

The Alan Turing Institute

Collaborative Coding

Levan Bokeria
Lydia France
Federico Nanni



Presentation Overview

- Slides and hands-on exercises

Presentation Overview

- Slides and hands-on exercises

Slido (anonymous):

- www.slido.com
- #4030909



Presentation Overview

- Slides and hands-on exercises
- Git basics:
 - Commits
 - Branches

Slido (anonymous):

- www.slido.com
- #4030909



Presentation Overview

- Slides and hands-on exercises
- Git basics:
 - Commits
 - Branches
- GitHub basics:
 - Forks
 - Issues and project boards
 - Pull Requests

Slido (anonymous):

- www.slido.com
- #4030909



Presentation Overview

- Slides and hands-on exercises
- Git basics:
 - Commits
 - Branches
- GitHub basics:
 - Forks
 - Issues and project boards
 - Pull Requests
- **The Git flow model**

Slido (anonymous):

- www.slido.com
- #4030909



Presentation Overview

- Slides and hands-on exercises
- Git basics:
 - Commits
 - Branches
- GitHub basics:
 - Forks
 - Issues and project boards
 - Pull Requests
- **The Git flow model**

We will not cover:

- Coding and programming examples.
Only markdown files.
- Git version control locally on your machine

Slido (anonymous):

- www.slido.com
- #4030909



Part 1: Git and GitHub

Git Version Control – the basics

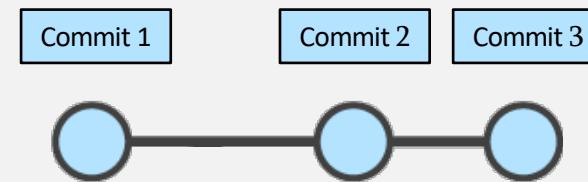
- **Version control**, or source control, involves tracking and managing changes to code.

Git Version Control – the basics

- **Version control**, or source control, involves tracking and managing changes to code.
- **Commits**: Every change is recorded as a "commit" and given an identifier (a hash code).

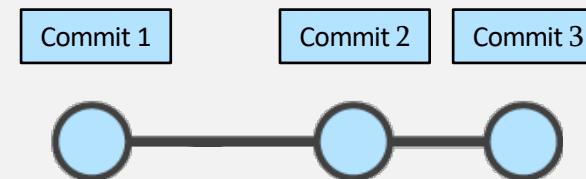
Git Version Control – the basics

- **Version control**, or source control, involves tracking and managing changes to code.
- **Commits**: Every change is recorded as a “commit” and given an identifier (a hash code).



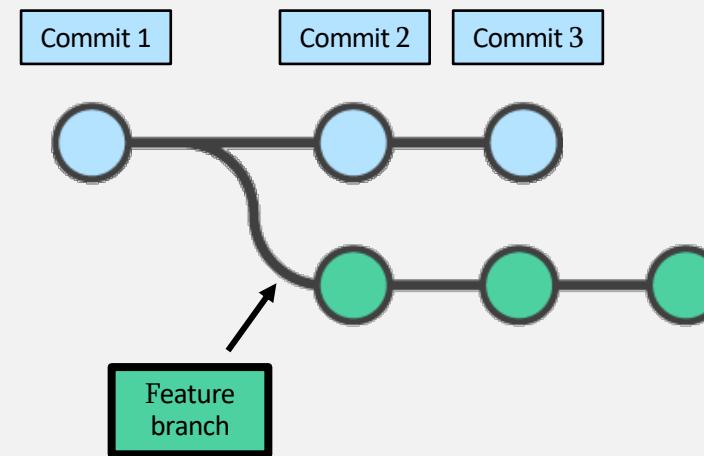
Git Version Control – the basics

- **Version control**, or source control, involves tracking and managing changes to code.
- **Commits**: Every change is recorded as a “commit” and given an identifier (a hash code).
- **Branches**: parallel versions of code.



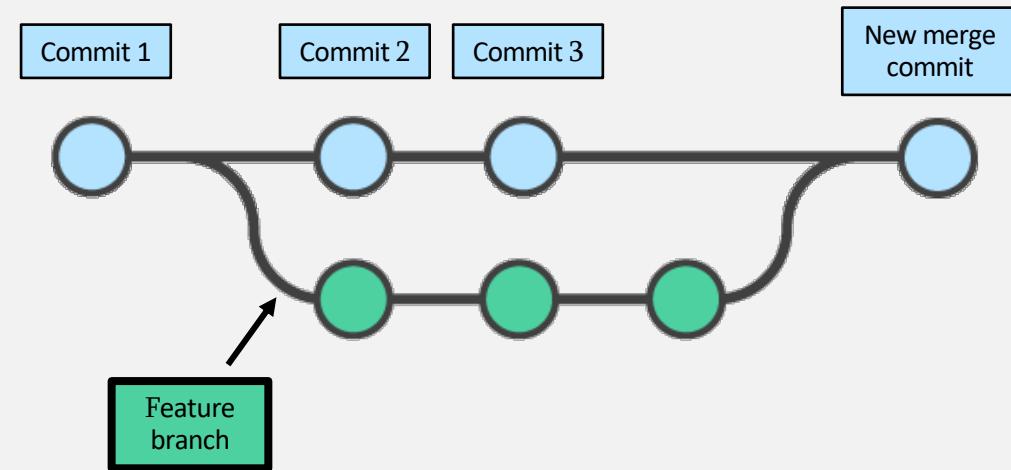
Git Version Control – the basics

- **Version control**, or source control, involves tracking and managing changes to code.
- **Commits**: Every change is recorded as a “commit” and given an identifier (a hash code).
- **Branches**: parallel versions of code.



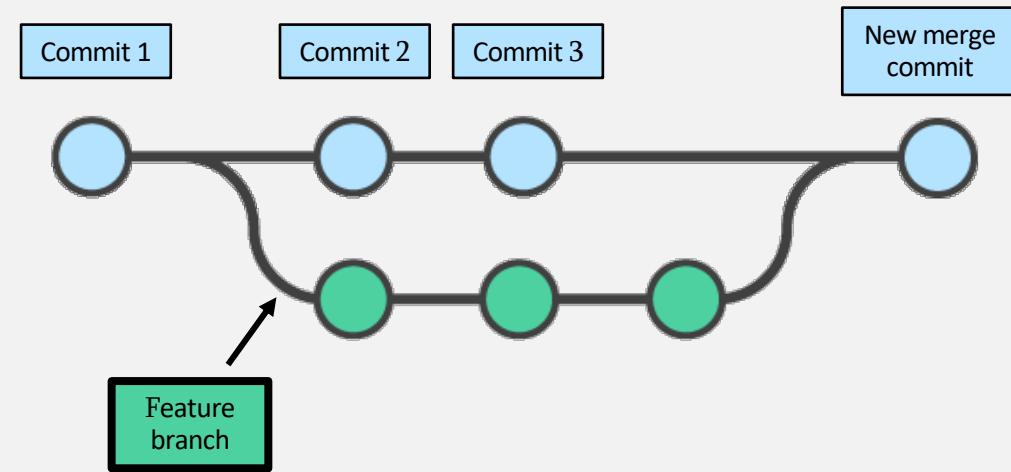
Git Version Control – the basics

- **Version control**, or source control, involves tracking and managing changes to code.
- **Commits**: Every change is recorded as a “commit” and given an identifier (a hash code).
- **Branches**: parallel versions of code.



Git Version Control – the basics

- **Version control**, or source control, involves tracking and managing changes to code.
- **Commits**: Every change is recorded as a “commit” and given an identifier (a hash code).
- **Branches**: parallel versions of code.
- **Merging branches**: fusing of code from two different branches, usually done via a “pull request” (also known as a “merge request”)



GitHub – a simple example repo

GitHub – a simple example repo

The screenshot shows a GitHub repository page for 'alan-turing-institute / github-example'. The repository is public and has 5 watches, 51 forks, and 0 stars. It contains 2 branches (main) and 2 tags. The main branch has 9 commits. Recent commits include 'Add Whernside' (3494fdd), 'Add Wales' (Pennines.md), 'Add Helvellyn' (Wales.md), and 'Include lakes in the scope' (lakeland.md). The repository has no description or website provided. The owner is 'hut23' with 5 watchers. There are 2 tags listed.

Code | **Issues** | **Pull requests** | **Actions** | **Projects** | **Security** | **Insights**

github-example Public

Watch 5 Fork 51 Star 0

main 2 branches 2 tags

Turing Developer Add Whernside 3494fdd on May 30 9 commits

Pennines.md Add Whernside 2 months ago

Wales.md Add wales 2 months ago

lakeland.md Add Helvellyn 2 months ago

test.md Include lakes in the scope 2 months ago

About

No description or website provided.

hut23

Activity

0 stars

5 watching

51 forks

Report repository

Releases

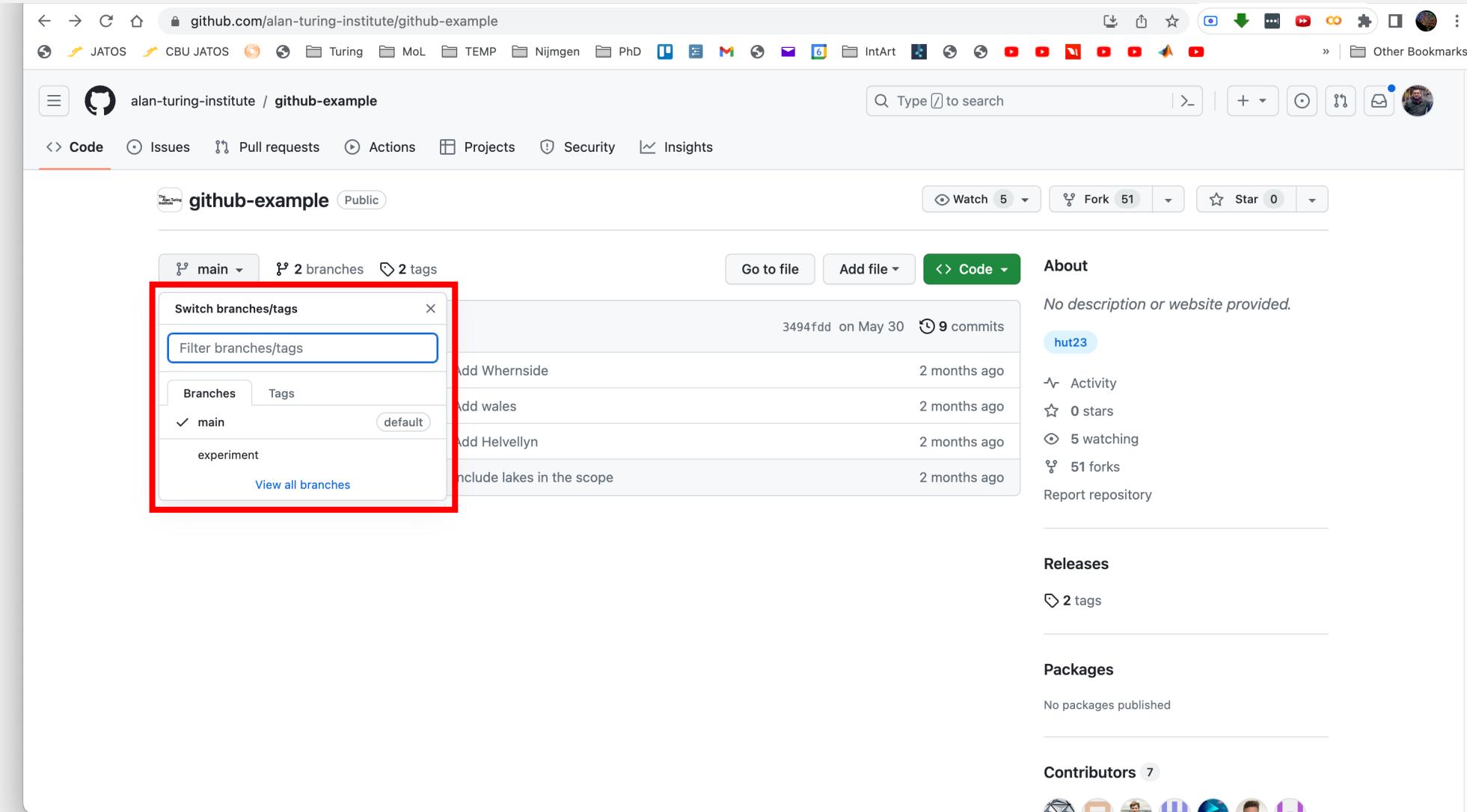
2 tags

Packages

No packages published

Contributors 7

GitHub – a simple example repo



The screenshot shows a GitHub repository page for 'alan-turing-institute/github-example'. The repository is public and has 5 watches, 51 forks, and 0 stars. It contains 2 branches (main and experiment) and 2 tags. A red box highlights the 'Switch branches/tags' dropdown menu, which includes a 'Filter branches/tags' input field and lists 'main' (selected) and 'experiment'. The main content area shows a commit history with 9 commits, all made by 'hut23' on May 30. The commits are: 'Add Whernside', 'Add wales', 'Add Helvellyn', and 'include lakes in the scope'. The 'About' section notes 'No description or website provided.' and lists the author 'hut23' with activity stats: 0 stars, 5 watching, and 51 forks. The 'Releases' section shows 2 tags. The 'Packages' section indicates 'No packages published'. The 'Contributors' section lists 7 contributors.

github.com/alan-turing-institute/github-example

alan-turing-institute / github-example

Type ⌘ to search

Code Issues Pull requests Actions Projects Security Insights

github-example Public

Watch 5 Fork 51 Star 0

main 2 branches 2 tags

Switch branches/tags

Filter branches/tags

Branches Tags

✓ main default

experiment

View all branches

3494fdd on May 30 9 commits

Add Whernside 2 months ago

Add wales 2 months ago

Add Helvellyn 2 months ago

include lakes in the scope 2 months ago

About

No description or website provided.

hut23

Activity

0 stars

5 watching

51 forks

Report repository

Releases

2 tags

Packages

No packages published

Contributors 7

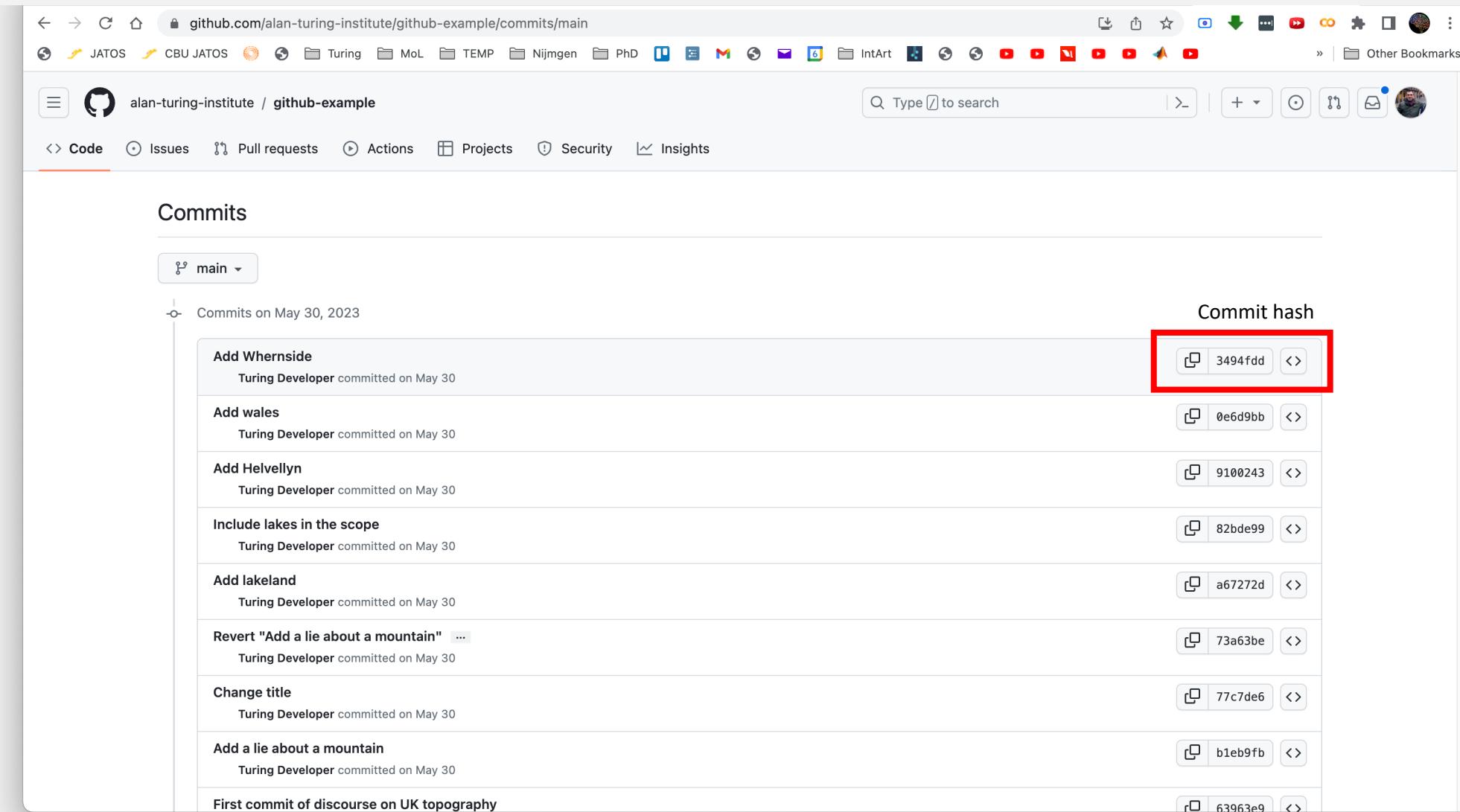
GitHub – a simple example repo

The screenshot shows a GitHub repository page for `alan-turing-institute/github-example`. The main navigation bar includes links for Code, Issues, Pull requests, Actions, Projects, Security, and Insights. The 'Code' tab is selected. Below the navigation, there's a search bar and a user profile icon.

The main content area is titled 'Commits' and shows the commit history for the 'main' branch. A dropdown menu indicates the branch is set to 'main'. The commits are listed in chronological order from May 30, 2023:

- Add Whernside** - Turing Developer committed on May 30. Commit hash: 3494fd
- Add wales** - Turing Developer committed on May 30. Commit hash: 0e6d9bb
- Add Helvellyn** - Turing Developer committed on May 30. Commit hash: 9100243
- Include lakes in the scope** - Turing Developer committed on May 30. Commit hash: 82bde99
- Add lakeland** - Turing Developer committed on May 30. Commit hash: a67272d
- Revert "Add a lie about a mountain"** - Turing Developer committed on May 30. Commit hash: 73a63be
- Change title** - Turing Developer committed on May 30. Commit hash: 77c7de6
- Add a lie about a mountain** - Turing Developer committed on May 30. Commit hash: b1eb9fb
- First commit of discourse on UK topography** - (no author information visible). Commit hash: 63963e9

GitHub – a simple example repo



The screenshot shows a GitHub repository page for 'alan-turing-institute / github-example'. The 'Code' tab is selected. The 'Commits' section displays a list of commits made on May 30, 2023. The first commit, 'First commit of discourse on UK topography', has its commit hash, '63963e9', highlighted with a red box. Other commits listed include 'Add Whernside', 'Add wales', 'Add Helvellyn', 'Include lakes in the scope', 'Add lakeland', 'Revert "Add a lie about a mountain"', 'Change title', 'Add a lie about a mountain', and 'First commit of discourse on UK topography'.

Commit hash
3494fd
0e6d9bb
9100243
82bde99
a67272d
73a63be
77c7de6
b1eb9fb
63963e9

Hands on part 1/2:

- Pairs of 2 people:
 - **Admin:** oversee the repo and review contributions.
 - **Collaborator:** address some specific issue.

Hands on part 1/2:

- Pairs of 2 people:
 - **Admin:** oversee the repo and review contributions.
 - **Collaborator:** address some specific issue.
- Topics covered:
 - Forking a repo
 - Creating an Issue
 - Creating a Project board to track issues.

Hands on part 1/2:

- Pairs of 2 people:
 - **Admin:** oversee the repo and review contributions.
 - **Collaborator:** address some specific issue.
- Topics covered:
 - Forking a repo
 - Creating an Issue
 - Creating a Project board to track issues.
- [Link to the step-by-step instructions document](#)
 - Complete sections: Fork, Commit, Collaborate, Kanban board, Issues.

Part 2: Git flow framework for collaboration

GitHub - a more complex example:

The screenshot shows a GitHub repository page for 'DeezyMatch'. The repository is public and has 526 commits. The 'About' section describes it as 'A Flexible Deep Learning Approach to Fuzzy String Matching'. It includes links to the project's website and various tags: nlp, machine-learning, natural-language-processing, deep-learning, hacktoberfest, hut23, and hut23-96. The 'Activity' section shows 106 stars, 4 watchers, and 29 forks. The 'Report repository' button is also present. The 'Releases' section lists 'DeezyMatch v1.3.4' as the latest release, updated on Sep 20, 2022.

Code Issues 26 Pull requests Actions Projects 1 Security Insights

DeezyMatch Public

master 7 branches 15 tags

Go to file Add file ▾ <> Code ▾

About

A Flexible Deep Learning Approach to Fuzzy String Matching

[living-with-machines.github.io/Deezy...](#)

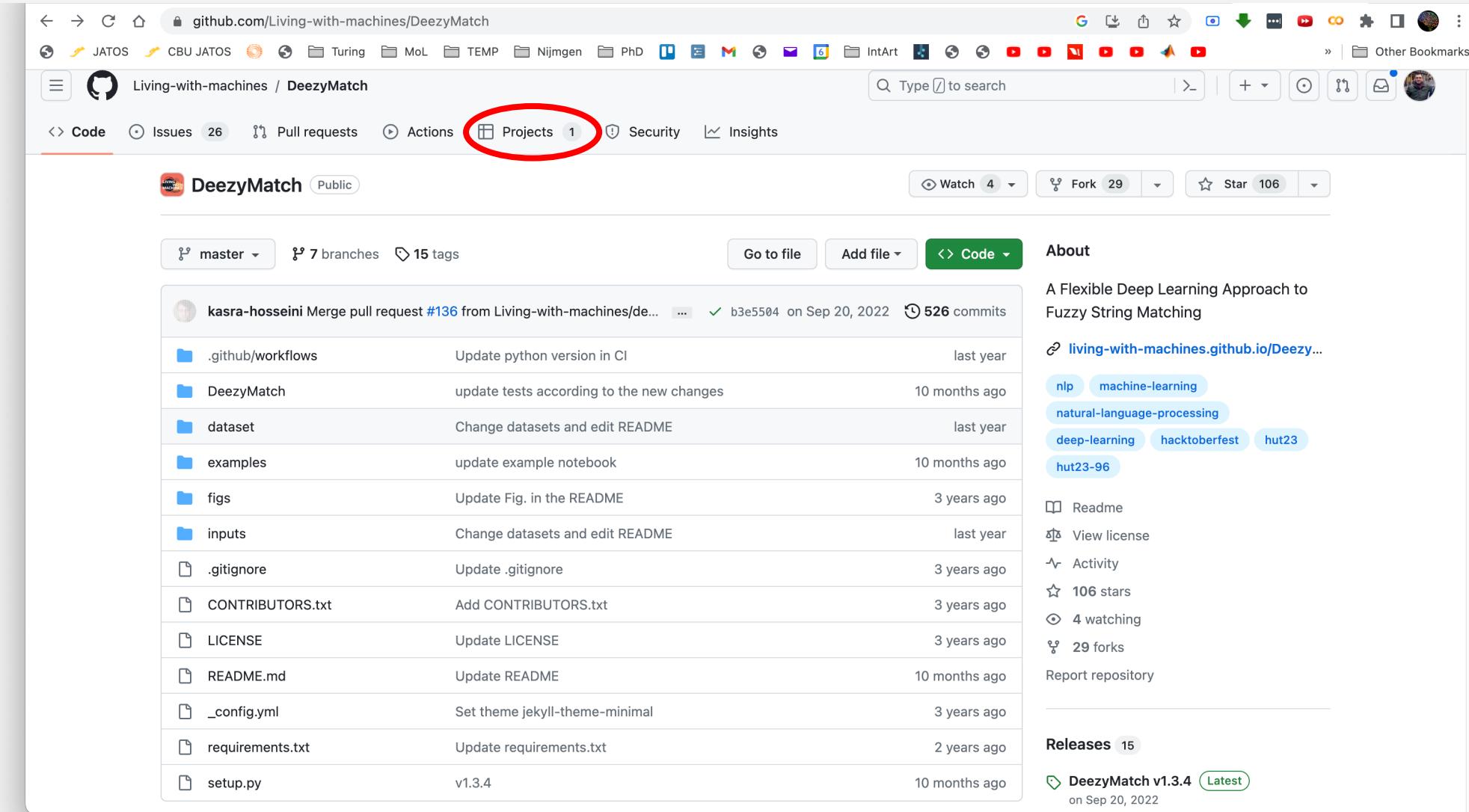
nlp machine-learning
natural-language-processing
deep-learning hacktoberfest hut23
hut23-96

Readme View license Activity 106 stars 4 watching 29 forks Report repository

Releases 15

DeezyMatch v1.3.4 Latest on Sep 20, 2022

GitHub - a more complex example:



The screenshot shows a GitHub repository page for 'DeezyMatch'. The top navigation bar includes links for Code, Issues (26), Pull requests, Actions, Projects (1, highlighted with a red circle), Security, and Insights. The main content area displays a list of commits from the 'master' branch, with 7 branches and 15 tags available. On the right side, there is an 'About' section with a description of the project as 'A Flexible Deep Learning Approach to Fuzzy String Matching', a link to the website 'living-with-machines.github.io/Deezy...', and various tags like nlp, machine-learning, natural-language-processing, deep-learning, hacktoberfest, hut23, and hut23-96. Below the 'About' section are links for Readme, View license, Activity, 106 stars, 4 watching, 29 forks, and Report repository. At the bottom, there is a 'Releases' section showing 'DeezyMatch v1.3.4' as the latest release, updated on Sep 20, 2022.

Code Issues 26 Pull requests Actions Projects 1 Security Insights

DeezyMatch Public

master 7 branches 15 tags Go to file Add file ▾

Commit	Message	Date
kasra-hosseini Merge pull request #136 from Living-with-machines/de...	b3e5504 on Sep 20, 2022	526 commits
.github/workflows	Update python version in CI	last year
DeezyMatch	update tests according to the new changes	10 months ago
dataset	Change datasets and edit README	last year
examples	update example notebook	10 months ago
figs	Update Fig. in the README	3 years ago
inputs	Change datasets and edit README	last year
.gitignore	Update .gitignore	3 years ago
CONTRIBUTORS.txt	Add CONTRIBUTORS.txt	3 years ago
LICENSE	Update LICENSE	3 years ago
README.md	Update README	10 months ago
_config.yml	Set theme jekyll-theme-minimal	3 years ago
requirements.txt	Update requirements.txt	2 years ago
setup.py	v1.3.4	10 months ago

About

A Flexible Deep Learning Approach to Fuzzy String Matching

[living-with-machines.github.io/Deezy...](#)

nlp machine-learning
natural-language-processing
deep-learning hacktoberfest hut23
hut23-96

Readme View license Activity 106 stars 4 watching 29 forks Report repository

Releases 15

DeezyMatch v1.3.4 Latest on Sep 20, 2022

GitHub – Kanban style project boards

The screenshot shows a GitHub project board for the 'DeezyMatch' repository. The board is organized into six columns:

- Backlog**: Contains 20 items, mostly tasks related to documentation and feature development.
- Upcoming**: Contains 2 items, both labeled as 'enhancement'.
- In progress**: Contains 3 items, including a tutorial and a feature request.
- Standing**: Contains 2 items, both labeled as 'enhancement'.
- For review**: Contains 1 item, labeled as 'enhancement'.
- Done**: Contains 18 items, mostly pull requests related to various bug fixes and feature enhancements.

Each card on the board includes a title, a brief description, the number of comments, and the user who added it. Labels like 'enhancement' and 'new feature' are visible on several cards.

GitHub – Kanban style project boards

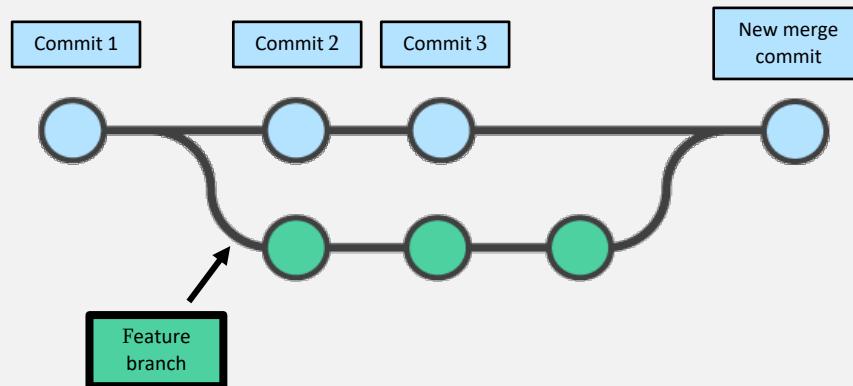
The screenshot shows a GitHub project board with the following columns and items:

- Backlog**:
 - Tasks we are considering working on soon, unordered
 - Scaling tests
 - [Tutorials] Issue with pytorch GPU
 - Improve documentation on generating train/valid/test datasets
 - Add option to extend the vocabulary when fine-tuning a model
 - Column 3 accepts (case-insensitive): [true, false, 0, 1], extend this to other cases: "Correct" "Wrong"
 - Installation on Windows
 - csv_sep is currently fixed to \t. Make this more general.
 - In addition to reading a tsv file, support list inputs for training and inference (create a pandas dataframe internally)
- Upcoming**:
 - Issues to be worked on next, in order (pull from the top)
 - Query/Candidate matching on-the-fly
- In progress**:
 - Maximum three items
 - Add OCR tutorial for DH2022
 - Add Heritage Gazetteer of Libya tutorial for DH2022
- Standing**:
 - Issues to be reviewed regularly at project meetings (or asynchronously)
 - Suggestions/comments on DeezyMatch during LinkedPasts
- For review**:
 - Can these issues be closed? Is there anything in the discussion that should be captured elsewhere
- Done**:
 - Allow disabling cosine similarity in candidate ranking
 - verbose flag. For each module, if verbose=False, we do not show any outputs.
 - Linting
 - Fix ranking metric documentation in candidateRanker
 - Fix the hardcoded multiplier in candidateRanker
 - Add post-processing filter to candidate ranking with maximum string length difference allowed
 - Ranking without prediction
 - Add specific datasets for the different DM functions and adapt test notebooks

A red box highlights the issue "Add Heritage Gazetteer of Libya tutorial for DH2022" in the In progress column. Red arrows point from left to right between the columns, indicating the flow of work.

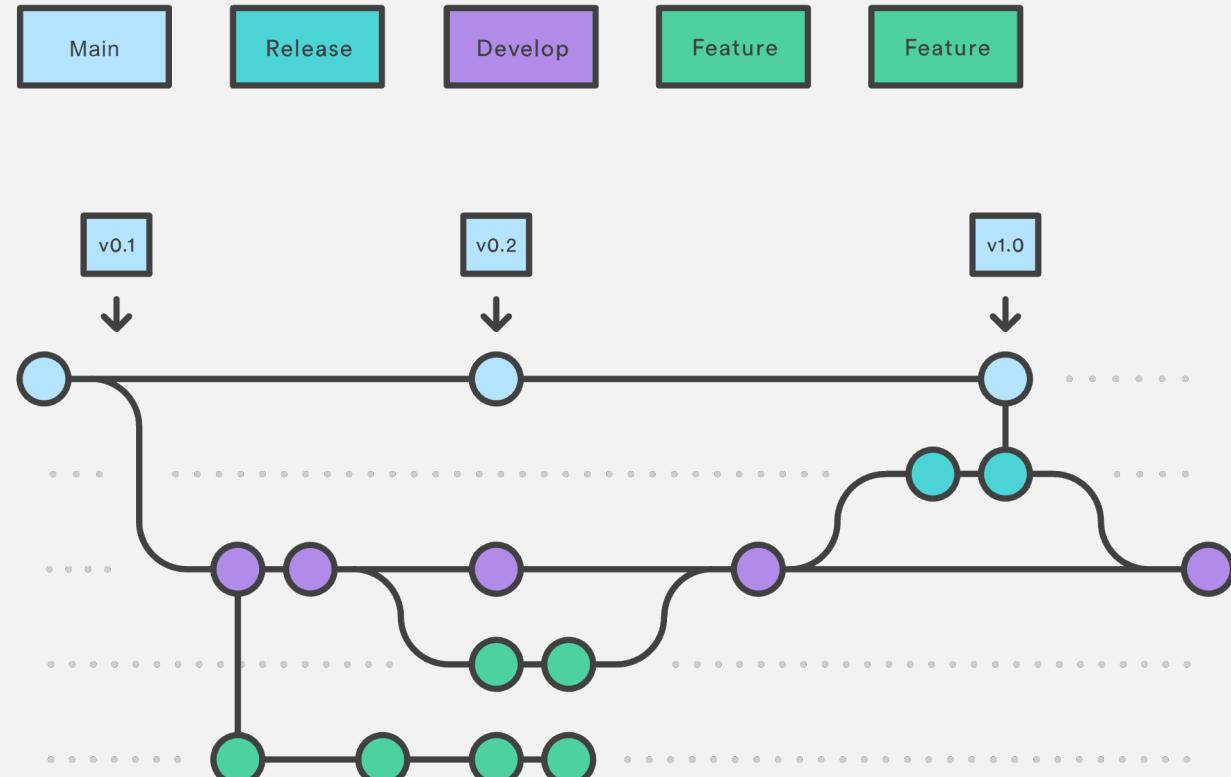
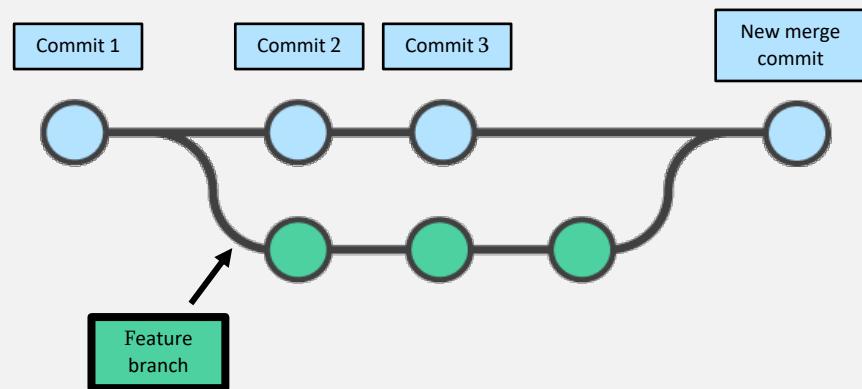
Git flow for collaboration

- A Git workflow



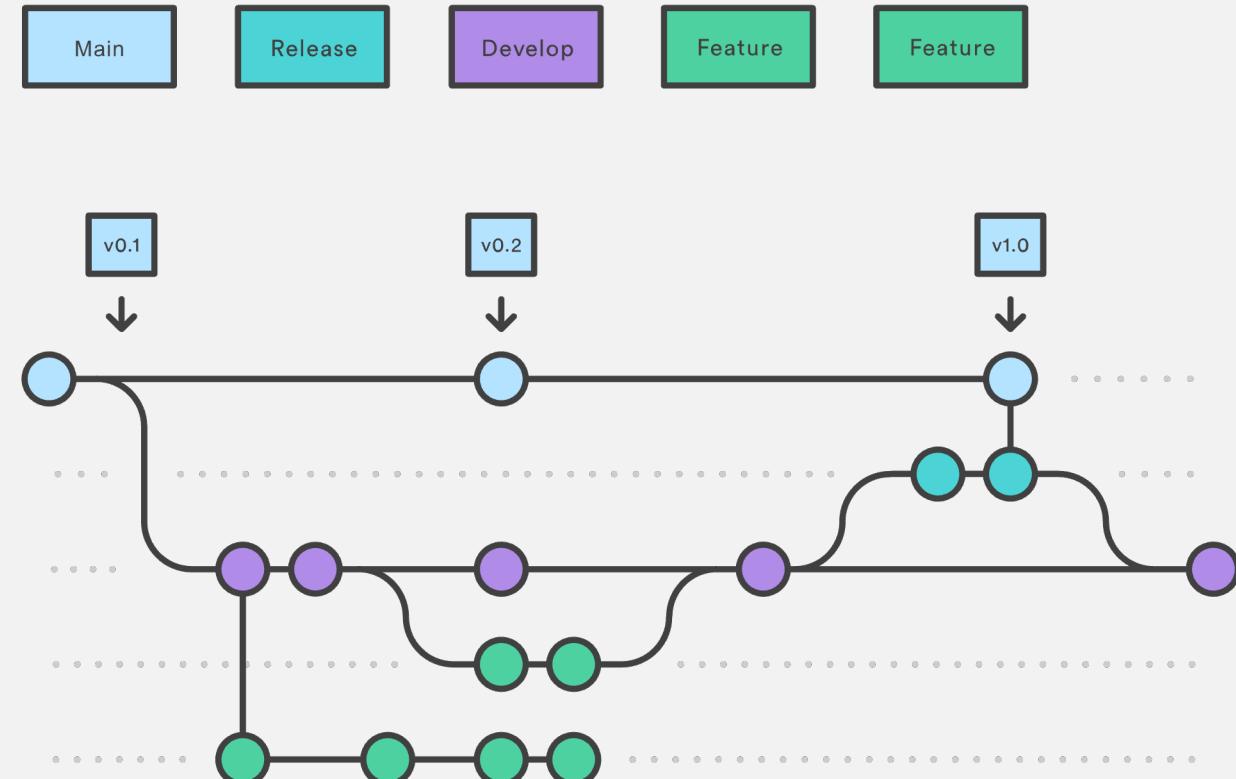
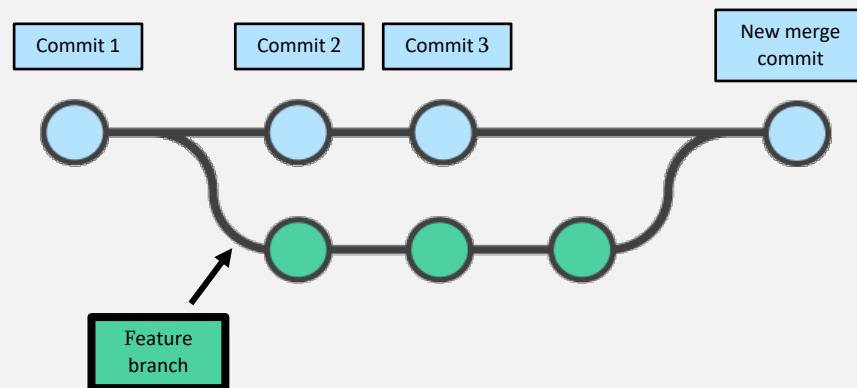
Git flow for collaboration

- A Git workflow



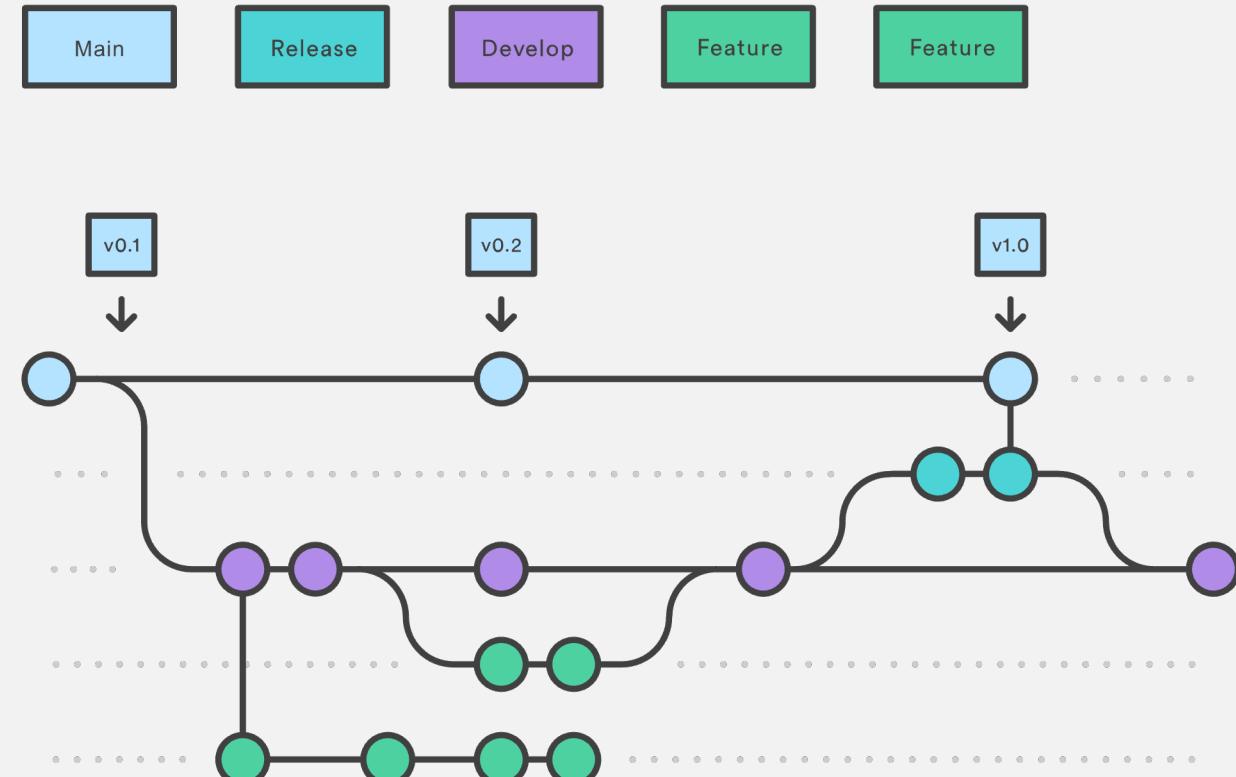
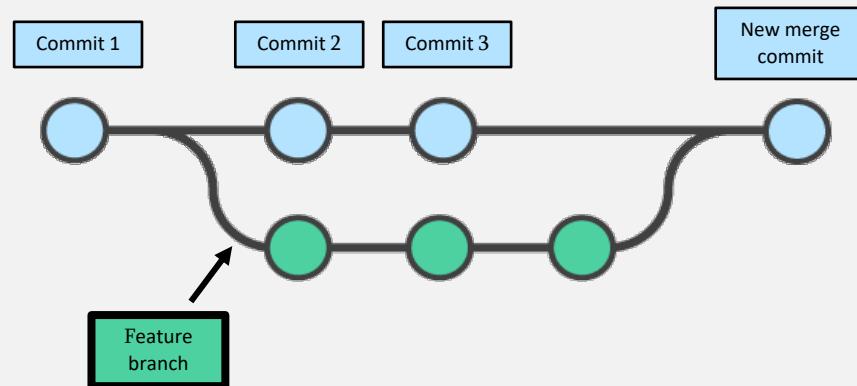
Git flow for collaboration

- A Git workflow
 - A branching model designed around the project release.



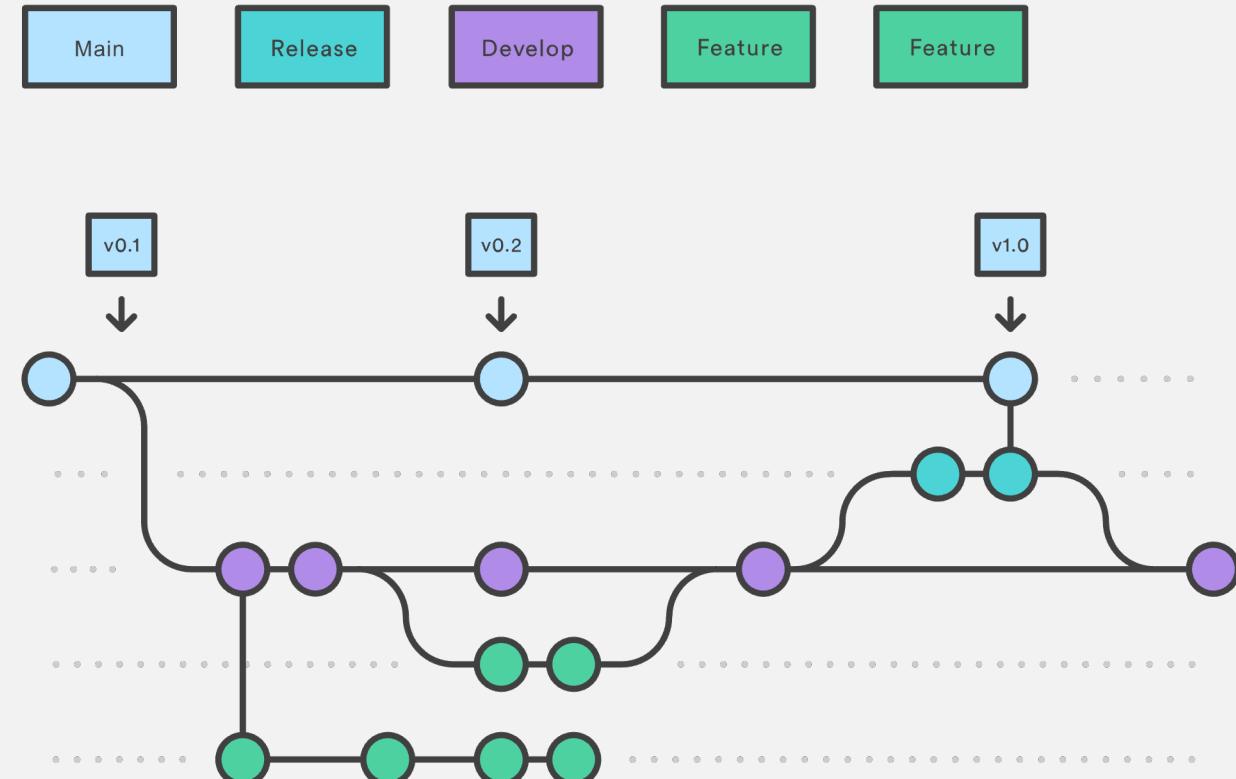
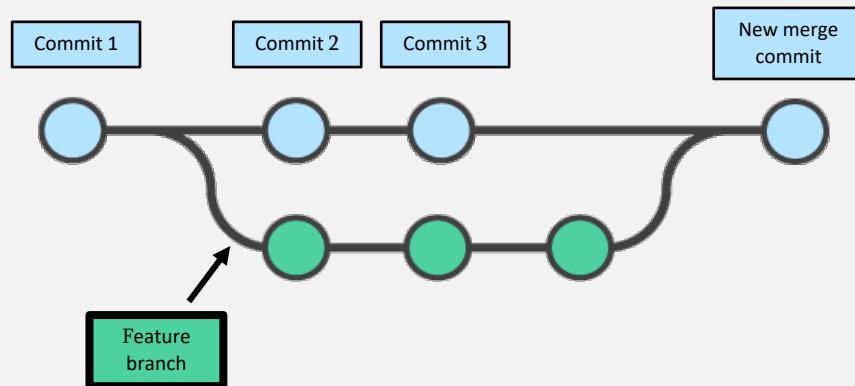
Git flow for collaboration

- A Git workflow
 - A branching model designed around the project release.
 - Specific roles to different branches



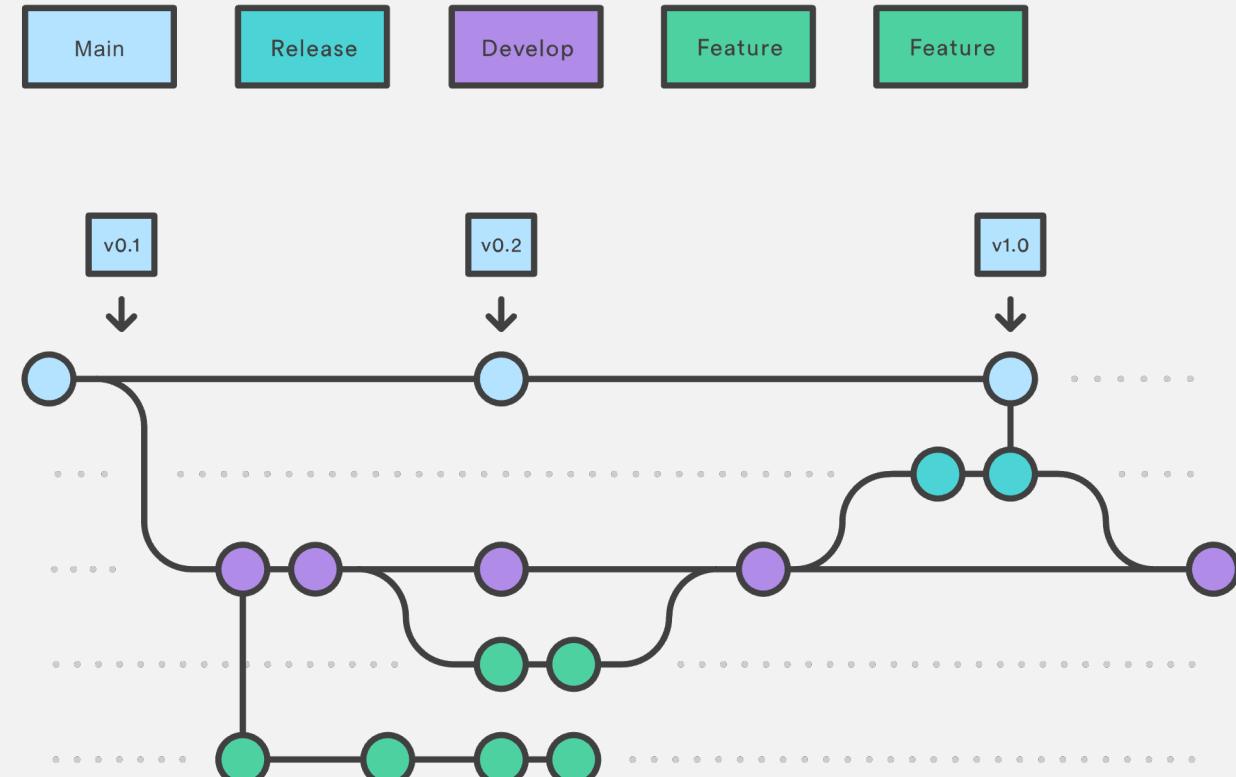
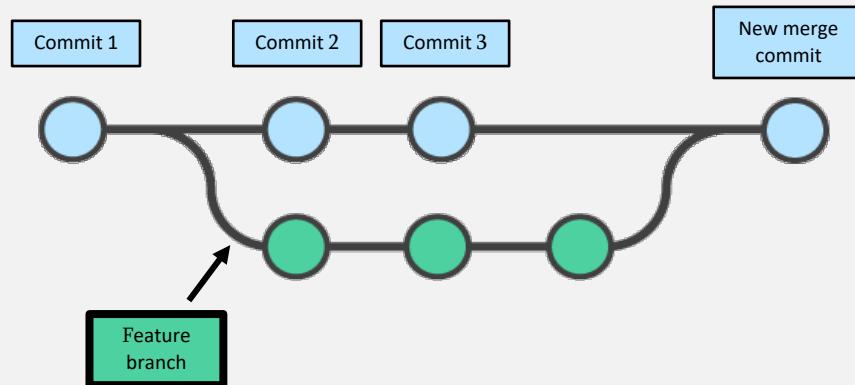
Git flow for collaboration

- A Git workflow
 - A branching model designed around the project release.
 - Specific roles to different branches
 - How and when they should interact



Git flow for collaboration

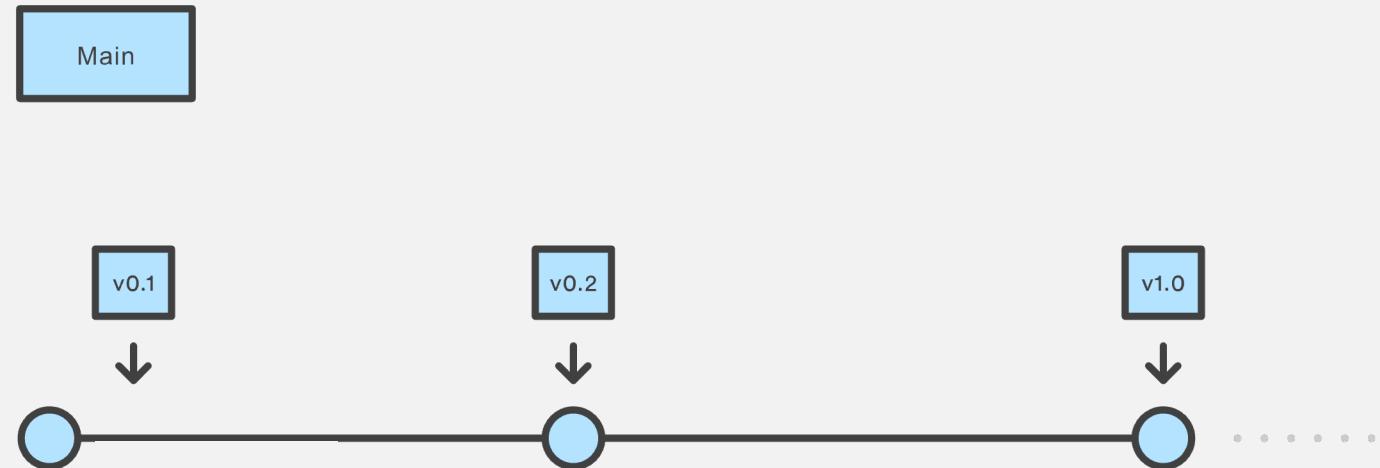
- A Git workflow
 - A branching model designed around the project release.
 - Specific roles to different branches
 - How and when they should interact
- Not the only workflow
 - e.g. GitHub workflow for continuous delivery



Git flow for collaboration

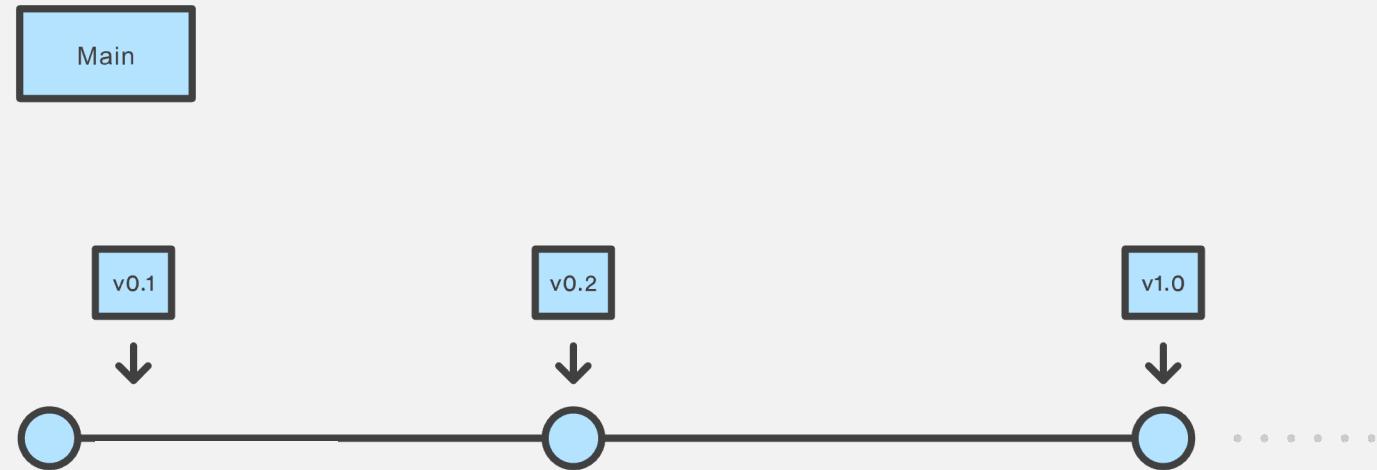
- Two branches to record the history of the project (with an infinite lifetime)

1. "Main" branch: release history.



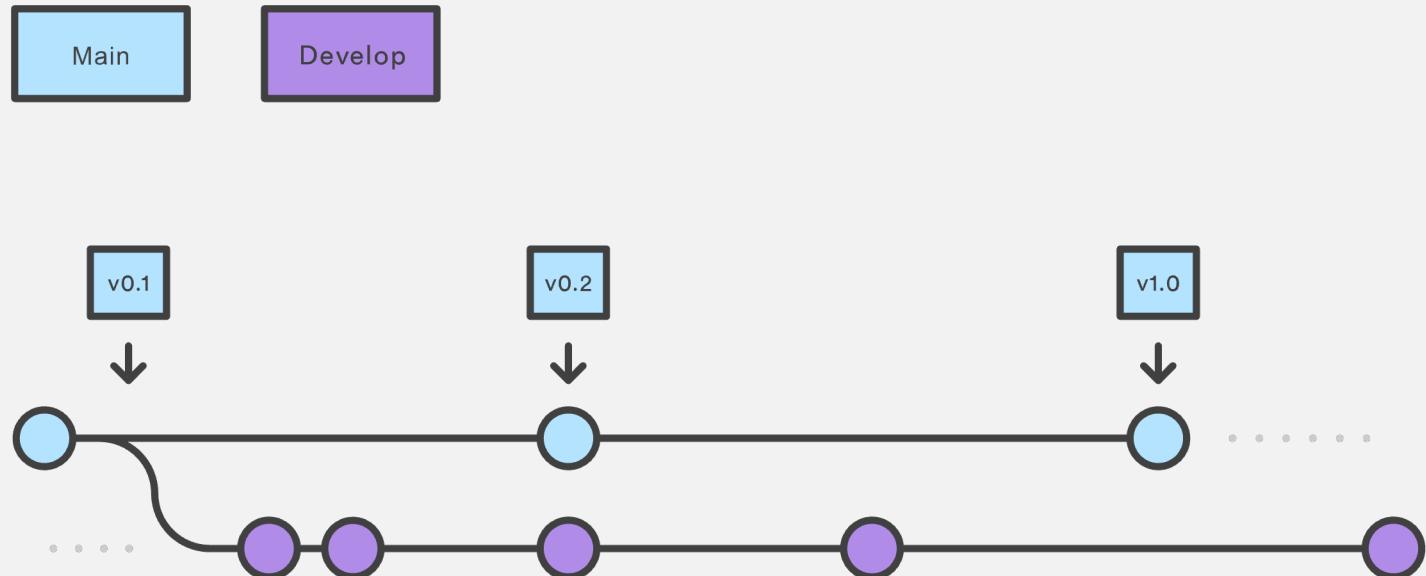
Git flow for collaboration

- Two branches to record the history of the project (with an infinite lifetime)
 1. "Main" branch: release history.
 2. "Develop" branch: integration branch for features.



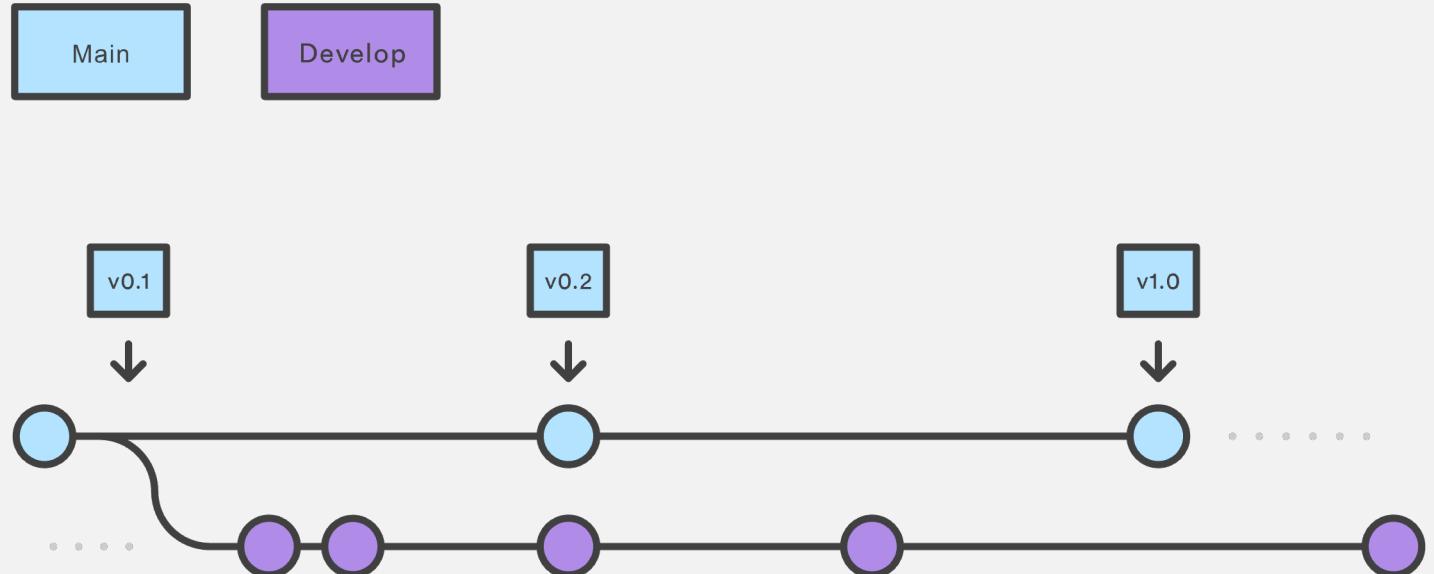
Git flow for collaboration

- Two branches to record the history of the project (with an infinite lifetime)
 1. "Main" branch: release history.
 2. "Develop" branch: integration branch for features.



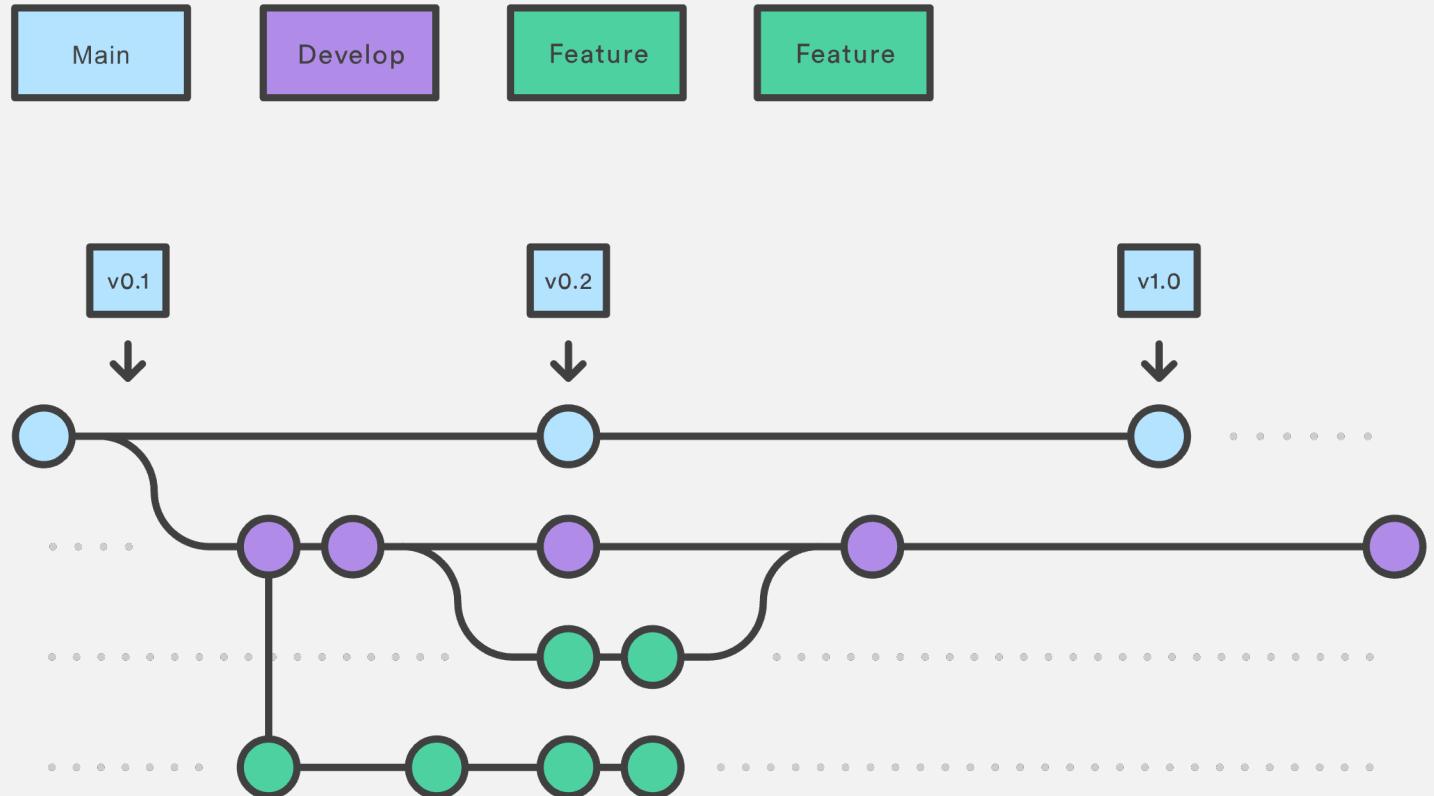
Git flow for collaboration

- Two branches to record the history of the project (with an infinite lifetime)
 1. "Main" branch: release history.
 2. "Develop" branch: integration branch for features.
- Each new feature has its own branch
 - Branch off Develop.
 - Merge back into Develop.

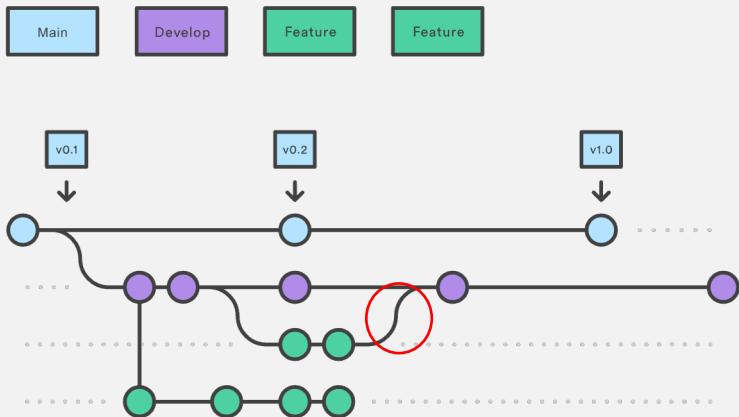


Git flow for collaboration

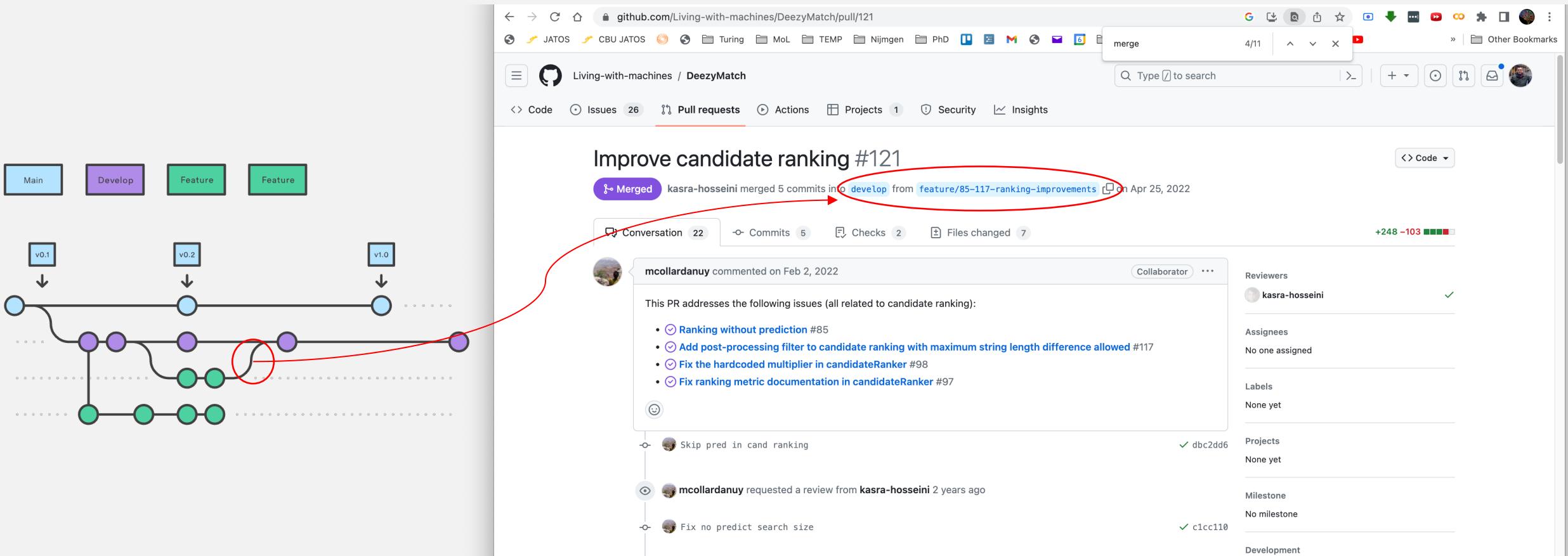
- Two branches to record the history of the project (with an infinite lifetime)
 1. "Main" branch: release history.
 2. "Develop" branch: integration branch for features.
- Each new feature has its own branch
 - Branch off Develop.
 - Merge back into Develop.



Git flow for collaboration



Git flow for collaboration



Hands-on part 2/2:

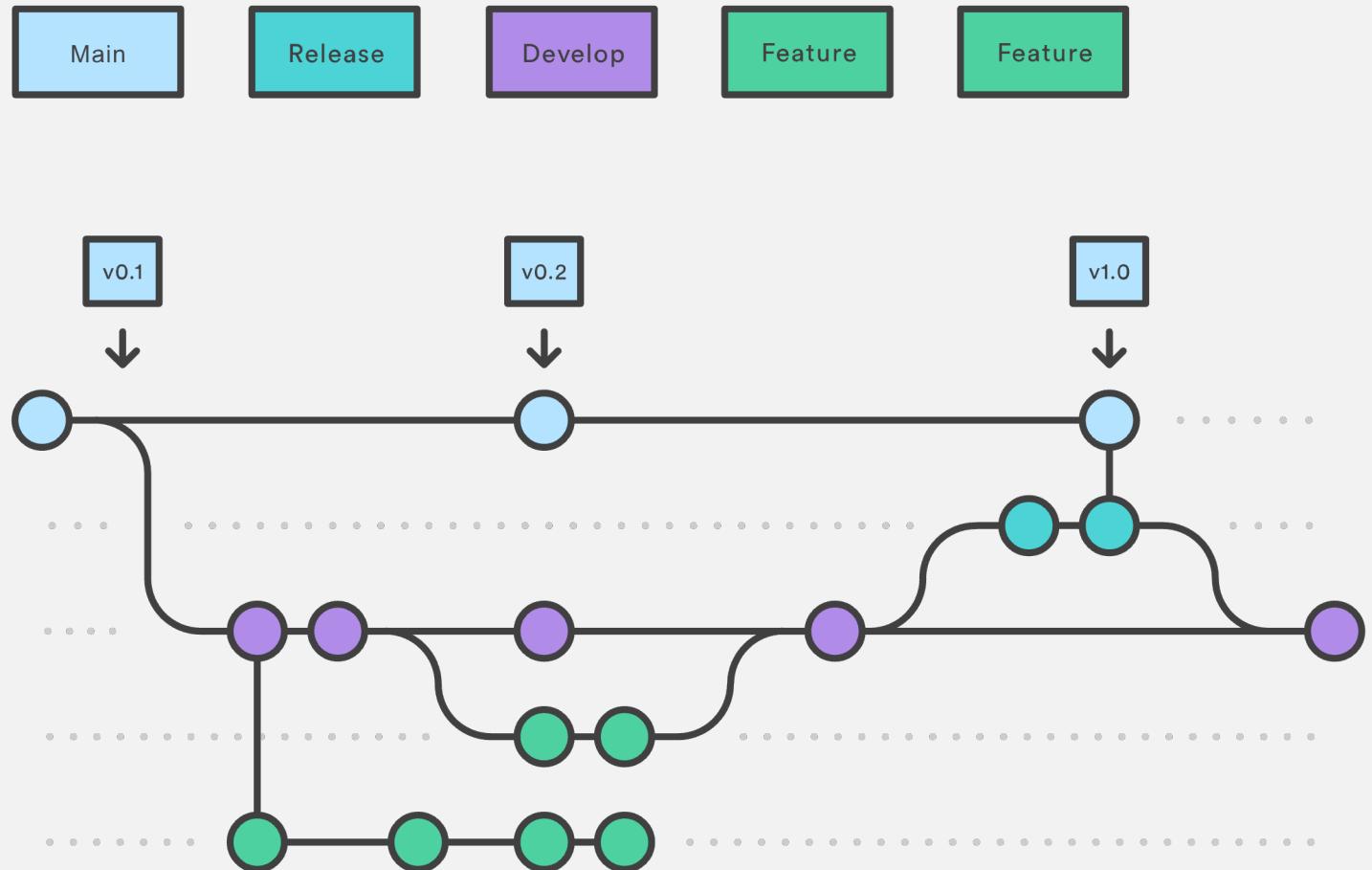
- Topics covered:
 - Creating branches
 - Creating Pull Requests
 - Reviewing Pull Requests
 - Managing merge conflicts

Hands-on part 2/2:

- Topics covered:
 - Creating branches
 - Creating Pull Requests
 - Reviewing Pull Requests
 - Managing merge conflicts
- [Link to the step-by-step instructions document](#)
 - Complete sections: Branches, Pull Requests, Review Pull Requests, Merging.

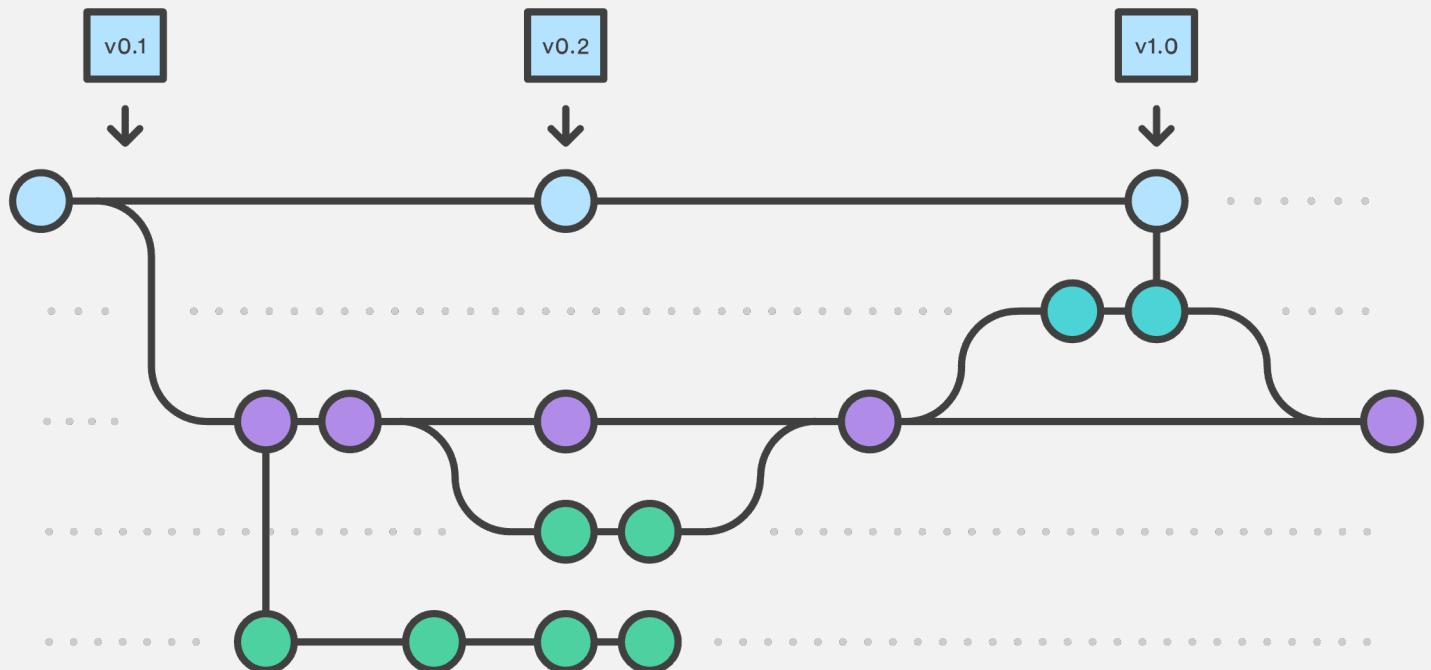
Git flow continued:
Release and Hotfix branches

Git flow: Release branches



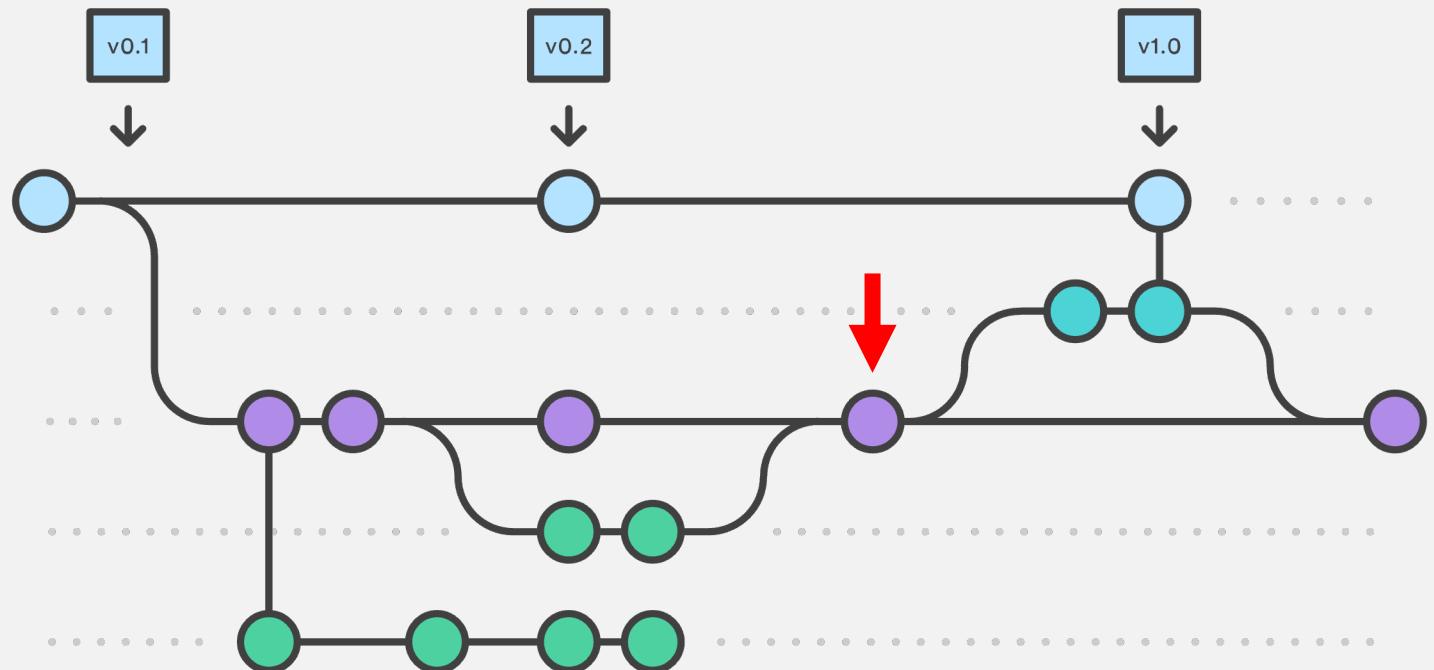
Git flow: Release branches

- When ready for a new release,
no new features.



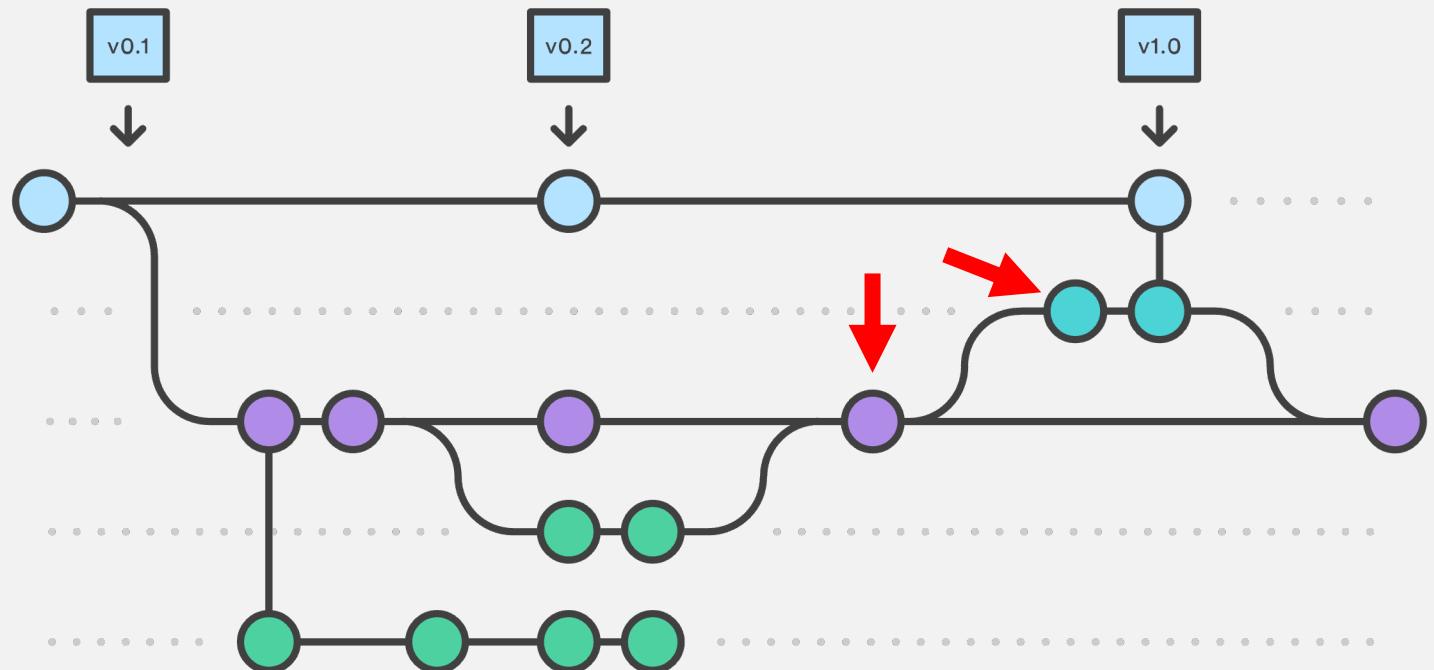
Git flow: Release branches

- When ready for a new release,
no new features.



Git flow: Release branches

- When ready for a new release,
no new features.
- Create a “release” branch.



Git flow: Release branches

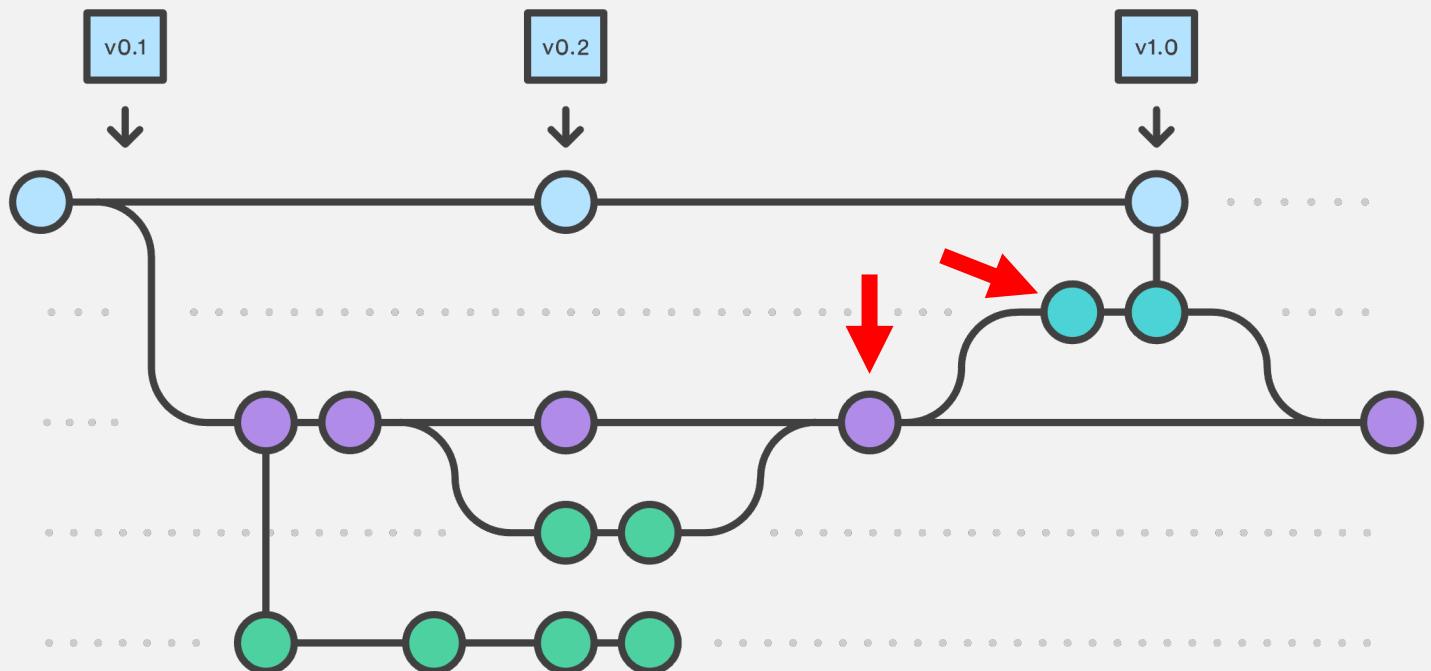
- When ready for a new release, no new features.



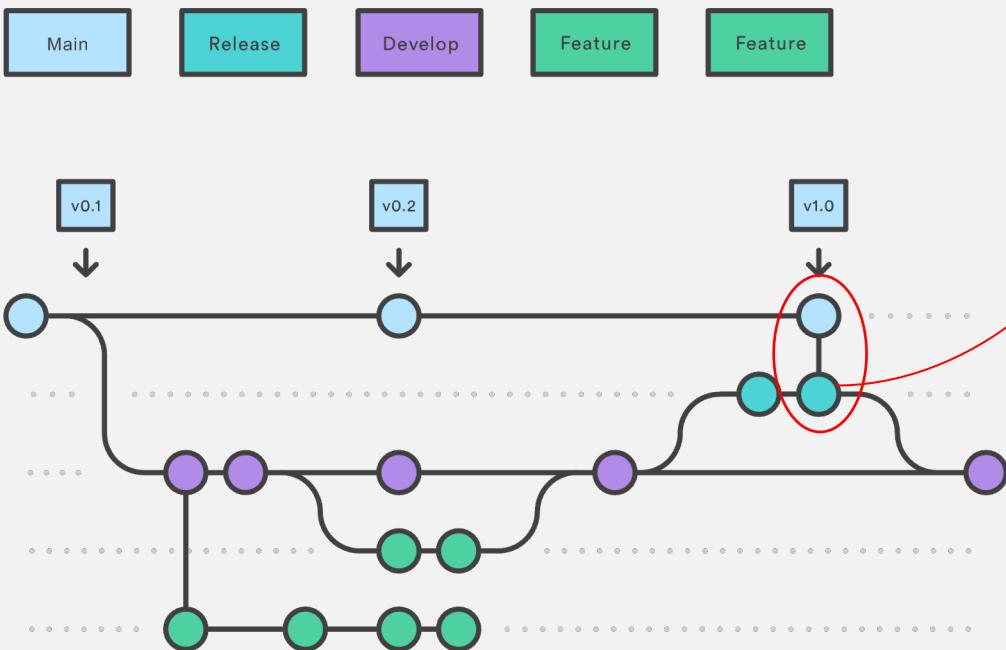
- Create a "release" branch.

Only:

- Documentation
- Minor bug fixes
- Meta-data for release
- etc.



Git flow: Release branches

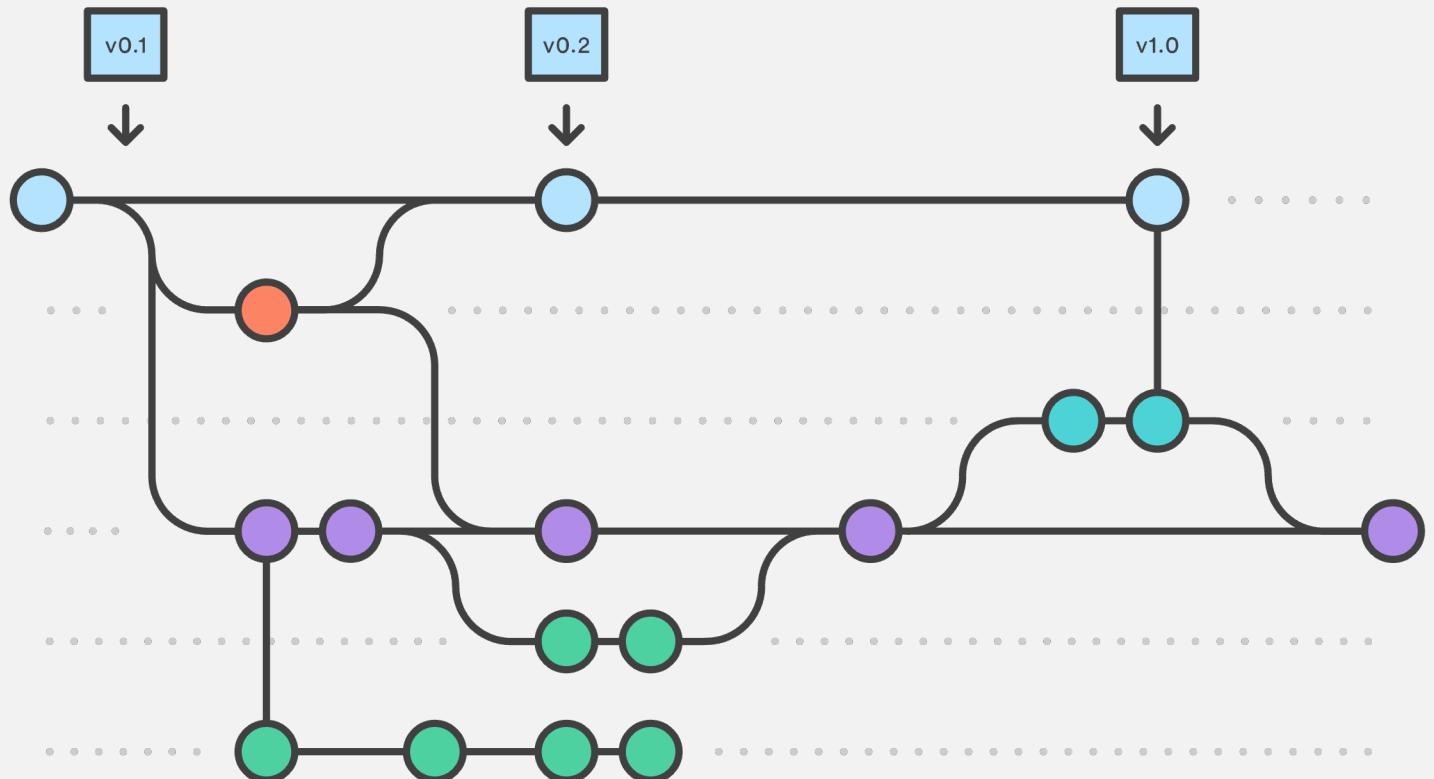


Review, discuss...

A screenshot of a GitHub pull request interface. The title bar says 'version 1.2.0 #75'. A red arrow points from the 'Review, discuss...' text in the diagram to the 'Conversation' tab in the GitHub interface, which shows 0 comments. Another red arrow points from the 'Merge' status in the GitHub interface back to the 'v1.0' release point in the diagram. The GitHub interface displays the merge commit 'kasra-hosseini merged 11 commits into master from develop on Sep 15, 2020'. The pull request details show 11 commits from 'kasra-hosseini' added 3 years ago, including changes to README files and input files. The pull request has been merged and is now closed. The GitHub interface also shows the repository's stats: +122 -58, 2 participants, and 2 reviews.

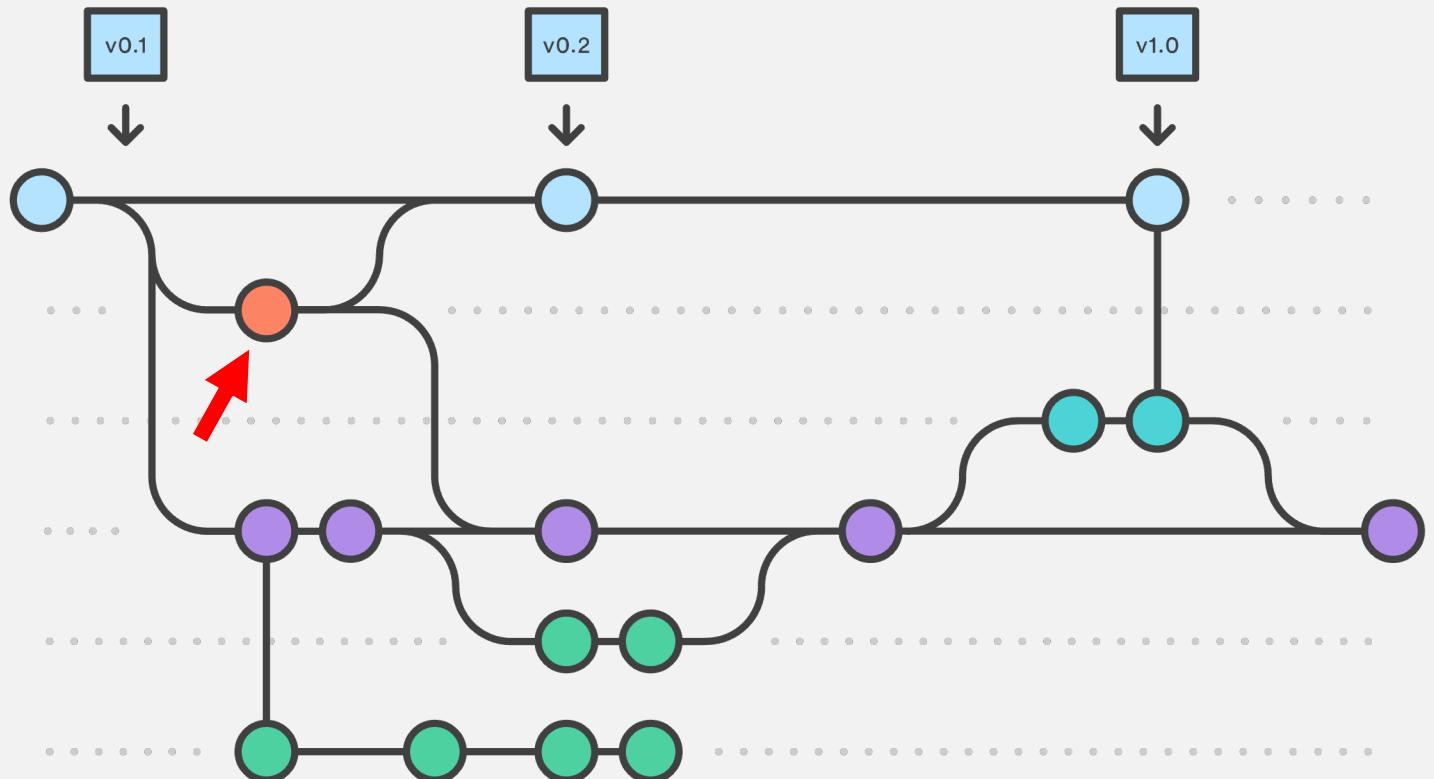
Git flow: Hotfix branches

- Maintenance or Hotfix branches to patch releases



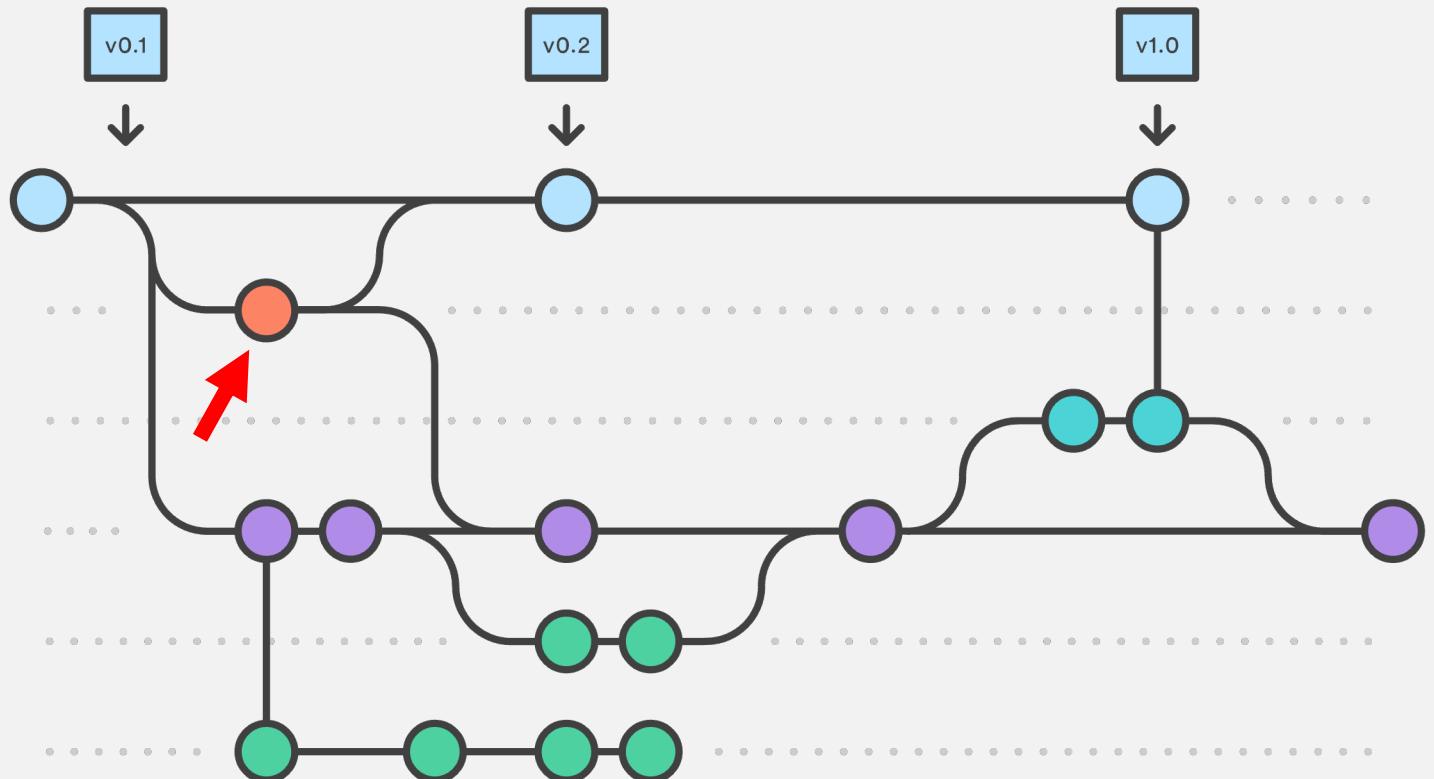
Git flow: Hotfix branches

- Maintenance or Hotfix branches to patch releases



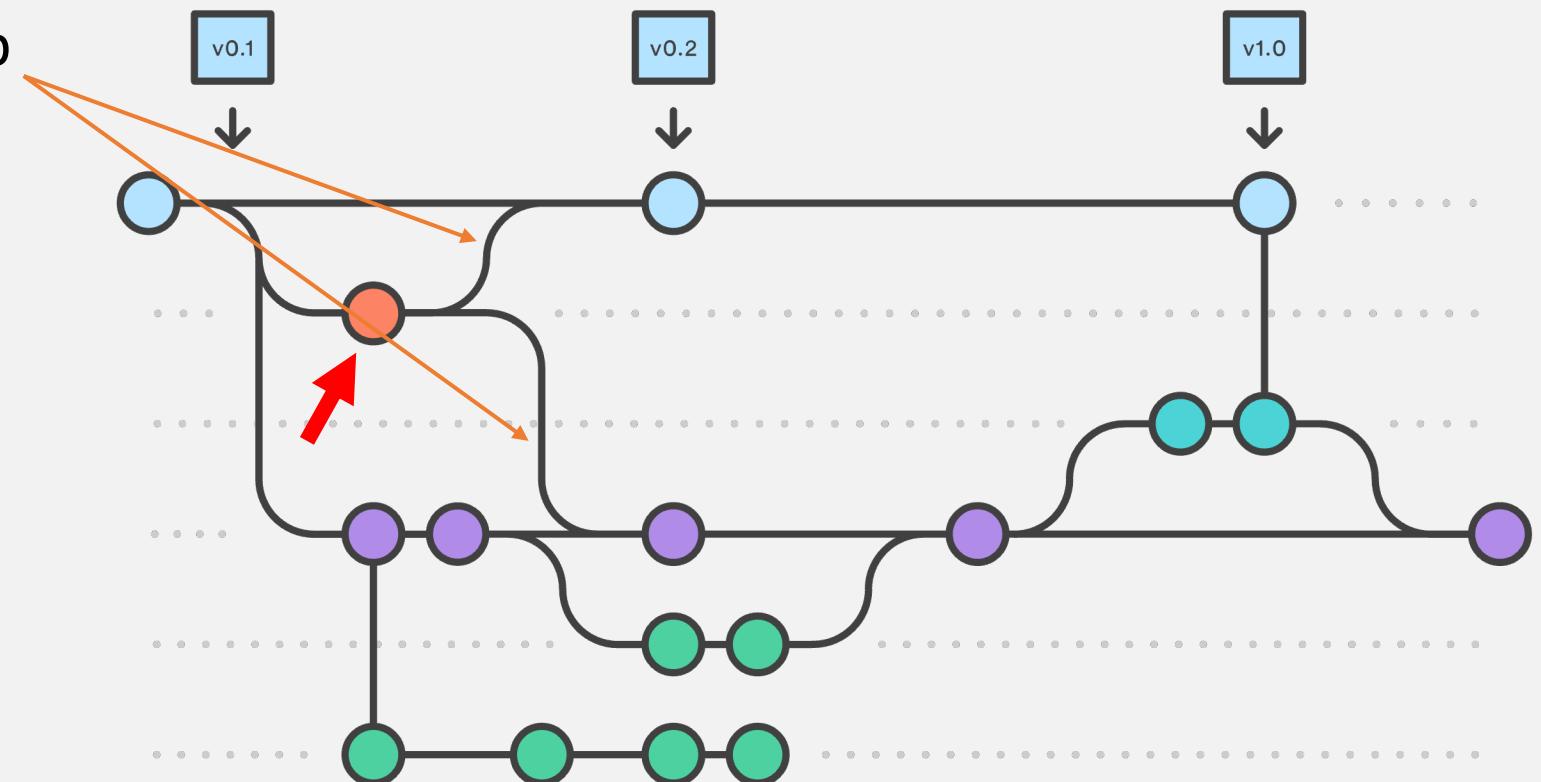
Git flow: Hotfix branches

- Maintenance or Hotfix branches to patch releases
- They are based on main



Git flow: Hotfix branches

- Maintenance or Hotfix branches to patch releases
- They are based on main
- Merge into Main and Develop (or release, if exists)



Git flow: versioning the release of your software

The image displays a Mac desktop with two browser windows open. Both windows have identical toolbars at the top, including back, forward, search, and download icons.

Left Window (GitHub Repository):

- Address bar: `github.com/Living-w...`
- Repository name: `Living-with-machines / DeezyMatch`
- Branch: `master`
- Code navigation buttons: `Go to file`, `Add file`, `<> Code`
- Project stats: `Watch 4`, `Fork 29`, `Star 106`
- Section: `About`
 - Description: A Flexible Deep Learning Approach to Fuzzy String Matching
 - Link: `living-with-machines.github.io/...`
 - Tags: `nlp`, `machine-learning`, `natural-language-processing`, `deep-learning`, `hacktoberfest`, `hut23`, `hut23-96`
 - Files: `Readme`, `View license`
 - Activity: `106 stars`, `4 watching`, `29 forks`
 - Actions: `Report repository`
- Section: `Releases`
 - Release: `DeezyMatch v1.3.4` (Latest)
on Sep 20, 2022
 - More releases: `+ 14 releases`

Right Window (Project Landing Page):

- Address bar: `living-with-machines.github.io/DeezyMatch`
- Title: **DeezyMatch**
- Description: A Flexible Deep Learning Approach to Fuzzy String Matching
- Text: View the Project on GitHub `Living-with-machines/DeezyMatch`
- Logo: A graphic featuring three stylized, colorful swirls (blue, purple, red) above the word "Deezy" and "Match".
- Section: **A Flexible Deep Neural Network Approach to Fuzzy String Matching**
- Links: `pypi v1.3.4` (circled in red), `License MIT`, `launch binder`, `Integration Tests passing`
- Text: DeezyMatch can be used in the following tasks:
 - Fuzzy string matching
 - Candidate ranking/selection
 - Query expansion
 - Toponym matching
- Text: Or as a component in tasks requiring fuzzy string matching and candidate ranking, such as:
 - Record linkage

Thank you!