©Bennet Becker, 2019

IP of the Max-Planck-Institute for Physics of Complex Systems (mpipks),

Nöthnitzer Straße 38, 01187 Dresden

DO NOT USE OUTSIDE THE MPI PKS

Images Copyrighted by their owners

Theme Metropolis CC-BY-SA 4.0 by github.com/matze

MPI PKS C++ Lecture

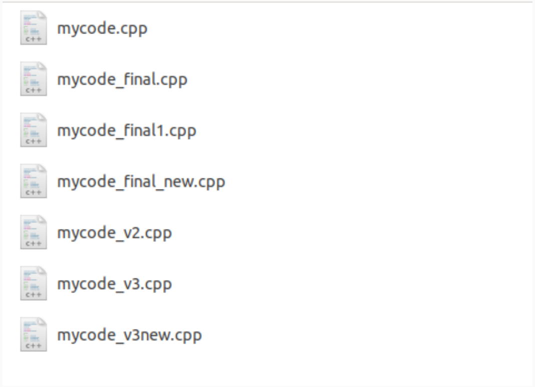
Bennet Becker

2019-09-09

MPI PKS

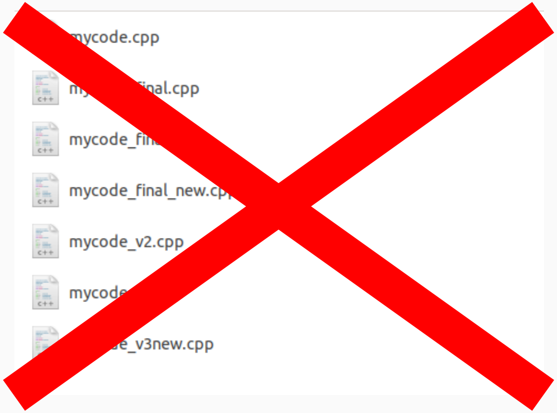
Git - Versioning your Code

What the git?? Version Control 🤨⁉️❓

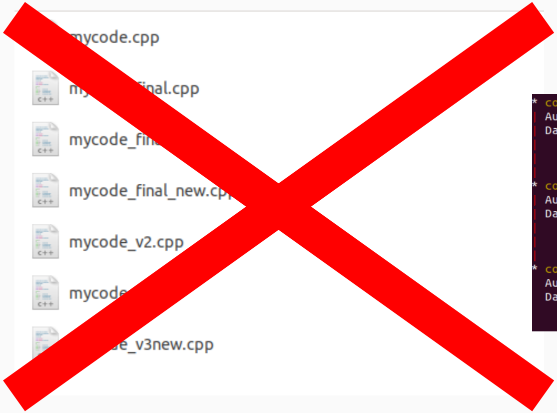


mycode.cpp
mycode_final.cpp
mycode_final1.cpp
mycode_final_new.cpp
mycode_v2.cpp
mycode_v3.cpp
mycode_v3new.cpp

What the git?? Version Control 🤨⁉❓


ycode.cpp
m final.cpp
mycode_fi
mycode_final_new.cp
mycode_v2.cpp
mycode
e_v3new.cpp

What the git?? Version Control 🤨⁉❓



mycode.cpp

m final.cpp

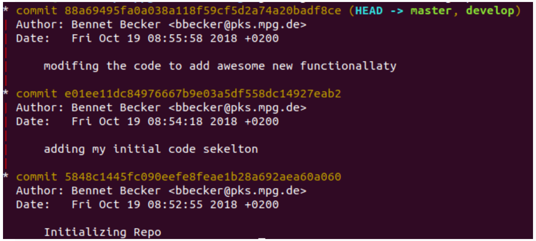
mycode_fin

mycode_final_new.cp

mycode_v2.cpp

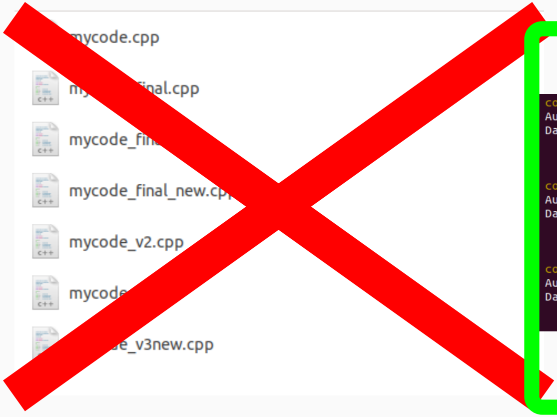
mycode

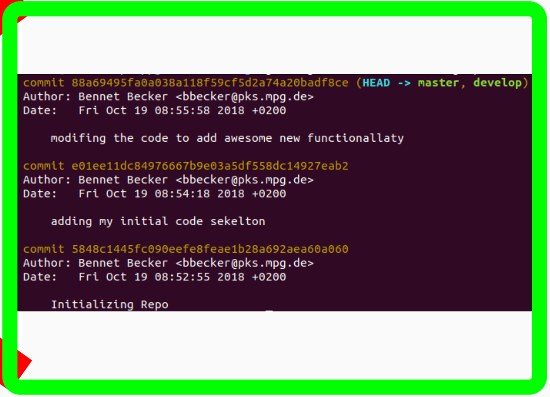
e_v3new.cpp



```
* commit 88a69495fa0a038a118f59cf5d2a74a20badf8ce (HEAD -> master, develop)
| Author: Bennet Becker <bbecker@pks.mpg.de>
| Date:   Fri Oct 19 08:55:58 2018 +0200
|
|     modifing the code to add awesome new functionallaty
|
* commit e01ee11dc84976667b9e03a5df558dc14927eab2
| Author: Bennet Becker <bbecker@pks.mpg.de>
| Date:   Fri Oct 19 08:54:18 2018 +0200
|
|     adding my initial code sekelton
|
* commit 5848c1445fc090eefe8feae1b28a692aea60a060
  Author: Bennet Becker <bbecker@pks.mpg.de>
  Date:   Fri Oct 19 08:52:55 2018 +0200

      Initializing Repo
```

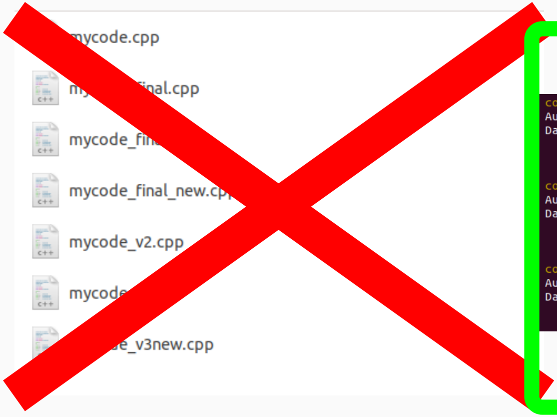
What the git?? Version Control 🤨⁉️❓



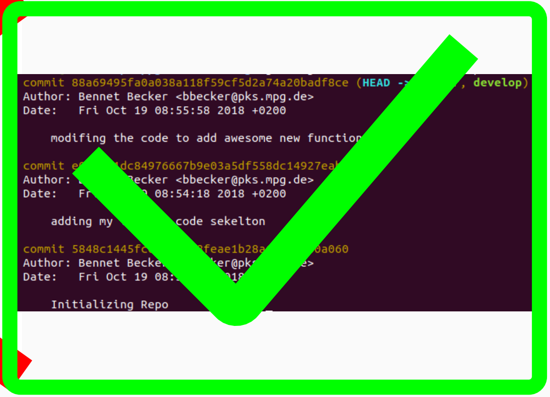


```
commit 88a69495fa0a038a118f59cf5d2a74a20badf8ce (HEAD -> master, develop)
Author: Bennet Becker <bbecker@pks.mpg.de>
Date:   Fri Oct 19 08:55:58 2018 +0200

    modifing the code to add awesome new functionallaty

commit e01ee11dc84976667b9e03a5df558dc14927eab2
Author: Bennet Becker <bbecker@pks.mpg.de>
Date:   Fri Oct 19 08:54:18 2018 +0200

    adding my initial code sekelton

commit 5848c1445fc090eefe8feae1b28a692aea60a060
Author: Bennet Becker <bbecker@pks.mpg.de>
Date:   Fri Oct 19 08:52:55 2018 +0200

    Initializing Repo
```

What the git?? Version Control 🤨⁉❓

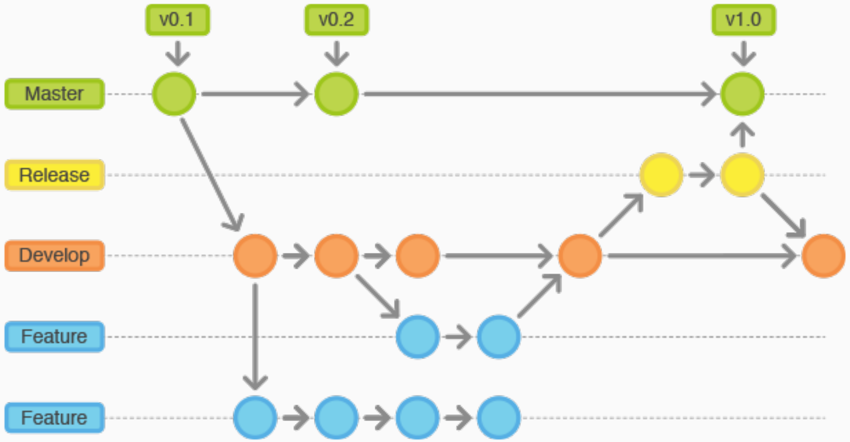


```
mycode.cpp
m    final.cpp
mycode_fi
mycode_final_new.cp
mycode_v2.cpp
mycode
    e_v3new.cpp
```

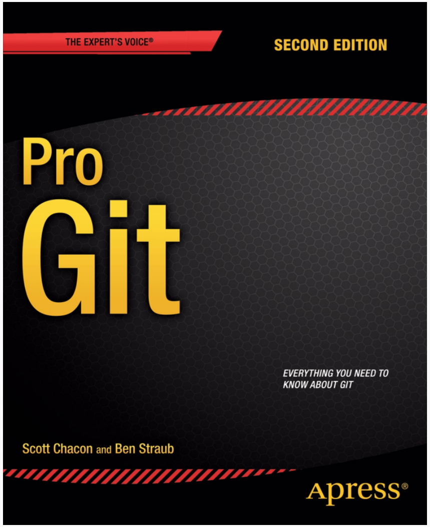


```
commit 88a69495fa0a038a118f59cf5d2a74a20badf8ce (HEAD -        , develop)
Author: Bennet Becker <bbecker@pks.mpg.de>
Date:   Fri Oct 19 08:55:58 2018 +0200

    modifing the code to add awesome new function

commit e     dc84976667b9e03a5df558dc14927ea
Author:       Becker <bbecker@pks.mpg.de>
Date:   Fr       08:54:18 2018 +0200

    adding my       code sekelton

commit 5848c1445fc        feae1b28a        0a060
Author: Bennet Becke       ker@pk        e>
Date:   Fri Oct 19 08:      018

    Initializing Repo
```

In an ideal World



Git - the whole story


THE EXPERT'S VOICE®
SECOND EDITION
Pro Git
EVERYTHING YOU NEED TO KNOW ABOUT GIT
Scott Chacon and Ben Straub
Apress®

Pro Git, Scott Chacon and Ben Straub

Print ISBN 978-1-4842-0077-3
Online ISBN 978-1-4842-0076-6

Free E-book
`https://doi.org/10.1007/978-1-4842-0076-6`

Git Versioning

- Keep track of changes to files, managed by Git

Git Versioning

- Keep track of changes to files, managed by Git
- No manual copy for each and every change required

Git Versioning

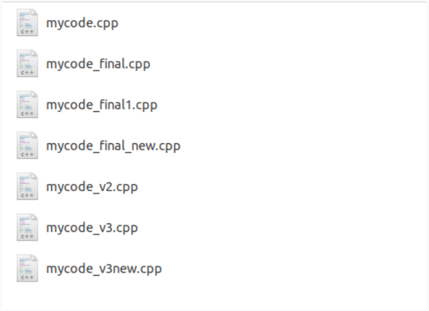
- Keep track of changes to files, managed by Git
- No manual copy for each and every change required

So you avoid this:

Git Versioning

- Keep track of changes to files, managed by Git
- No manual copy for each and every change required

So you avoid this:



mycode.cpp
mycode_final.cpp
mycode_final1.cpp
mycode_final_new.cpp
mycode_v2.cpp
mycode_v3.cpp
mycode_v3new.cpp

How does it work

- behaves like a miniature filesystem, saving a “snapshot” of it with each commit

*except deleting the entire repo before saving it somewhere else or trying to manipulate git history

How does it work

- behaves like a miniature filesystem, saving a “snapshot” of it with each commit
- but if a file has not changed, it is not stored again

*except deleting the entire repo before saving it somewhere else or trying to manipulate git history

How does it work

- behaves like a miniature filesystem, saving a “snapshot” of it with each commit
- but if a file has not changed, it is not stored again
- if a snapshot has been commited to git it is nearly impossible to lose this change*

*except deleting the entire repo before saving it somewhere else or trying to manipulate git history

How does it work

- behaves like a miniature filesystem, saving a “snapshot” of it with each commit
- but if a file has not changed, it is not stored again
- if a snapshot has been commited to git it is nearly impossible to lose this change*

- nearly all operations are local

*except deleting the entire repo before saving it somewhere else or trying to manipulate git history

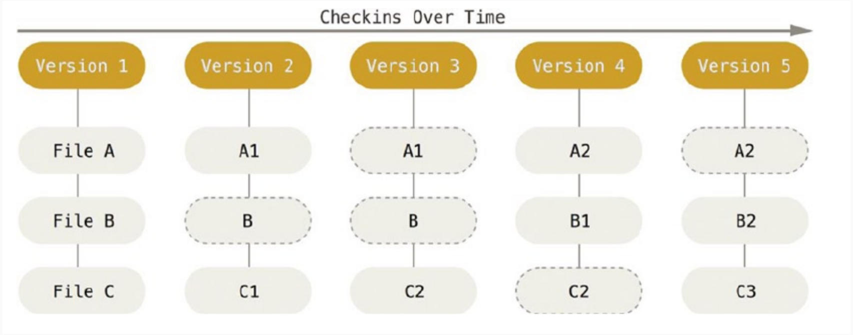
How does it work

- behaves like a miniature filesystem, saving a “snapshot” of it with each commit
- but if a file has not changed, it is not stored again
- if a snapshot has been commited to git it is nearly impossible to lose this change*

- nearly all operations are local
- files referenced by hashes for integrity

*except deleting the entire repo before saving it somewhere else or trying to manipulate git history

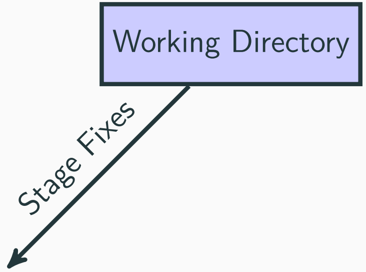
How does it work ii



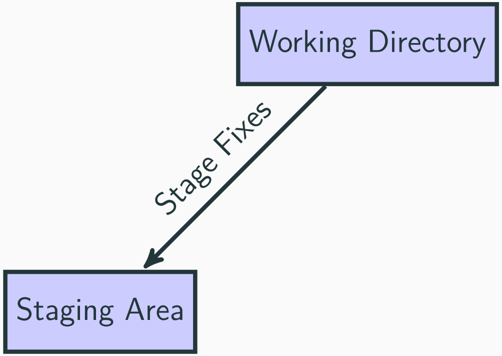
3 Areas in Git

3 Areas in Git

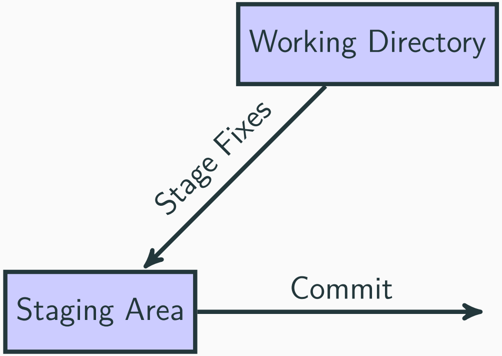
3 Areas in Git



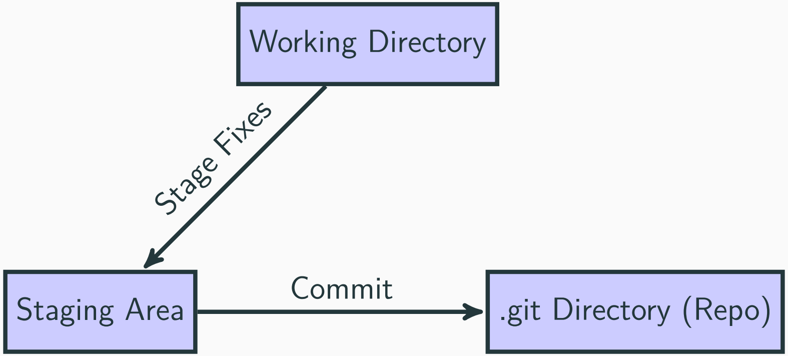
3 Areas in Git



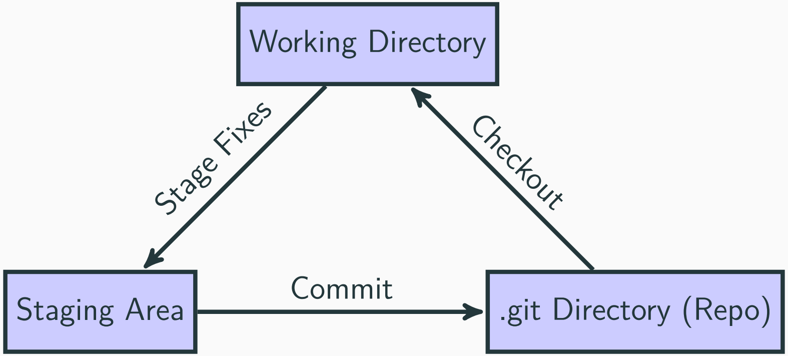
3 Areas in Git



3 Areas in Git



3 Areas in Git



Setup your git

- tell git your identity

  ```
  git config --global user.name "John Doe"
  git config --global user.email johndoe@example.com
  ```

- choose your editor

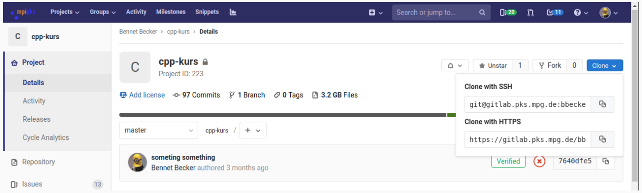
  ```
  git config --global core.editor vim
  ```

  `$EDITOR` and /etc/alternatives should be honored too

Create a Repo (locally)

```
mkdir myrepo
cd myrepo/
git init
git add *.c
git add LICENSE
git commit -m 'initial project version'
```

Or Clone a Repo i

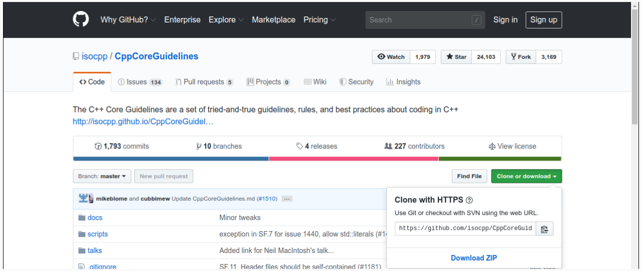


cpp-kurs
Project ID: 223

Clone with SSH
`git@gitlab.pks.mpg.de:bbecke`

Clone with HTTPS
`https://gitlab.pks.mpg.de/bb`

someting something
Bennet Becker authored 3 months ago
7640dfe5



isocpp / CppCoreGuidelines

The C++ Core Guidelines are a set of tried-and-true guidelines, rules, and best practices about coding in C++
http://isocpp.github.io/CppCoreGuidel…

mikeblome and cubbimew Update CppCoreGuidelines.md (#1510)

docs	Minor tweaks
scripts	exception in SF.7 for issue 1440, allow std::literals (#1
talks	Added link for Neil MacIntosh's talk...
.gitignore	SF.11. Header files should be self-contained (#1181)

Clone with HTTPS
Use Git or checkout with SVN using the web URL.
`https://github.com/isocpp/CppCoreGuid`

Or Clone a Repo ii

```
git clone --recursive git@gitlab.pks.mpg.de:user/repo.git
```

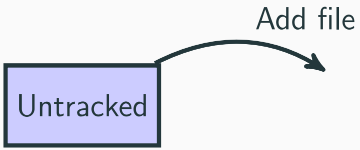
Protocols in Git

- HTTPS (`https://gitserver/user/repo.git`)
  - is often easier
  - username/password authentication
  - HTTPS is a very common protocol, so it works in most network setups
- SSH (`git@gitserver:/user/repo.git`)
  - easy server setup
  - authentication via public-keys
  - no anonymous access to repository
- git (`git://gitserver/user/repo.git`)
  - special daemon with special port
  - no authentication

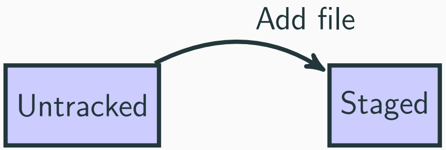
Recording Changes to the Repo

Recording Changes to the Repo

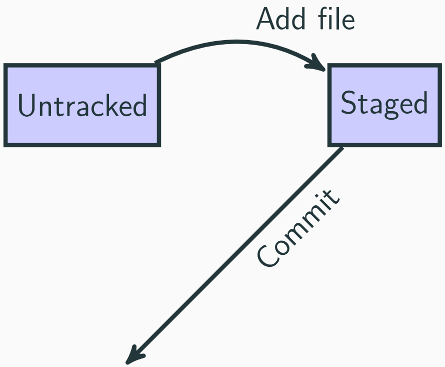
Recording Changes to the Repo



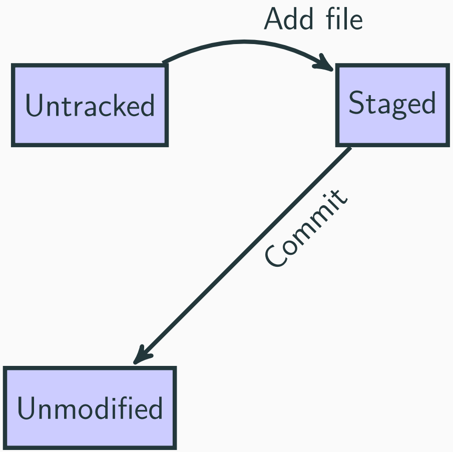
Recording Changes to the Repo



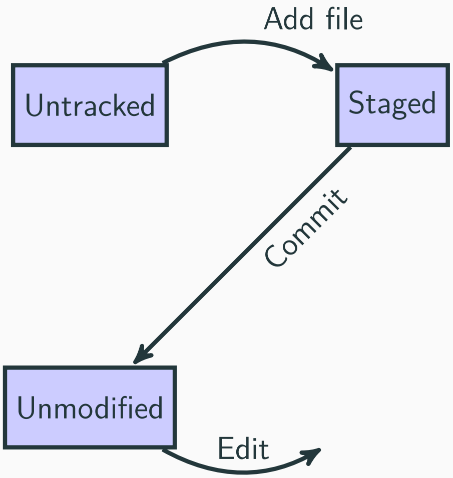
Recording Changes to the Repo



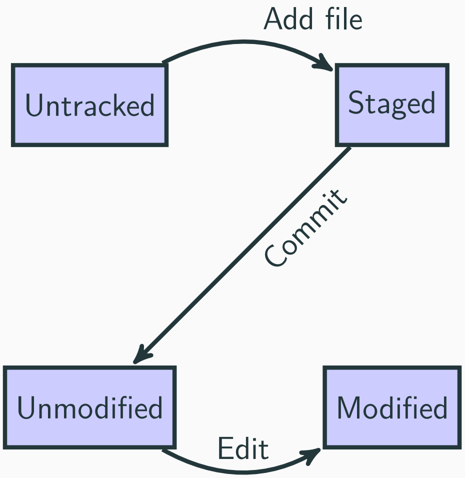
Recording Changes to the Repo



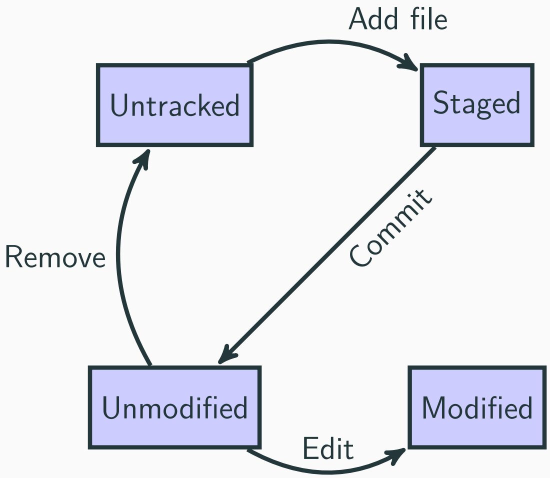
Recording Changes to the Repo



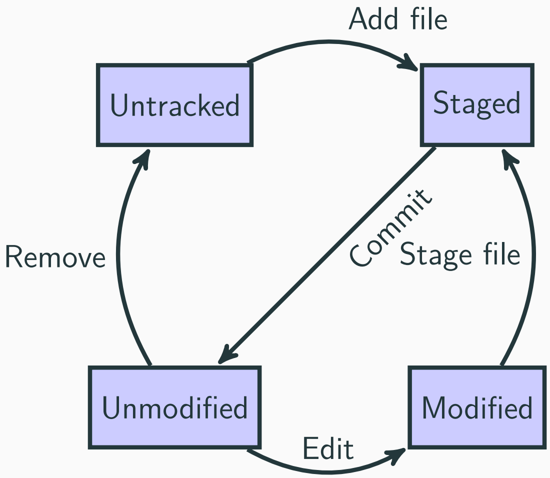
Recording Changes to the Repo



Recording Changes to the Repo



Recording Changes to the Repo

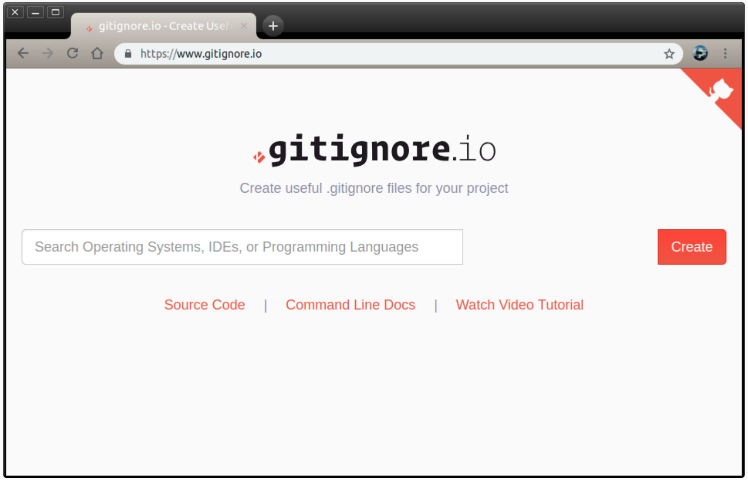

$$\begin{tikzcd}
\text{Untracked} \arrow[r, bend left, "\text{Add file}"] & \text{Staged} \arrow[dl, "\text{Commit}"] \\
\text{Unmodified} \arrow[u, bend left, "\text{Remove}"] \arrow[r, bend right, "\text{Edit}"'] & \text{Modified} \arrow[u, bend right, "\text{Stage file}"']
\end{tikzcd}$$

Ignoring Files i

- hide files you don't want to track at all
- Binaries / Large stuff
- Local configs
- confidential stuff
- automatically generated files

```
> cat .gitignore
*.[oa]
*~
```

Ignoring Files ii



gitignore.io - Create Usef

https://www.gitignore.io

gitignore.io

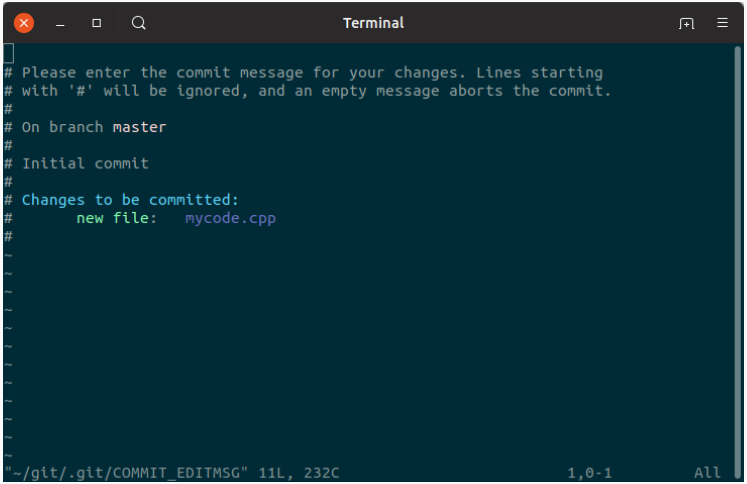
Create useful .gitignore files for your project

Search Operating Systems, IDEs, or Programming Languages
Create
Source Code | Command Line Docs | Watch Video Tutorial

https://www.gitignore.io/

Commiting your changes

```
$ git commit
```

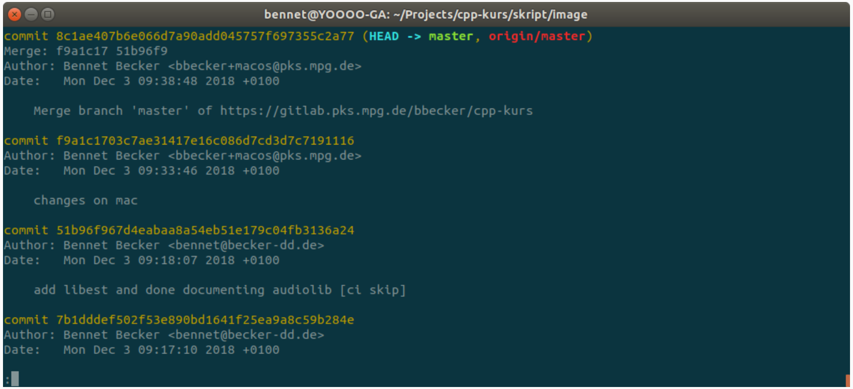


```
Terminal

# Please enter the commit message for your changes. Lines starting
# with '#' will be ignored, and an empty message aborts the commit.
#
# On branch master
#
# Initial commit
#
# Changes to be committed:
#       new file:   mycode.cpp
#
~
~
~
~
~
~
~
~
~
~
~
~
"~/git/.git/COMMIT_EDITMSG" 11L, 232C                     1,0-1         All
```

```
$ git commit -m "Your commit message"
```

Viewing the history i

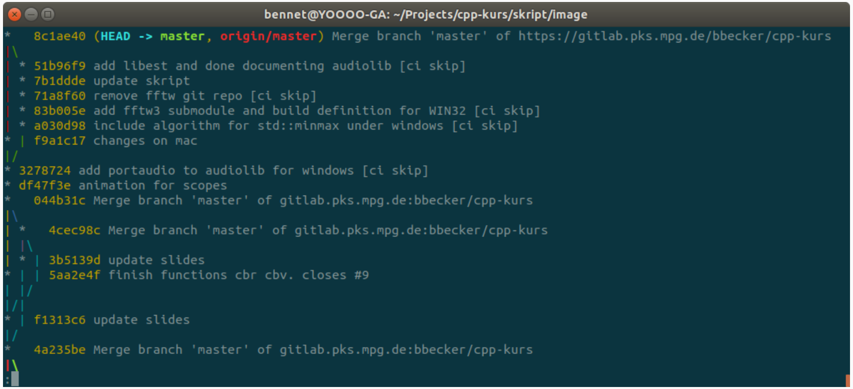
```
$ git log
```



```
commit 8c1ae407b6e066d7a90add045757f697355c2a77 (HEAD -> master, origin/master)
Merge: f9a1c17 51b96f9
Author: Bennet Becker <bbecker+macos@pks.mpg.de>
Date:   Mon Dec 3 09:38:48 2018 +0100

    Merge branch 'master' of https://gitlab.pks.mpg.de/bbecker/cpp-kurs

commit f9a1c1703c7ae31417e16c086d7cd3d7c7191116
Author: Bennet Becker <bbecker+macos@pks.mpg.de>
Date:   Mon Dec 3 09:33:46 2018 +0100

    changes on mac

commit 51b96f967d4eabaa8a54eb51e179c04fb3136a24
Author: Bennet Becker <bennet@becker-dd.de>
Date:   Mon Dec 3 09:18:07 2018 +0100

    add libest and done documenting audiolib [ci skip]

commit 7b1dddef502f53e890bd1641f25ea9a8c59b284e
Author: Bennet Becker <bennet@becker-dd.de>
Date:   Mon Dec 3 09:17:10 2018 +0100

:
```

Viewing the history ii

```
$ git log --all --decorate --oneline --graph
```



```
bennet@YOOOO-GA: ~/Projects/cpp-kurs/skript/image
*   8c1ae40 (HEAD -> master, origin/master) Merge branch 'master' of https://gitlab.pks.mpg.de/bbecker/cpp-kurs
|\
| * 51b96f9 add libest and done documenting audiolib [ci skip]
| * 7b1ddde update skript
| * 71a8f60 remove fftw git repo [ci skip]
| * 83b005e add fftw3 submodule and build definition for WIN32 [ci skip]
| * a030d98 include algorithm for std::minmax under windows [ci skip]
* | f9a1c17 changes on mac
|/
* 3278724 add portaudio to audiolib for windows [ci skip]
* df47f3e animation for scopes
*   044b31c Merge branch 'master' of gitlab.pks.mpg.de:bbecker/cpp-kurs
|\
| *   4cec98c Merge branch 'master' of gitlab.pks.mpg.de:bbecker/cpp-kurs
| |\
| * | 3b5139d update slides
* | | 5aa2e4f finish functions cbr cbv. closes #9
| |/
|/|
* | f1313c6 update slides
|/
*   4a235be Merge branch 'master' of gitlab.pks.mpg.de:bbecker/cpp-kurs
|\
:
```

Viewing the history iii

I DON'T DO GIT LOG ALL THE TIME, BUT WHEN I DO JUST REMEMBER
(A DOG) --ALL --DECORATE --ONELINE --GRAPH
memegenerator.net

Undoing Things

- Undo Commit (not pushed) `git commit --amend`
- Unstage a file `git reset HEAD filename`
- Unmodify a file `git checkout -- filename`

Remotes

- when used `git clone`, a remote called `origin` is already set
- add remote `git remote add origin https://gitsrv/user/repo.git`
- modify url `git remote set-url origin https://gitsrv/user/repo.git`
- push to remote `git push origin branch`
- repository can have multiple remotes, you decide on push where to push to.
    - `git push -u origin branch` sets the remote as default for this branch, so you can just use `git push` later

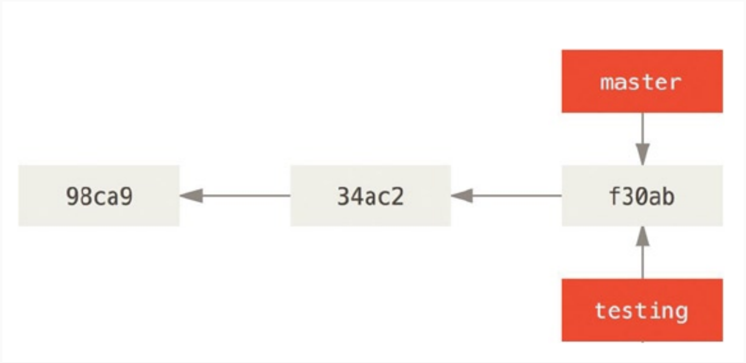
Branching

- (recap) git stores series of snapshots
- commit contains a pointer to the snapshot of the staged content
- commit also contains author information (name, email), the commit-message, a pointers to it's parents (zero for inital, one for "normal" commit, multiple for merge commits)

Branching

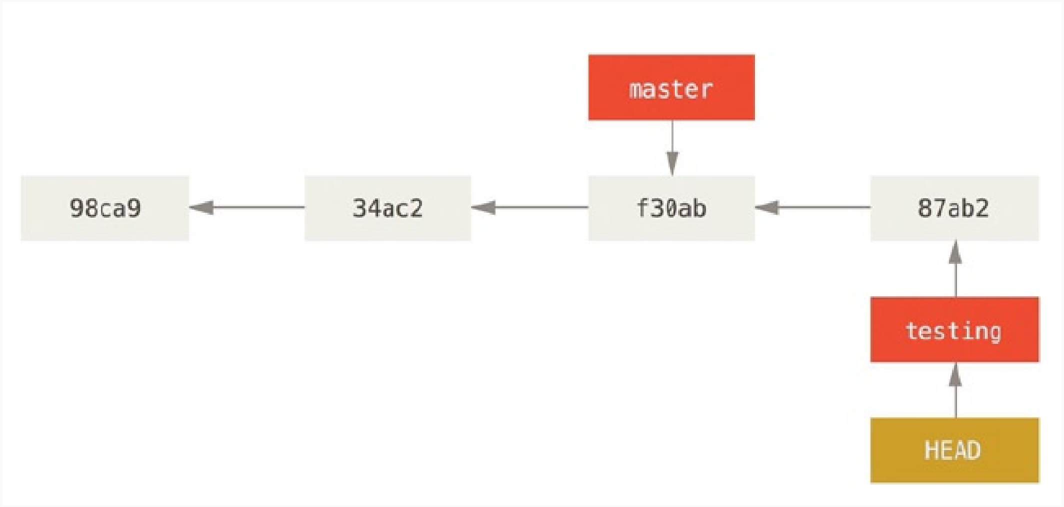
$\boxed{\text{commit}}$

Create a new Branch i



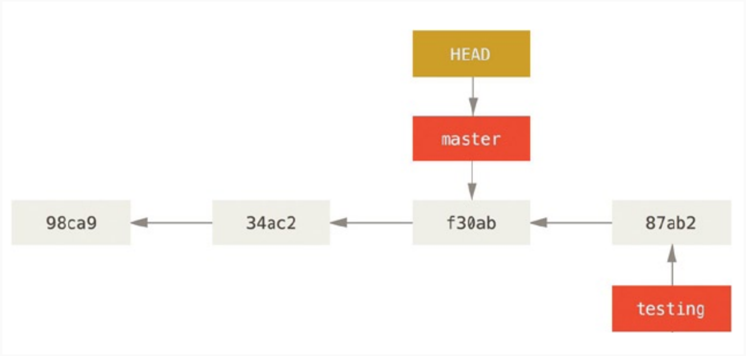
```
git branch testing
```

Create a new Branch ii

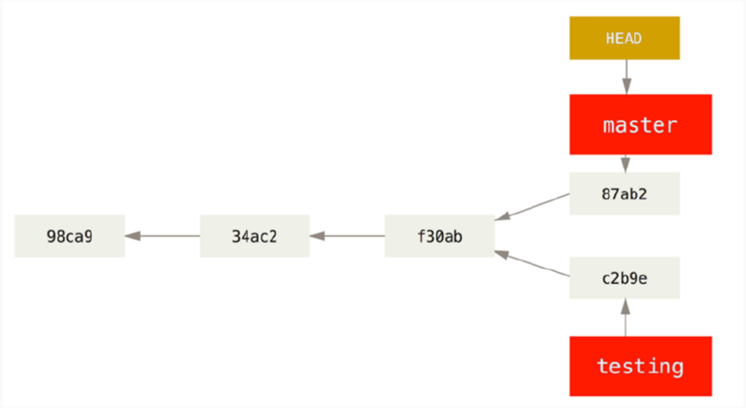


Switching branches i

```
git checkout branch
```

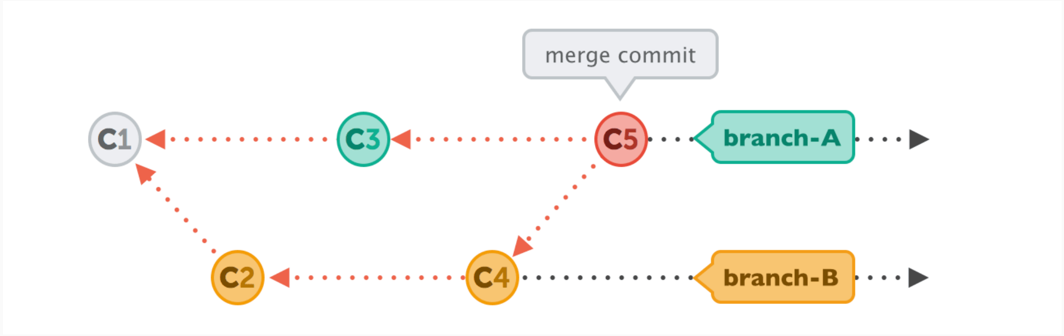


Switching branches ii

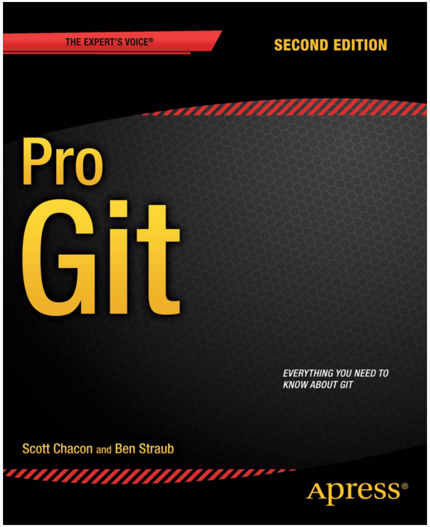


Merge Branches

`git merge`



More on Git


THE EXPERT'S VOICE®
SECOND EDITION
Pro Git
EVERYTHING YOU NEED TO KNOW ABOUT GIT
Scott Chacon and Ben Straub
Apress®

Pro Git, Scott Chacon and Ben Straub

Print ISBN 978-1-4842-0077-3
Online ISBN 978-1-4842-0076-6

Free E-book
`https://doi.org/10.1007/978-1-4842-0076-6`