

The Alan Turing Institute

Collaborative Coding

Federico Nanni
Kaspar Beelen
Ed Chalstrey
Kalle Westerling



Slides adapted from: <https://drive.google.com/file/d/1n8vQ9sZ3M6QDxiE24rYE7PXviTKBxOWD/view>

Presentation Overview

- Slides and hands-on exercises

Presentation Overview

- Slides and hands-on exercises

Slido (anonymous):

- www.slido.com
- #4869941



Presentation Overview

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

Slido (anonymous):

- www.slido.com
- #4869941



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
- #4869941



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
- #4869941



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
- #4869941



Part 1: Git and GitHub

Git Version Control – the basics

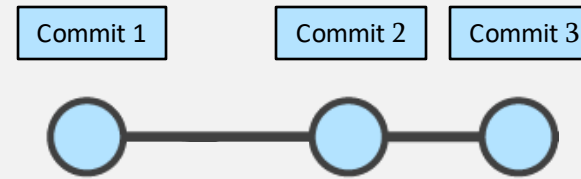
- **Version control**, or source control, involves tracking and managing changes to code.

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).

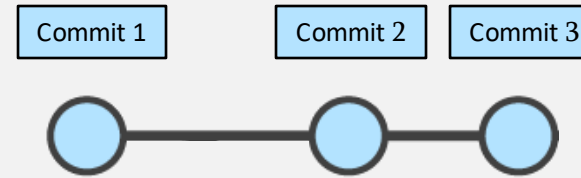
Git Version Control – the basics

- **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).



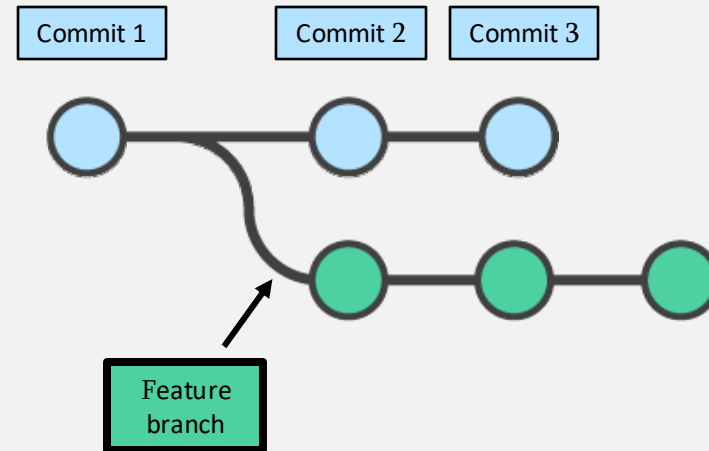
Git Version Control – the basics

- **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.



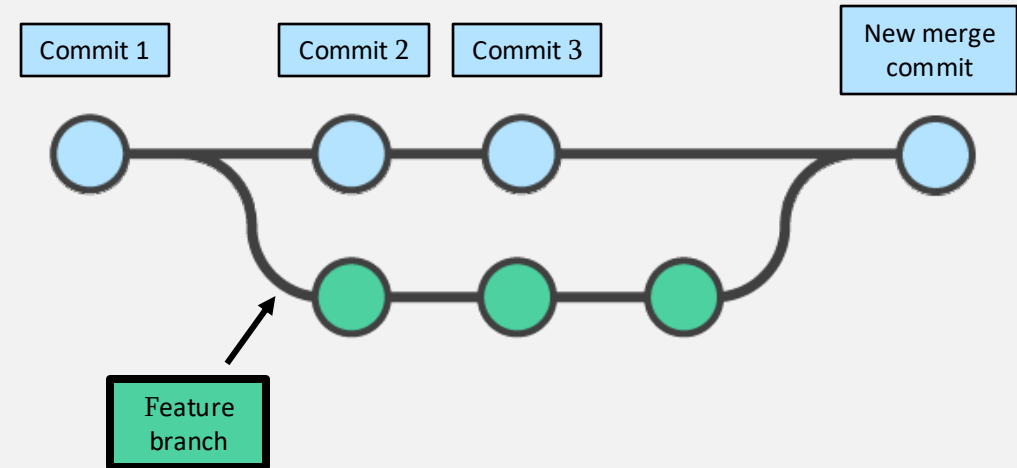
Git Version Control – the basics

- **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.



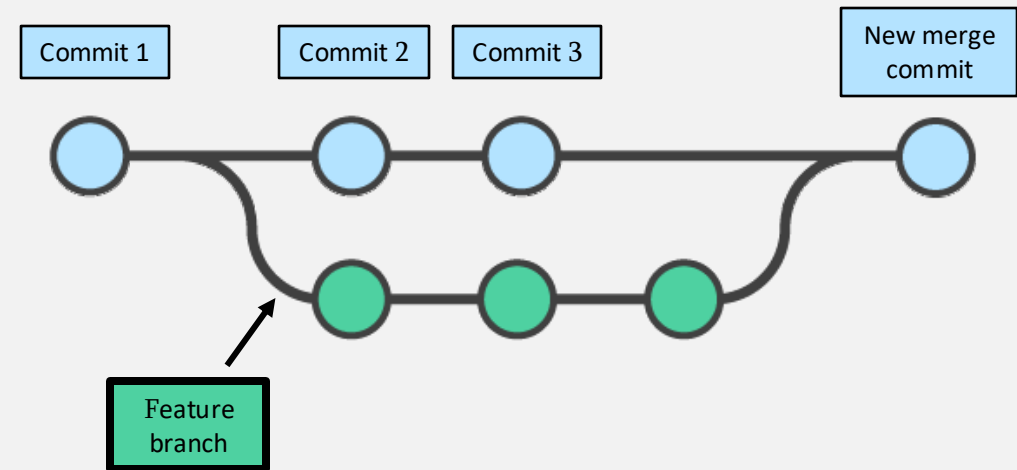
Git Version Control – the basics

- **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 web browser displaying the GitHub repository page for 'github-example' by 'alan-turing-institute'. The browser's address bar shows the URL 'github.com/alan-turing-institute/github-example'. The repository is public and has 51 forks, 5 watchers, and 0 stars. The main branch is 'main', and there are 2 branches and 2 tags. The repository contains a file named 'Pennines.md' with the commit message 'Add Whernside' and a commit hash of '3494fdd' on May 30, 2023. Other files listed are 'Wales.md', 'lakeland.md', and 'test.md'. The right sidebar shows the repository's activity, including 0 stars, 5 watchers, and 51 forks. The 'About' section states 'No description or website provided.' The 'Releases' section shows 2 tags. The 'Packages' section shows 'No packages published.' The 'Contributors' section shows 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

Go to file Add file <> Code

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 the GitHub interface for the repository 'github-example' by 'alan-turing-institute'. The repository is public and has 51 forks and 0 stars. A modal window titled 'Switch branches/tags' is open, showing a list of branches: 'main' (selected, default) and 'experiment'. The modal also includes a search bar for 'Filter branches/tags' and a link to 'View all branches'. The background shows the repository's commit history and right-hand sidebar with sections for 'About', 'Releases', 'Packages', and '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

Go to file Add file <> Code

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

Switch branches/tags

Filter branches/tags

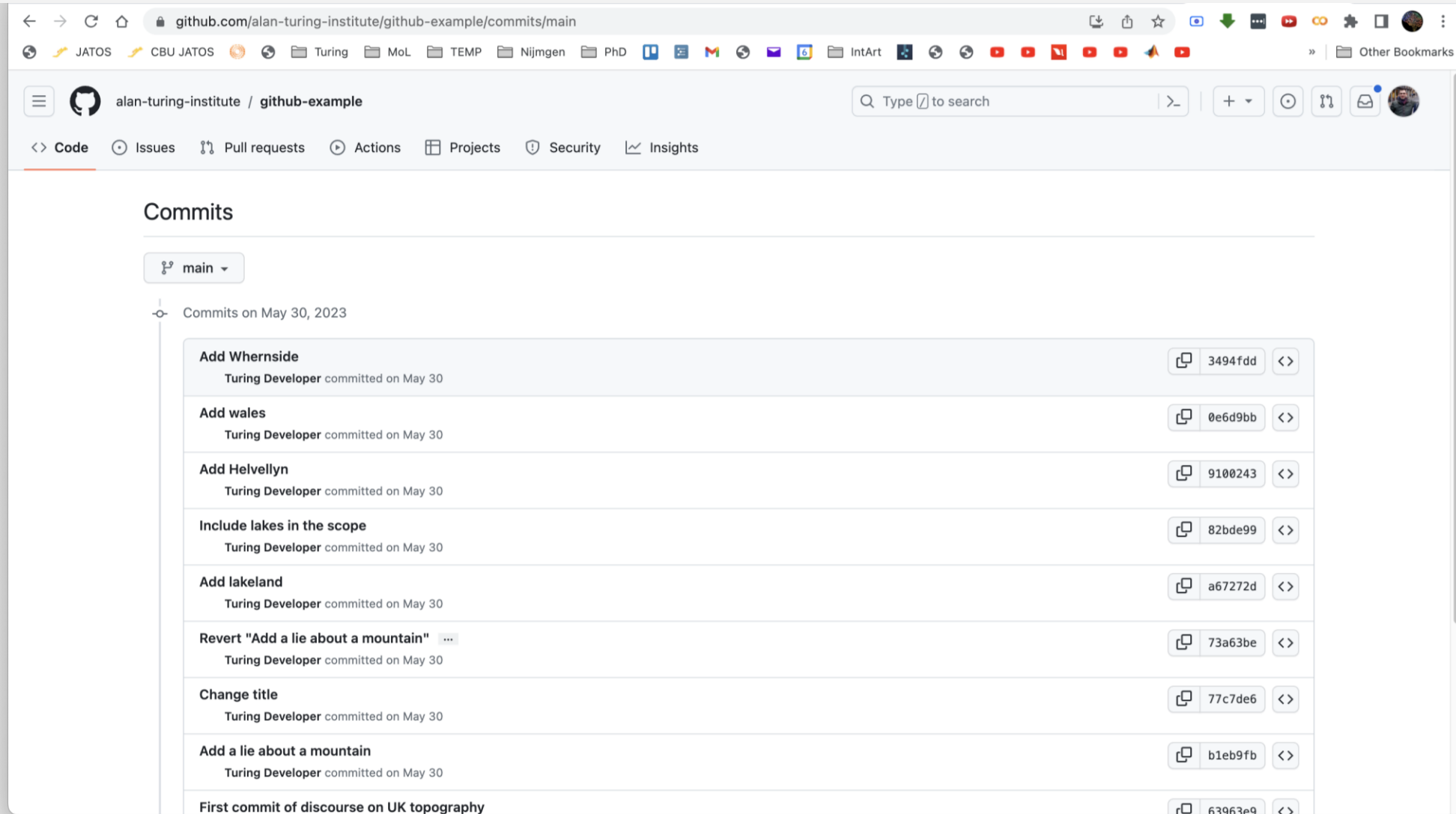
Branches Tags

✓ main default

experiment

View all branches

GitHub – a simple example repo



The screenshot shows a web browser displaying the GitHub repository page for 'alan-turing-institute / github-example'. The URL in the address bar is 'github.com/alan-turing-institute/github-example/commits/main'. The page header includes the repository name, a search bar, and navigation links for Code, Issues, Pull requests, Actions, Projects, Security, and Insights. The main content area is titled 'Commits' and shows a list of commits on the 'main' branch, filtered for 'Commits on May 30, 2023'. Each commit entry includes the commit message, the author 'Turing Developer', the commit date 'committed on May 30', a copy icon, the commit hash, and a code icon.

Commit Message	Author	Date	Hash
Add Whernside	Turing Developer	committed on May 30	3494fdd
Add wales	Turing Developer	committed on May 30	0e6d9bb
Add Helvellyn	Turing Developer	committed on May 30	9100243
Include lakes in the scope	Turing Developer	committed on May 30	82bde99
Add lakeland	Turing Developer	committed on May 30	a67272d
Revert "Add a lie about a mountain"	Turing Developer	committed on May 30	73a63be
Change title	Turing Developer	committed on May 30	77c7de6
Add a lie about a mountain	Turing Developer	committed on May 30	b1eb9fb
First commit of discourse on UK topography			63963e9

GitHub – a simple example repo

The screenshot shows the GitHub interface for the repository 'alan-turing-institute / github-example'. The 'Commits' tab is selected, and the 'main' branch is chosen. A filter for 'Commits on May 30, 2023' is applied. A list of commits is displayed, with the first commit, 'Add Whernside', highlighted. The commit hash '3494fdd' is highlighted with a red box, and a label 'Commit hash' points to it. The commit message is 'Add Whernside' and it was committed by 'Turing Developer' on May 30.

Commit Message	Author	Commit Hash
Add Whernside	Turing Developer	3494fdd
Add wales	Turing Developer	0e6d9bb
Add Helvellyn	Turing Developer	9100243
Include lakes in the scope	Turing Developer	82bde99
Add lakeland	Turing Developer	a67272d
Revert "Add a lie about a mountain"	Turing Developer	73a63be
Change title	Turing Developer	77c7de6
Add a lie about a mountain	Turing Developer	b1eb9fb
First commit of discourse on UK topography	Turing Developer	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 the GitHub repository page for **Living-with-machines / MapReader**. The repository is public and has 72 stars, 8 forks, and 6 watchers. The main branch is **main**, with 9 branches and 22 tags. The repository description is "A computer vision pipeline for exploring and analyzing images at scale". The repository includes a README, license, code of conduct, and a citation link. The repository also has a list of files and folders, including .github, assets/css, conda, docs, figs, mapreader, paper, tests, worked_examples, .all-contributorsrc, and _all_contributors_rc.

Repository: **MapReader** (Public)

Stats: 72 Stars, 8 Forks, 6 Watchers

Branches: 9, Tags: 22

Search: Go to file

Buttons: Edit Pins, Watch, Fork, Star, Code

About: A computer vision pipeline for exploring and analyzing images at scale

Topics: machine-learning, computer-vision, deep-learning, article, maps, pytorch, digital-humanities, spatial-data, hut23, hut23-96

Readme, View license, Code of conduct, Cite this repository, Activity, Custom properties, 72 stars, 6 watching, 8 forks

File/Folder	Description	Time
.github	Update publish-to-test-pypi.yml	3 weeks ago
assets/css	Add style	2 years ago
conda	v. minor tidy-up of conda specification	last year
docs	update test instructions	3 weeks ago
figs	Add files via upload	last year
mapreader	Merge pull request #434 from Living-with-machines/426-...	2 weeks ago
paper	Add data/culture grant no. to paper	last week
tests	update annotator to fix warnings	28 days ago
worked_examples	update y-labels in metric plots	3 weeks ago
.all-contributorsrc	Update .all-contributorsrc	5 months ago
_all_contributors_rc	Create _all_contributors_rc	5 months ago

GitHub - a more complex example:

github.com/Living-with-machines/MapReader

WhatsApp Notifications Project Board Hut23 Reginald LWM Read the Docs HackMD Utils absentia_2024_hols Home Page - Math... Admin

Living-with-machines / MapReader

Type to search

<> Code Issues 84 Pull requests 5 Discussions Actions **Projects 1** Wiki Security Insights Settings

MapReader Public

Edit Pins Watch 6 Fork 8 Star 72

main 9 Branches 22 Tags Go to file Add file <> Code

rwood-97 Add data/culture grant no. to paper ✓ cc73913 · last week 1,453 Commits

.github	Update publish-to-test-pypi.yml	3 weeks ago
assets/css	Add style	2 years ago
conda	v. minor tidy-up of conda specification	last year
docs	update test instructions	3 weeks ago
figs	Add files via upload	last year
mapreader	Merge pull request #434 from Living-with-machines/426-...	2 weeks ago
paper	Add data/culture grant no. to paper	last week
tests	update annotator to fix warnings	28 days ago
worked_examples	update y-labels in metric plots	3 weeks ago
.all-contributorsrc	Update .all-contributorsrc	5 months ago
_all_contributors_rc	Create _all_contributors_rc	5 months ago

About

A computer vision pipeline for exploring and analyzing images at scale

mapreader.readthedocs.io/en/latest/

machine-learning computer-vision deep-learning article maps pytorch digital-humanities spatial-data hut23 hut23-96

Readme View license Code of conduct Cite this repository Activity Custom properties 72 stars 6 watching 8 forks

GitHub – Kanban style project boards

The screenshot displays the 'MapReader Project Board' on GitHub, organized into five columns representing different stages of the project workflow:

- Backlog (34):** Tasks before end of the year 2024, ORDERED. Includes items like MapReader #295 (Submit to Reviews in DH), MapReader #344 (Consider adding further filtering rules in context-based post processing), MapReader #234 (Commandline tool to (re)create worked_examples dir), MapReader #135 (synthesize workshop comments + make new discussion topics), MapReader #245 (Deployment to test.pypi.org fails in some cases), and MapReader #162 (Fix code for workflow).
- Upcoming (28):** Issues to be worked on within 2 weeks, in order (pull from the top). Includes items like MapReader #407 (show metadata), MapReader #360 (outline MapReader LwM book chapter), MapReader #367 (Create change log in docs), MapReader #390 (Work out method for training/evaluating models), and MapReader #294 (Add documentation for reuse of mapreader).
- In progress (now) (5):** Includes items like MapReader #358 (plan sprint for integrating text detection & recognition), MapReader #417 (Update workshop materials based on feedback from April), MapReader #404 (Add duplicate detection to text spotting code), MapReader #405 (Add method for dealing with cut-off at patch edges in text spotting work), and MapReader #393 (Create a sample/examples folder with some highlighted maps/patches).
- Done (review) (6):** Done but needs review in a meeting. Includes items like MapReader #406 (Update annotated_images.show_sample() example), MapReader #408 (Broken URL in the Workshop_AprilMay2024_part2.ipynb notebook), MapReader #409 (Invalid syntax error on the calculate_add_metrics cell), MapReader #411 (Add the device argument for GPU support on the ClassifierContainer()), and MapReader #420 (review notebooks and provide suggestions for how to frame "your turn" sections).
- Done (177):** Includes items like MapReader #64 (commandline script fails), MapReader #401 (Annotation tool broken in python 3.12), MapReader #386 (fix documentation and directory for worked examples), MapReader #385 (Update tests - fix warnings and add docs for testing), MapReader #389 (Fix typos in docs), MapReader #394 (Make notebook for April 2024 workshop), and data-culture-newspapers #36 (Fill out AHRC data survey).

Each item card includes a title, description, and relevant labels (e.g., community, LwM-outputs, enhancement, new feature, bug, documentation, workshop, workshop2024, paper, RAM). The board also features a filter bar at the top and a '+ Add item' button at the bottom of each column.

GitHub – Kanban style project boards

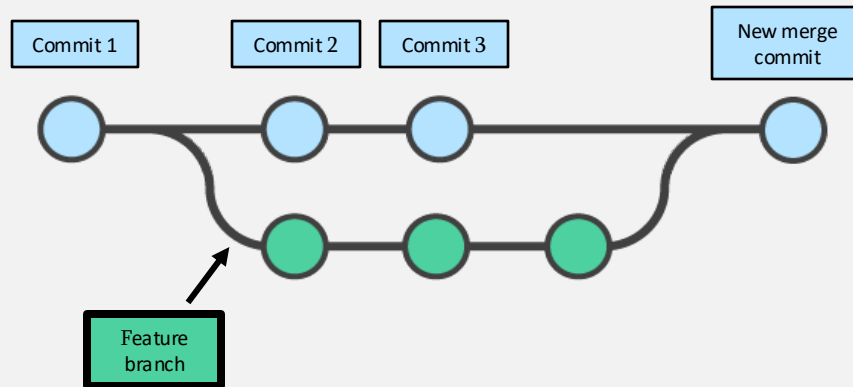
The screenshot displays the 'MapReader Project Board' on GitHub, organized in a Kanban style. The board is divided into five columns, each representing a stage in the project workflow. Red arrows indicate the flow from left to right between the columns.

- Backlog (34):** Tasks before end of the year 2024, ORDERED.
 - MapReader #295: Submit to Reviews in DH (community, LwM-outputs)
 - MapReader #344: Consider adding further filtering rules in context-based post processing (enhancement, new feature, post-processing)
 - MapReader #234: Commandline tool to (re)create worked_examples dir somewhere user accessible (enhancement, new feature)
 - MapReader #135: synthesize workshop comments + make new discussion topics (project management)
 - MapReader #245: Deployment to test.pypi.org fails in some cases (bug)
 - MapReader #162: Fix code for workflow
- Upcoming (28):** Issues to be worked on within 2 weeks, in order (pull from the top).
 - Draft: multimodal tasks with text on maps
 - MapReader #407: show metadata (e.g pixel values, geometry) in the annotator view (enhancement, new feature)
 - MapReader #360: outline MapReader LwM book chapter (paper)
 - MapReader #367: Create change log in docs (documentation)
 - MapReader #390: Work out method for training/evaluating models using gold standard recogito and/or MapKurator outputs
 - MapReader #294: Add documentation for reuse of mapreader (documentation)
 - MapReader #292
- In progress (now) (5):**
 - MapReader #358: plan sprint for integrating text detection & recognition (new feature, text on maps)
 - MapReader #417: Update workshop materials based on feedback from April (community, documentation, RAM, workshop, workshop2024)
 - MapReader #404: Add duplicate detection to text spotting code (enhancement, new feature)
 - MapReader #405: Add method for dealing with cut-off at patch edges in text spotting work (enhancement, new feature)
 - MapReader #393: Create a sample/examples folder with some highlighted maps/patches which are good for workshops (workshop, workshop2024)
- Done (review) (6):** Done but needs review in a meeting.
 - MapReader #406: Update annotated_images.show_sample() example (documentation)
 - MapReader #408: Broken URL in the Workshop_AprilMay2024_part2.ipynb notebook (bug)
 - MapReader #409: Invalid syntax error on the calculate_add_metrics cell (documentation)
 - MapReader #411: Add the device argument for GPU support on the ClassifierContainer() (documentation)
 - MapReader #420: review notebooks and provide suggestions for how to frame "your turn" sections (workshop, workshop2024)
 - MapReader #410: Displaying scores per class
- Done (177):**
 - MapReader #64: commandline script fails (bug)
 - MapReader #401: Annotation tool broken in python 3.12 (bug)
 - MapReader #386: fix documentation and directory for worked examples (paper)
 - MapReader #385: Update tests - fix warnings and add docs for testing
 - MapReader #389: Fix typos in docs (documentation)
 - MapReader #394: Make notebook for April 2024 workshop
 - data-culture-newspapers #36: Fill out AHRC data survey

At the top right, there are buttons for 'Add status update', 'Discard', and 'Save'. At the bottom of each column, there is a '+ Add item' button.

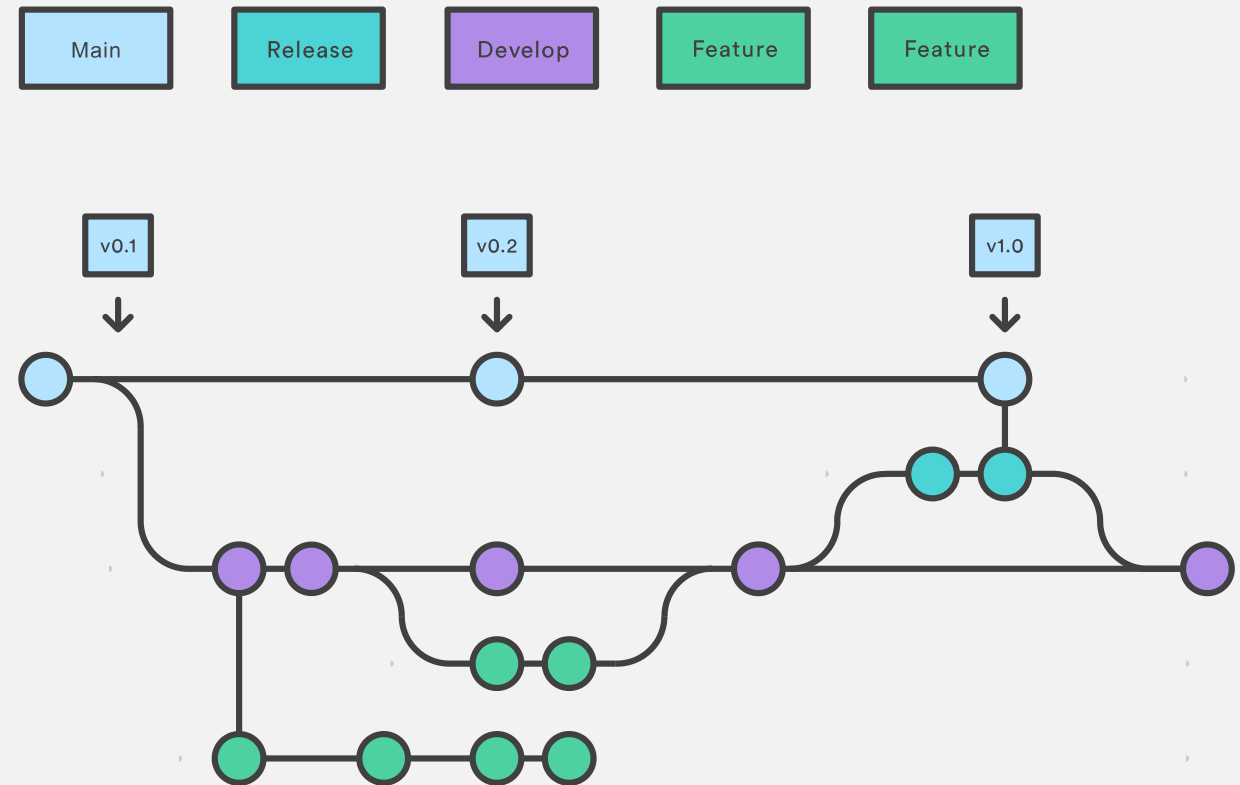
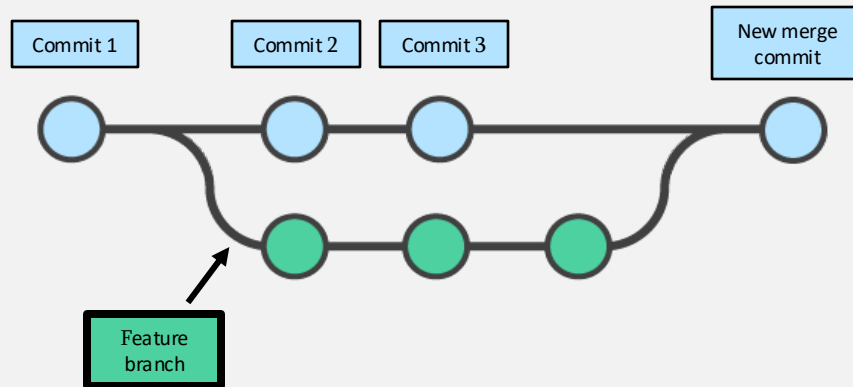
Git flow for collaboration

- A Git workflow



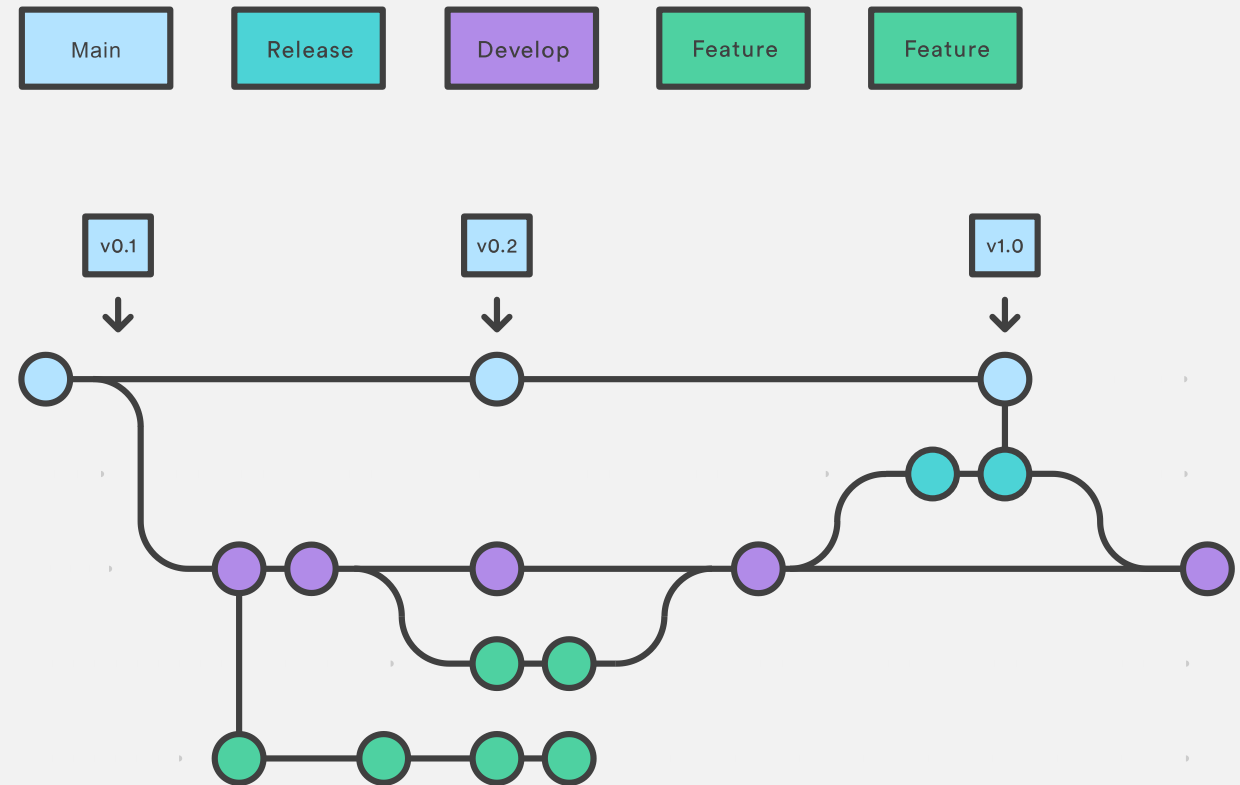
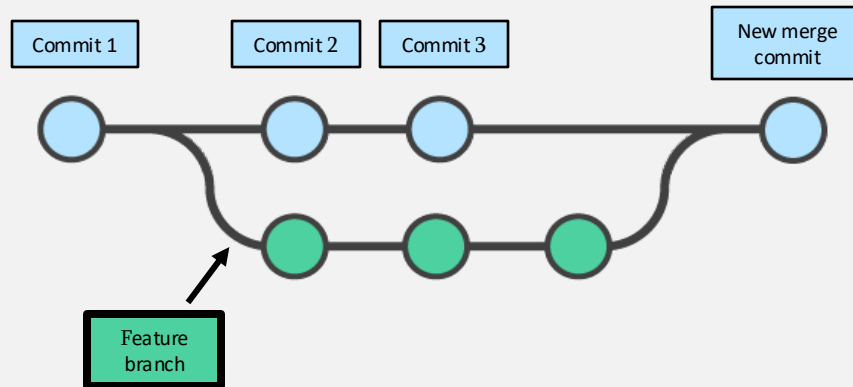
Git flow for collaboration

- A Git workflow



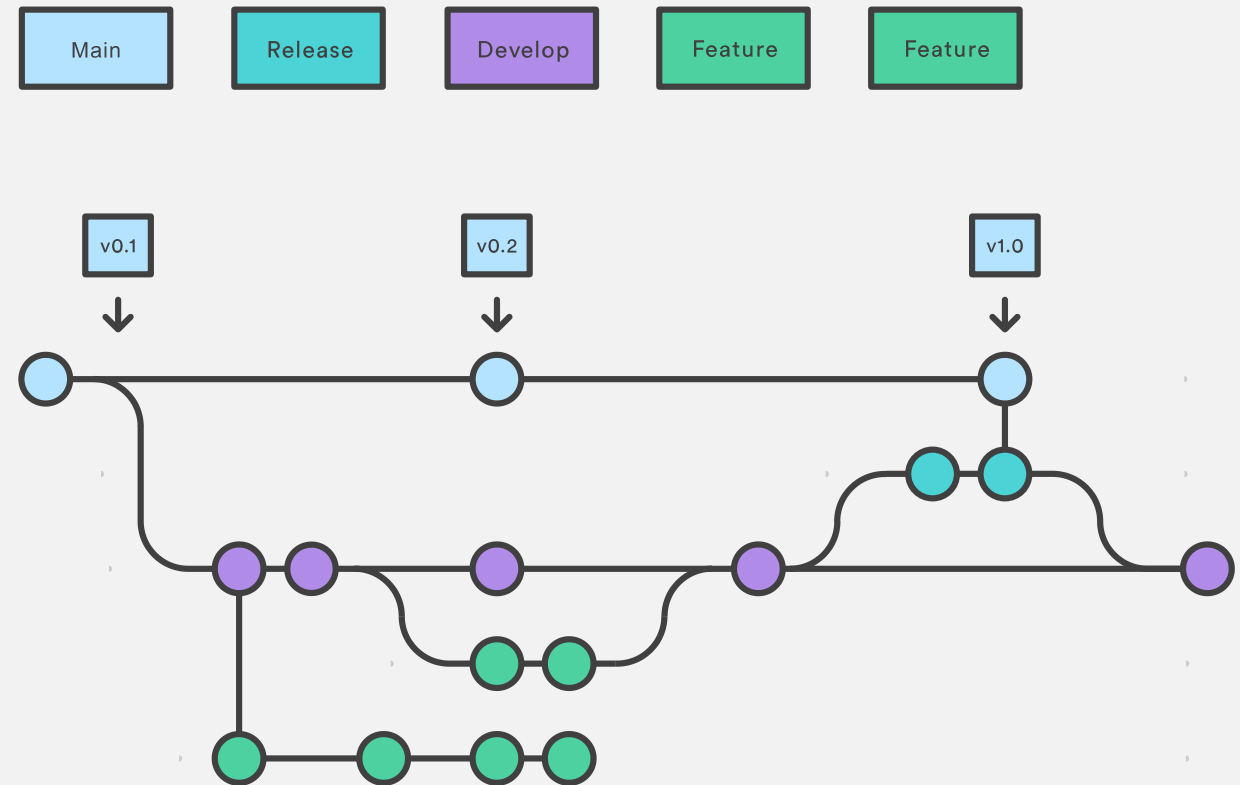
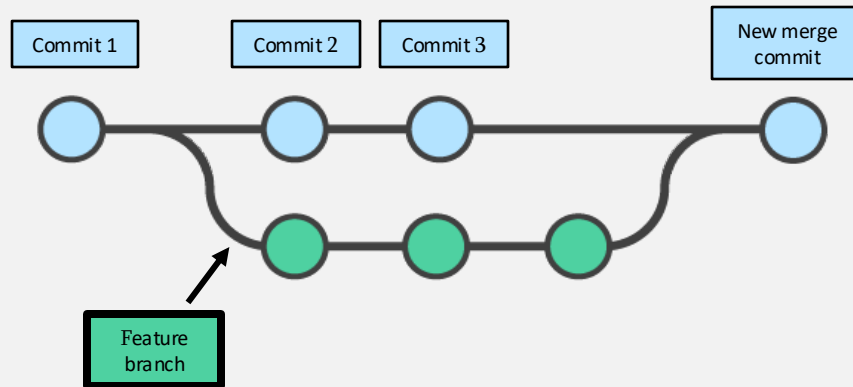
Git flow for collaboration

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



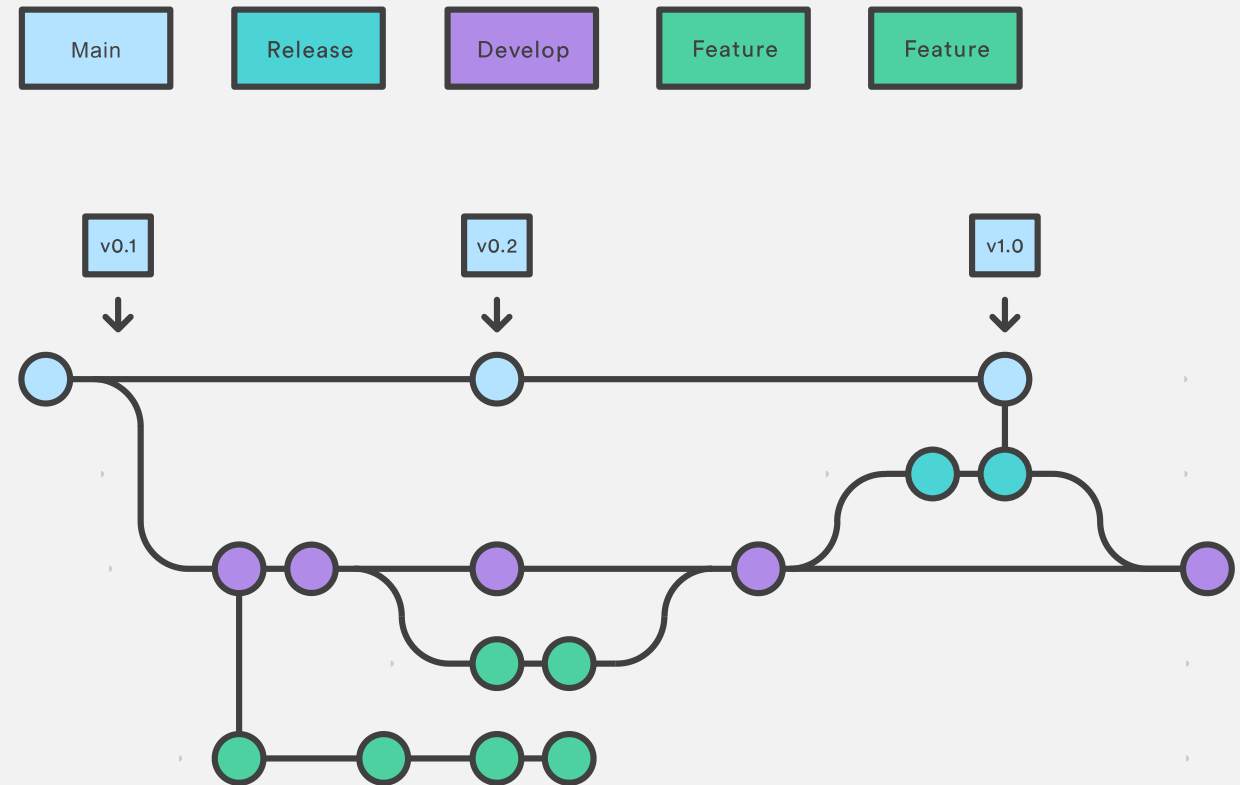
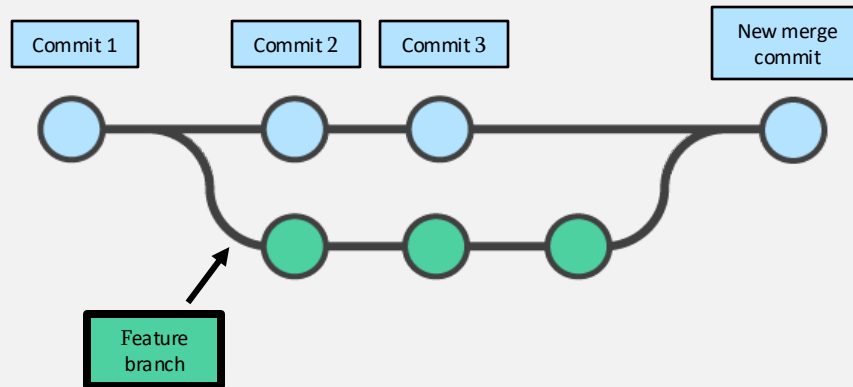
Git flow for collaboration

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



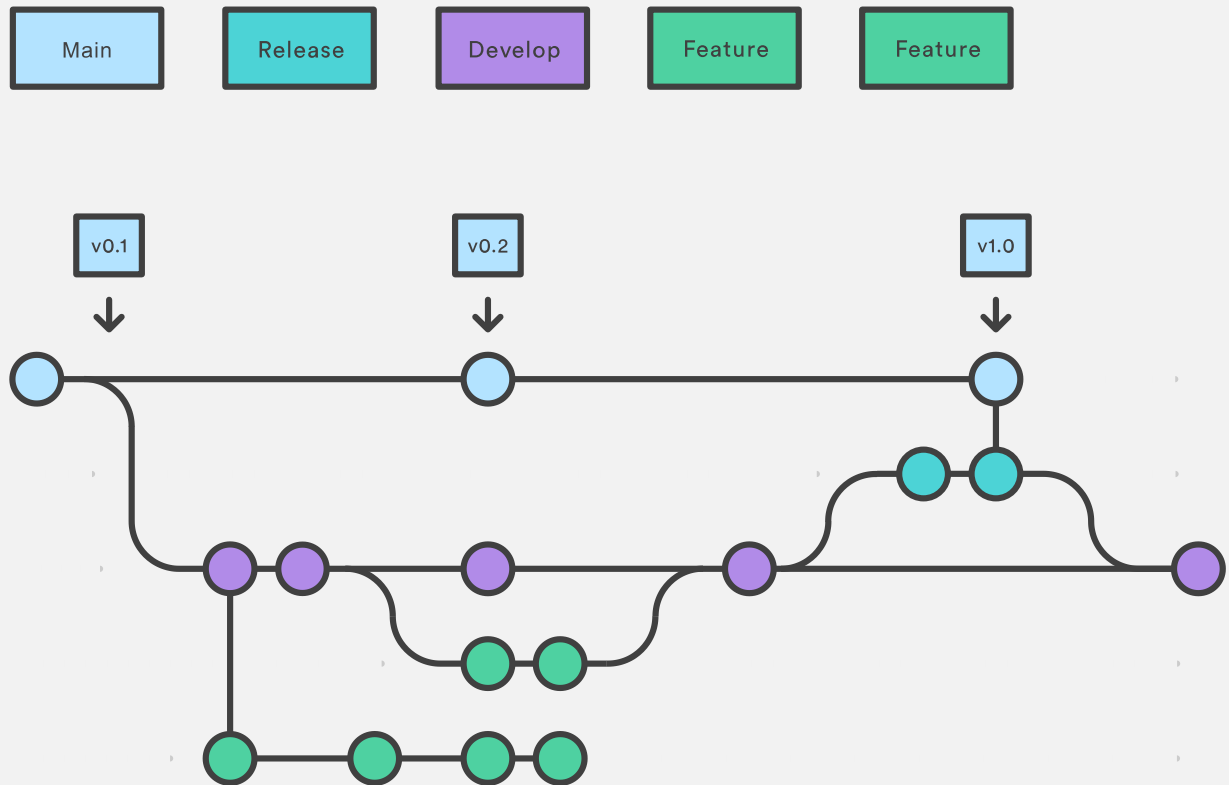
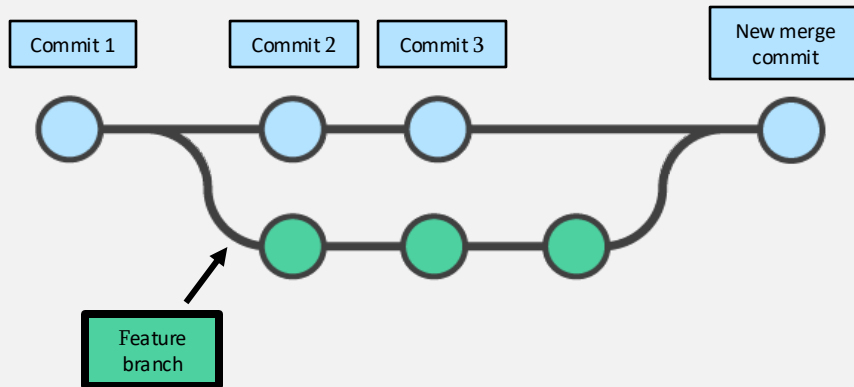
Git flow for collaboration

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



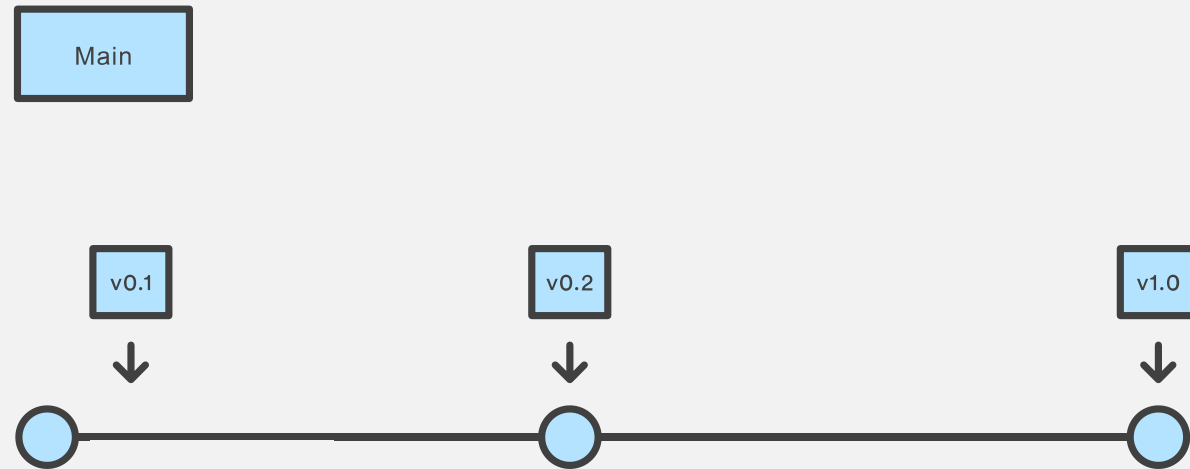
Git flow for collaboration

- 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



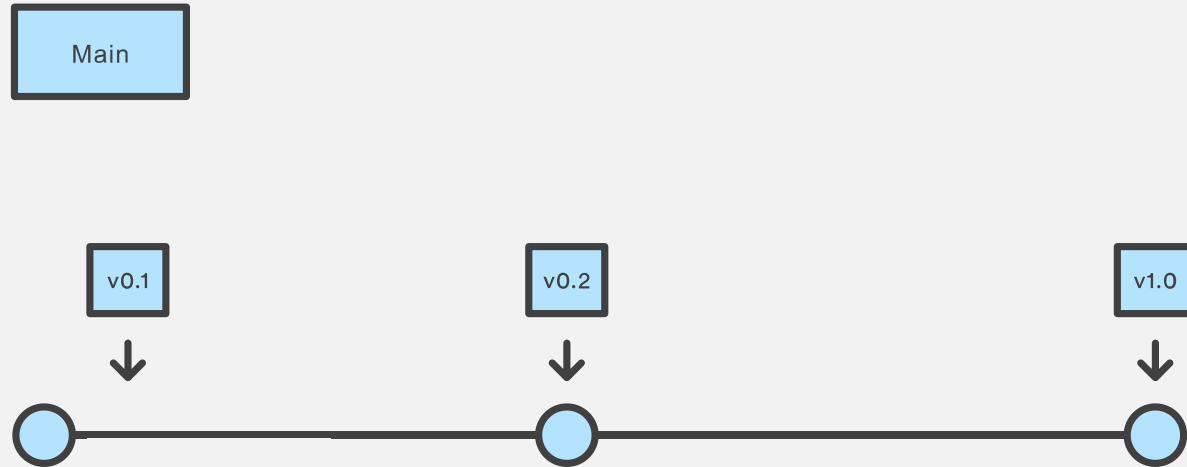
Git flow for collaboration

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



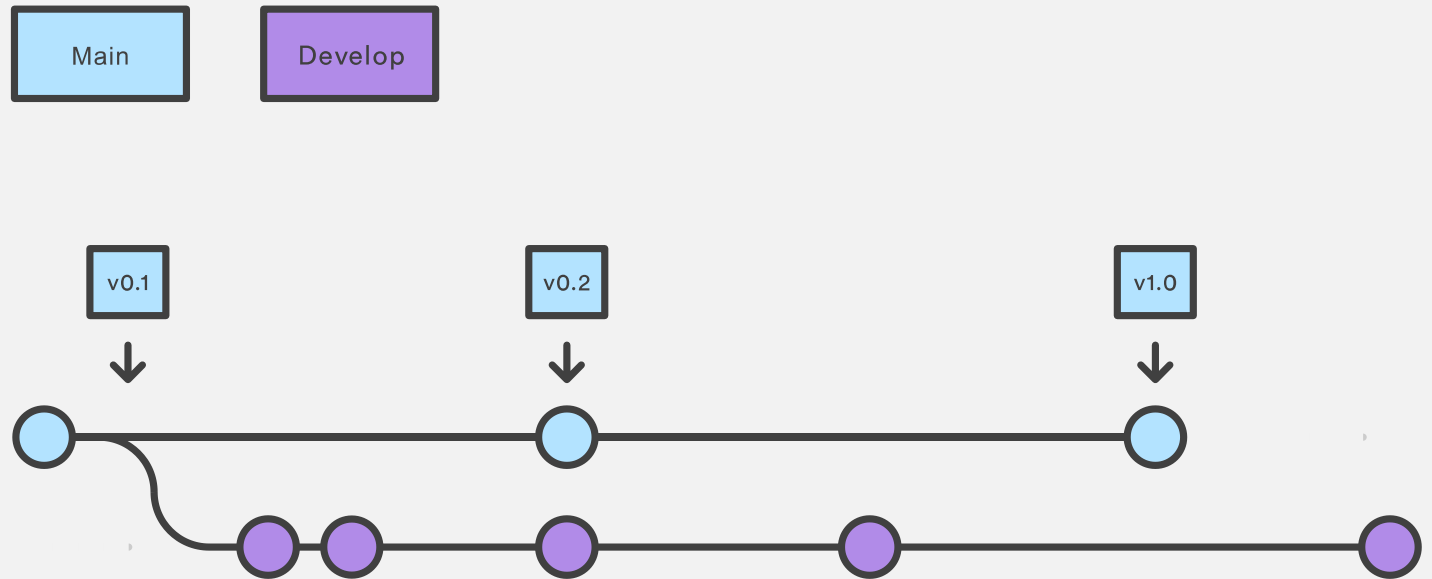
Git flow for collaboration

- 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.



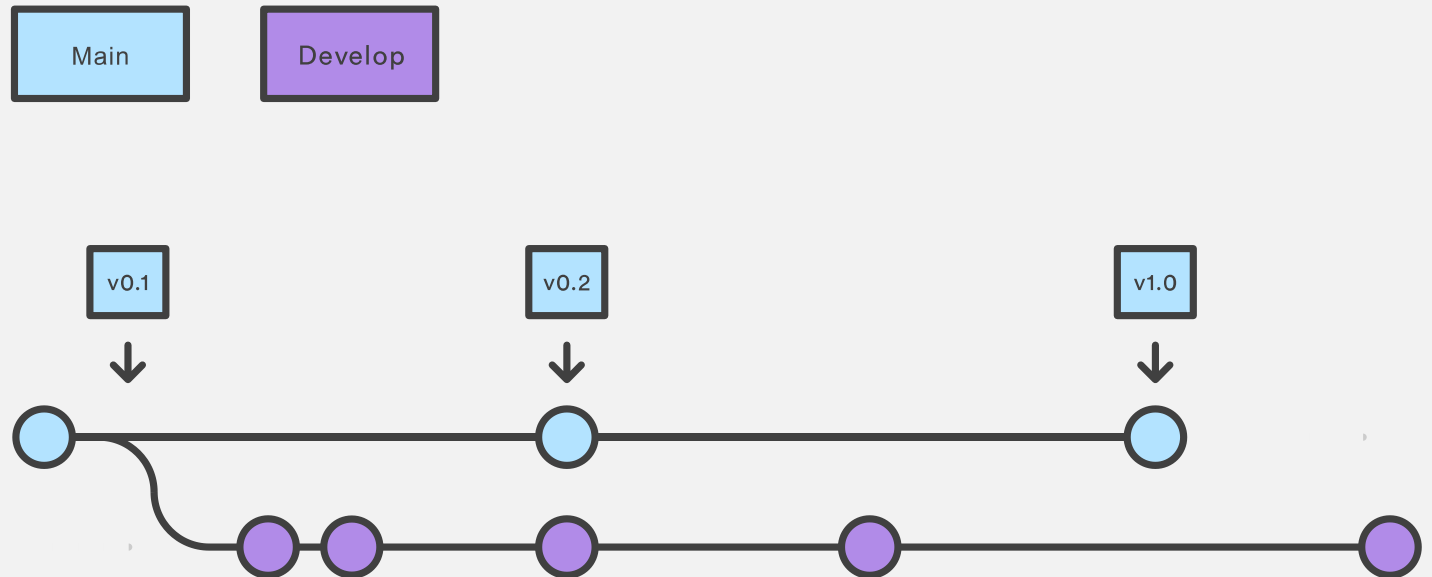
Git flow for collaboration

- 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.



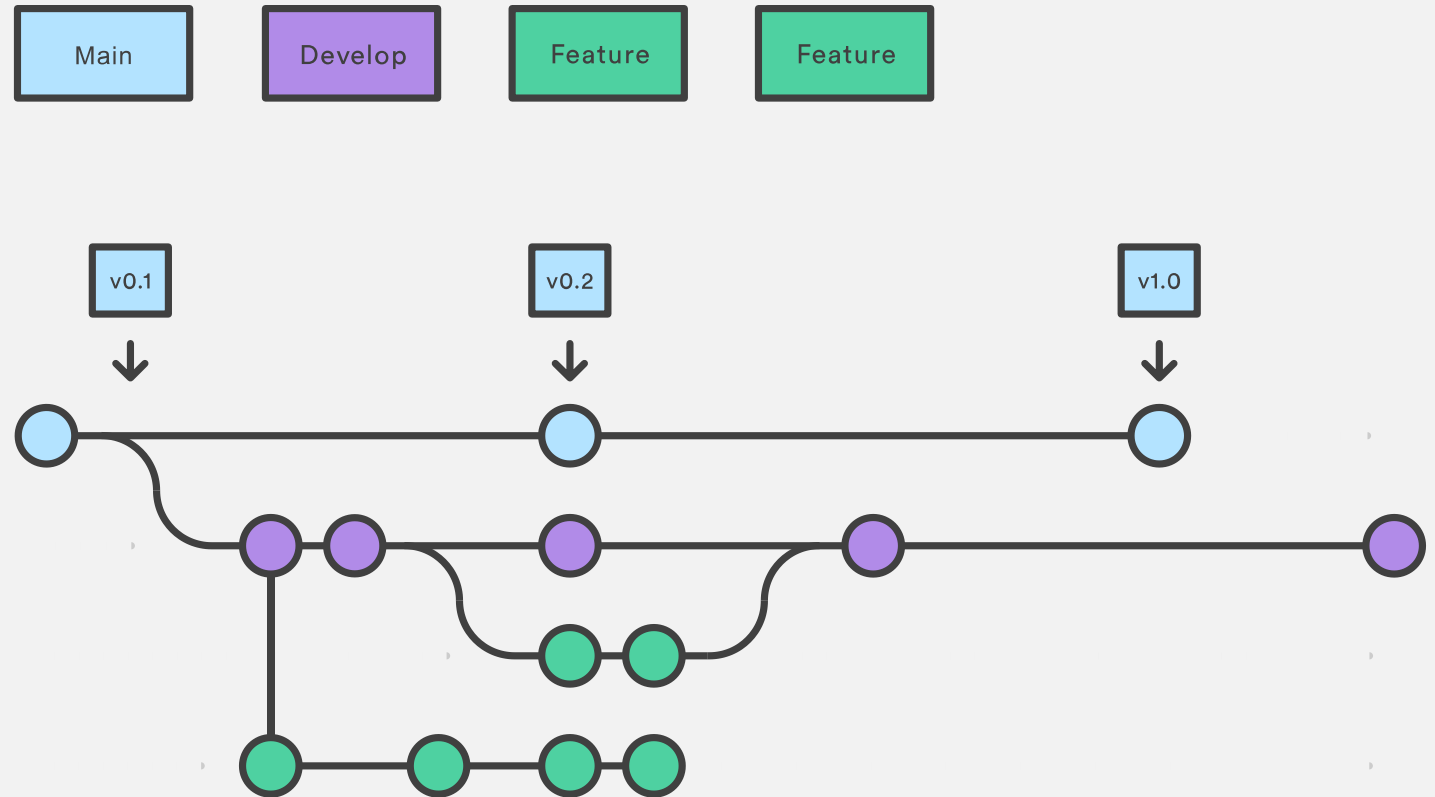
Git flow for collaboration

- 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.

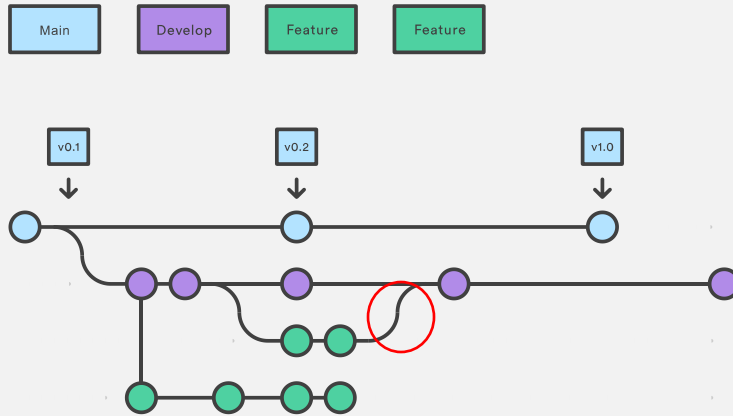


Git flow for collaboration

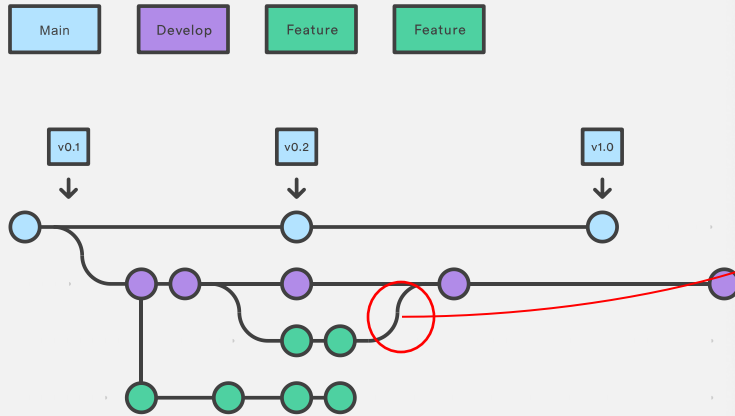
- Two branches to record the history of the project (with an infinite lifetime)
 1. "Main" branch: release history.
 2. "Develop" branch: integration branch for features.
- Each new feature has its own branch
 - Branch off Develop.
 - Merge back into Develop.



Git flow for collaboration



Git flow for collaboration



github.com/Living-with-machines/DeezyMatch/pull/121

merge 4/11

Living-with-machines / DeezyMatch

<> Code Issues 26 Pull requests Actions Projects 1 Security Insights

Improve candidate ranking #121

Merged kasra-hosseini merged 5 commits into develop from feature/85-117-ranking-improvements on Apr 25, 2022

Conversation 22 Commits 5 Checks 2 Files changed 7 +248 -103

mcollardanuy commented on Feb 2, 2022 Collaborator

This PR addresses the following issues (all related to candidate ranking):

- Ranking without prediction #85
- Add post-processing filter to candidate ranking with maximum string length difference allowed #117
- Fix the hardcoded multiplier in candidateRanker #98
- Fix ranking metric documentation in candidateRanker #97

Skip pred in cand ranking ✓ dbc2dd6

mcollardanuy requested a review from kasra-hosseini 2 years ago

Fix no predict search size ✓ c1cc110

Reviewers: kasra-hosseini ✓

Assignees: No one assigned

Labels: None yet

Projects: None yet

Milestone: No milestone

Development

Hands-on part 2/2:

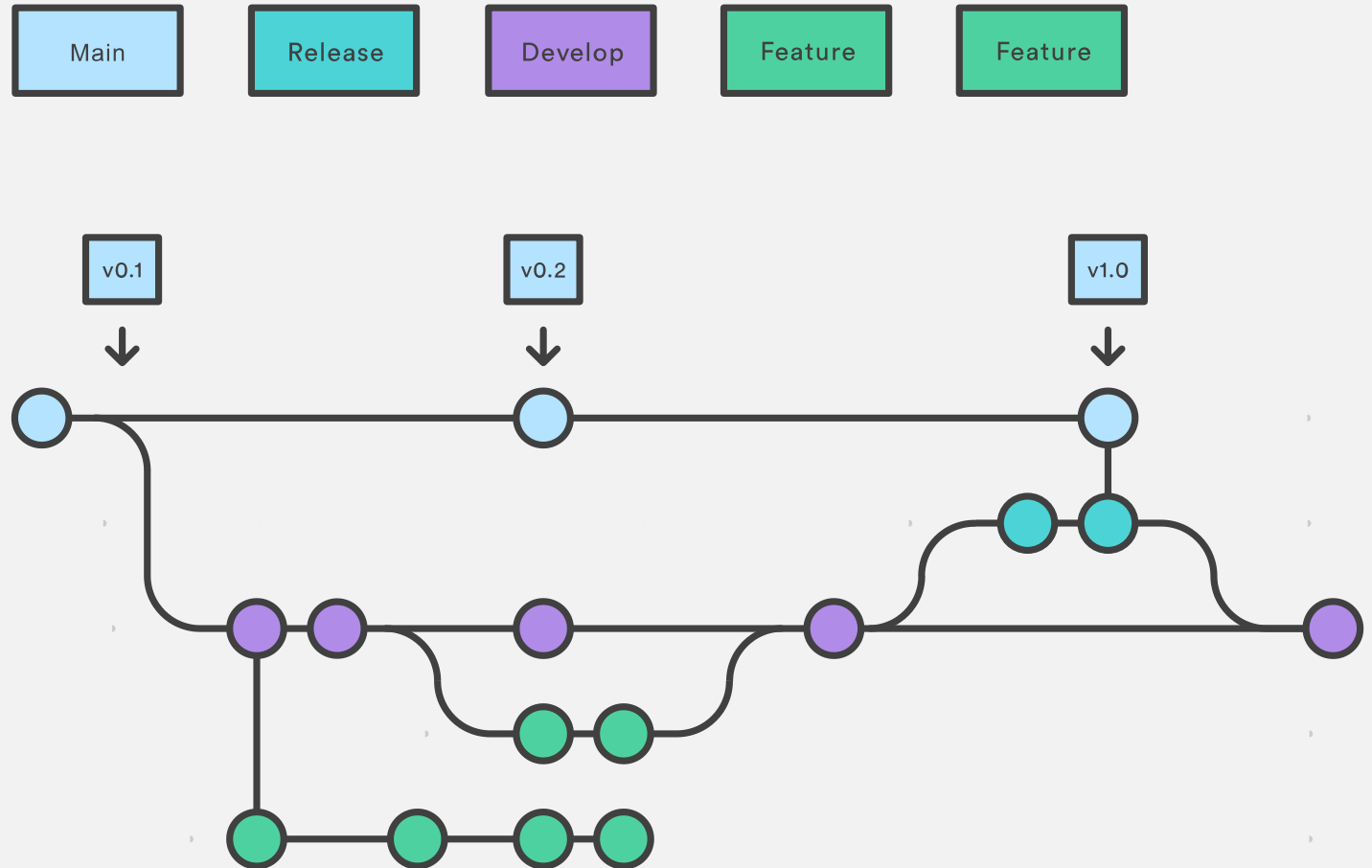
- Topics covered:
 - Creating branches
 - Creating Pull Requests
 - Reviewing Pull Requests
 - Managing merge conflicts

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.

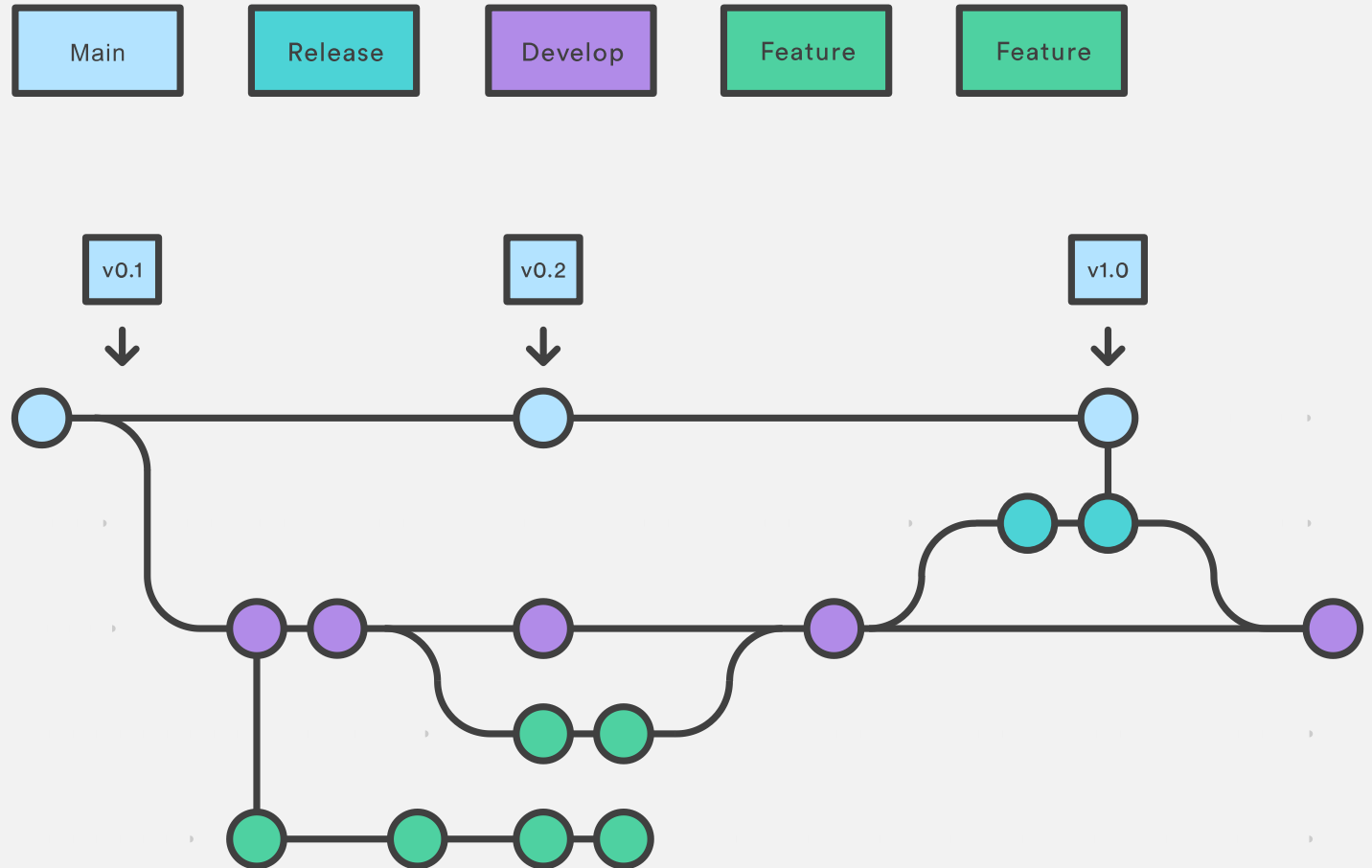
Git flow continued:
Release and Hotfix branches

Git flow: Release branches



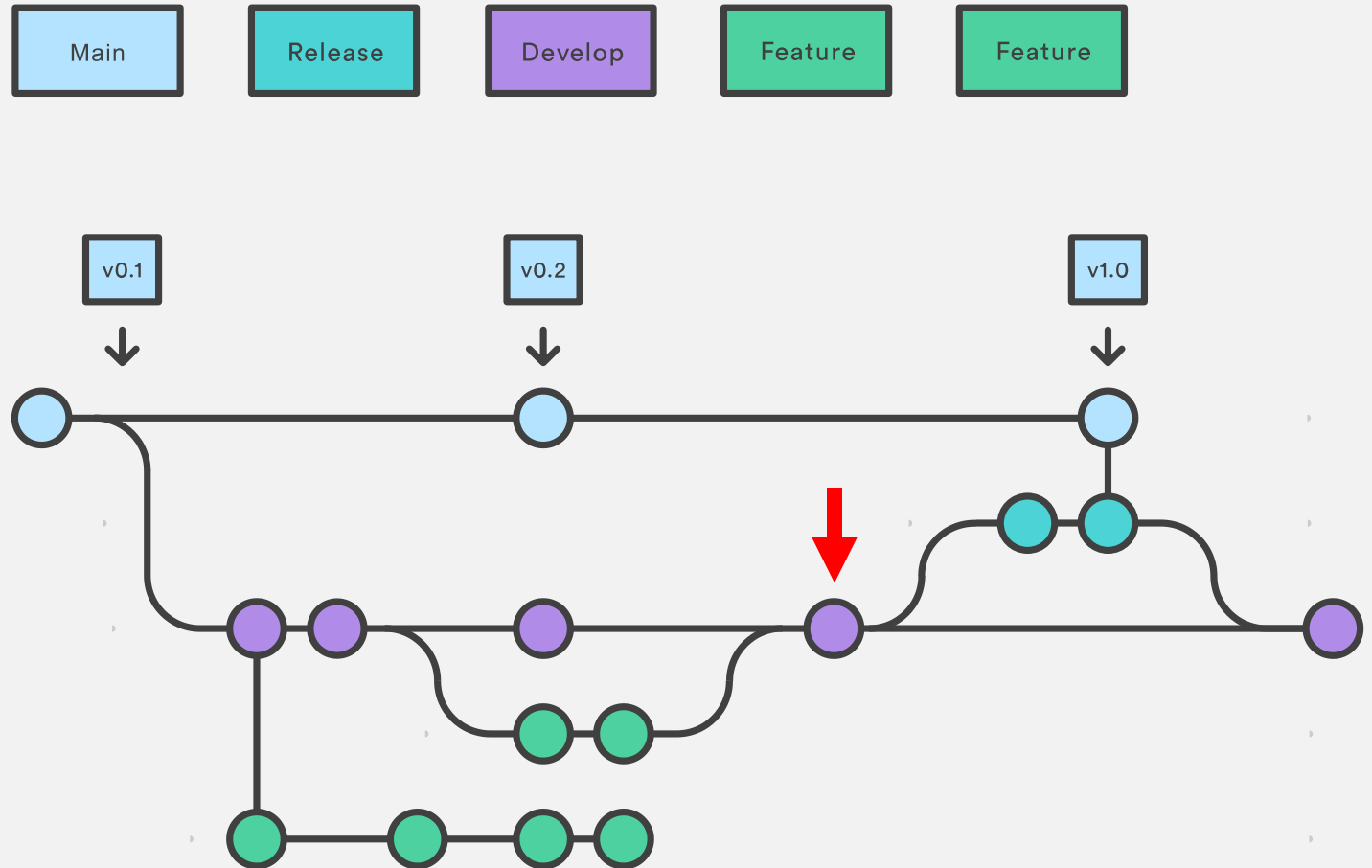
Git flow: Release branches

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



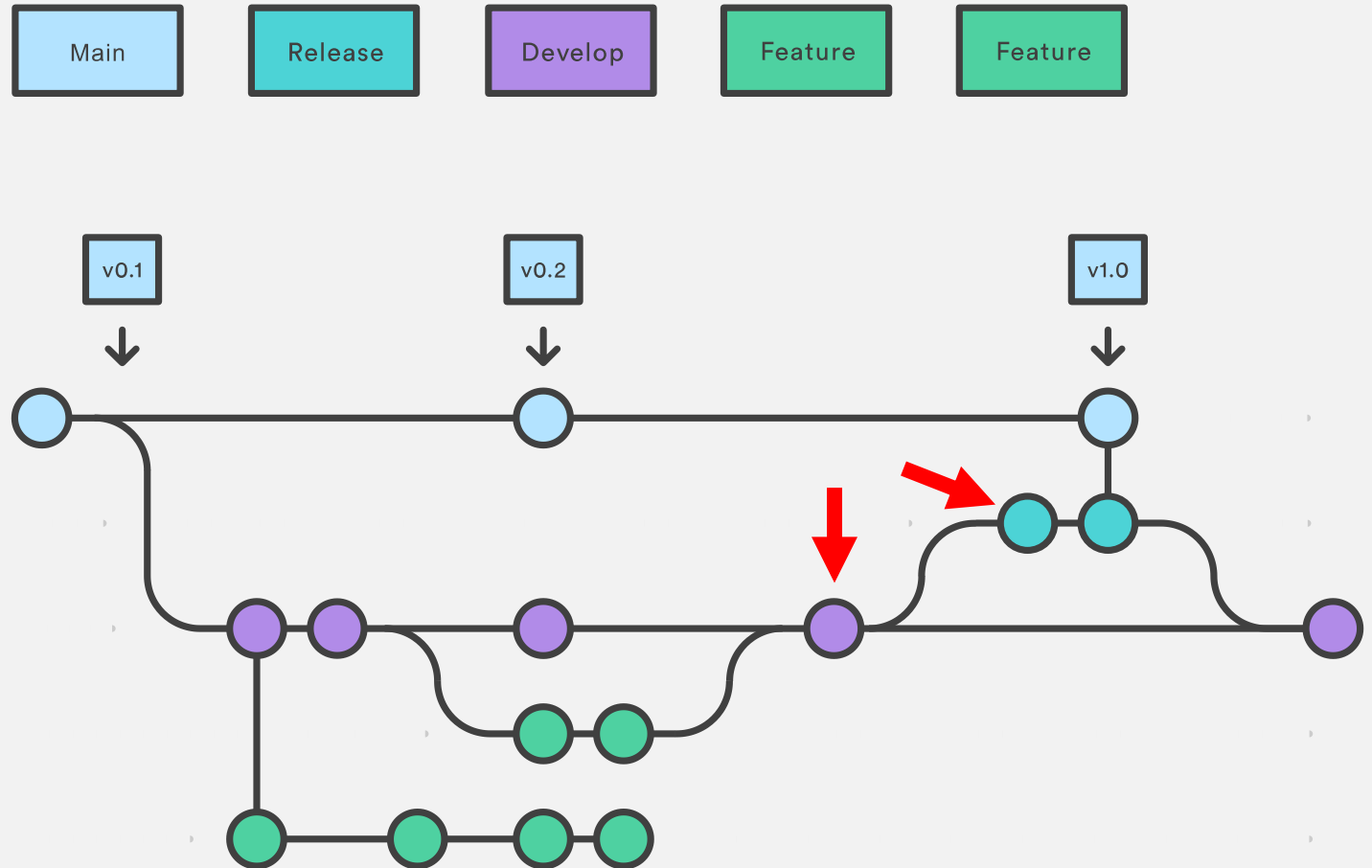
Git flow: Release branches

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



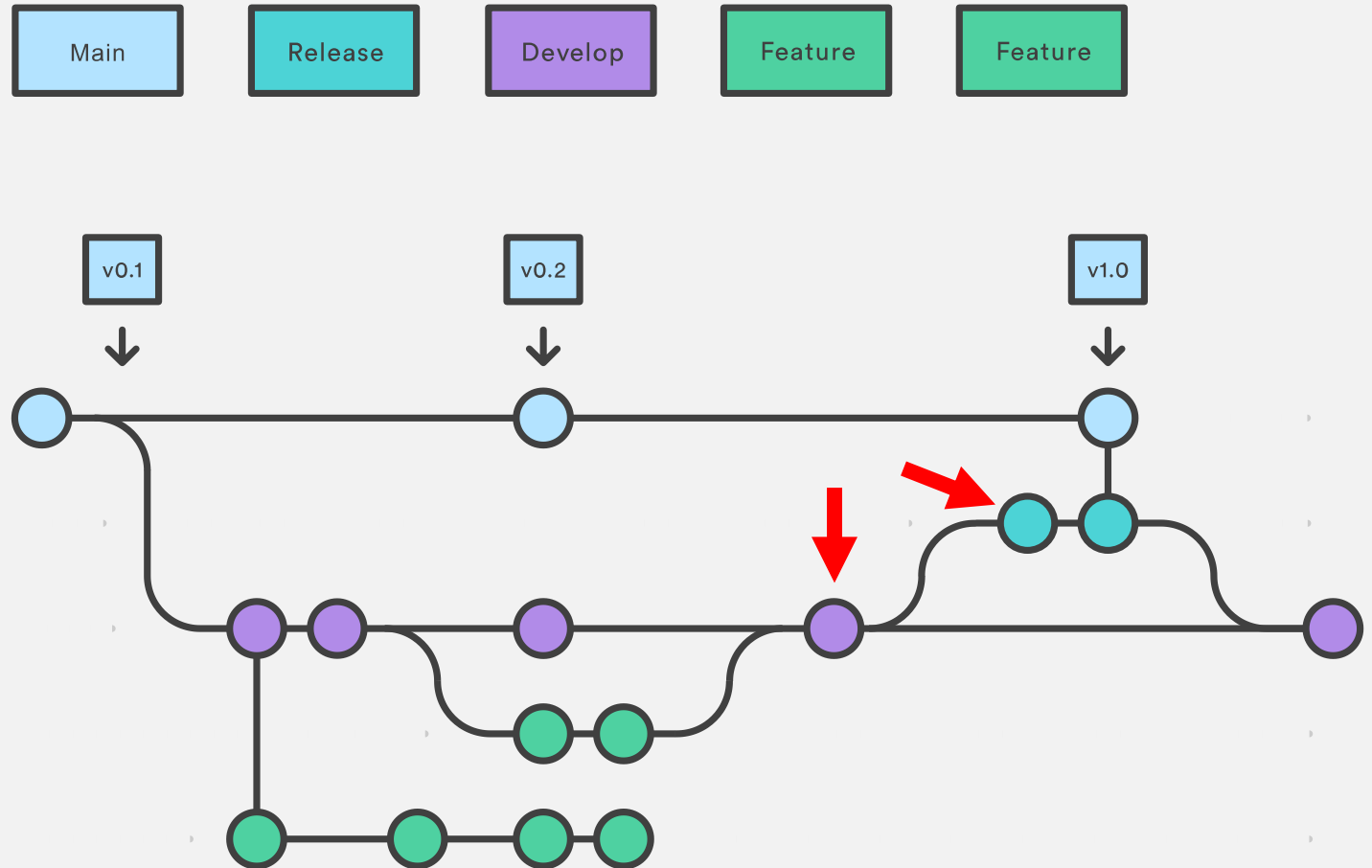
Git flow: Release branches

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

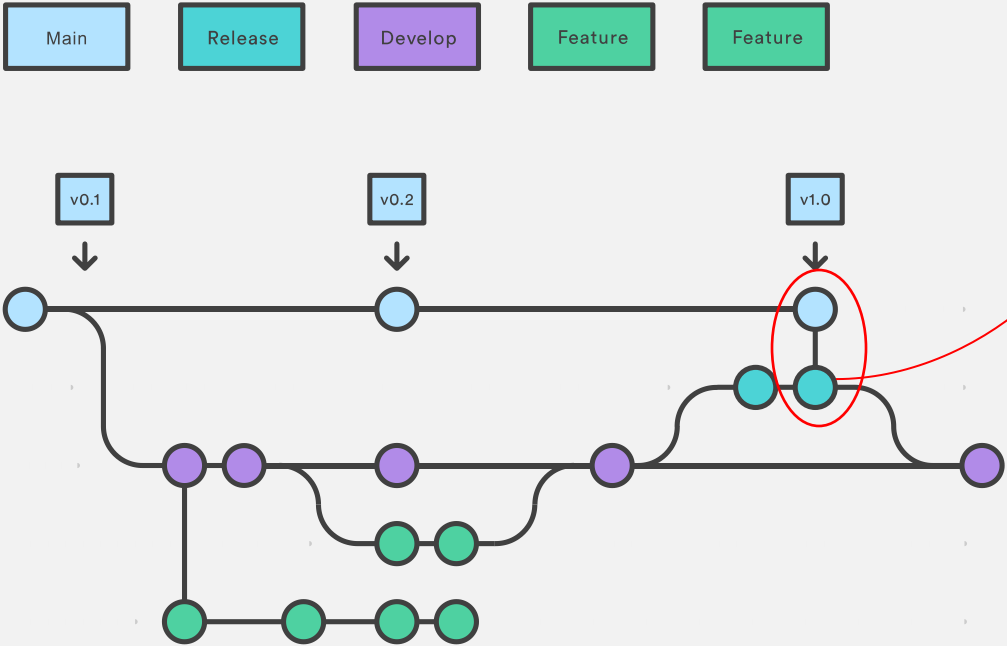


Git flow: Release branches

- 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...



version 1.2.0 #75

Merged kasra-hosseini merged 11 commits into master from develop on Sep 15, 2020

Conversation 0 Commits 11 Checks 0 Files changed 6 +122 -58

kasra-hosseini commented on Sep 15, 2020

- early stopping
- EMNLP citation
- Fix the logo

kasra-hosseini added 11 commits 3 years ago

- Early stopping, two parameters: ...
- Remove early_stopping_delta for simplicity ...
- Update input_dfm_notebook_001 accordingly
- DeezyMatch saves models for each epoch, i.e., checkpoints (unless `va...
- Update README
- Update input files
- Merge pull request #74 from Living-with-machines/feature/24-early-sto...
- Update README.md
- Update README.md
- Version 1.2.0
- Update README.md

kasra-hosseini merged commit 38d8142 into master on Sep 15, 2020

Write Preview H B I

Leave a comment

Attach files by dragging & dropping, selecting or pasting them.

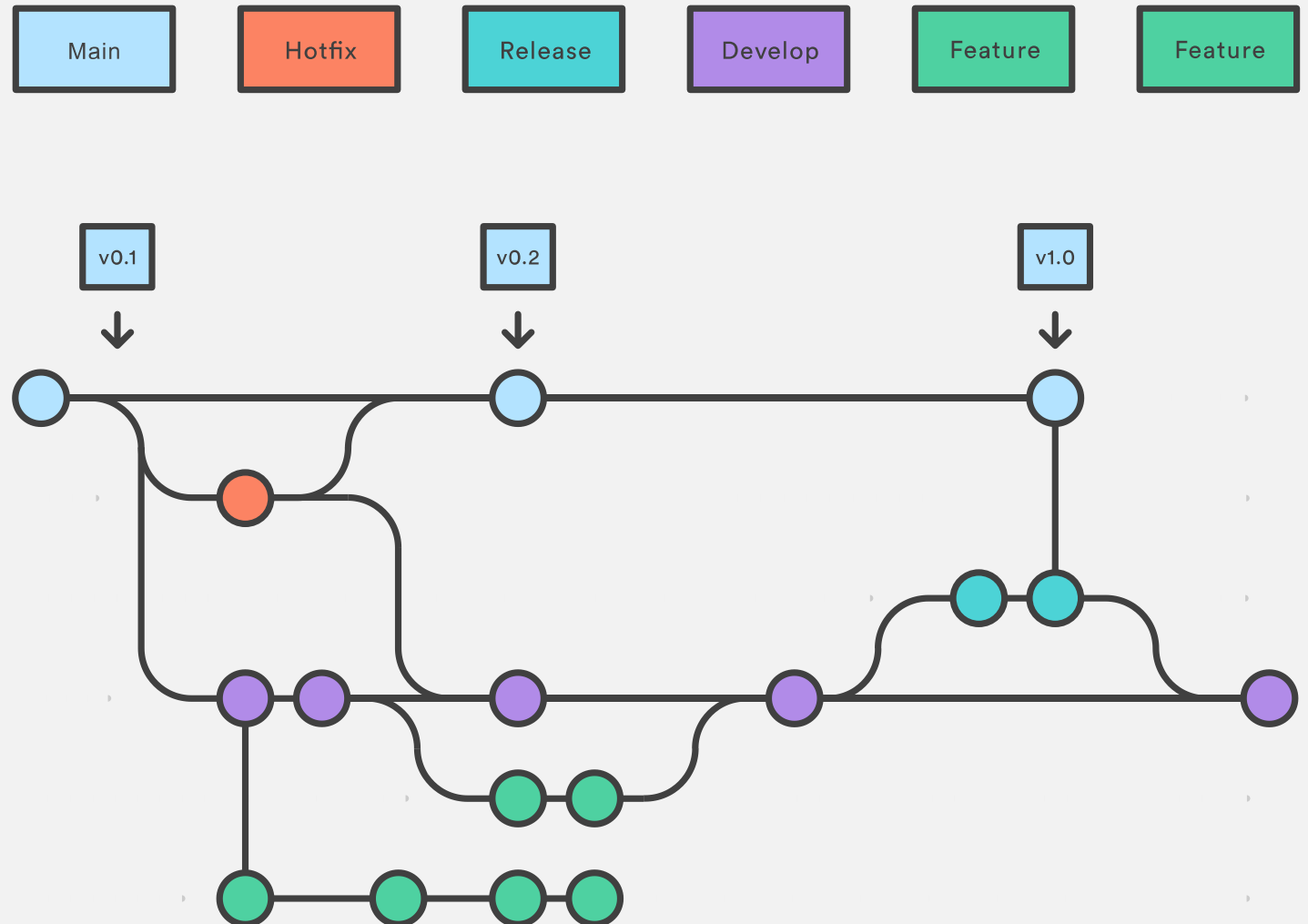
Comment

Remember, contributions to this repository should follow our GitHub Community Guidelines.

ProTip! Add .patch or .diff to the end of URLs for Git's plaintext views.

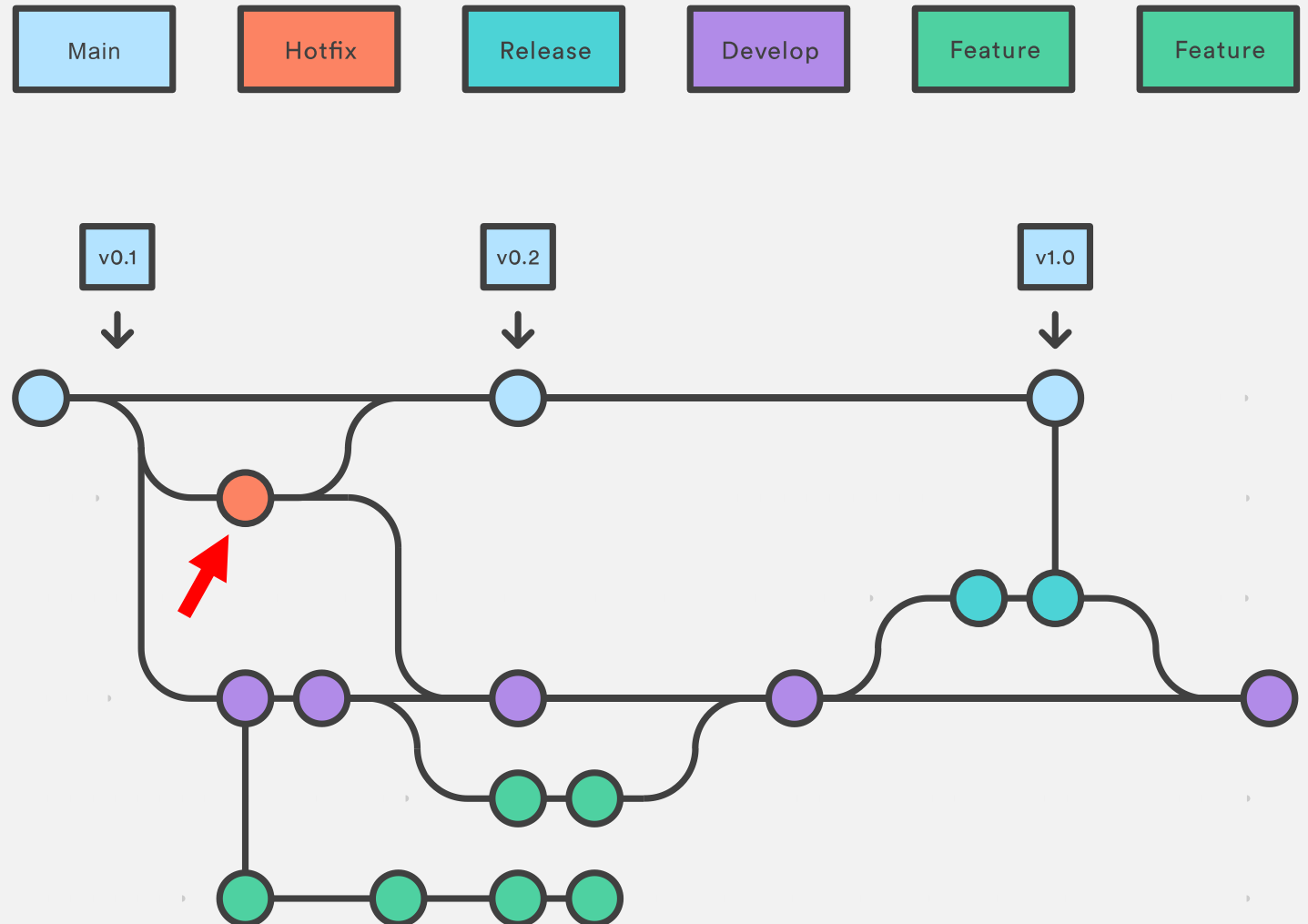
Git flow: Hotfix branches

- Maintenance or Hotfix branches to patch releases



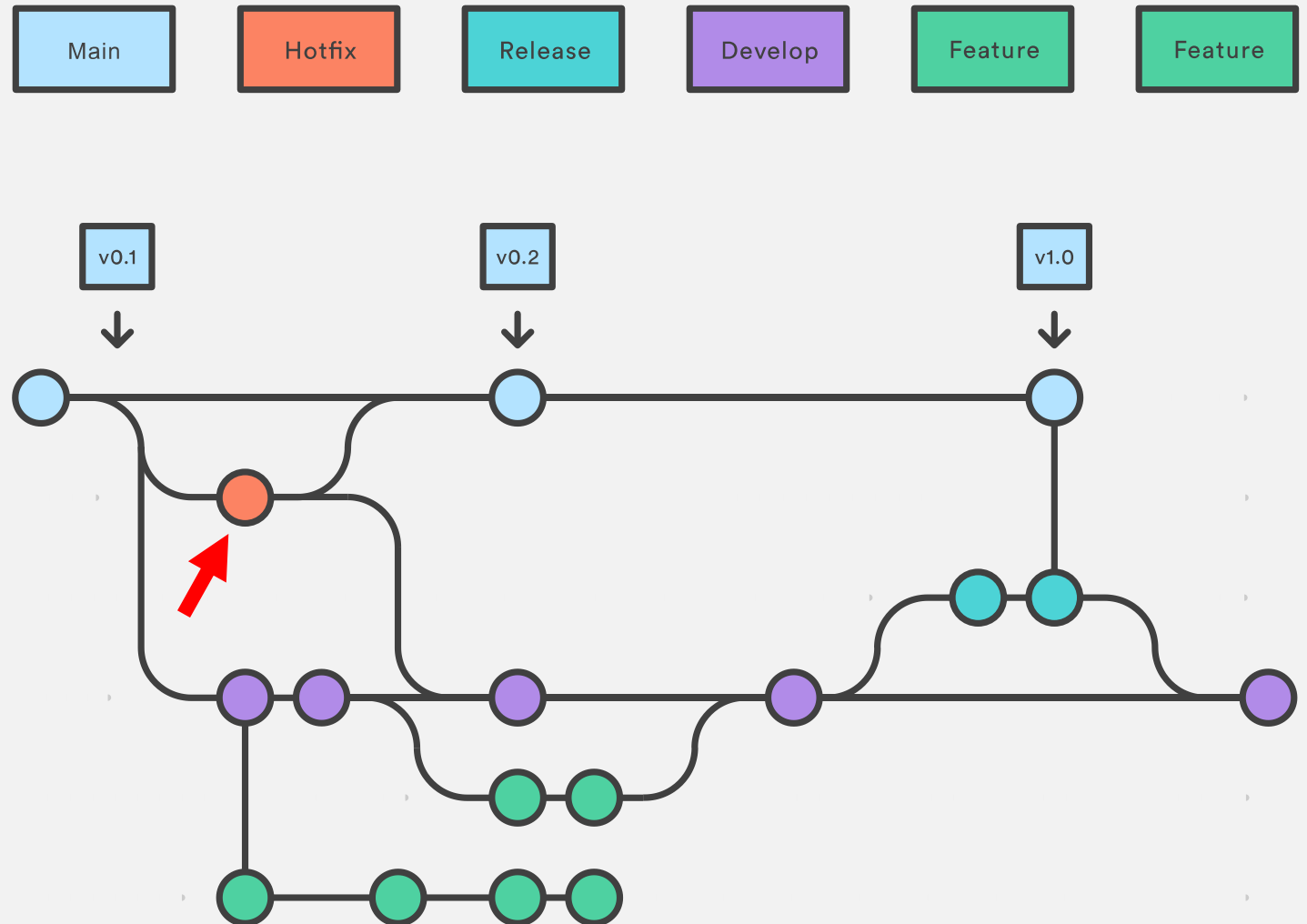
Git flow: Hotfix branches

- Maintenance or Hotfix branches to patch releases



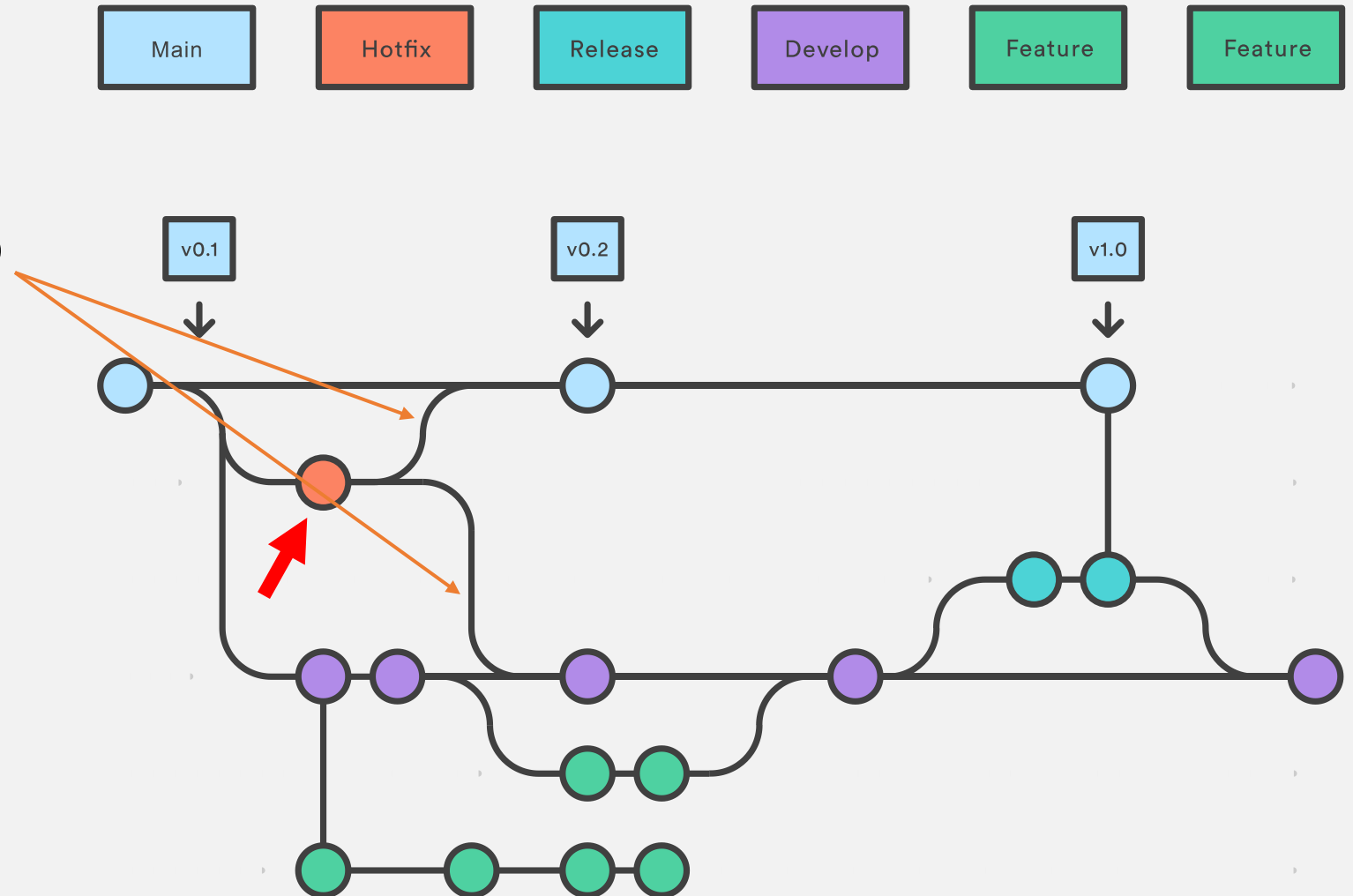
Git flow: Hotfix branches

- 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 two browser windows side-by-side, illustrating the Git flow process for versioning software releases.

Left Window (GitHub Repository):

- URL: `github.com/Living-with-machines/DeezyMatch`
- Repository: **DeezyMatch** (Public)
- Branches: `master` (selected)
- Files: `.github/work...`, `DeezyMatch`, `dataset`, `examples`, `figs`, `inputs`, `.gitignore`, `CONTRIBUT...`, `LICENSE`, `README.md`, `_config.yml`, `requirement...`, `setup.py`
- Releases: **DeezyMatch v1.3.4** (Latest) on Sep 20, 2022

Right Window (Project Landing Page):

- URL: `living-with-machines.github.io/DeezyMatch`
- Project Name: **DeezyMatch**
- Description: A Flexible Deep Learning Approach to Fuzzy String Matching
- View the Project on GitHub: `Living-with-machines/DeezyMatch`
- Logo: DeezyMatch
- Tags: `nlp`, `machine-learning`, `natural-language-processing`, `deep-learning`, `hacktoberfest`, `hut23`, `hut23-96`
- Readme: A Flexible Deep Neural Network Approach to Fuzzy String Matching
- Integration Tests: `passing`
- Tasks: Fuzzy string matching, Candidate ranking/selection, Query expansion, Toponym matching
- Record linkage

No code was damaged
in the making of this workshop