Advanced git

Created by Guy Van den Nieuwenhof
https://github.com/guyvdn

Advanced Git

- Properly configure Git
- Advanced Git commands
- Customize Git with Git Hooks
- Further Reading

Advanced Git

- **Properly configure Git**
- Advanced Git commands
- Customize Git with Git Hooks
- Further Reading

Properly configure Git

1. Files
2. Your Identity
3. Git Credential Manager
4. Editors
5. Checking Your Settings

Files

1. **/etc/gitconfig**

   System-wide configuration file.

   Pass the option **--system** to git config.

2. **~/.gitconfig** or **~/.config/git/config**

   User-specific configuration file.

   Pass the option **--global** to git config.

3. **.git/config** (default)

   Repository specific configuration file.

   Pass the option **--local** to git config.

Your Identity

Global Identity

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

Identity for customer specific repository

```
$ git config --local user.email johndoe@customer.com
```

Git Credential Manager

Git Credential Managers simplify authentication with your Git repos. They also provide support two-factor authentication.

Windows
Download and run the latest Git for Windows installer, which includes the Git Credential Manager for Windows.
https://git-scm.com/download/win

macOS and Linux

```
> git-credential-manager install
```

Editors

Text Editor

```
$ git config --global core.editor ✂
"'C:/Program Files (x86)/Notepad++/notepad++.exe' ✂
-multiInst -notabbar -nosession"
```

Merge and Diff Tool

```
$ git config --global --add merge.tool kdiff3
$ git config --global --add mergetool.kdiff3.path "C:/../kdiff3.exe"
$ git config --global --add mergetool.kdiff3.trustExitCode false

$ git config --global --add diff.guitool kdiff3
$ git config --global --add difftool.kdiff3.path "C:/../kdiff3.exe"
$ git config --global --add difftool.kdiff3.trustExitCode false
```

Checking Your Settings

If you want to check your configuration settings, you can use the `git config --list` command to list all the settings Git can find at that point.

```
$ git config --list
user.name=John Doe
user.email=johndoe@example.com
..
```

You may see keys more than once, because Git reads the same key from different files (`~/.gitconfig` and `.git/config`, for example). In this case, Git uses the last value for each unique key it sees.

Checking Your Settings

You can also check what Git thinks a specific key’s value is by typing `git config <key>`

```
$ git config user.name
John Doe
```

Advanced Git

- Properly configure Git
- **Advanced Git commands**
- Customize Git with Git Hooks
- Further Reading

Advanced Git commands

1. git log
2. git reset
3. git rebase interactive
4. git worktree
5. git alias

git log

Visualising the git history and branches is often an argument to use a git GUI tool but this can perfectly be done from the command line using `git log` with these commands:

--all
Shows all commits in the history of branches, tags and other refs

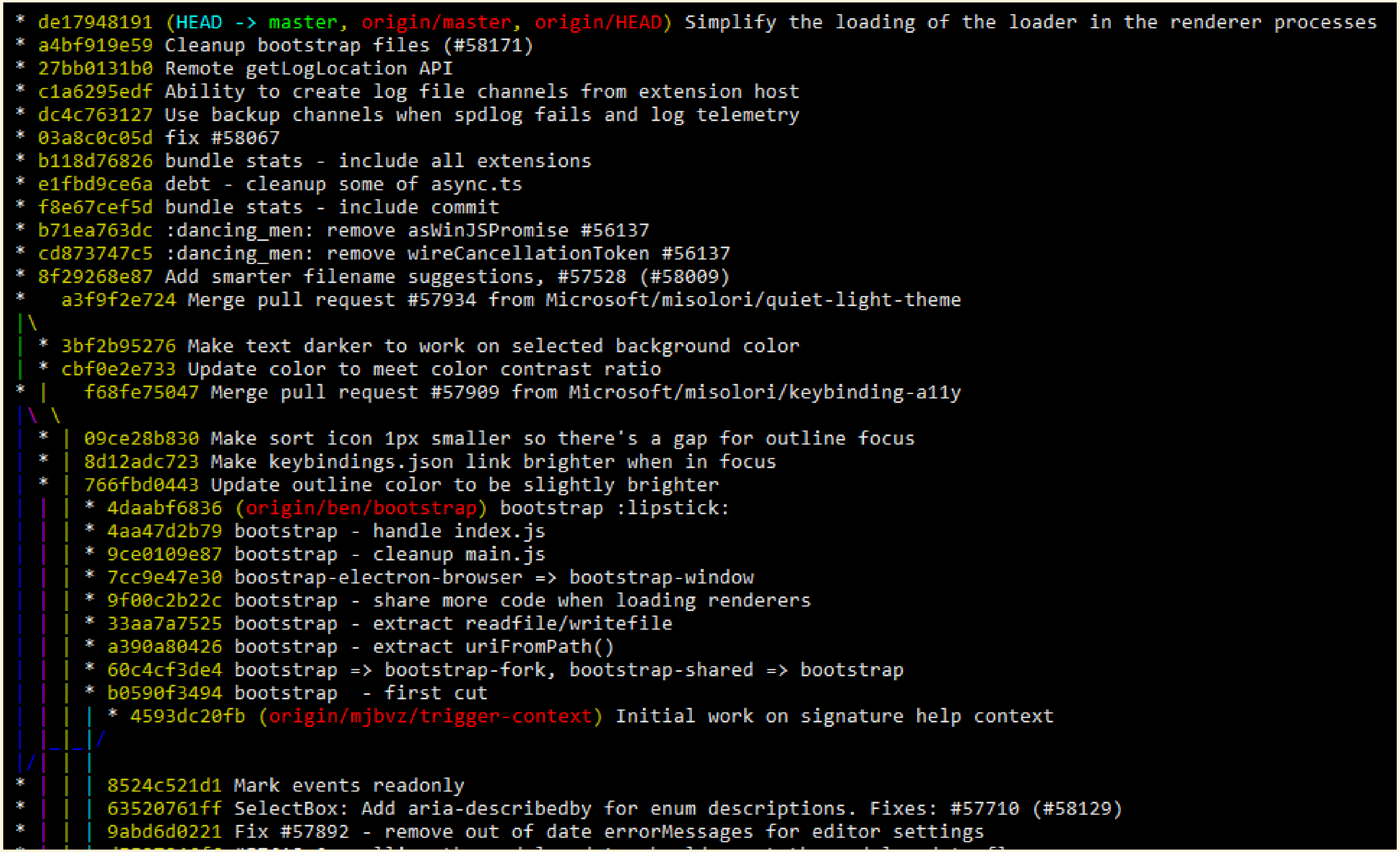
--graph
Text-based graphical representation of the commit history.

--pretty
Pretty-print the contents of the commit logs.
Can be oneline, short, medium, full, fuller, email, raw, format:<string>

--abbrev-commit
Show partial prefix of the 40-byte hexadecimal commit object name.

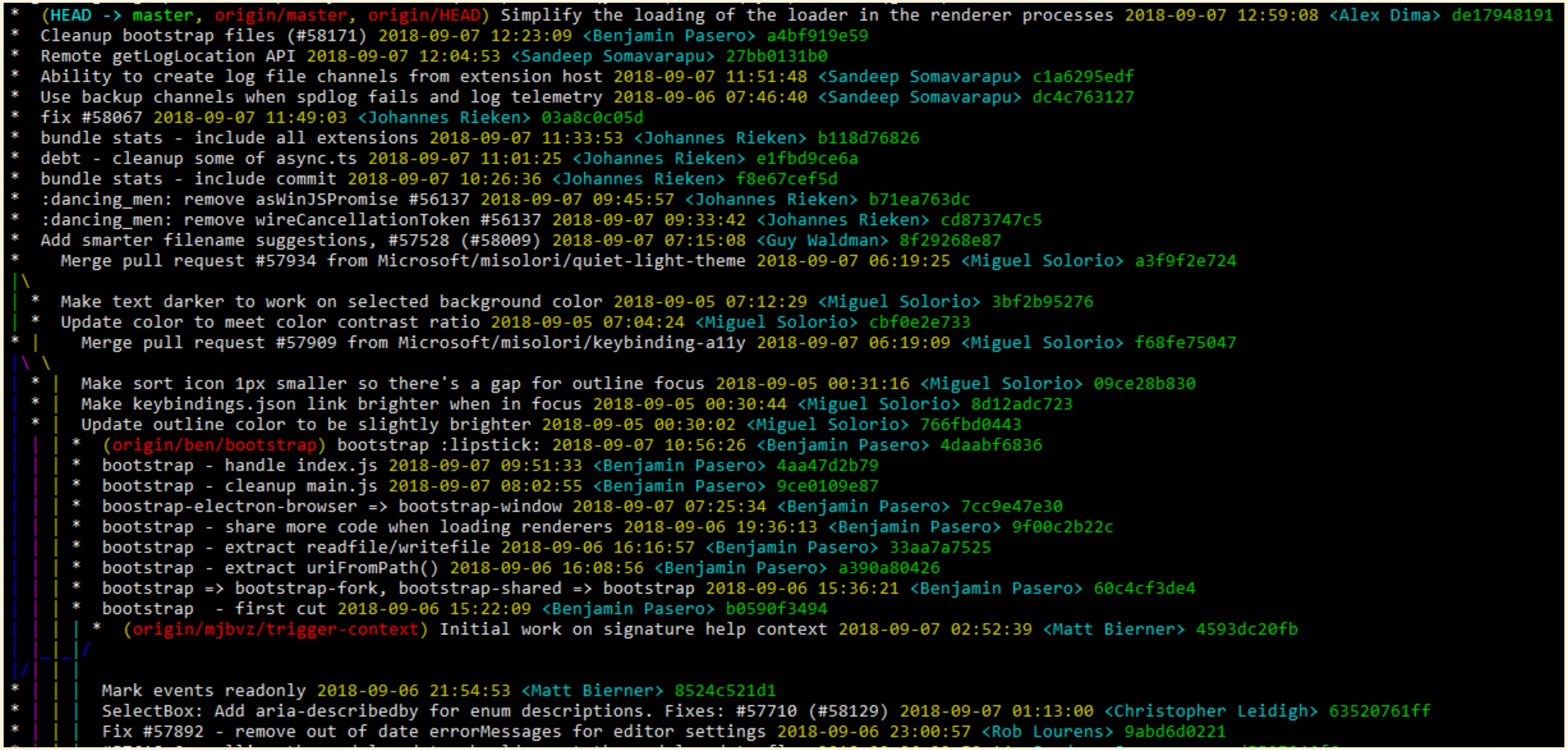
--oneline
This is a shorthand for "--pretty=oneline --abbrev-commit" used together.

$ git log --graph --all --oneline



```
* de17948191 (HEAD -> master, origin/master, origin/HEAD) Simplify the loading of the loader in the renderer processes
* a4bf919e59 Cleanup bootstrap files (#58171)
* 27bb0131b0 Remote getLogLocation API
* c1a6295edf Ability to create log file channels from extension host
* dc4c763127 Use backup channels when spdlog fails and log telemetry
* 03a8c0c05d fix #58067
* b118d76826 bundle stats - include all extensions
* e1fbd9ce6a debt - cleanup some of async.ts
* f8e67cef5d bundle stats - include commit
* b71ea763dc :dancing_men: remove asWinJSPromise #56137
* cd873747c5 :dancing_men: remove wireCancellationToken #56137
* 8f29268e87 Add smarter filename suggestions, #57528 (#58009)
*   a3f9f2e724 Merge pull request #57934 from Microsoft/misolori/quiet-light-theme
|\
| * 3bf2b95276 Make text darker to work on selected background color
| * cbf0e2e733 Update color to meet color contrast ratio
* |   f68fe75047 Merge pull request #57909 from Microsoft/misolori/keybinding-a11y
|\ \
| * | 09ce28b830 Make sort icon 1px smaller so there's a gap for outline focus
| * | 8d12adc723 Make keybindings.json link brighter when in focus
| * | 766fbd0443 Update outline color to be slightly brighter
| | | * 4daabf6836 (origin/ben/bootstrap) bootstrap :lipstick:
| | | * 4aa47d2b79 bootstrap - handle index.js
| | | * 9ce0109e87 bootstrap - cleanup main.js
| | | * 7cc9e47e30 boostrap-electron-browser => bootstrap-window
| | | * 9f00c2b22c bootstrap - share more code when loading renderers
| | | * 33aa7a7525 bootstrap - extract readfile/writefile
| | | * a390a80426 bootstrap - extract uriFromPath()
| | | * 60c4cf3de4 bootstrap => bootstrap-fork, bootstrap-shared => bootstrap
| | | * b0590f3494 bootstrap  - first cut
| | | | * 4593dc20fb (origin/mjbvz/trigger-context) Initial work on signature help context
| |_|_|/
|/| | |
* | | | 8524c521d1 Mark events readonly
* | | | 63520761ff SelectBox: Add aria-describedby for enum descriptions. Fixes: #57710 (#58129)
* | | | 9abd6d0221 Fix #57892 - remove out of date errorMessages for editor settings
```

```
$ git log --graph --all ✂
--pretty=format:"%C(auto)%d %s %C(yellow)%ad %C(cyan)<%an> %C(green)%h" ✂
--date="format-local:%Y-%m-%d %H:%M:%S"
```

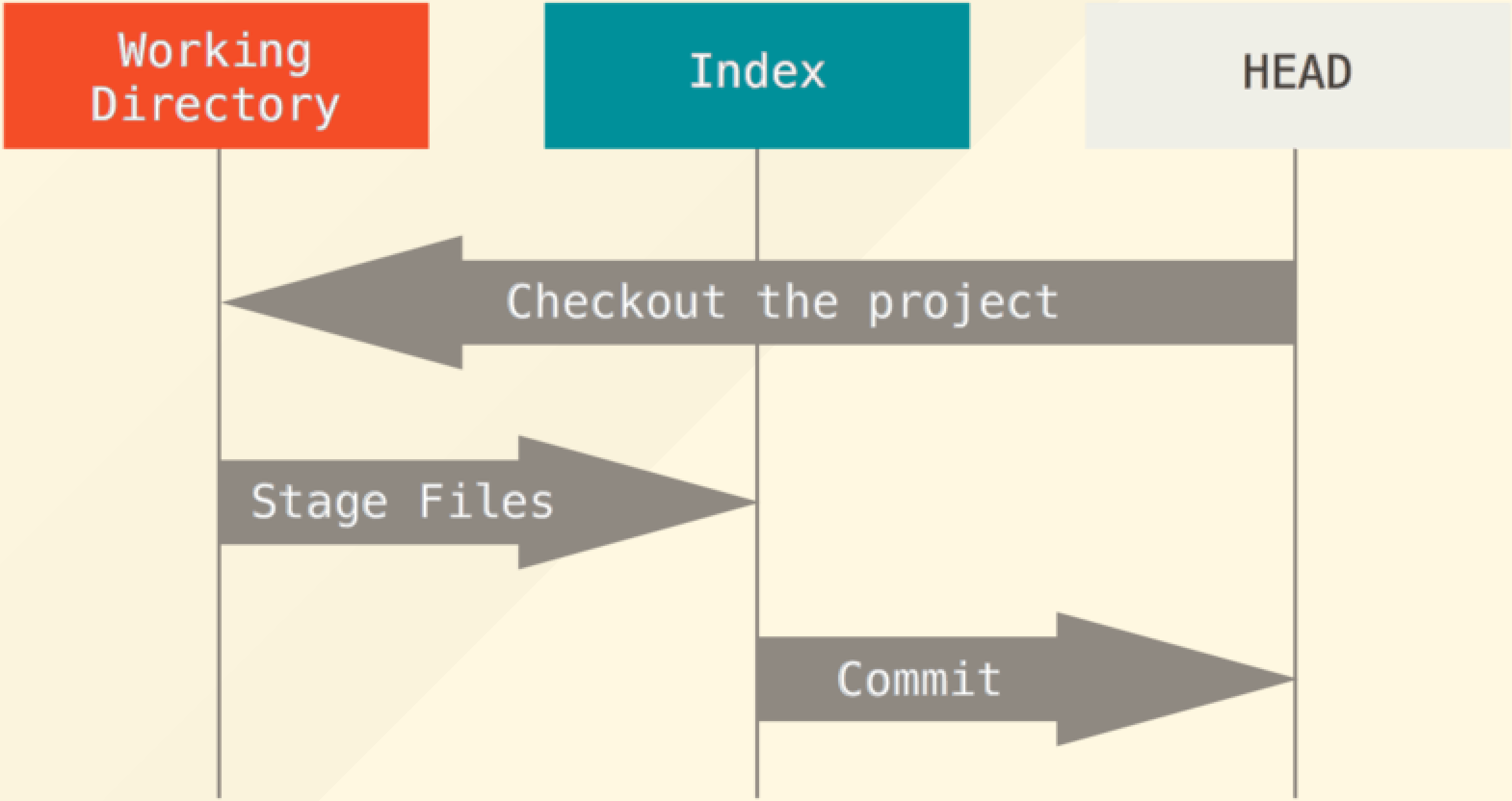


```
*  (HEAD -> master, origin/master, origin/HEAD) Simplify the loading of the loader in the renderer processes 2018-09-07 12:59:08 <Alex Dima> de17948191
*  Cleanup bootstrap files (#58171) 2018-09-07 12:23:09 <Benjamin Pasero> a4bf919e59
*  Remote getLogLocation API 2018-09-07 12:04:53 <Sandeep Somavarapu> 27bb0131b0
*  Ability to create log file channels from extension host 2018-09-07 11:51:48 <Sandeep Somavarapu> c1a6295edf
*  Use backup channels when spdlog fails and log telemetry 2018-09-06 07:46:40 <Sandeep Somavarapu> dc4c763127
*  fix #58067 2018-09-07 11:49:03 <Johannes Rieken> 03a8c0c05d
*  bundle stats - include all extensions 2018-09-07 11:33:53 <Johannes Rieken> b118d76826
*  debt - cleanup some of async.ts 2018-09-07 11:01:25 <Johannes Rieken> e1fbd9ce6a
*  bundle stats - include commit 2018-09-07 10:26:36 <Johannes Rieken> f8e67cef5d
*  :dancing_men: remove asWinJSPromise #56137 2018-09-07 09:45:57 <Johannes Rieken> b71ea763dc
*  :dancing_men: remove wireCancellationToken #56137 2018-09-07 09:33:42 <Johannes Rieken> cd873747c5
*  Add smarter filename suggestions, #57528 (#58009) 2018-09-07 07:15:08 <Guy Waldman> 8f29268e87
*    Merge pull request #57934 from Microsoft/misolori/quiet-light-theme 2018-09-07 06:19:25 <Miguel Solorio> a3f9f2e724
|\
| *  Make text darker to work on selected background color 2018-09-05 07:12:29 <Miguel Solorio> 3bf2b95276
| *  Update color to meet color contrast ratio 2018-09-05 07:04:24 <Miguel Solorio> cbf0e2e733
* |    Merge pull request #57909 from Microsoft/misolori/keybinding-a11y 2018-09-07 06:19:09 <Miguel Solorio> f68fe75047
|\ \
| * |  Make sort icon 1px smaller so there's a gap for outline focus 2018-09-05 00:31:16 <Miguel Solorio> 09ce28b830
| * |  Make keybindings.json link brighter when in focus 2018-09-05 00:30:44 <Miguel Solorio> 8d12adc723
| * |  Update outline color to be slightly brighter 2018-09-05 00:30:02 <Miguel Solorio> 766fbd0443
| | | *  (origin/ben/bootstrap) bootstrap :lipstick: 2018-09-07 10:56:26 <Benjamin Pasero> 4daabf6836
| | | *  bootstrap - handle index.js 2018-09-07 09:51:33 <Benjamin Pasero> 4aa47d2b79
| | | *  bootstrap - cleanup main.js 2018-09-07 08:02:55 <Benjamin Pasero> 9ce0109e87
| | | *  boostrap-electron-browser => bootstrap-window 2018-09-07 07:25:34 <Benjamin Pasero> 7cc9e47e30
| | | *  bootstrap - share more code when loading renderers 2018-09-06 19:36:13 <Benjamin Pasero> 9f00c2b22c
| | | *  bootstrap - extract readfile/writefile 2018-09-06 16:16:57 <Benjamin Pasero> 33aa7a7525
| | | *  bootstrap - extract uriFromPath() 2018-09-06 16:08:56 <Benjamin Pasero> a390a80426
| | | *  bootstrap => bootstrap-fork, bootstrap-shared => bootstrap 2018-09-06 15:36:21 <Benjamin Pasero> 60c4cf3de4
| | | *  bootstrap  - first cut 2018-09-06 15:22:09 <Benjamin Pasero> b0590f3494
| | | | *  (origin/mjbvz/trigger-context) Initial work on signature help context 2018-09-07 02:52:39 <Matt Bierner> 4593dc20fb
| |_|_|/
|/| | |
* | | |  Mark events readonly 2018-09-06 21:54:53 <Matt Bierner> 8524c521d1
* | | |  SelectBox: Add aria-describedby for enum descriptions. Fixes: #57710 (#58129) 2018-09-07 01:13:00 <Christopher Leidigh> 63520761ff
* | | |  Fix #57892 - remove out of date errorMessages for editor settings 2018-09-06 23:00:57 <Rob Lourens> 9abd6d0221
```

git reset

Can be used to undo changes that are local to your computer

git reset - trees



git reset - commands

git reset overwrites these trees in a specific order

git reset --soft
move the branch HEAD points to
can be used to undo commits

git reset --mixed (default)
also make the Index look like HEAD
can be used to unstage changes

git reset --hard
also make the Working Directory look HEAD
can be used to undo changes

git reset - demo

git rebase interactive

git rebase --interactive
git rebase -i

Make a list of the commits which are about to be rebased. Let the user edit that list before rebasing.

```
Do not rebase commits that exist outside your repository.
If you follow that guideline, you’ll be fine.
If you don’t, people will hate you.
```

git-rebase-todo file

```
pick 2a6ed8c v4
pick d5a5839 v5
pick 010a3f0 v6

# Rebase fba439a..010a3f0 onto fba439a (3 commands)
#
# p, pick <commit> = use commit
# r, reword <commit> = use commit, but edit the commit message
# e, edit <commit> = use commit, but stop for amending
# s, squash <commit> = use commit, but meld into previous commit
# f, fixup <commit> = like "squash", but discard this commit's log message
# x, exec <command> = run command (the rest of the line) using shell
# d, drop <commit> = remove commit
# l, label <label> = label current HEAD with a name
# t, reset <label> = reset HEAD to a label
# m, merge [-C <commit> | -c <commit>] <label> [# <oneline>]
# .       create a merge commit using the original merge commit's
# .       message (or the oneline, if no original merge commit was
# .       specified). Use -c <commit> to reword the commit message.
# These lines can be re-ordered; they are executed from top to bottom.
#
# If you remove a line here THAT COMMIT WILL BE LOST.
#
#       However, if you remove everything, the rebase will be aborted.
```

git rebase interactive - commands

pick
use commit
default, use the commit as is

reword
use commit, but edit the commit message
do not change the message in the todo file, you will be prompted for a new message

edit
use commit, but stop for amending
can be used to add, remove or change files in the commit
you will have to **git commit --amend** followed by **git rebase --continue** to complete this step

git rebase interactive - commands

squash

use commit, but meld into previous commit

can be used to combine multiple commits into one

you will be prompted for which commit message to keep

fixup

like "squash", but discard this commit's log message

skip the prompt for the commit message if the first message is the one you would like to keep

git rebase interactive - commands

drop
remove commit
e.g. remove merge commits that become irrelevant after rebasing

label
label current HEAD with a name
to create a reference that can later be used

git worktree

Manage multiple working trees attached to the same repository.

A git repository can support multiple working trees, allowing you to check out more than one branch at a time

This new working tree is called a "**linked working tree**" as opposed to the "**main working tree**" prepared by "**git init**" or "**git clone**"

A repository has one main working tree and zero or more linked working trees.

git worktree - commands

add

```
git worktree add <path> <remote>/<branch>
```

Create <path> and checkout into it. The new working directory is linked to the current repository, sharing everything except working directory specific files such as HEAD, index, etc.

Every branch can only exist in one worktree. You can create a new branch with the option `-b <branch>`

list

List details of each worktree. The main worktree is listed first, followed by each of the linked worktrees.

git worktree - commands

lock

Prevent it from being pruned automatically in case it is on a portable device or networkshare which is not always mounted.
Also prevents it from being moved or deleted.

unlock

Unlock a working tree, allowing it to be pruned, moved or deleted.

git worktree - commands

move

Move a working tree to a new location.

remove

Remove a working tree. Only clean working trees (no untracked files and no modification in tracked files) can be removed. Unclean working trees or ones with submodules can be removed with --force. The main working tree cannot be removed.

prune

Prune working tree information in $GIT_DIR/worktrees.

git alias

Now that we have seen some advanced commands we can make it easier to use them wit `git alias`.

For example, to correct the usability problem you encountered with unstaging a file you can add your own unstage alias to Git:

```
$ git config --global alias.unstage 'reset HEAD --'
```

This makes the following two commands equivalent:

```
$ git unstage fileA
$ git reset HEAD -- fileA
```

git alias

Some more commands

Alias	Command
s	status -sb
last	log -1 HEAD
tree	log --all --oneline --graph
branches	branch -a
stashes	stash list
aliases	!git config -l | grep alias | cut -c 7-
echo	!echo "$1" && :
chs	!git checkout && git s && :

git alias

Share aliases in team

You can put all aliases inside a separate config file

```
[alias]
  a = add
  ..
```

Add this file to your `.config` file as such

```
[include]
    path = aliases.txt
```

Advanced Git

- Properly configure Git
- Advanced Git commands
- **Customize Git with Git Hooks**
- Further Reading

Git hooks

Git has a way to fire off custom scripts when certain important actions occur.

Client-side hooks are triggered by operations such as committing and merging.

Server-side hooks run on network operations such as receiving pushed commits.

Git hooks - samples

Some samples can already be found in the .git\hooks folder

```
applypatch-msg.sample
commit-msg.sample
fsmonitor-watchman.sample
post-update.sample
pre-applypatch.sample
pre-commit.sample
pre-push.sample
pre-rebase.sample
pre-receive.sample
prepare-commit-msg.sample
update.sample
```

Git hooks - commit-msg example

A sample ruby script that will check if the commit message starts with a Jira ticket number.

```ruby
#!/usr/bin/env ruby
message_file = ARGV[0]
message = File.read(message_file)

$regex = /JIRA-(\d+) .+/

if !$regex.match(message)
  puts "[ERROR] Your message should start with a Jira ticket number."
  exit 1
end
```

When executed in the console it will look like this and the commit is not completed.

```
$ git commit -m blah
[ERROR] Your message should start with a Jira ticket number.
```

Stash unstaged changes before running tests

Ensure that code that isn’t part of the commit isn’t tested within your pre-commit script.

```
# pre-commit.sh
STASH_NAME="pre-commit-$(date +%s)"
git stash save -q --keep-index $STASH_NAME

# Test prospective commit
...

STASHES=$(git stash list)
if [[ $STASHES == "$STASH_NAME" ]]; then
  git stash pop -q
fi
```

Skip the pre-commit hook

```
git commit --no-verify
```

This option bypasses the pre-commit and commit-msg hooks

Advanced Git

- Properly configure Git
- Advanced Git commands
- Customize Git with Git Hooks
- **Further Reading**

Further Reading

Git Reference Manual

https://git-scm.com/docs

The official and comprehensive man pages that are included in the Git package itself

GitHub Cheat Sheet

https://services.github.com/on-demand/downloads/github-git-cheat-sheet.pdf

Summarizes commonly used Git command line instructions for quick reference

Visual Git Cheat Sheet

http://ndpsoftware.com/git-cheatsheet.html

An interactive cheat sheet from NDP Software

Pro Git book

https://git-scm.com/book

Free book by Scott Chacon and Ben Straub. Also available on amazon.com

Pluralsight - Mastering Git Paolo Perrotta

https://app.pluralsight.com/library/courses/mastering-git/