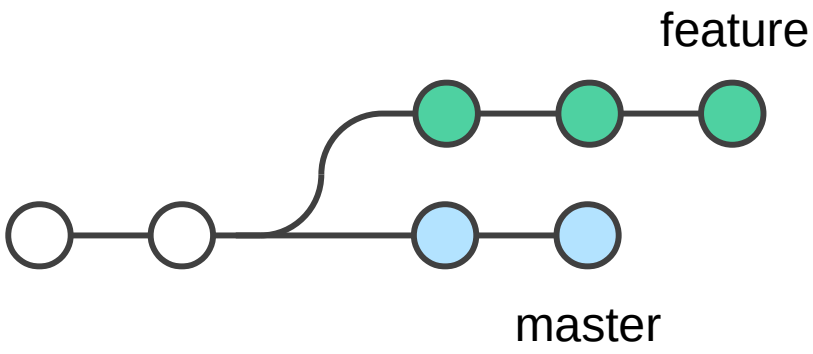
the NeRd Talks

git rebase --interactive

Relatore: Giuseppe Sucameli

git rebase (recap)

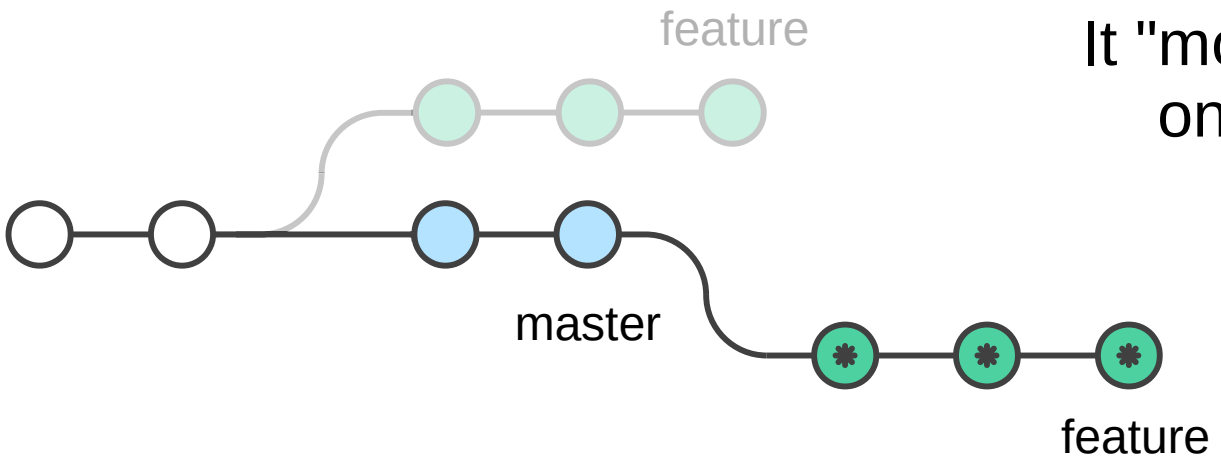
Rebase is a way to reintregrate changes from a branch (e.g. ***master***) branch into a different branch (e.g. ***feature***).



git rebase (recap)

Rebase is a way to reintregrate changes
from a branch (e.g. ***master***)
into a different branch (e.g. ***feature***).

= brand new commit



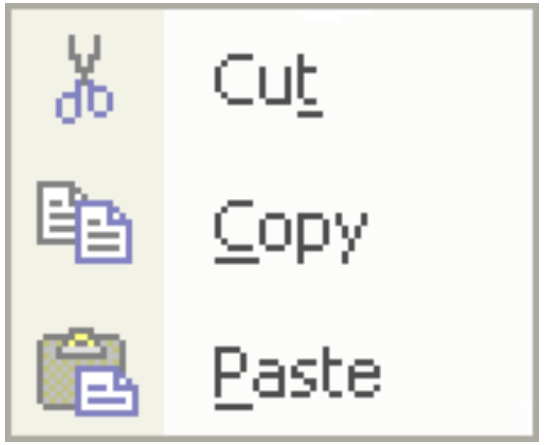
It "moves" ***feature*** branch to start
on tip of ***master*** branch

Rebase **re-writes** the history
by creating brand new commits.

```
git checkout feature
git rebase master
```

git rebase --interactive

Rebasing interactively means that you have a chance to edit the commits which are rebased.



-i, --interactive

Make a list of the commits which are about to be rebased.

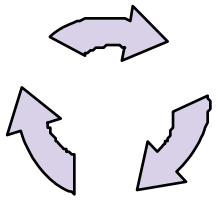
Let the user edit that list before rebasing.

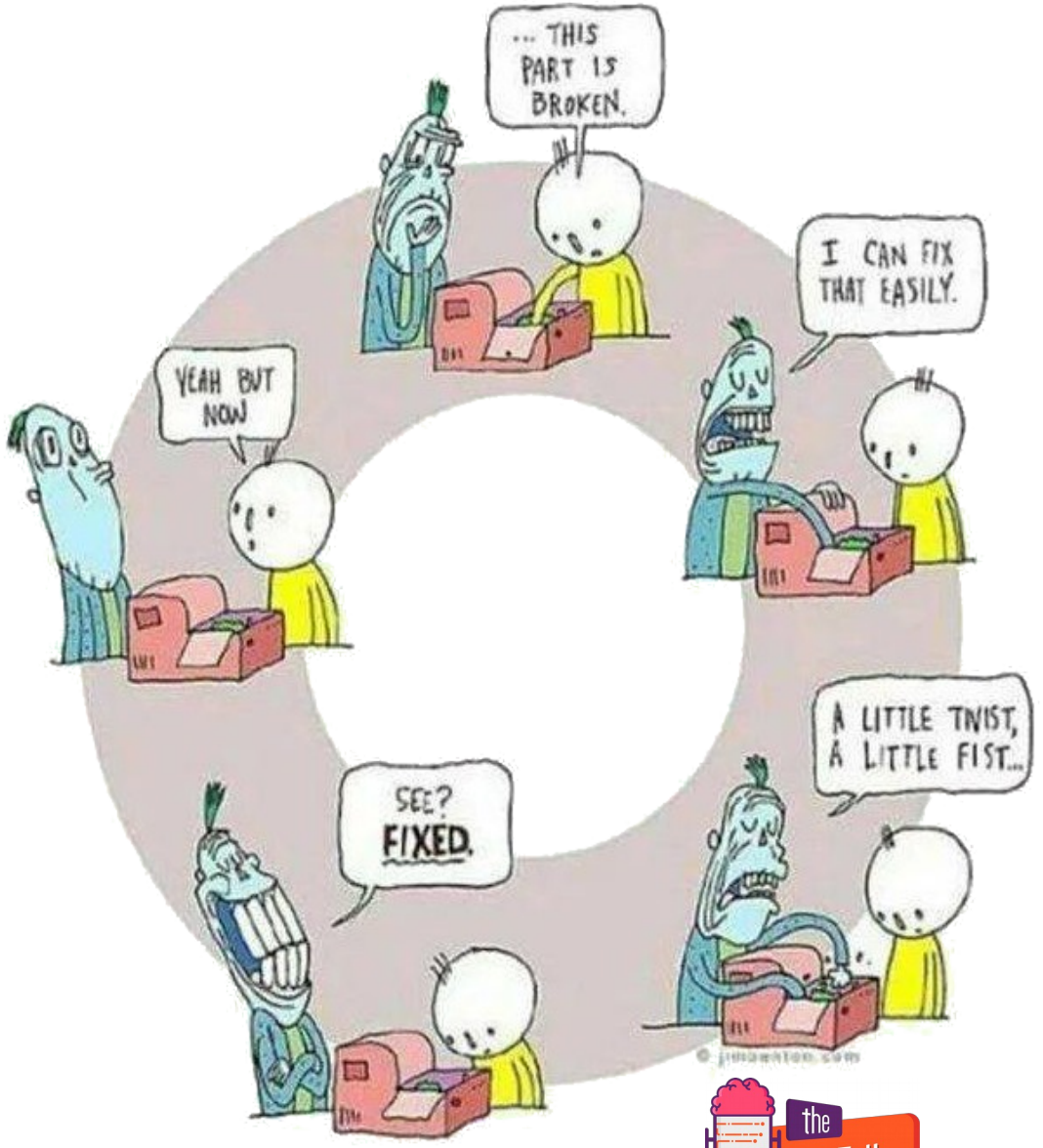
This mode can also be used to split commits.

The Coding Cycle

Steps

Coding
→ Try… Fail
→ → Fix
→ Try again… OK
Coding
→ Try… OK
Coding
→ Try… Fail
→ → Fix
→ Try… Fail
→ → Fix
Coding
…

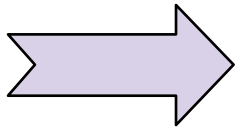




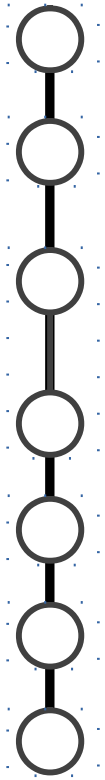
The Coding Cycle

Steps

```
Coding
→ Try… Fail
→ → Fix
→ Try again… OK
Coding
→ Try… OK
Coding
→ Try… Fail
→ → Fix
→ Try… Fail
→ → Fix
Coding
…
```



Commits

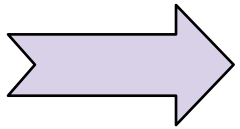


```
First HTTP client impl (GET)
   Fix: increase conn timeout
Add POST method
Add SSL support
   Fix: need newer libopenssl
   Fix increase conn timeout
Configure conn timeout
  …
```

The Coding Cycle

Steps

```
Coding
→ Try… Fail
→ → Fix
→ Try again… OK
Coding
→ Try… OK
Coding
→ Try… Fail
→ → Fix
→ Try… Fail
→ → Fix
Coding
…
```



Commits

First HTTP client impl (GET)

Fix: increase conn timeout

Add POST method

Add SSL support

Fix: need newer libopenssl

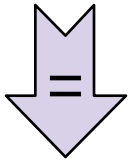
Fix increase conn timeout

Configure conn timeout

…

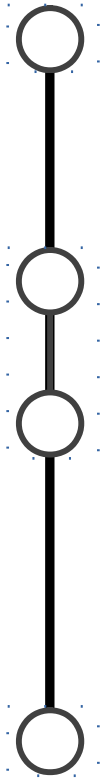
The Coding Cycle

Reorder commits
+
Join related ones



Clean history
+
Easier and faster reviews

Commits



`First HTTP client impl (GET)`

`Add POST method`

`Add SSL support`

`Configure conn timeout`

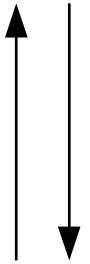
…

git rebase --interactive

Usage: `git rebase -i <after-this-commit>`

git log --oneline

```
6887e58 (HEAD -> master) ignore retry files
d50b001 (origin/master, origin/HEAD) init     KCN w/ fw 4.9.9: post upgr + HF1
e12fdda add playbooks to be executed post fw 4.9.9 upgrade
1502dc4 WI-16980: HF1 add hotel billing tables to cdrdb repset
fdf64c9 make PostFwUpgrade playbooks using reusable tasks
7580765 create reusable tasks
cd9ea80 set ansible_user for WI staging KCNs
61a8cb5 update ansible readme.md
89ceb20 move .gitignore to the repo root
84e2eb5 add ansible config and playbooks from WI env
ce76861 (initial) Add README.md
```



git rebase -i initial

```
pick 84e2eb5 add ansible config and playbooks from WI env
pick 89ceb20 move .gitignore to the repo root
pick 61a8cb5 update ansible readme.md
pick cd9ea80 set ansible_user for WI staging KCNs
pick 7580765 create reusable tasks
pick fdf64c9 make PostFwUpgrade playbooks using reusable tasks
pick 1502dc4 WI-16980: HF1 add hotel billing tables to cdrdb repset
pick e12fdda add playbooks to be executed post fw 4.9.9 upgrade
pick d50b001 init     KCN w/ fw 4.9.9: post upgr + HF1
pick 6887e58 ignore retry files

# Rebase ce76861..6887e58 onto ce76861 (10 commands)
```

Note: *git rebase -i* displays commits reversed compared to *git log*

git rebase --interactive

Interactive rebase allows to:

- Reorder commits
- Drop commits
- Join commits
- Edit commits
  - Insert new commits
  - Modify an existent commit (its content and/or its commit message)
  - Split a commit into multiple ones

git rebase --interactive

Interactive rebase commands:

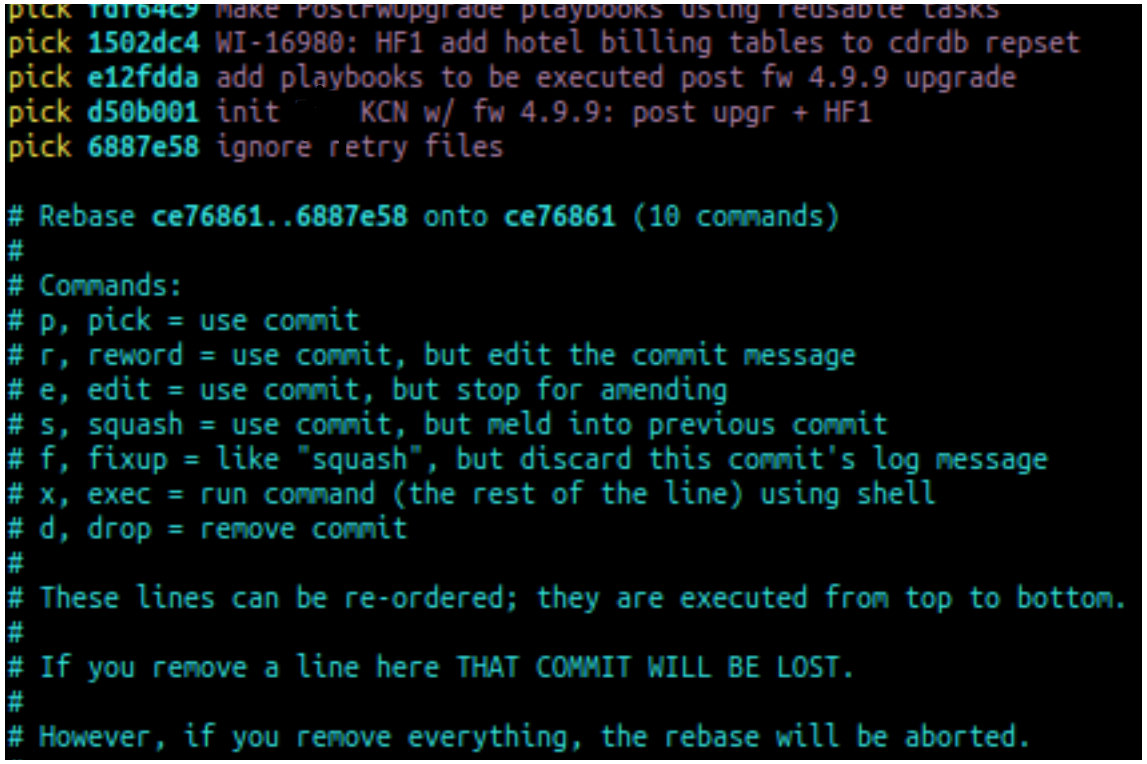
- **pick (p)**: keep the commit
- **drop (d)**: remove the commit
- **edit (e)**: stop rebase, allow to change/amend commit
- **fixup (f)**: meld commit into previous one
- **squash (s)**: similar to “fixup”, but keep the commit message
- **reword (r)**: edit the commit message

git rebase --interactive

Don’t worry!

There’s not need to remember all the commands.

git helps you :)



git rebase --interactive

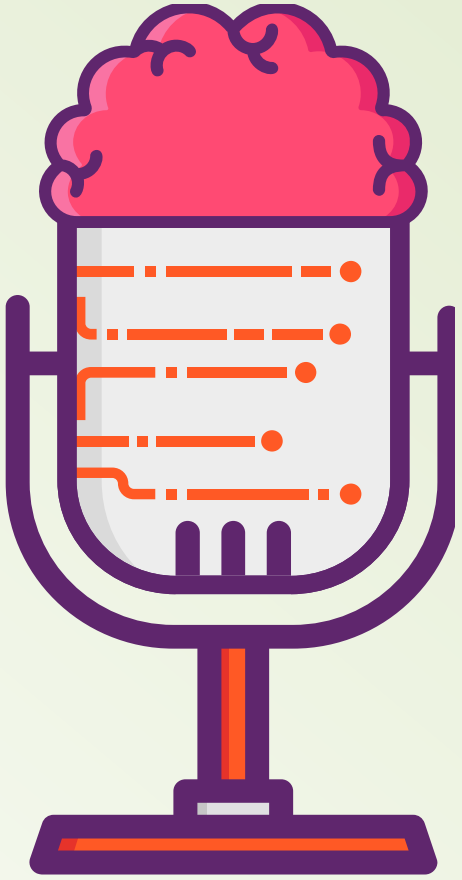
the NeRd Talks

git rebase --interactive

Remember: rebase cannot be reverted.

Don’t forget to **create a branch** before doing it!

```
git branch pre-rebase
git rebase -i HEAD~5 # last 5 commits
```



git rebase --interactive

That’s all folks!

Questions?