

Git for
beginners

Gianfranco
Gallizia

Brief history
of Git.

Git basic
commands.

Alias and
other tricks.

Unified diff.

Branches,
merges and
conflicts.

Git for beginners

Module 1

Gianfranco Gallizia

2025



CC-BY-SA 4.0

Brief history of Git.

Version Control Systems

Git for
beginners

Gianfranco
Gallizia

Brief history
of Git.

Git basic
commands.

Alias and
other tricks.

Unified diff.

Branches,
merges and
conflicts.

Brief history of Git.

Version Control Systems

Git for
beginners

Gianfranco
Gallizia

Brief history
of Git.

Git basic
commands.

Alias and
other tricks.

Unified diff.

Branches,
merges and
conflicts.

Version Control System (VCS)

A set of software tools and procedures dedicated to keep an history of changes regarding one or more files.

Brief history of Git.

Version Control Systems

Git for
beginners

Gianfranco
Gallizia

Brief history
of Git.

Git basic
commands.

Alias and
other tricks.

Unified diff.

Branches,
merges and
conflicts.

Version Control System (VCS)

A set of software tools and procedures dedicated to keep an history of changes regarding one or more files.

For example

Brief history of Git.

Version Control Systems

Git for
beginners

Gianfranco
Gallizia

Brief history
of Git.

Git basic
commands.

Alias and
other tricks.

Unified diff.

Branches,
merges and
conflicts.

Version Control System (VCS)

A set of software tools and procedures dedicated to keep an history of changes regarding one or more files.

For example

- SCCS - Source Code Control System (1973).

Brief history of Git.

Version Control Systems

Git for
beginners

Gianfranco
Gallizia

Brief history
of Git.

Git basic
commands.

Alias and
other tricks.

Unified diff.

Branches,
merges and
conflicts.

Version Control System (VCS)

A set of software tools and procedures dedicated to keep an history of changes regarding one or more files.

For example

- SCCS - Source Code Control System (1973).
- RCS - Revision Control System (1982).

Brief history of Git.

Version Control Systems

Git for
beginners

Gianfranco
Gallizia

Brief history
of Git.

Git basic
commands.

Alias and
other tricks.

Unified diff.

Branches,
merges and
conflicts.

Version Control System (VCS)

A set of software tools and procedures dedicated to keep an history of changes regarding one or more files.

For example

- SCCS - Source Code Control System (1973).
- RCS - Revision Control System (1982).
- CVS - Concurrent Versions System (1990).

Brief history of Git.

Version Control Systems

Git for
beginners

Gianfranco
Gallizia

Brief history
of Git.

Git basic
commands.

Alias and
other tricks.

Unified diff.

Branches,
merges and
conflicts.

Version Control System (VCS)

A set of software tools and procedures dedicated to keep an history of changes regarding one or more files.

For example

- SCCS - Source Code Control System (1973).
- RCS - Revision Control System (1982).
- CVS - Concurrent Versions System (1990).
- SVN - Subversion (2000).

Brief history of Git.

Version Control Systems

Git for
beginners

Gianfranco
Gallizia

Brief history
of Git.

Git basic
commands.

Alias and
other tricks.

Unified diff.

Branches,
merges and
conflicts.

Version Control System (VCS)

A set of software tools and procedures dedicated to keep an history of changes regarding one or more files.

For example

- SCCS - Source Code Control System (1973).
- RCS - Revision Control System (1982).
- CVS - Concurrent Versions System (1990).
- SVN - Subversion (2000).
- Git (2005).

Brief history of Git.

Version Control Systems

Git for
beginners

Gianfranco
Gallizia

Brief history
of Git.

Git basic
commands.

Alias and
other tricks.

Unified diff.

Branches,
merges and
conflicts.

VCS	Percent
Git	93,87%
SVN	5,18%
None	4,31%
Mercurial	1,13%

Fonte: Stack Overflow Developer Survey 2022

Brief history of Git.

Version Control Systems

Git for
beginners

Gianfranco
Gallizia

Brief history
of Git.

Git basic
commands.

Alias and
other tricks.

Unified diff.

Branches,
merges and
conflicts.

Method of interaction	Percent
Command line	83,57%
Code Editor	54,49%
Web UI	28,44%
Dedicated GUI Application	26,37%

Fonte: Stack Overflow Developer Survey 2022

Brief history of Git.

Why Git?

Git for
beginners

Gianfranco
Gallizia

Brief history
of Git.

Git basic
commands.

Alias and
other tricks.

Unified diff.

Branches,
merges and
conflicts.

Brief history of Git.

Why Git?

Git for
beginners

Gianfranco
Gallizia

Brief history
of Git.

Git basic
commands.

Alias and
other tricks.

Unified diff.

Branches,
merges and
conflicts.

The origins

Git was created in 2005 by Linus Torvalds following his frustration regarding the previous VCSs and the controversies regarding BitKeeper.

Brief history of Git.

Why Git?

Git for
beginners

Gianfranco
Gallizia

Brief history
of Git.

Git basic
commands.

Alias and
other tricks.

Unified diff.

Branches,
merges and
conflicts.

Use case (1/2)

Support the development of the Linux kernel:

Brief history of Git.

Why Git?

Git for
beginners

Gianfranco
Gallizia

Brief history
of Git.

Git basic
commands.

Alias and
other tricks.

Unified diff.

Branches,
merges and
conflicts.

Use case (1/2)

Support the development of the Linux kernel:

- Huge project consisting of millions of lines of code.

Brief history of Git.

Why Git?

Git for
beginners

Gianfranco
Gallizia

Brief history
of Git.

Git basic
commands.

Alias and
other tricks.

Unified diff.

Branches,
merges and
conflicts.

Use case (1/2)

Support the development of the Linux kernel:

- Huge project consisting of millions of lines of code.
- Thousands of developers around the World.

Brief history of Git.

Why Git?

Git for
beginners

Gianfranco
Gallizia

Brief history
of Git.

Git basic
commands.

Alias and
other tricks.

Unified diff.

Branches,
merges and
conflicts.

Use case (1/2)

Support the development of the Linux kernel:

- Huge project consisting of millions of lines of code.
- Thousands of developers around the World.
- Everything is developed in public mailing lists.

Brief history of Git.

Why Git?

Git for
beginners

Gianfranco
Gallizia

Brief history
of Git.

Git basic
commands.

Alias and
other tricks.

Unified diff.

Branches,
merges and
conflicts.

Use case (2/2)

Support the development of the Linux kernel (cont.):

Brief history of Git.

Why Git?

Git for
beginners

Gianfranco
Gallizia

Brief history
of Git.

Git basic
commands.

Alias and
other tricks.

Unified diff.

Branches,
merges and
conflicts.

Use case (2/2)

Support the development of the Linux kernel (cont.):

- There's no real central authority.

Brief history of Git.

Why Git?

Git for
beginners

Gianfranco
Gallizia

Brief history
of Git.

Git basic
commands.

Alias and
other tricks.

Unified diff.

Branches,
merges and
conflicts.

Use case (2/2)

Support the development of the Linux kernel (cont.):

- There's no real central authority.
- There's no central HQ.

Brief history of Git.

Why Git?

Git for
beginners

Gianfranco
Gallizia

Brief history
of Git.

Git basic
commands.

Alias and
other tricks.

Unified diff.

Branches,
merges and
conflicts.

Use case (2/2)

Support the development of the Linux kernel (cont.):

- There's no real central authority.
- There's no central HQ.
- There's no working time.

Brief history of Git.

Why Git?

Git for
beginners

Gianfranco
Gallizia

Brief history
of Git.

Git basic
commands.

Alias and
other tricks.

Unified diff.

Branches,
merges and
conflicts.

Use case (2/2)

Support the development of the Linux kernel (cont.):

- There's no real central authority.
- There's no central HQ.
- There's no working time.
- There's no stable connection to a central server.

Brief history of Git.

Why Git?

Git for
beginners

Gianfranco
Gallizia

Brief history
of Git.

Git basic
commands.

Alias and
other tricks.

Unified diff.

Branches,
merges and
conflicts.

Summarizing

Brief history of Git.

Why Git?

Git for
beginners

Gianfranco
Gallizia

Brief history
of Git.

Git basic
commands.

Alias and
other tricks.

Unified diff.

Branches,
merges and
conflicts.

Summarizing

- No central server.

Brief history of Git.

Why Git?

Git for
beginners

Gianfranco
Gallizia

Brief history
of Git.

Git basic
commands.

Alias and
other tricks.

Unified diff.

Branches,
merges and
conflicts.

Summarizing

- No central server.
- The developers act independently and submit their changes to the maintainers through patches sent to the mailing list.

Brief history of Git.

Why Git?

Git for
beginners

Gianfranco
Gallizia

Brief history
of Git.

Git basic
commands.

Alias and
other tricks.

Unified diff.

Branches,
merges and
conflicts.

Summarizing

- No central server.
- The developers act independently and submit their changes to the maintainers through patches sent to the mailing list.
- All the changes are performed locally then propagated, conflicts are solved in the integration phase.

Brief history of Git.

Why Git?

Git for
beginners

Gianfranco
Gallizia

Brief history
of Git.

Git basic
commands.

Alias and
other tricks.

Unified diff.

Branches,
merges and
conflicts.

Summarizing

- No central server.
- The developers act independently and submit their changes to the maintainers through patches sent to the mailing list.
- All the changes are performed locally then propagated, conflicts are solved in the integration phase.
- The system has been designed by the lead maintainer of the Linux kernel and it's closer to a filesystem rather than a database of changes.

Git basic commands.

Create an empty repository.

Git for
beginners

Gianfranco
Gallizia

Brief history
of Git.

**Git basic
commands.**

Alias and
other tricks.

Unified diff.

Branches,
merges and
conflicts.

Git basic commands.

Create an empty repository.

Git for
beginners

Gianfranco
Gallizia

Brief history
of Git.

Git basic
commands.

Alias and
other tricks.

Unified diff.

Branches,
merges and
conflicts.

Repository

Set of files and related changes plus corresponding metadata (author, date and time, optional file metadata, etc. etc.).

The inner structure of a repository is the main differentiation between VCSs.

Git basic commands.

Create an empty repository.

Git for
beginners

Gianfranco
Gallizia

Brief history
of Git.

**Git basic
commands.**

Alias and
other tricks.

Unified diff.

Branches,
merges and
conflicts.

Create an empty repository in the current directory:

```
git init
```

Git basic commands.

Create an empty repository.

Git for
beginners

Gianfranco
Gallizia

Brief history
of Git.

Git basic
commands.

Alias and
other tricks.

Unified diff.

Branches,
merges and
conflicts.

Create an empty repository in the current directory:

```
git init
```

Create an empty repository in the specified directory:

```
git init myrepo
```

Git basic commands.

Create an empty repository.

Git for
beginners

Gianfranco
Gallizia

Brief history
of Git.

Git basic
commands.

Alias and
other tricks.

Unified diff.

Branches,
merges and
conflicts.

Create an empty repository in the current directory:

```
git init
```

Create an empty repository in the specified directory:

```
git init myrepo
```

Set the name of the main branch:

```
git init --initial-branch=trunk
```


Git basic commands.

Getting ready for the first commit.

Git for
beginners

Gianfranco
Gallizia

Brief history
of Git.

**Git basic
commands.**

Alias and
other tricks.

Unified diff.

Branches,
merges and
conflicts.

Git basic commands.

Getting ready for the first commit.

Git for
beginners

Gianfranco
Gallizia

Brief history
of Git.

**Git basic
commands.**

Alias and
other tricks.

Unified diff.

Branches,
merges and
conflicts.

Set your name:

```
git config user.name "Fester Addams"
```

Git basic commands.

Getting ready for the first commit.

Git for
beginners

Gianfranco
Gallizia

Brief history
of Git.

Git basic
commands.

Alias and
other tricks.

Unified diff.

Branches,
merges and
conflicts.

Set your name:

```
git config user.name "Fester Addams"
```

Set your email:

```
git config user.email "fester.addams@example.com"
```

Git basic commands.

Getting ready for the first commit.

Git for
beginners

Gianfranco
Gallizia

Brief history
of Git.

Git basic
commands.

Alias and
other tricks.

Unified diff.

Branches,
merges and
conflicts.

Set your name:

```
git config user.name "Fester Addams"
```

Set your email:

```
git config user.email "fester.addams@example.com"
```

Setting both for each repository created by our user:

```
git config --global user.name "Fester Addams"  
git config --global user.email \  
"fester.addams@example.com"
```

Git basic commands.

Interlude: staging area

Git for
beginners

Gianfranco
Gallizia

Brief history
of Git.

**Git basic
commands.**

Alias and
other tricks.

Unified diff.

Branches,
merges and
conflicts.

Git basic commands.

Interlude: staging area

Git for
beginners

Gianfranco
Gallizia

Brief history
of Git.

Git basic
commands.

Alias and
other tricks.

Unified diff.

Branches,
merges and
conflicts.

Staging area

Changes in git are not added directly to the repository, instead they are added to a buffer file (called *INDEX*) before being added to the commit tree.

The content of the INDEX together with metadata (author, timestamp, message, etc.) and the SHA-1 hash of both INDEX and metadata will become the commit.

Git basic commands.

Interlude: staging area

Git for
beginners

Gianfranco
Gallizia

Brief history
of Git.

Git basic
commands.

Alias and
other tricks.

Unified diff.

Branches,
merges and
conflicts.

Staging area

Changes in git are not added directly to the repository, instead they are added to a buffer file (called *INDEX*) before being added to the commit tree.

The content of the INDEX together with metadata (author, timestamp, message, etc.) and the SHA-1 hash of both INDEX and metadata will become the commit.

From that moment on, Git will treat them as read-only data and the only way to modify them will be to destroy the commit and recreate it, altering the history of the repository and **making the modification obvious in the case of commits that have already been distributed to others.**

Git basic commands.

Getting ready for the first commit.

Git for
beginners

Gianfranco
Gallizia

Brief history
of Git.

**Git basic
commands.**

Alias and
other tricks.

Unified diff.

Branches,
merges and
conflicts.

Git basic commands.

Getting ready for the first commit.

Git for
beginners

Gianfranco
Gallizia

Brief history
of Git.

Git basic
commands.

Alias and
other tricks.

Unified diff.

Branches,
merges and
conflicts.

Adding the files to the staging area:

```
git add README.md src/main.c python/test.py
```

Git basic commands.

Getting ready for the first commit.

Git for
beginners

Gianfranco
Gallizia

Brief history
of Git.

Git basic
commands.

Alias and
other tricks.

Unified diff.

Branches,
merges and
conflicts.

Adding the files to the staging area:

```
git add README.md src/main.c python/test.py
```

Analyze the situation:

```
git status
```

Git basic commands.

Getting ready for the first commit.

Git for
beginners

Gianfranco
Gallizia

Brief history
of Git.

Git basic
commands.

Alias and
other tricks.

Unified diff.

Branches,
merges and
conflicts.

Adding the files to the staging area:

```
git add README.md src/main.c python/test.py
```

Analyze the situation:

```
git status
```

Perform the first commit:

```
git commit -m "Initial commit."
```

Git basic commands.

From the second commit onward.

Git for
beginners

Gianfranco
Gallizia

Brief history
of Git.

**Git basic
commands.**

Alias and
other tricks.

Unified diff.

Branches,
merges and
conflicts.

Git basic commands.

From the second commit onward.

Git for
beginners

Gianfranco
Gallizia

Brief history
of Git.

**Git basic
commands.**

Alias and
other tricks.

Unified diff.

Branches,
merges and
conflicts.

Workflow (1/2)

Git basic commands.

From the second commit onward.

Git for
beginners

Gianfranco
Gallizia

Brief history
of Git.

**Git basic
commands.**

Alias and
other tricks.

Unified diff.

Branches,
merges and
conflicts.

Workflow (1/2)

- 1 Make some changes.

Git basic commands.

From the second commit onward.

Git for
beginners

Gianfranco
Gallizia

Brief history
of Git.

Git basic
commands.

Alias and
other tricks.

Unified diff.

Branches,
merges and
conflicts.

Workflow (1/2)

- 1 Make some changes.
- 2 Analyze the situation with:
`git status`

Git basic commands.

From the second commit onward.

Git for
beginners

Gianfranco
Gallizia

Brief history
of Git.

Git basic
commands.

Alias and
other tricks.

Unified diff.

Branches,
merges and
conflicts.

Workflow (1/2)

- 1 Make some changes.
- 2 Analyze the situation with:
`git status`
- 3 Add some changes selectively:
`git add -i -p`

Git basic commands.

From the second commit onward.

Git for
beginners

Gianfranco
Gallizia

Brief history
of Git.

Git basic
commands.

Alias and
other tricks.

Unified diff.

Branches,
merges and
conflicts.

Workflow (1/2)

- 1 Make some changes.
- 2 Analyze the situation with:
`git status`
- 3 Add some changes selectively:
`git add -i -p`
- 4 Review the situation:
`git status`
`git diff`
`git diff --staged`

Git basic commands.

From the second commit onward.

Git for
beginners

Gianfranco
Gallizia

Brief history
of Git.

**Git basic
commands.**

Alias and
other tricks.

Unified diff.

Branches,
merges and
conflicts.

Workflow (2/2)

Git basic commands.

From the second commit onward.

Git for
beginners

Gianfranco
Gallizia

Brief history
of Git.

Git basic
commands.

Alias and
other tricks.

Unified diff.

Branches,
merges and
conflicts.

Workflow (2/2)

- 5 Perform the partial commit (opens \$EDITOR):

```
git commit
```

Git basic commands.

From the second commit onward.

Git for
beginners

Gianfranco
Gallizia

Brief history
of Git.

Git basic
commands.

Alias and
other tricks.

Unified diff.

Branches,
merges and
conflicts.

Workflow (2/2)

- 5 Perform the partial commit (opens \$EDITOR):

```
git commit
```

- 6 Add and commit all the changes (**dangerous**):

```
git add -A  
git commit -m "YOLO!"
```

Git basic commands.

Reviewing what has been done so far.

Git for
beginners

Gianfranco
Gallizia

Brief history
of Git.

**Git basic
commands.**

Alias and
other tricks.

Unified diff.

Branches,
merges and
conflicts.

Git basic commands.

Reviewing what has been done so far.

Git for
beginners

Gianfranco
Gallizia

Brief history
of Git.

**Git basic
commands.**

Alias and
other tricks.

Unified diff.

Branches,
merges and
conflicts.

Read the commits with their relative messages:

```
git log
```

Git basic commands.

Reviewing what has been done so far.

Git for
beginners

Gianfranco
Gallizia

Brief history
of Git.

Git basic
commands.

Alias and
other tricks.

Unified diff.

Branches,
merges and
conflicts.

Read the commits with their relative messages:

```
git log
```

Compact form log:

```
git log --oneline
```

Git basic commands.

Reviewing what has been done so far.

Git for
beginners

Gianfranco
Gallizia

Brief history
of Git.

Git basic
commands.

Alias and
other tricks.

Unified diff.

Branches,
merges and
conflicts.

Read the commits with their relative messages:

```
git log
```

Compact form log:

```
git log --oneline
```

Single file log:

```
git log -- README.md
```


Git basic commands.

Reviewing what has been done so far.

Git for
beginners

Gianfranco
Gallizia

Brief history
of Git.

Git basic
commands.

Alias and
other tricks.

Unified diff.

Branches,
merges and
conflicts.

Analyze in detail the changes made in the last commit (HEAD):

```
git show
```

```
git show HEAD
```

Git basic commands.

Reviewing what has been done so far.

Git for
beginners

Gianfranco
Gallizia

Brief history
of Git.

Git basic
commands.

Alias and
other tricks.

Unified diff.

Branches,
merges and
conflicts.

Analyze in detail the changes made in the last commit (HEAD):

```
git show  
git show HEAD
```

Analyze in detail the changes made in the second-to-last commit:

```
git show HEAD~1  
git show HEAD~1
```

Git basic commands.

Reviewing what has been done so far.

Git for
beginners

Gianfranco
Gallizia

Brief history
of Git.

Git basic
commands.

Alias and
other tricks.

Unified diff.

Branches,
merges and
conflicts.

Analyze a file row-by-row with commit annotations:

```
git blame -- README.md
```

Git basic commands.

Reviewing what has been done so far.

Git for
beginners

Gianfranco
Gallizia

Brief history
of Git.

Git basic
commands.

Alias and
other tricks.

Unified diff.

Branches,
merges and
conflicts.

Analyze a file row-by-row with commit annotations:

```
git blame -- README.md
```

Same as before with author's email instead of author's name:

```
git blame -e -- README.md
```

Alias and other tricks.

Configuration files.

Git for
beginners

Gianfranco
Gallizia

Brief history
of Git.

Git basic
commands.

**Alias and
other tricks.**

Unified diff.

Branches,
merges and
conflicts.

Alias and other tricks.

Configuration files.

Git for
beginners

Gianfranco
Gallizia

Brief history
of Git.

Git basic
commands.

Alias and
other tricks.

Unified diff.

Branches,
merges and
conflicts.

Configuration files

There are three default places where git searches its configuration files:

Alias and other tricks.

Configuration files.

Git for
beginners

Gianfranco
Gallizia

Brief history
of Git.

Git basic
commands.

Alias and
other tricks.

Unified diff.

Branches,
merges and
conflicts.

Configuration files

There are three default places where git searches its configuration files:

- `$GIT_DIR/config`: valid for the current repository.

Alias and other tricks.

Configuration files.

Git for
beginners

Gianfranco
Gallizia

Brief history
of Git.

Git basic
commands.

Alias and
other tricks.

Unified diff.

Branches,
merges and
conflicts.

Configuration files

There are three default places where git searches its configuration files:

- `$GIT_DIR/config`: valid for the current repository.
- `$HOME/.gitconfig`: valid for the current user.

Alias and other tricks.

Configuration files.

Git for
beginners

Gianfranco
Gallizia

Brief history
of Git.

Git basic
commands.

Alias and
other tricks.

Unified diff.

Branches,
merges and
conflicts.

Configuration files

There are three default places where git searches its configuration files:

- `$GIT_DIR/config`: valid for the current repository.
- `$HOME/.gitconfig`: valid for the current user.
- `/etc/gitconfig`: valid for all the users of the local machine.

Alias and other tricks.

Configuration files.

Git for
beginners

Gianfranco
Gallizia

Brief history
of Git.

Git basic
commands.

Alias and
other tricks.

Unified diff.

Branches,
merges and
conflicts.

Configuration files

There are three default places where git searches its configuration files:

- `$GIT_DIR/config`: valid for the current repository.
- `$HOME/.gitconfig`: valid for the current user.
- `/etc/gitconfig`: valid for all the users of the local machine.

Nota bene

If `/etc/gitconfig` or `$HOME/.gitconfig` do not exist git config does not return any error.
`$GIT_DIR/config` is created by `git init`.

Alias and other tricks.

Configuration files.

Git for
beginners

Gianfranco
Gallizia

Brief history
of Git.

Git basic
commands.

Alias and
other tricks.

Unified diff.

Branches,
merges and
conflicts.

Configuration files format

Git configuration files are formatted like INI files.

They are divided in sections delimited by:

```
[section name]
```

options are written one per line in the following format:

```
option = value
```

Alias and other tricks.

Configuration example.

Git for
beginners

Gianfranco
Gallizia

Brief history
of Git.

Git basic
commands.

Alias and
other tricks.

Unified diff.

Branches,
merges and
conflicts.

```
$HOME/.gitconfig
```

```
[user]
```

```
name = Fester Addams
```

```
email = fester.addams@example.com
```

Alias and other tricks.

Ignoring files.

Git for
beginners

Gianfranco
Gallizia

Brief history
of Git.

Git basic
commands.

**Alias and
other tricks.**

Unified diff.

Branches,
merges and
conflicts.

Alias and other tricks.

Ignoring files.

Git for
beginners

Gianfranco
Gallizia

Brief history
of Git.

Git basic
commands.

Alias and
other tricks.

Unified diff.

Branches,
merges and
conflicts.

*.o *.pyc and other binaries

There are some files that we **DO NOT** want to place under git control:

Alias and other tricks.

Ignoring files.

Git for
beginners

Gianfranco
Gallizia

Brief history
of Git.

Git basic
commands.

Alias and
other tricks.

Unified diff.

Branches,
merges and
conflicts.

*.o *.pyc and other binaries

There are some files that we **DO NOT** want to place under git control:

- Compiler's output (*.o)

Alias and other tricks.

Ignoring files.

Git for
beginners

Gianfranco
Gallizia

Brief history
of Git.

Git basic
commands.

Alias and
other tricks.

Unified diff.

Branches,
merges and
conflicts.

*.o *.pyc and other binaries

There are some files that we **DO NOT** want to place under git control:

- Compiler's output (*.o)
- Python Bytecode (*.pyc)

Alias and other tricks.

Ignoring files.

Git for
beginners

Gianfranco
Gallizia

Brief history
of Git.

Git basic
commands.

Alias and
other tricks.

Unified diff.

Branches,
merges and
conflicts.

*.o *.pyc and other binaries

There are some files that we **DO NOT** want to place under git control:

- Compiler's output (*.o)
- Python Bytecode (*.pyc)
- Whole directories recreated by the build system (./bin, ./build, ...)

Alias and other tricks.

Ignoring files.

Git for
beginners

Gianfranco
Gallizia

Brief history
of Git.

Git basic
commands.

Alias and
other tricks.

Unified diff.

Branches,
merges and
conflicts.

`*.o *.pyc` and other binaries

We can add a file called `.gitignore` to the working directory and write in that file the patterns to ignore.

Alias and other tricks.

Ignoring files.

Git for
beginners

Gianfranco
Gallizia

Brief history
of Git.

Git basic
commands.

Alias and
other tricks.

Unified diff.

Branches,
merges and
conflicts.

```
.gitignore
```

```
# Ignore all .pyc
# (regardless where they are)
*.pyc

# Ignore all .o
*.o

# Ignore bin and all its content
bin/
```

Alias and other tricks.

Alias

Git for
beginners

Gianfranco
Gallizia

Brief history
of Git.

Git basic
commands.

**Alias and
other tricks.**

Unified diff.

Branches,
merges and
conflicts.

Alias and other tricks.

Alias

Git for
beginners

Gianfranco
Gallizia

Brief history
of Git.

Git basic
commands.

Alias and
other tricks.

Unified diff.

Branches,
merges and
conflicts.

Alias

Aliases in git are like aliases in bash: sequences of characters that are expanded into predefined sequences of commands plus flags.

Alias and other tricks.

Alias

Git for
beginners

Gianfranco
Gallizia

Brief history
of Git.

Git basic
commands.

Alias and
other tricks.

Unified diff.

Branches,
merges and
conflicts.

Alias

Aliases in git are like aliases in bash: sequences of characters that are expanded into predefined sequences of commands plus flags.

Shorten `git checkout` in `git co`:

```
git config --global alias.co checkout
```

Alias and other tricks.

Alias

Git for
beginners

Gianfranco
Gallizia

Brief history
of Git.

Git basic
commands.

Alias and
other tricks.

Unified diff.

Branches,
merges and
conflicts.

Aliases in .gitconfig

It is possible to edit the `$HOME/.gitconfig` file instead of issuing the `git config` command in order to add aliases or to modify git options.

Alias and other tricks.

Alias

Git for
beginners

Gianfranco
Gallizia

Brief history
of Git.

Git basic
commands.

Alias and
other tricks.

Unified diff.

Branches,
merges and
conflicts.

Aliases in .gitconfig

It is possible to edit the `$HOME/.gitconfig` file instead of issuing the `git config` command in order to add aliases or to modify git options.

E.G.:

```
[user]
name = Fester Addams
email = fester.addams@example.com
[init]
defaultBranch = main
[alias]
co = checkout
ci = commit
```


Alias and other tricks.

Cleaning up.

Git for
beginners

Gianfranco
Gallizia

Brief history
of Git.

Git basic
commands.

**Alias and
other tricks.**

Unified diff.

Branches,
merges and
conflicts.

Alias and other tricks.

Cleaning up.

Git for
beginners

Gianfranco
Gallizia

Brief history
of Git.

Git basic
commands.

Alias and
other tricks.

Unified diff.

Branches,
merges and
conflicts.

Restoring a file to the latest commit:

```
git restore -- README.md
```

Alias and other tricks.

Cleaning up.

Git for
beginners

Gianfranco
Gallizia

Brief history
of Git.

Git basic
commands.

Alias and
other tricks.

Unified diff.

Branches,
merges and
conflicts.

Restoring a file to the latest commit:

```
git restore -- README.md
```

Restoring all the files in the current working directory:

```
git restore .
```

Alias and other tricks.

Cleaning up.

Git for
beginners

Gianfranco
Gallizia

Brief history
of Git.

Git basic
commands.

Alias and
other tricks.

Unified diff.

Branches,
merges and
conflicts.

Restoring a file to the latest commit:

```
git restore -- README.md
```

Restoring all the files in the current working directory:

```
git restore .
```

Restoring all the Python source files:

```
git restore '*.py'
```

Alias e altri trucchi.

Fare pulizia

Git for
beginners

Gianfranco
Gallizia

Brief history
of Git.

Git basic
commands.

Alias and
other tricks.

Unified diff.

Branches,
merges and
conflicts.

Restoring the whole working directory:

```
git restore :/
```

Alias and other tricks.

Cleaning up.

Git for
beginners

Gianfranco
Gallizia

Brief history
of Git.

Git basic
commands.

Alias and
other tricks.

Unified diff.

Branches,
merges and
conflicts.

`git clean`

Sometimes you realize that your working directory is too deteriorated and you want to clean all the changes made by an automated tool or a compiler.

Other than `git restore` there's the `git clean` command.

Alias and other tricks.

Cleaning up.

Git for
beginners

Gianfranco
Gallizia

Brief history
of Git.

Git basic
commands.

Alias and
other tricks.

Unified diff.

Branches,
merges and
conflicts.

`git clean`

Sometimes you realize that your working directory is too deteriorated and you want to clean all the changes made by an automated tool or a compiler.

Other than `git restore` there's the `git clean` command.

CAUTION!

`git clean` deletes the files in the working directory that are **NOT** under git control.

Alias and other tricks.

Cleaning up.

Git for
beginners

Gianfranco
Gallizia

Brief history
of Git.

Git basic
commands.

**Alias and
other tricks.**

Unified diff.

Branches,
merges and
conflicts.

Dry run (does not delete):

```
git clean -n
```


Alias and other tricks.

Cleaning up.

Git for
beginners

Gianfranco
Gallizia

Brief history
of Git.

Git basic
commands.

Alias and
other tricks.

Unified diff.

Branches,
merges and
conflicts.

Dry run (does not delete):

```
git clean -n
```

Starting an interactive cleaning session:

```
git clean -i
```

Alias and other tricks.

Cleaning up.

Git for
beginners

Gianfranco
Gallizia

Brief history
of Git.

Git basic
commands.

Alias and
other tricks.

Unified diff.

Branches,
merges and
conflicts.

Dry run (does not delete):

```
git clean -n
```

Starting an interactive cleaning session:

```
git clean -i
```

Force delete (**very dangerous**):

```
git clean -f
```

```
#Also delete files present in .gitignore
```

```
git clean -f -x
```

Unified diff.

Unified diff format.

Git for
beginners

Gianfranco
Gallizia

Brief history
of Git.

Git basic
commands.

Alias and
other tricks.

Unified diff.

Branches,
merges and
conflicts.

Unified diff.

Unified diff format.

Git for
beginners

Gianfranco
Gallizia

Brief history
of Git.

Git basic
commands.

Alias and
other tricks.

Unified diff.

Branches,
merges and
conflicts.

Comparing files

One of the most common operations performed with a VCS is comparing the differences between two revisions of one or more files.

Git uses the unified diff format, the same format you get through the `diff -u file1 file2` command and the same input format for the `patch` command.

Unified diff.

Unified diff format.

Git for
beginners

Gianfranco
Gallizia

Brief history
of Git.

Git basic
commands.

Alias and
other tricks.

Unified diff.

Branches,
merges and
conflicts.

Header

Unified diffs start with a variable-length header. The header ends with the following two lines:

```
--- original_file  
+++ new_file
```

Unified diff.

Unified diff format.

Git for
beginners

Gianfranco
Gallizia

Brief history
of Git.

Git basic
commands.

Alias and
other tricks.

Unified diff.

Branches,
merges and
conflicts.

Hunks

After the header there are the blocks of modified text (hunks).
A hunk has the following header:

```
@@ -line_number,length +line_number,length @@ opt
```

Unified diff.

Unified diff format.

Git for
beginners

Gianfranco
Gallizia

Brief history
of Git.

Git basic
commands.

Alias and
other tricks.

Unified diff.

Branches,
merges and
conflicts.

Hunks

After the header there are the blocks of modified text (hunks).
A hunk has the following header:

```
@@ -line_number,length +line_number,length @@ opt
```

E.G.:

```
@@ -565,6 +565,45 @@ git clean -f -x
```

Unified diff.

Unified diff format.

Git for
beginners

Gianfranco
Gallizia

Brief history
of Git.

Git basic
commands.

Alias and
other tricks.

Unified diff.

Branches,
merges and
conflicts.

Hunks (cont.)

After the hunk header there are the block lines. Rows starting with a - have been removed, those starting with a + have been added and those that start with a space are context lines (unchanged between the two revisions).

Unified diff.

Unified diff format.

Git for
beginners

Gianfranco
Gallizia

Brief history
of Git.

Git basic
commands.

Alias and
other tricks.

Unified diff.

Branches,
merges and
conflicts.

Hunks (end)

```
I'll stay here.  
-I'm going.  
+I'll arrive.
```

Unified diff.

Unified diff format.

Git for
beginners

Gianfranco
Gallizia

Brief history
of Git.

Git basic
commands.

Alias and
other tricks.

Unified diff.

Branches,
merges and
conflicts.

Putting it all together:

All I'm writing here will be ignored by patch.

Here comes the important part:

```
--- a/test.txt
+++ b/test.txt
@@ -1,2 +1,2 @@
    I'll stay here.
-I'm going.
+I'll arrive.
```

Unified diff.

Examples.

Git for
beginners

Gianfranco
Gallizia

Brief history
of Git.

Git basic
commands.

Alias and
other tricks.

Unified diff.

Branches,
merges and
conflicts.

Other than `git diff` (and `git show`) git uses the unified diff format also in the following:

Unified diff.

Examples.

Git for
beginners

Gianfranco
Gallizia

Brief history
of Git.

Git basic
commands.

Alias and
other tricks.

Unified diff.

Branches,
merges and
conflicts.

Other than `git diff` (and `git show`) git uses the unified diff format also in the following:

- `git add -i -p`: Add portions of files to the staging area.

Unified diff.

Examples.

Git for
beginners

Gianfranco
Gallizia

Brief history
of Git.

Git basic
commands.

Alias and
other tricks.

Unified diff.

Branches,
merges and
conflicts.

Other than `git diff` (and `git show`) git uses the unified diff format also in the following:

- `git add -i -p`: Add portions of files to the staging area.
- `git restore -p`: Restore portions of files from the last commit.

Unified diff.

Examples.

Git for
beginners

Gianfranco
Gallizia

Brief history
of Git.

Git basic
commands.

Alias and
other tricks.

Unified diff.

Branches,
merges and
conflicts.

Other than `git diff` (and `git show`) git uses the unified diff format also in the following:

- `git add -i -p`: Add portions of files to the staging area.
- `git restore -p`: Restore portions of files from the last commit.
- `git checkout -p`: Recover portions of files from a previous commit.

Branches, merges and conflicts.

Development branches.

Git for
beginners

Gianfranco
Gallizia

Brief history
of Git.

Git basic
commands.

Alias and
other tricks.

Unified diff.

**Branches,
merges and
conflicts.**

Branches, merges and conflicts.

Development branches.

Git for
beginners

Gianfranco
Gallizia

Brief history
of Git.

Git basic
commands.

Alias and
other tricks.

Unified diff.

Branches,
merges and
conflicts.

Development branches

Software development is not always a linear activity, sometimes there's the need to detach temporarily or permanently from the main development line in order to progress from a common code base.

Branches, merges and conflicts.

Development branches.

Git for
beginners

Gianfranco
Gallizia

Brief history
of Git.

Git basic
commands.

Alias and
other tricks.

Unified diff.

Branches,
merges and
conflicts.

Development branches

Software development is not always a linear activity, sometimes there's the need to detach temporarily or permanently from the main development line in order to progress from a common code base.

E.G.:

Branches, merges and conflicts.

Development branches.

Git for
beginners

Gianfranco
Gallizia

Brief history
of Git.

Git basic
commands.

Alias and
other tricks.

Unified diff.

Branches,
merges and
conflicts.

Development branches

Software development is not always a linear activity, sometimes there's the need to detach temporarily or permanently from the main development line in order to progress from a common code base.

E.G.:

- Adding a new feature not retro-compatible.

Branches, merges and conflicts.

Development branches.

Git for
beginners

Gianfranco
Gallizia

Brief history
of Git.

Git basic
commands.

Alias and
other tricks.

Unified diff.

Branches,
merges and
conflicts.

Development branches

Software development is not always a linear activity, sometimes there's the need to detach temporarily or permanently from the main development line in order to progress from a common code base.

E.G.:

- Adding a new feature not retro-compatible.
- Creating a new stable release that will not see new features but only bugfixes.

Branches, merges and conflicts.

Development branches.

Git for
beginners

Gianfranco
Gallizia

Brief history
of Git.

Git basic
commands.

Alias and
other tricks.

Unified diff.

Branches,
merges and
conflicts.

Development branches

Software development is not always a linear activity, sometimes there's the need to detach temporarily or permanently from the main development line in order to progress from a common code base.

E.G.:

- Adding a new feature not retro-compatible.
- Creating a new stable release that will not see new features but only bugfixes.
- Integrating conflicting changes from other developers.

Branches, merges and conflicts.

Structure of a commit.

Git for
beginners

Gianfranco
Gallizia

Brief history
of Git.

Git basic
commands.

Alias and
other tricks.

Unified diff.

**Branches,
merges and
conflicts.**

Branches, merges and conflicts.

Structure of a commit.

Git for
beginners

Gianfranco
Gallizia

Brief history
of Git.

Git basic
commands.

Alias and
other tricks.

Unified diff.

Branches,
merges and
conflicts.

A git commit is a data structure containing (among other things):

Branches, merges and conflicts.

Structure of a commit.

Git for
beginners

Gianfranco
Gallizia

Brief history
of Git.

Git basic
commands.

Alias and
other tricks.

Unified diff.

Branches,
merges and
conflicts.

A git commit is a data structure containing (among other things):

- The author's name.

Branches, merges and conflicts.

Structure of a commit.

Git for
beginners

Gianfranco
Gallizia

Brief history
of Git.

Git basic
commands.

Alias and
other tricks.

Unified diff.

Branches,
merges and
conflicts.

A git commit is a data structure containing (among other things):

- The author's name.
- The author's email.

Branches, merges and conflicts.

Structure of a commit.

Git for
beginners

Gianfranco
Gallizia

Brief history
of Git.

Git basic
commands.

Alias and
other tricks.

Unified diff.

Branches,
merges and
conflicts.

A git commit is a data structure containing (among other things):

- The author's name.
- The author's email.
- Date and time of the commit's creation.

Branches, merges and conflicts.

Structure of a commit.

Git for
beginners

Gianfranco
Gallizia

Brief history
of Git.

Git basic
commands.

Alias and
other tricks.

Unified diff.

Branches,
merges and
conflicts.

A git commit is a data structure containing (among other things):

- The author's name.
- The author's email.
- Date and time of the commit's creation.
- A pointer to a snapshot of the repository at that moment.

Branches, merges and conflicts.

Structure of a commit.

Git for
beginners

Gianfranco
Gallizia

Brief history
of Git.

Git basic
commands.

Alias and
other tricks.

Unified diff.

Branches,
merges and
conflicts.

A git commit is a data structure containing (among other things):

- The author's name.
- The author's email.
- Date and time of the commit's creation.
- A pointer to a snapshot of the repository at that moment.
- A pointer to the commit (or commits) that precede this commit.

Branches, merges and conflicts.

Structure of a commit.

Git for
beginners

Gianfranco
Gallizia

Brief history
of Git.

Git basic
commands.

Alias and
other tricks.

Unified diff.

Branches,
merges and
conflicts.

A git commit is a data structure containing (among other things):

- The author's name.
- The author's email.
- Date and time of the commit's creation.
- A pointer to a snapshot of the repository at that moment.
- A pointer to the commit (or commits) that precede this commit.
- The SHA-1 hash of all these metadata.

Branches, merges and conflicts.

Structure of a commit.

Git for
beginners

Gianfranco
Gallizia

Brief history
of Git.

Git basic
commands.

Alias and
other tricks.

Unified diff.

Branches,
merges and
conflicts.

The first commit of a repository doesn't have a preceding commit, the commits we did so far without branching and merging have saved internally the hash of the single commit preceding them.

Branches, merges and conflicts.

Development branches.

Git for
beginners

Gianfranco
Gallizia

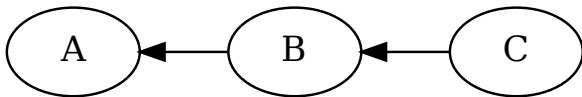
Brief history
of Git.

Git basic
commands.

Alias and
other tricks.

Unified diff.

Branches,
merges and
conflicts.



Branches, merges and conflicts.

Development branches.

Git for
beginners

Gianfranco
Gallizia

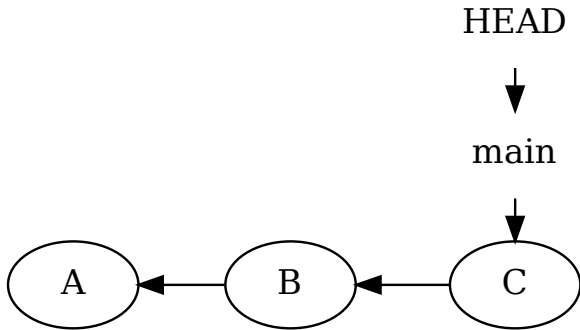
Brief history
of Git.

Git basic
commands.

Alias and
other tricks.

Unified diff.

Branches,
merges and
conflicts.



Branches, merges and conflicts.

Development branches.

Git for
beginners

Gianfranco
Gallizia

Brief history
of Git.

Git basic
commands.

Alias and
other tricks.

Unified diff.

Branches,
merges and
conflicts.

Creating a new branch

```
git branch test
```


Branches, merges and conflicts.

Development branches.

Git for
beginners

Gianfranco
Gallizia

Brief history
of Git.

Git basic
commands.

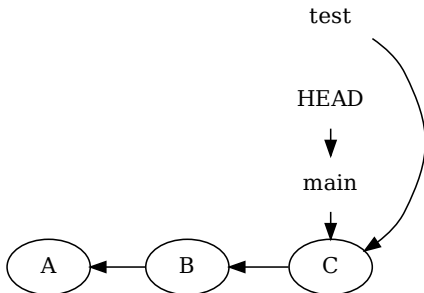
Alias and
other tricks.

Unified diff.

Branches,
merges and
conflicts.

Creating a new branch

```
git branch test
```



Branches, merges and conflicts.

Development branches.

Git for
beginners

Gianfranco
Gallizia

Brief history
of Git.

Git basic
commands.

Alias and
other tricks.

Unified diff.

Branches,
merges and
conflicts.

Moving to the new branch

```
git checkout test
```

Branches, merges and conflicts.

Development branches.

Git for
beginners

Gianfranco
Gallizia

Brief history
of Git.

Git basic
commands.

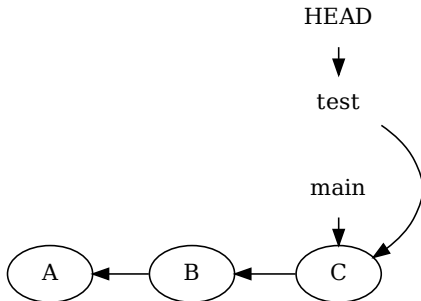
Alias and
other tricks.

Unified diff.

Branches,
merges and
conflicts.

Moving to the new branch

```
git checkout test
```



Branches, merges and conflicts.

Development branches.

Git for
beginners

Gianfranco
Gallizia

Brief history
of Git.

Git basic
commands.

Alias and
other tricks.

Unified diff.

Branches,
merges and
conflicts.

Making some changes and committing them:

```
git add -- fancy.py
git commit -m "Added fancy Python script."
```

Branches, merges and conflicts.

Development branches.

Git for
beginners

Gianfranco
Gallizia

Brief history
of Git.

Git basic
commands.

Alias and
other tricks.

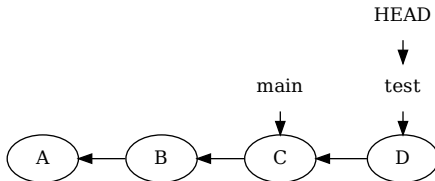
Unified diff.

Branches,
merges and
conflicts.

Making some changes and committing them:

```
git add -- fancy.py
```

```
git commit -m "Added fancy Python script."
```



Branches, merges and conflicts.

Development branches.

Git for
beginners

Gianfranco
Gallizia

Brief history
of Git.

Git basic
commands.

Alias and
other tricks.

Unified diff.

**Branches,
merges and
conflicts.**

After other commits. . .

Branches, merges and conflicts.

Development branches.

Git for
beginners

Gianfranco
Gallizia

Brief history
of Git.

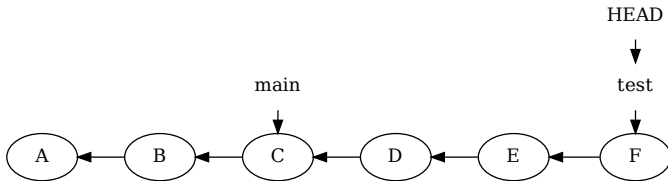
Git basic
commands.

Alias and
other tricks.

Unified diff.

Branches,
merges and
conflicts.

After other commits. . .



Branches, merges and conflicts.

Development branches.

Git for
beginners

Gianfranco
Gallizia

Brief history
of Git.

Git basic
commands.

Alias and
other tricks.

Unified diff.

Branches,
merges and
conflicts.

Going back to main:

```
git checkout main
```


Branches, merges and conflicts.

Development branches.

Git for
beginners

Gianfranco
Gallizia

Brief history
of Git.

Git basic
commands.

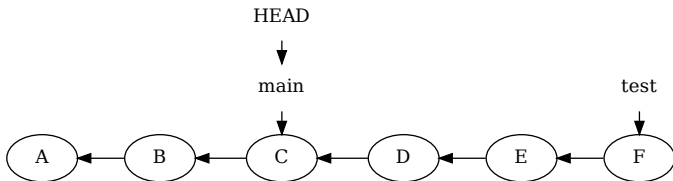
Alias and
other tricks.

Unified diff.

Branches,
merges and
conflicts.

Going back to main:

```
git checkout main
```



Branches, merges and conflicts.

Development branches.

Git for
beginners

Gianfranco
Gallizia

Brief history
of Git.

Git basic
commands.

Alias and
other tricks.

Unified diff.

Branches,
merges and
conflicts.

Integrating the changes:

```
git merge test
```

Branches, merges and conflicts.

Development branches.

Git for
beginners

Gianfranco
Gallizia

Brief history
of Git.

Git basic
commands.

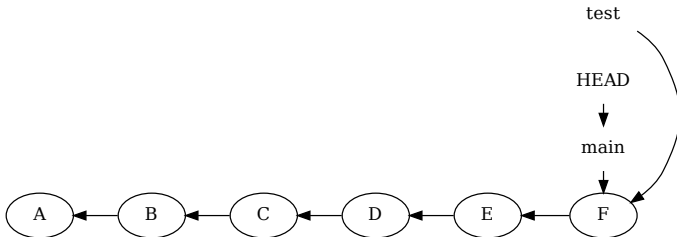
Alias and
other tricks.

Unified diff.

Branches,
merges and
conflicts.

Integrating the changes:

```
git merge test
```



Branches, merges and conflicts.

Development branches.

Git for
beginners

Gianfranco
Gallizia

Brief history
of Git.

Git basic
commands.

Alias and
other tricks.

Unified diff.

Branches,
merges and
conflicts.

Let's get back before the merge. . .

Branches, merges and conflicts.

Development branches.

Git for
beginners

Gianfranco
Gallizia

Brief history
of Git.

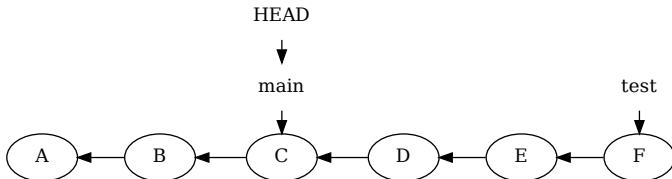
Git basic
commands.

Alias and
other tricks.

Unified diff.

Branches,
merges and
conflicts.

Let's get back before the merge. . .



Branches, merges and conflicts.

Development branches.

Git for
beginners

Gianfranco
Gallizia

Brief history
of Git.

Git basic
commands.

Alias and
other tricks.

Unified diff.

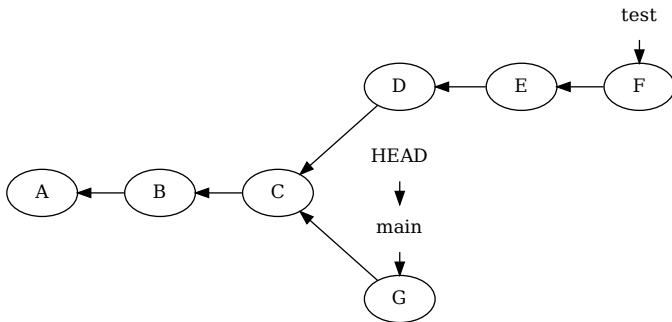
Branches,
merges and
conflicts.

Let's commit a hotfix on `main...`

Branches, merges and conflicts.

Development branches.

Let's commit a hotfix on `main`...



Git for
beginners

Gianfranco
Gallizia

Brief history
of Git.

Git basic
commands.

Alias and
other tricks.

Unified diff.

Branches,
merges and
conflicts.

Branches, merges and conflicts.

Development branches.

Git for
beginners

Gianfranco
Gallizia

Brief history
of Git.

Git basic
commands.

Alias and
other tricks.

Unified diff.

Branches,
merges and
conflicts.

At this point a *fast-forward* is no longer possible, we need to create a merge commit: a commit that has both F and G as parents and that collects the changes from both branches.

Branches, merges and conflicts.

Development branches.

Git for
beginners

Gianfranco
Gallizia

Brief history
of Git.

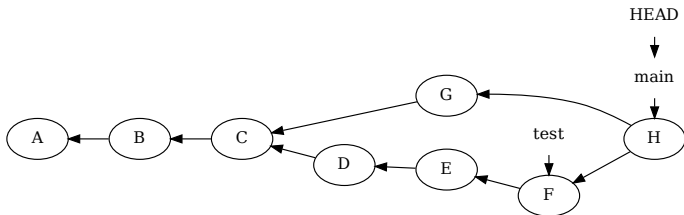
Git basic
commands.

Alias and
other tricks.

Unified diff.

Branches,
merges and
conflicts.

At this point a *fast-forward* is no longer possible, we need to create a merge commit: a commit that has both F and G as parents and that collects the changes from both branches.



Branches, merges and conflicts.

Development branches - conflicts.

Git for
beginners

Gianfranco
Gallizia

Brief history
of Git.

Git basic
commands.

Alias and
other tricks.

Unified diff.

Branches,
merges and
conflicts.

Conflicts

Let's assume we're in a situation where a conflict arises. How can we manage it when merging two branches in git?

Branches, merges and conflicts.

Development branches - conflicts.

Git for
beginners

Gianfranco
Gallizia

Brief history
of Git.

Git basic
commands.

Alias and
other tricks.

Unified diff.

Branches,
merges and
conflicts.

Conflicts

Let's assume we're in a situation where a conflict arises. How can we manage it when merging two branches in git?

```
git init --initial-branch=main conflict
cd conflict
echo -e "A\nB\nC" > README.txt
git add README.txt
git commit -m "Initial Commit."
```

Branches, merges and conflicts.

Development branches - conflicts.

Git for
beginners

Gianfranco
Gallizia

Brief history
of Git.

Git basic
commands.

Alias and
other tricks.

Unified diff.

Branches,
merges and
conflicts.

Conflicts (cont.)

```
git checkout -b change_A
tr A-Z a-z < README.txt > tmp.txt
mv tmp.txt README.txt
git add README.txt
git commit -m "Lowercase."
```

Branches, merges and conflicts.

Development branches - conflicts.

Git for
beginners

Gianfranco
Gallizia

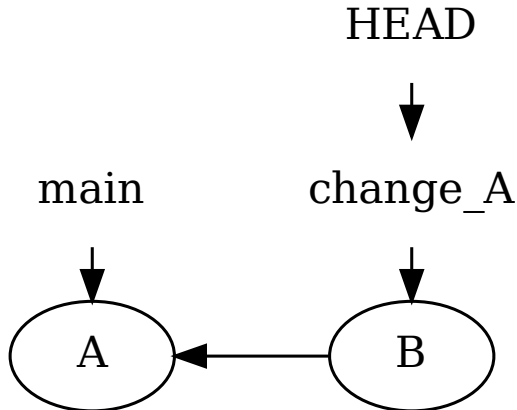
Brief history
of Git.

Git basic
commands.

Alias and
other tricks.

Unified diff.

Branches,
merges and
conflicts.



Branches, merges and conflicts.

Development branches - conflicts.

Git for
beginners

Gianfranco
Gallizia

Brief history
of Git.

Git basic
commands.

Alias and
other tricks.

Unified diff.

Branches,
merges and
conflicts.

Conflicts (cont.)

```
git checkout main
sed -i -r 's/^(([A-Z]))/\1\1/' README.txt
git add README.txt
git commit -m "Double the letters."
```

Branches, merges and conflicts.

Development branches - conflicts.

Git for
beginners

Gianfranco
Gallizia

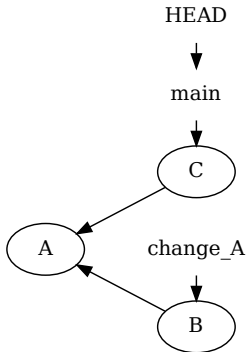
Brief history
of Git.

Git basic
commands.

Alias and
other tricks.

Unified diff.

Branches,
merges and
conflicts.



Branches, merges and conflicts.

Development branches - conflicts.

Git for
beginners

Gianfranco
Gallizia

Brief history
of Git.

Git basic
commands.

Alias and
other tricks.

Unified diff.

Branches,
merges and
conflicts.

Conflicts (cont.)

```
git checkout -b change_B
```


Branches, merges and conflicts.

Development branches - conflicts.

Git for
beginners

Gianfranco
Gallizia

Brief history
of Git.

Git basic
commands.

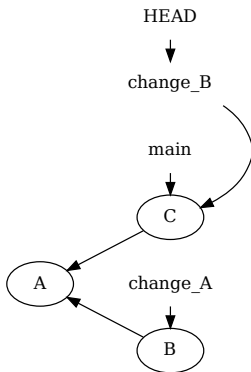
Alias and
other tricks.

Unified diff.

Branches,
merges and
conflicts.

Conflicts (cont.)

```
git checkout -b change_B
```



Branches, merges and conflicts.

Development branches - conflicts.

Git for
beginners

Gianfranco
Gallizia

Brief history
of Git.

Git basic
commands.

Alias and
other tricks.

Unified diff.

Branches,
merges and
conflicts.

Conflicts (cont.)

```
git merge change_A
```

```
Auto-merging README.txt
```

```
CONFLICT (content): Merge conflict in README.txt
```

```
Automatic merge failed; fix conflicts and then commit
```

Branches, merges and conflicts.

Development branches - conflicts.

Git for
beginners

Gianfranco
Gallizia

Brief history
of Git.

Git basic
commands.

Alias and
other tricks.

Unified diff.

Branches,
merges and
conflicts.

First let's check the status:

```
git status
```

```
On branch change_B
```

```
You have unmerged paths.
```

```
    (fix conflicts and run "git commit")
```

```
    (use "git merge --abort" to abort the merge)
```

```
Unmerged paths:
```

```
    (use "git add <file>..." to mark resolution)
```

```
both modified:   README.txt
```

```
no changes added to commit
```

```
(use "git add" and/or "git commit -a")
```

Branches, merges and conflicts.

Development branches - conflicts.

Git for
beginners

Gianfranco
Gallizia

Brief history
of Git.

Git basic
commands.

Alias and
other tricks.

Unified diff.

Branches,
merges and
conflicts.

Let's check the changes in detail:

```
git diff -- README.txt
diff --cc README.txt
index 21c9ecf,de98044..0000000
--- a/README.txt
+++ b/README.txt
@@@ -1,3 -1,3 +1,9 @@@
++<<<<<< HEAD
+AA
+BB
+CC
++=====
+ a
+ b
+ c
++>>>>>> change_A
```

Branches, merges and conflicts.

Development branches - conflicts.

Git for
beginners

Gianfranco
Gallizia

Brief history
of Git.

Git basic
commands.

Alias and
other tricks.

Unified diff.

Branches,
merges and
conflicts.

README.txt

<<<<<<< HEAD

AA

BB

CC

=====

a

b

c

>>>>>>> change_A

Branches, merges and conflicts.

Development branches - conflicts.

Git for
beginners

Gianfranco
Gallizia

Brief history
of Git.

Git basic
commands.

Alias and
other tricks.

Unified diff.

Branches,
merges and
conflicts.

README.txt

aA

bB

cC

Branches, merges and conflicts.

Development branches - conflicts.

Git for
beginners

Gianfranco
Gallizia

Brief history
of Git.

Git basic
commands.

Alias and
other tricks.

Unified diff.

Branches,
merges and
conflicts.

Let's resolve the conflict

```
git commit -a
git log --all --decorate --oneline --graph
*   8b393c8 (HEAD -> change_B) Merge branch...
|\
| * 5855fe7 (change_A) Lowercase.
* | bd1130e (main) Double the letters.
|/
* d1f0f78 Initial Commit.
```

Branches, merges and conflicts.

Development branches - conflicts.

Git for
beginners

Gianfranco
Gallizia

Brief history
of Git.

Git basic
commands.

Alias and
other tricks.

Unified diff.

Branches,
merges and
conflicts.

Let's resolve the conflict

```
git diff main
diff --git a/README.txt b/README.txt
index 21c9ecf..a359da0 100644
--- a/README.txt
+++ b/README.txt
@@ -1,3 +1,3 @@
-AA
-BB
-CC
+aA
+bB
+cC
```


Branches, merges and conflicts.

Development branches - conflicts.

Git for
beginners

Gianfranco
Gallizia

Brief history
of Git.

Git basic
commands.

Alias and
other tricks.

Unified diff.

Branches,
merges and
conflicts.

Let's resolve the conflict

```
git diff change_A
diff --git a/README.txt b/README.txt
index de98044..a359da0 100644
--- a/README.txt
+++ b/README.txt
@@ -1,3 +1,3 @@
-a
-b
-c
+aA
+bB
+cC
```

End of module 1

Git for
beginners

Gianfranco
Gallizia

Brief history
of Git.

Git basic
commands.

Alias and
other tricks.

Unified diff.

Branches,
merges and
conflicts.

Thank you for your attention!

End of module 1

Git for
beginners

Gianfranco
Gallizia

Brief history
of Git.

Git basic
commands.

Alias and
other tricks.

Unified diff.

Branches,
merges and
conflicts.

Thank you for your attention!

Any questions?