

# Git & GitHub

Francesco Frassinelli, NINA



Living Norway  
Ecological Data Network

# What is Git?

---

# What is Git?

## Git

From Wikipedia, the free encyclopedia

*For other uses, see [Git \(disambiguation\)](#).*

*Not to be confused with [GitHub](#).*

**Git** ([/ɡɪt/](#))<sup>[9]</sup> is software for tracking changes in any set of [files](#), usually used for coordinating work among [programmers](#) collaboratively developing [source code](#) during [software development](#). Its goals include speed, [data integrity](#), and support for distributed, non-linear workflows (thousands of parallel branches running on different systems).

<sup>[10][11][12]</sup>

# What is Git?

---

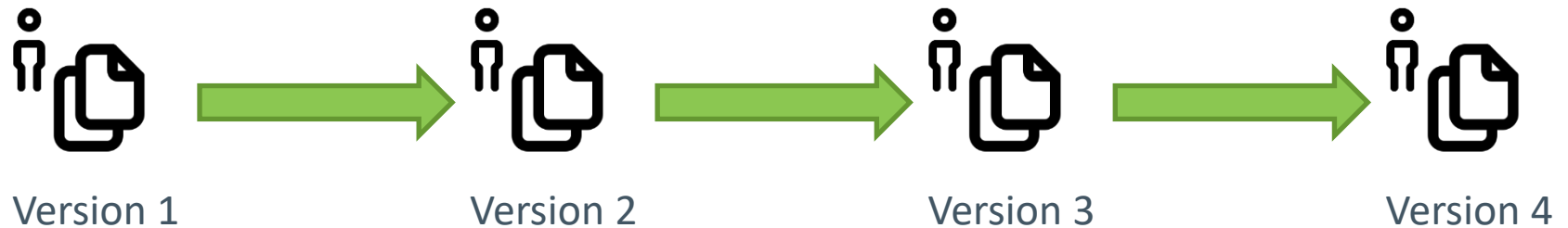
**Git** ([/git/](#))<sup>[9]</sup> is software for tracking changes in any set of [files](#), usually used for coordinating work among [programmers](#) collaboratively developing [source code](#) during [software development](#). Its goals include speed, [data integrity](#), and support for distributed, non-linear workflows (thousands of parallel branches running on different systems).  
[\[10\]](#)[\[11\]](#)[\[12\]](#)

# What is Git?

---

**Git** ([/git/](#))<sup>[9]</sup> is software for tracking changes in any set of files, usually used for coordinating work among [programmers](#) collaboratively developing [source code](#) during [software development](#). Its goals include speed, [data integrity](#), and support for distributed, non-linear workflows (thousands of parallel branches running on different systems).  
[\[10\]](#)[\[11\]](#)[\[12\]](#)

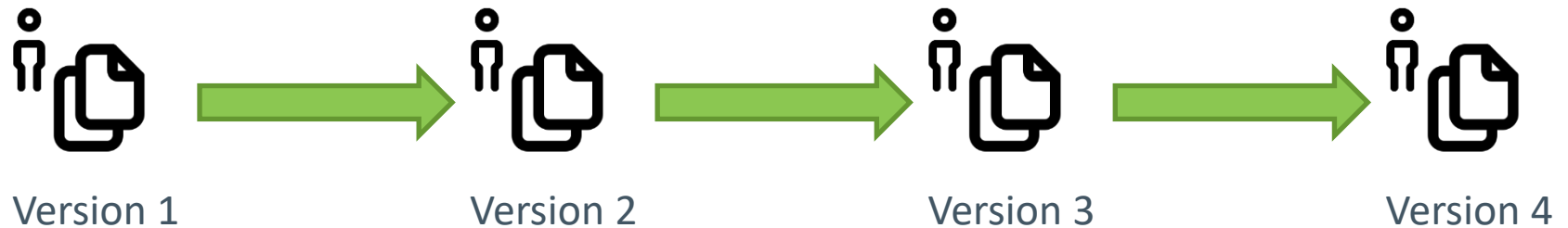
# What is Git?



**Git** ([/git/](https://git-scm.com/))<sup>[9]</sup> is software for tracking changes in any set of files, usually used for coordinating work among [programmers](#) collaboratively developing [source code](#) during [software development](#). Its goals include speed, [data integrity](#), and support for distributed, non-linear workflows (thousands of parallel branches running on different systems).

[10][11][12]

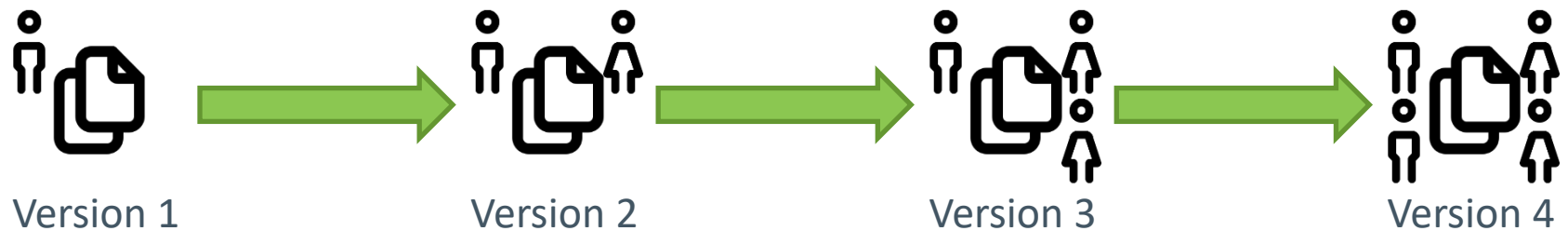
# What is Git?



**Git** ([/git/](#))<sup>[9]</sup> is software for tracking changes in any set of [files](#), usually used for coordinating work among [programmers](#) collaboratively developing [source code](#) during [software development](#). Its goals include speed, [data integrity](#), and support for distributed, non-linear workflows (thousands of parallel branches running on different systems).

[10][11][12]

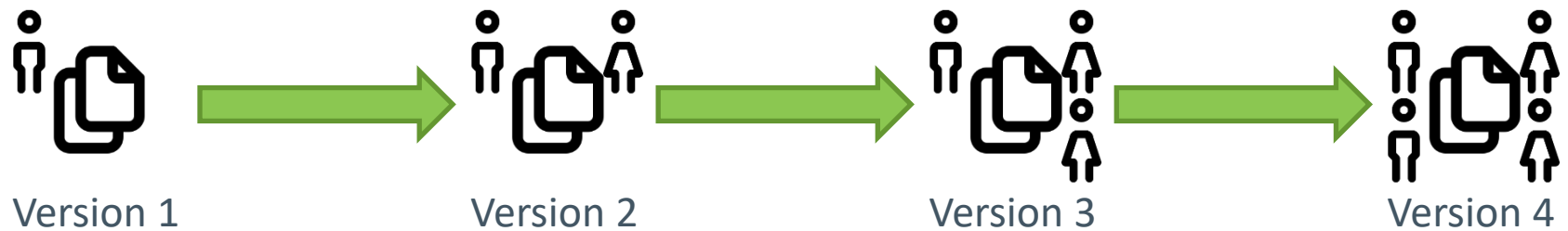
# What is Git?



**Git** ([/git/](https://git-scm.com/))<sup>[9]</sup> is software for tracking changes in any set of [files](#), usually used for coordinating work among [programmers](#) collaboratively developing [source code](#) during [software development](#). Its goals include speed, [data integrity](#), and support for distributed, non-linear workflows (thousands of parallel branches running on different systems).

[10][11][12]

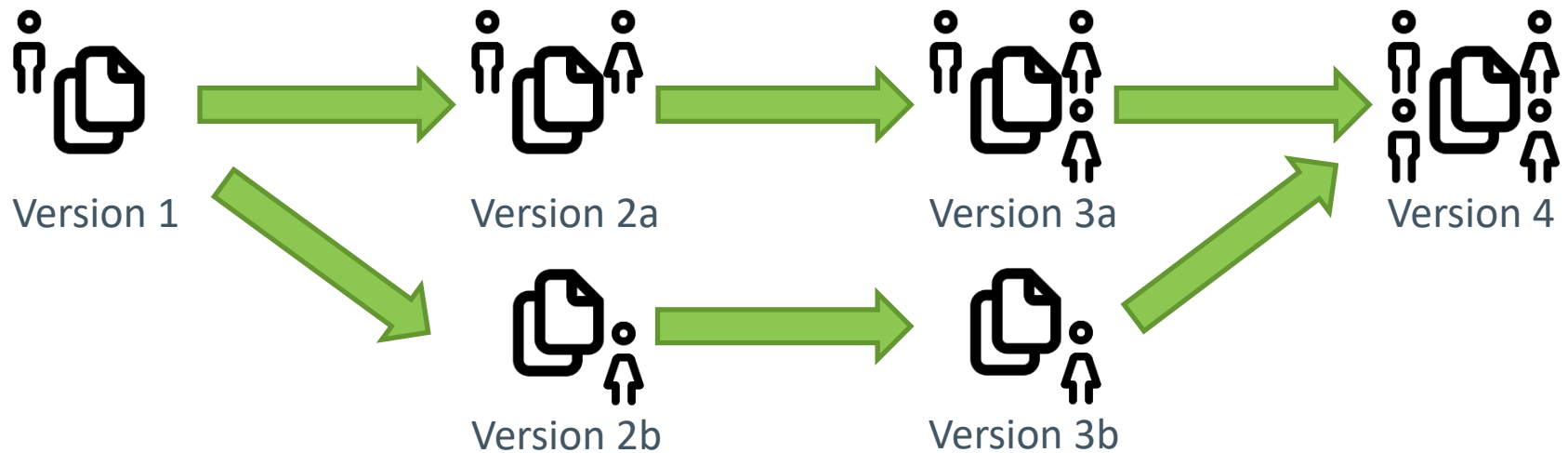
# What is Git?



[software development](#). Its goals include speed, [data integrity](#), and support for distributed, non-linear workflows (thousands of parallel branches running on different systems).

[\[10\]](#)[\[11\]](#)[\[12\]](#)

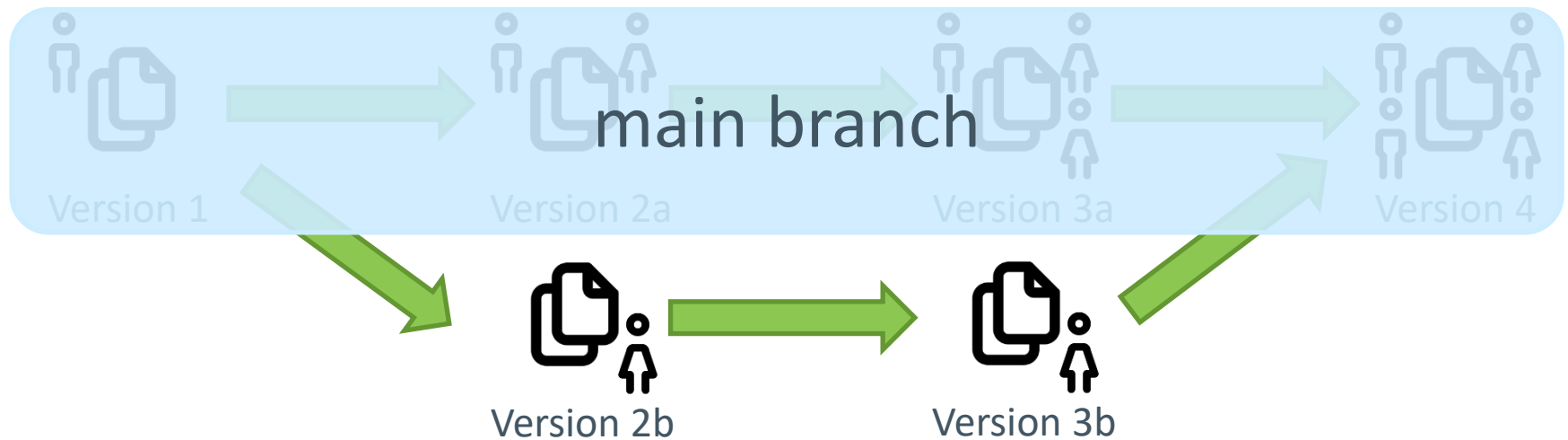
# What is Git?



software development. Its goals include speed, data integrity, and support for distributed, non-linear workflows (thousands of parallel branches running on different systems).

[10][11][12]

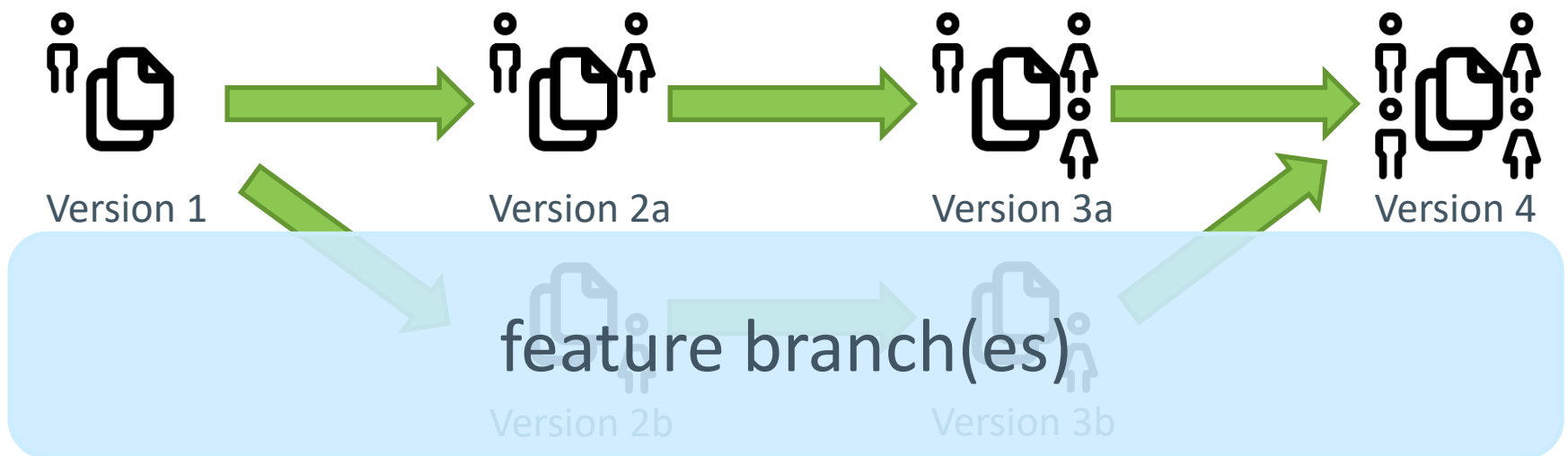
# What is Git?



software development. Its goals include speed, data integrity, and support for distributed, non-linear workflows (thousands of parallel branches running on different systems).

[10][11][12]

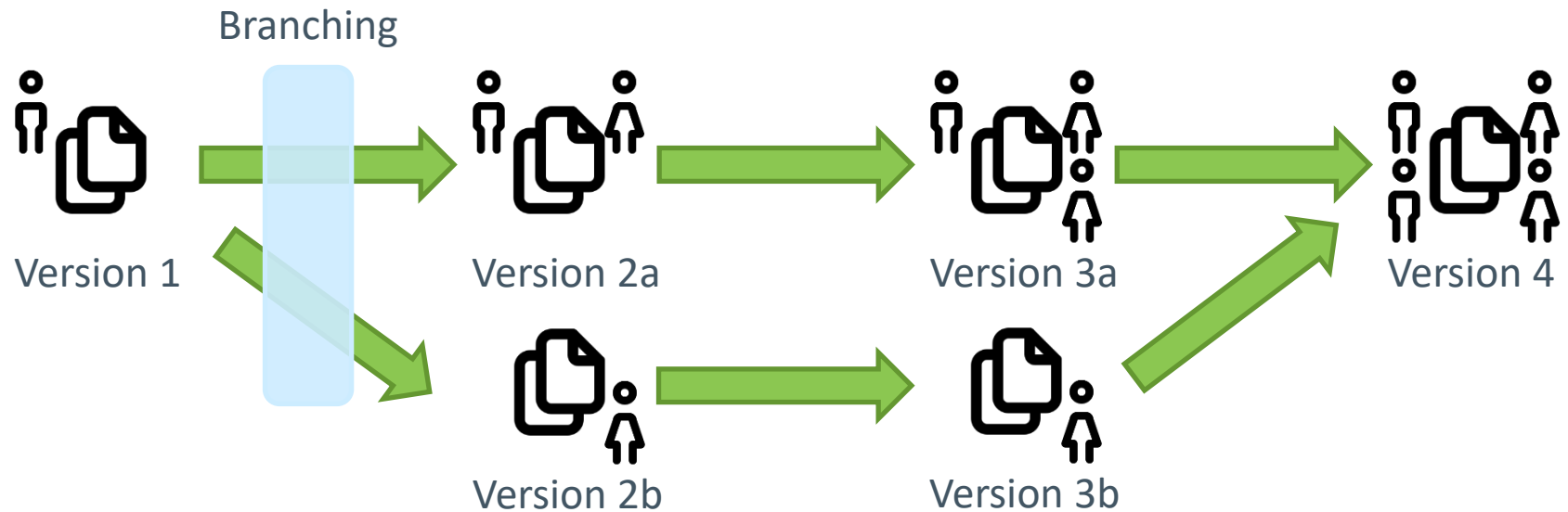
# What is Git?



software development. Its goals include speed, data integrity, and support for distributed, non-linear workflows (thousands of parallel branches running on different systems).

[10][11][12]

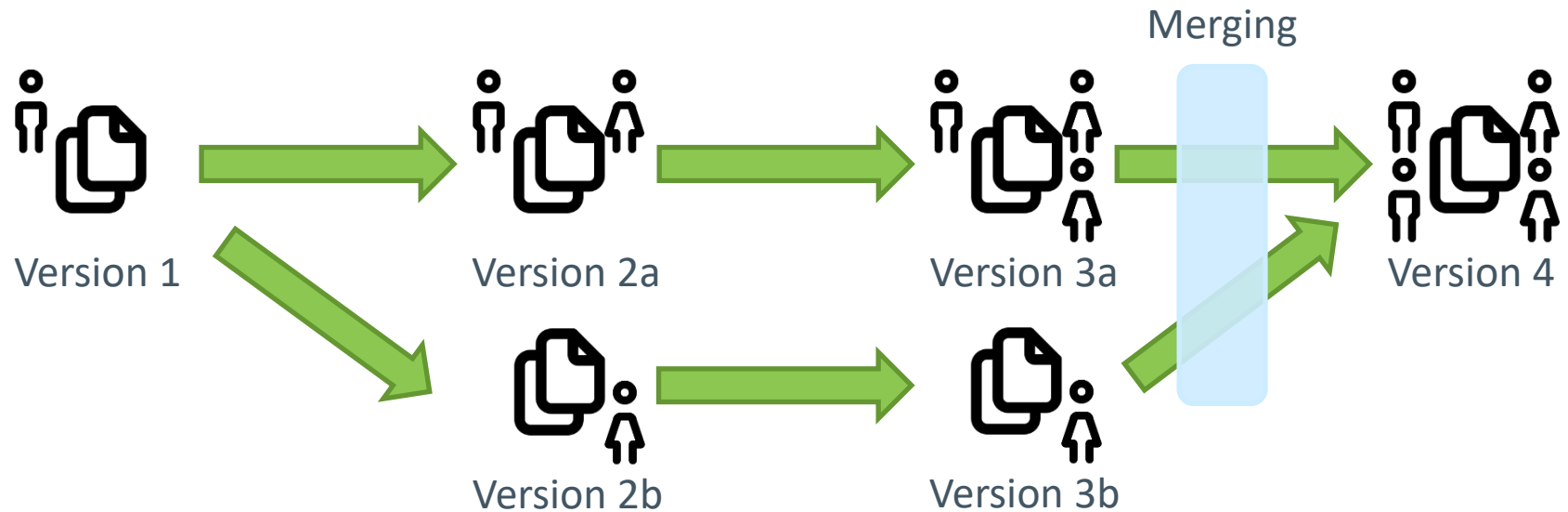
# What is Git?



software development. Its goals include speed, data integrity, and support for distributed, non-linear workflows (thousands of parallel branches running on different systems).

[10][11][12]

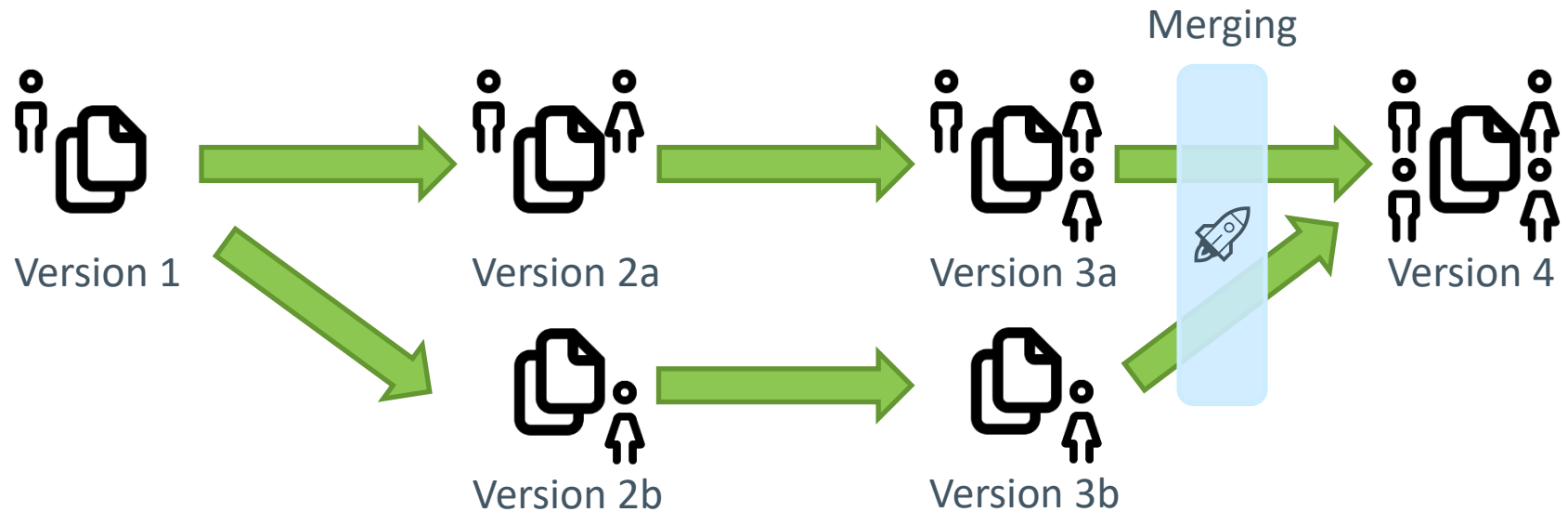
# What is Git?



software development. Its goals include speed, data integrity, and support for distributed, non-linear workflows (thousands of parallel branches running on different systems).

[10][11][12]

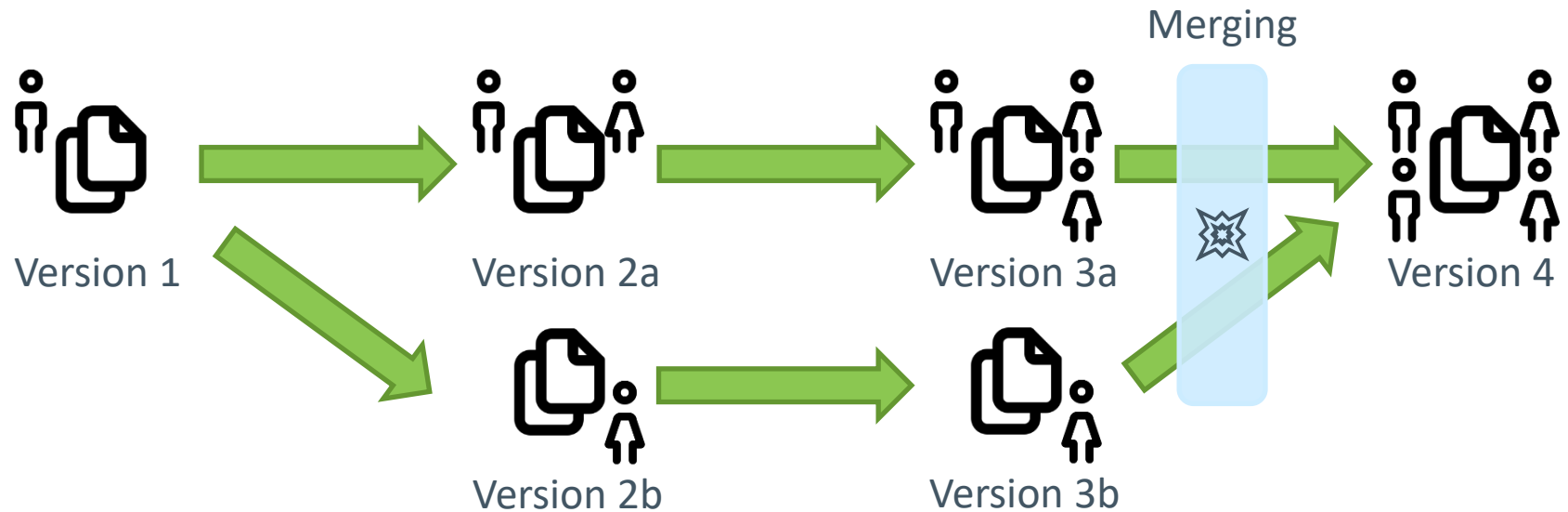
# What is Git?



software development. Its goals include speed, data integrity, and support for distributed, non-linear workflows (thousands of parallel branches running on different systems).

[10][11][12]

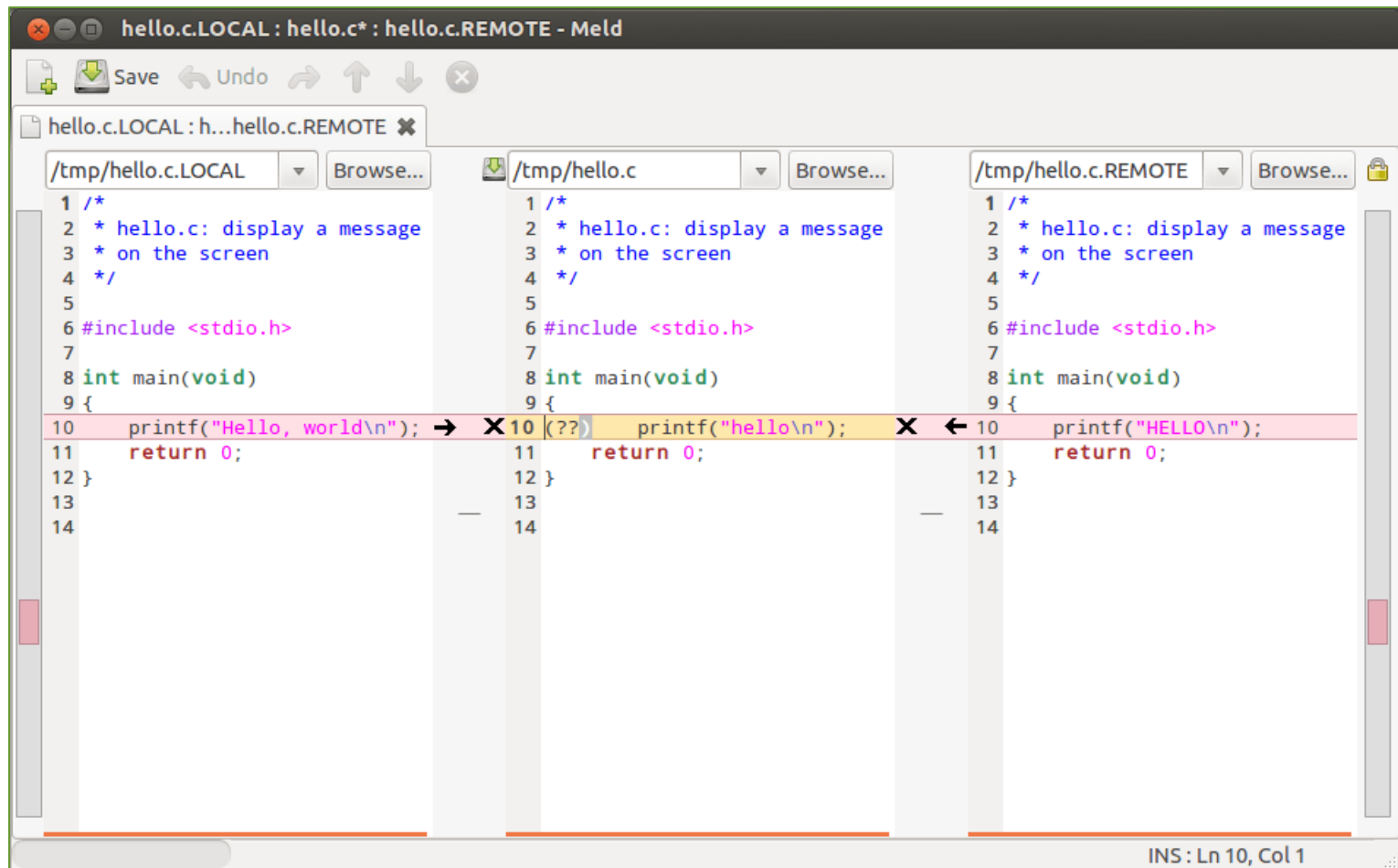
# What is Git?



software development. Its goals include speed, data integrity, and support for distributed, non-linear workflows (thousands of parallel branches running on different systems).

[10][11][12]

# Conflicts



hello.c.LOCAL : hello.c\* : hello.c.REMOTE - Meld

Save Undo

hello.c.LOCAL : h...hello.c.REMOTE X

/tmp/hello.c.LOCAL Browse... /tmp/hello.c Browse... /tmp/hello.c.REMOTE Browse...

```
1 /*
2  * hello.c: display a message
3  * on the screen
4  */
5
6 #include <stdio.h>
7
8 int main(void)
9 {
10  printf("Hello, world\n");
11  return 0;
12 }
13
14
```

```
1 /*
2  * hello.c: display a message
3  * on the screen
4  */
5
6 #include <stdio.h>
7
8 int main(void)
9 {
10  (??) printf("hello\n");
11  return 0;
12 }
13
14
```

```
1 /*
2  * hello.c: display a message
3  * on the screen
4  */
5
6 #include <stdio.h>
7
8 int main(void)
9 {
10  printf("HELLO\n");
11  return 0;
12 }
13
14
```

INS : Ln 10, Col 1

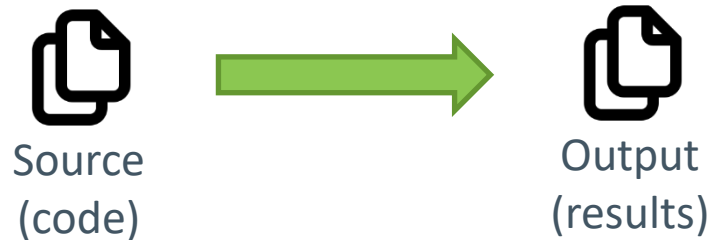
# What should I store in Git?

---

**Git** ([/git/](#))<sup>[9]</sup> is software for tracking changes in any set of [files](#), usually used for coordinating work among [programmers](#) collaboratively developing [source code](#) during [software development](#). Its goals include speed, [data integrity](#), and support for distributed, non-linear workflows (thousands of parallel branches running on different systems).

[\[10\]](#)[\[11\]](#)[\[12\]](#)

# What should I store in Git?



**Git** ([/git/](#))<sup>[9]</sup> is software for tracking changes in any set of [files](#), usually used for coordinating work among [programmers](#) collaboratively developing [source code](#) during [software development](#). Its goals include speed, [data integrity](#), and support for distributed, non-linear workflows (thousands of parallel branches running on different systems).

[\[10\]](#)[\[11\]](#)[\[12\]](#)

# Which files are most suitable?

## ☺ Text data

Python  
.py

R  
.r

Julia  
.jl

Plain text  
.txt

HTML  
.html

Markdown  
.md

LaTeX  
.tex

YAML  
.yaml

Jupyter  
.ipynb

# Which files are most suitable?

## 😊 Text data

Python  
.py

R  
.r

Julia  
.jl

Plain text  
.txt

HTML  
.html

Markdown  
.md

LaTeX  
.tex

YAML  
.yaml

Jupyter  
.ipynb

## 🤖 Binary/large data

Images  
.jpg/.png

Text data  
.csv/.tsv

Minified  
code

# Which files are most suitable?

## 😊 Text data

Python  
.py

R  
.r

Julia  
.jl

Plain text  
.txt

HTML  
.html

Markdown  
.md

LaTeX  
.tex

YAML  
.yaml

Jupyter  
.ipynb

## 🤖 Binary/large data

Images  
.jpg/.png

PDF  
.pdf

Software  
.exe

Text data  
.csv/.tsv

Audio  
.mp3/.wav

Video  
.mp4

Minified  
code

Office  
.docx/.xlsx

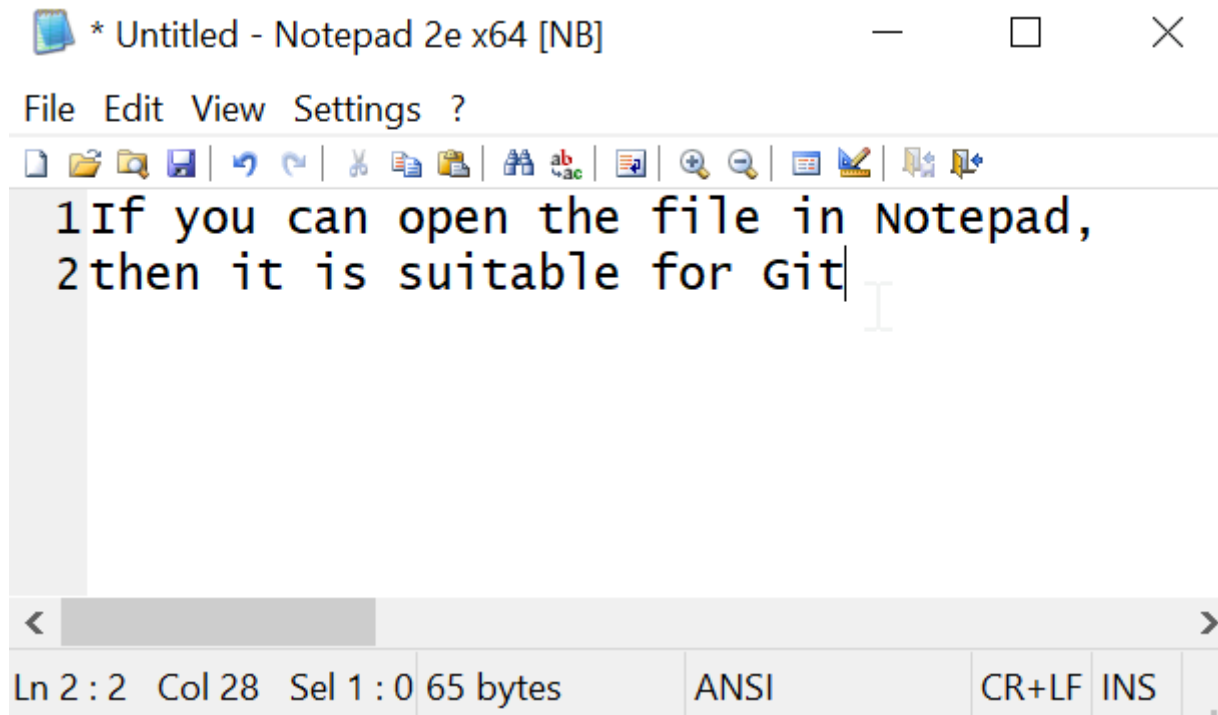
Archives  
.zip

# Which files are most suitable?

---

Rule of thumb

# Which files are most suitable?



# Why is Git important?

---



# Why is Git important?

-  Better reproducibility

# Why is Git important?

-  Better reproducibility
  - ▶  «I am using revision a3e264b»

# Why is Git important?

-  Better reproducibility
  - ▶  «I am using revision a3e264b»
-  History

# Why is Git important?

-  Better reproducibility
  - ▶  «I am using revision a3e264b»
-  History
  - ▶  «Who changed this line?»

# Why is Git important?

-  Better reproducibility
  - ▶  «I am using revision a3e264b»
-  History
  - ▶  «Who changed this line?» *git blame* 

# Why is Git important?

-  Better reproducibility
  - ▶  «I am using revision a3e264b»
-  History
  - ▶  «Who changed this line?» *git blame* 

# Why is Git important?

-  Better reproducibility
  - ▶  «I am using revision a3e264b»
-  History
  - ▶  «Who changed this line?» *git blame* 
  - ▶  «Can we undo that change?»

# Why is Git important?

-  Better reproducibility
  - ▶  «I am using revision a3e264b»
-  History
  - ▶  «Who changed this line?» *git blame* 
  - ▶  «Can we undo that change?» *git revert*

# Why is Git important?

-  Better reproducibility
  - ▶  «I am using revision a3e264b»
-  History
  - ▶  «Who changed this line?» *git blame* 
  - ▶  «Can we undo that change?» *git revert*
-  Better collaboration

# Why is Git important?

-  Better reproducibility
  - ▶  «I am using revision a3e264b»
-  History
  - ▶  «Who changed this line?» *git blame* 
  - ▶  «Can we undo that change?» *git revert*
-  Better collaboration
  - ▶  «What do you think about this change?»

# Why is Git important?

-  Better reproducibility
  - ▶  «I am using revision a3e264b»
-  History
  - ▶  «Who changed this line?» *git blame* 
  - ▶  «Can we undo that change?» *git revert*
-  Better collaboration
  - ▶  «What do you think about this change?»
  - ▶  «Your change is great!»

# Why is Git important?

-  Better reproducibility
  - ▶  «I am using revision a3e264b»
-  History
  - ▶  «Who changed this line?» *git blame* 
  - ▶  «Can we undo that change?» *git revert*
-  Better collaboration
  - ▶  «What do you think about this change?»
  - ▶  «Your change is great!»
  - ▶  «...but it needs some minor improvements»

# What is GitHub?

---

# What is GitHub?

## Git

From Wikipedia, the free encyclopedia

*For other uses, see [Git \(disambiguation\)](#).*

*Not to be confused with [GitHub](#).*

**Git** ([/ɡɪt/](#))<sup>[9]</sup> is software for tracking changes in any set of [files](#), usually used for coordinating work among [programmers](#) collaboratively developing [source code](#) during [software development](#). Its goals include speed, [data integrity](#), and support for distributed, non-linear workflows (thousands of parallel branches running on different systems).

[\[10\]](#)[\[11\]](#)[\[12\]](#)

# What is GitHub?

## Git

From Wikipedia, the free encyclopedia

*For other uses, see [Git \(disambiguation\)](#).*

*Not to be confused with [GitHub](#).*

**Git** ([/ɡɪt/](#))<sup>[9]</sup> is software for tracking changes in any set of [files](#), usually used for coordinating work among [programmers](#) collaboratively developing [source code](#) during [software development](#). Its goals include speed, [data integrity](#), and support for distributed, non-linear workflows (thousands of parallel branches running on different systems).

[\[10\]](#)[\[11\]](#)[\[12\]](#)

# Git vs GitHub.com



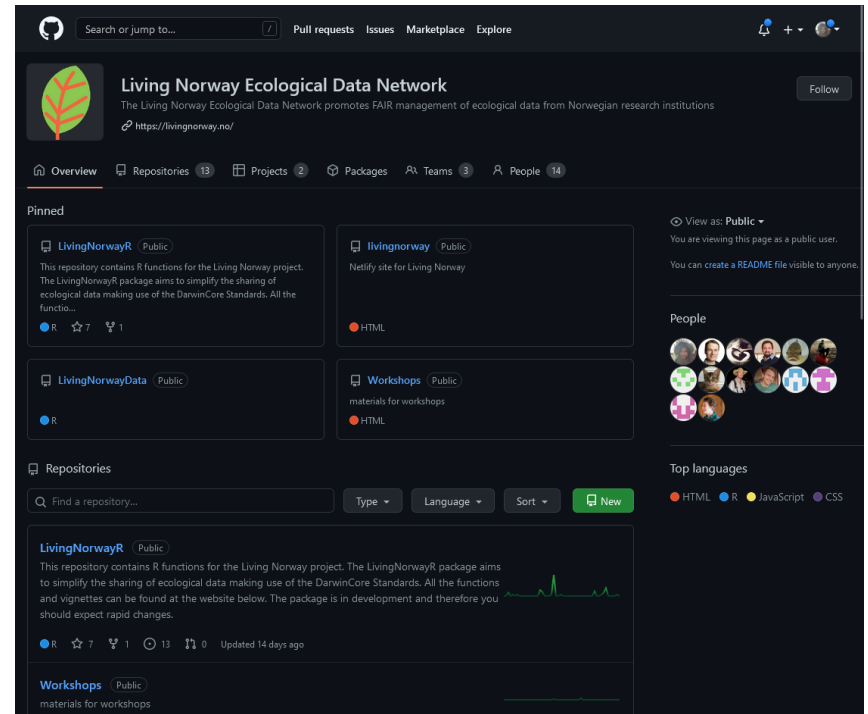
**Git is the software**



**GitHub is a service**

```
MINGW64; c:/Users/francesco.frassinelli/project
francesco.frassinelli@NINAPC756 MINGW64 ~/project (master)
$ git status
on branch master
changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   README.md

francesco.frassinelli@NINAPC756 MINGW64 ~/project (master)
$
```



# What does GitHub provides?

---

# What does GitHub provides?

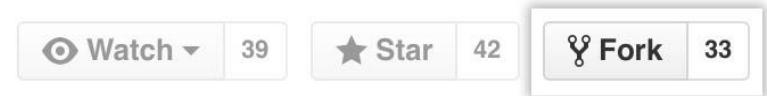
---

- Simplified collaboration using:

# What does GitHub provides?

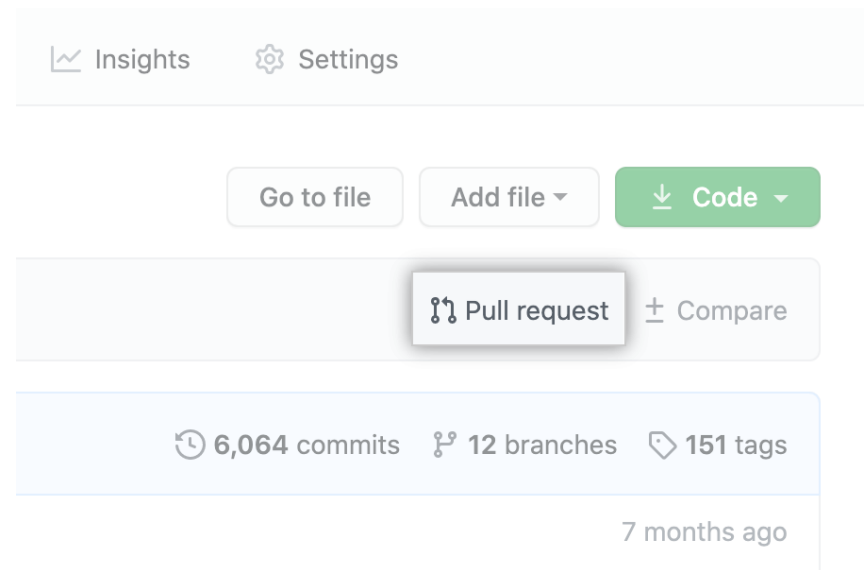
---

- Simplified collaboration using:
  - ▶ Forks



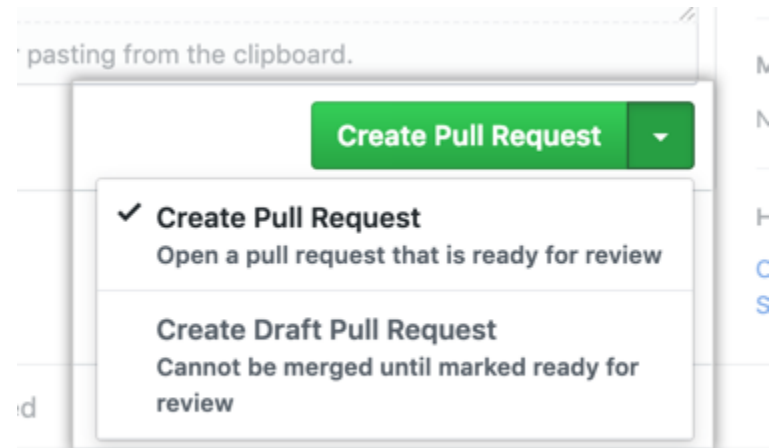
# What does GitHub provides?

- Simplified collaboration using:
  - ▶ Forks
  - ▶ Pull requests (PR)



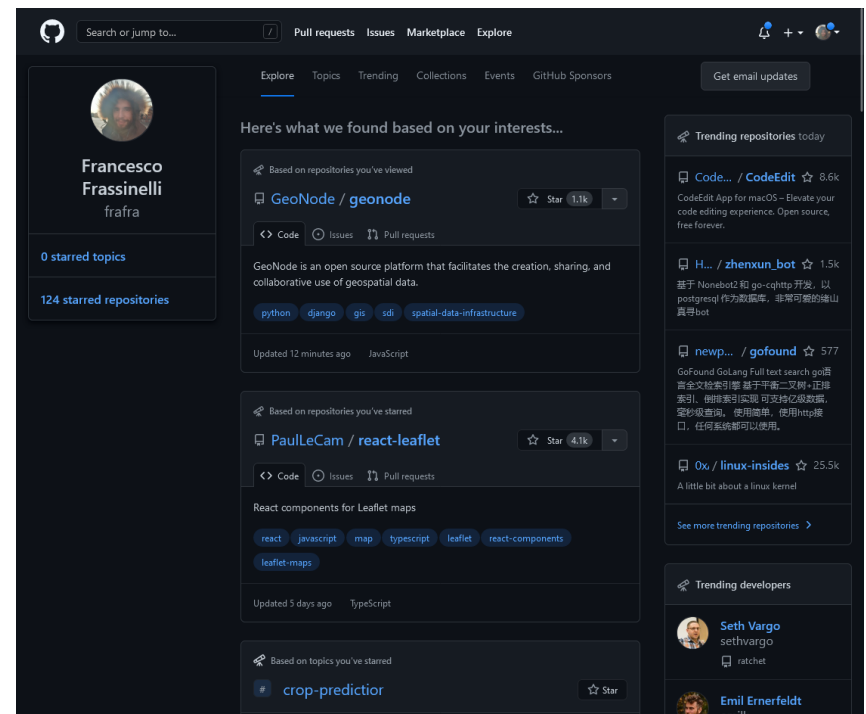
# What does GitHub provides?

- Simplified collaboration using:
  - ▶ Forks
  - ▶ Pull requests (PR)



# What does GitHub provides?

- Simplified collaboration using:
  - ▶ Forks
  - ▶ Pull requests (PR)
- Software catalog
  - ▶ Explore



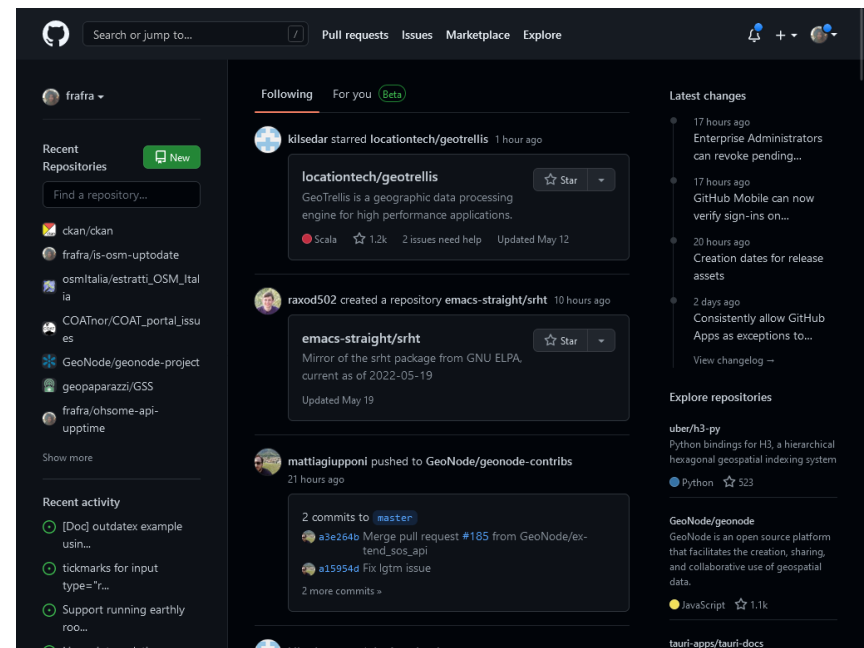
# What does GitHub provides?

- Simplified collaboration using:
  - ▶ Forks
  - ▶ Pull requests (PR)
- Software catalog
  - ▶ Explore
  - ▶ Favourites



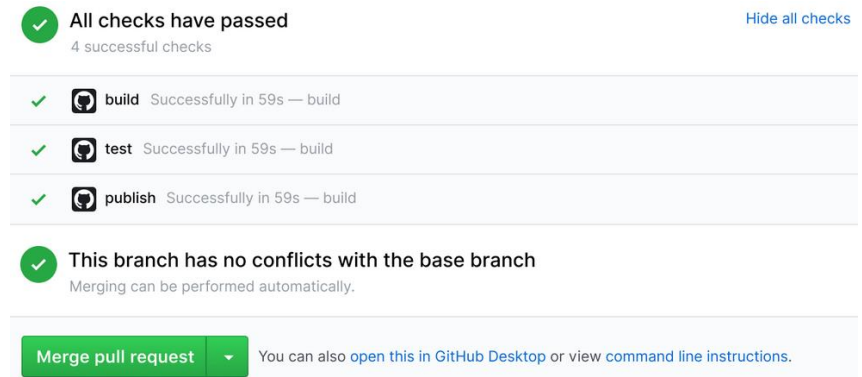
# What does GitHub provides?

- Simplified collaboration using:
  - ▶ Forks
  - ▶ Pull requests (PR)
- Software catalog
  - ▶ Explore
  - ▶ Favourites
  - ▶ Follow

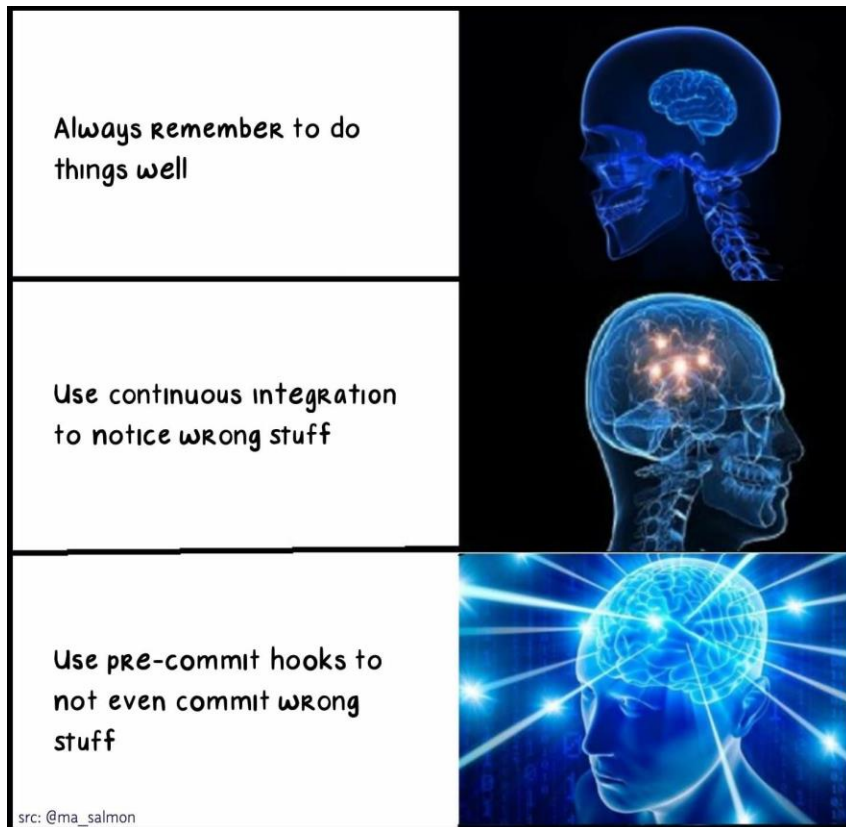


# What does GitHub provides?

- Simplified collaboration using:
  - ▶ Forks
  - ▶ Pull requests (PR)
- Software catalog
  - ▶ Explore
  - ▶ Favourites
  - ▶ Follow
- Continuous integration via GitHub Actions



# pre-commit



Main software:

- [pre-commit.com](https://pre-commit.com)

R-specific hooks:

- [lorenzwalther.github.io/precommit/](https://lorenzwalther.github.io/precommit/)

Automatic formatting:

- [styler.r-lib.org](https://styler.r-lib.org)

# | Questions?



Living Norway  
Ecological Data Network

# Suggested workflow

---

1.  Create a **new branch**

# Suggested workflow

1.  Create a **new branch**
2.  **Edit**

# Suggested workflow

1.  Create a **new branch**
2.  **Edit**
3.  **Commit** relevant changes only

# Suggested workflow

1.  Create a **new branch**
  2.  **Edit**
  3.  **Commit** relevant changes only
  4.  **Repeat**
- 

# Suggested workflow

1.  Create a **new branch**
  2.  **Edit**
  3.  **Commit** relevant changes only
  4.  **Repeat** until the branch is ready
- 

# Suggested workflow

1.  Create a **new branch**
2.  **Edit**
3.  **Commit** relevant changes only
4.  **Repeat** until the branch is ready
5.  Create a **pull request**

# Suggested workflow

1.  Create a **new branch**
2.  **Edit**
3.  **Commit** relevant changes only
4.  **Repeat** until the branch is ready
5.  Create a **pull request**
6.  Get some **feedback**

# Suggested workflow

1.  Create a **new branch**
2.  **Edit**
3.  **Commit** relevant changes only
4.  **Repeat** until the branch is ready
5.  Create a **pull request**
6.  Get some **feedback** (  or  )

# Suggested workflow

1.  Create a **new branch**
2.  **Edit**
3.  **Commit** relevant changes only
4.  **Repeat** until the branch is ready
5.  Create a **pull request**
6.  Get some **feedback** (  or  )
7.  **Merge**