current directory
git init
```

How to Start a Project

```
mkdir PROJECT-NAME
cd PROJECT-NAME

# initialize a repo in the current directory
git init

# rename the default branch `master`
git branch -m main
```

Adding Files and Committing

```
git add FILE1 FILE2 ...
git commit -m 'Commit message'
```

Example: Adding and Committing

```
# Write, save, compile, and test a Hello World program
# (say hello.c); then:

git add hello.c
git commit -m 'Basic version of Hello World'
```

Example: Adding and Committing (Stage 1)

`git status` **after creation**:

```
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be commi
    hello.c
```

Example: Adding and Committing (Stage 2)

`git status` **after `git add hello.c`**:

```
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   hello.c
```

Example: Adding and Committing (Stage 3)

`git status` **after commit**:

```
On branch master
nothing to commit, working tree clean
```

Example: Adding and Committing

```
# Now edit hello.c to add some feature (say, color),
# save, and test again; then:

git add hello.c
git commit -m 'Feature: add color to the output'
```

Example: Adding and Committing

`git status` **after modifying hello.c, before adding and committing**:

```
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committe
  (use "git restore <file>..." to discard changes in worki
    modified:   hello.c

no changes added to commit (use "git add" and/or "git comm
```

State Changes

Time	Action	Working Tree	Index	History
0		empty	clean	empty

State Changes

Time	Action	Working Tree	Index	History
0		empty	clean	empty
1	Create hello.c	hello.c (dirty)	clean	empty

State Changes

Time	Action	Working Tree	Index	History
0		empty	clean	empty
1	Create hello.c	hello.c (dirty)	clean	empty
2	`git add hello.c`	hello.c	hello.c	empty

State Changes

Time	Action	Working Tree	Index	History
0		empty	clean	empty
1	Create hello.c	hello.c (dirty)	clean	empty
2	`git add hello.c`	hello.c	hello.c	empty
3	`git commit hello.c`	hello.c	clean	hello.c

State Changes

Time	Action	Working Tree	Index	History
0		empty	clean	empty
1	Create hello.c	hello.c (dirty)	clean	empty
2	`git add hello.c`	hello.c	hello.c	empty
3	`git commit hello.c`	hello.c	clean	hello.c
4	Modify hello.c	hello.c (dirty)	clean	hello.c (old)

Quiz

You committed the initial version of hello.c, modified it, and committed again. Will you be able to access the initial version now?

Branching and Merging

Usual workflow:

```
git branch -v
git checkout -b feature-x
```

Branching and Merging

Usual workflow:

```
git branch -v
git checkout -b feature-x

# Modify, add, and commit files
# like we've already seen; then:
```

Branching and Merging

Usual workflow:

```
git branch -v
git checkout -b feature-x

# Modify, add, and commit files
# like we've already seen; then:

git checkout main
git merge feature-x
git branch -d feature-x
```

Branching and Merging

```
...
git checkout main          # (1)
git merge feature-x        # (2)
git branch -d feature-x
```

- hello.c gets restored after (1) (i.e., feature-x gone), gets updated again after (2)

Branching and Merging

```
...

git checkout main          # (1)
git merge feature-x        # (2)
git branch -d feature-x
```

- hello.c gets restored after (1) (i.e., feature-x gone), gets updated again after (2)

Connected Workflow

```
+--------+
|        | -------- clone, pull ----------> You
|        | <---------- push ----------------'
| Remote |
|        | -------- clone, pull ----------> Your friend
|        | <---------- push ----------------'
+--------+
```

Connected Workflow

- `git clone URL` – create a local clone of the repo pointed by URL
- `git clone --depth=1 URL` – shallow clone (no history); useful if not contributing
- `git pull` – incorporate changes from the remote to the current local branch
- `git push` – upload changes from all local branches to the remote
- `git remote -v` – list remotes

Standard GitHub Workflow

1. File/pick an issue

Standard GitHub Workflow

1. File/pick an issue
2. Fork and clone your fork

Standard GitHub Workflow

1. File/pick an issue
2. Fork and clone your fork
3. Branch

Standard GitHub Workflow

1. File/pick an issue
2. Fork and clone your fork
3. Branch
4. Commit your changes and push
5. File *Pull Request* (aka *Merge Request*)

Standard GitHub Workflow

1. File/pick an issue
2. Fork and clone your fork
3. Branch
4. Commit your changes and push
5. File *Pull Request* (aka *Merge Request*)
6. Upstream reviews, accepts/rejects the PR

Standard GitHub Workflow

1. File/pick an issue
2. Fork and clone your fork
3. Branch
4. Commit your changes and push
5. File *Pull Request* (aka *Merge Request*)
6. Upstream reviews, accepts/rejects the PR

Some of these actions are offered by the platform, not Git.

Example: Cloning

```
git clone \
  https://gitlab.gnome.org/GNOME/gnome-calculator.git

cd gnome-calculator

# Now create a branch and start working
# on your feature or fix
```

Git Counterparts of Standard Commands

- `git mv`
- `git rm`
- `git grep`

Don't forget to commit after `git mv` and `git rm`.

Reset, Restore and Revert

Based on the section “Reset, restore and revert" from the man page:

- `git revert` - make a new commit that reverts other commits (affects history)
- `git restore` - restore files in the working tree or the index[1] from either the index or another commit
- `git reset` - move “the tip in order to add or remove commits from the branch" (affects history)

The man page says: “git reset can also be used to restore the index, overlapping with git restore."

[1] `--staged`

Best Practices

Make sure the following are not added:

- Secrets (passwords, keys, etc.)

Best Practices

Make sure the following are not added:

- Secrets (passwords, keys, etc.)
- Executable binary files

Best Practices

Make sure the following are not added:

- Secrets (passwords, keys, etc.)
- Executable binary files
- Anything that can be easily generated from the source that is already included

Best Practices

Make sure the following are not added:

- Secrets (passwords, keys, etc.)
- Executable binary files
- Anything that can be easily generated from the source that is already included
- Proprietary files that you don't have rights to

Best Practices

- Do not commit without checking status and diff

Best Practices

- Do not commit without checking status and diff
- Use meaningful commit messages

Best Practices

- Do not commit without checking status and diff
- Use meaningful commit messages
- Set and follow a convention (tense, lowercase, prefixes like “fix:", etc.)

Best Practices

- Do not commit without checking status and diff
- Use meaningful commit messages
- Set and follow a convention (tense, lowercase, prefixes like “fix:", etc.)
- Always branch

Best Practices

- Do not commit without checking status and diff
- Use meaningful commit messages
- Set and follow a convention (tense, lowercase, prefixes like “fix:", etc.)
- Always branch
- Pull before you start working

Best Practices

- Do not commit without checking status and diff
- Use meaningful commit messages
- Set and follow a convention (tense, lowercase, prefixes like “fix:", etc.)
- Always branch
- Pull before you start working
- Avoid altering history after pushing; avoid force pushses in general

.gitignore

Demo

Referencing

- **HEAD** - Points to the latest commit in the branch history
- **HEAD^n** - *n*th parent commit of HEAD
- **HEAD~n** - *n*th predecessor commit of HEAD
- **HEAD@{n}** - where HEAD used to be *n* moves ago (you can move HEAD around)

Learn Git

- Man pages: giteveryday(7), gittutorial(7), gitworkflows(7)
- Official documentation on https://git-scm.com/
- Interactive resources from platforms like GitHub and GitLab
- https://learngitbranching.js.org/ (game)