

```
## rebase
```

```
#           master
#  ---o---o---o
#      \
#      A---B topic
```

```
(topic)
```

```
$ git fetch
```

```
$ git rebase master
```

```
#           master
#  ---o---o---o
#      \       \
#      A---B   A'---B' topic
```

## why rebase?

```
#                                     master
#   ---o---o---o---o---x
#       \           \       /
#       A---B---x---C topic
```

```
(topic)
$ git commit
$ git commit
$ git merge master
$ git commit
(master)
$ git merge topic
```

# merge commits do not add value  
# difficult to read history

```
(topic)
$ git rebase master
#                                     master
#   ---o---o---o---o
#       \                   \
#       A---B---C   A'---B'---C' topic
```

```
## rebase -i
```

```
# Commit often, perfect later.
```

```
#           master
#  ---o---o---o
#           \
#           A---B---C---D topic
```

```
(topic)
```

```
$ git rebase -i master|head~4
```

```
pick A
```

```
pick B
```

```
pick C
```

```
pick D
```

```
# Commands:
```

```
# p, pick = use commit
```

```
# r, reword = use commit, but edit the commit message
```

```
# e, edit = use commit, but stop for amending
```

```
# s, squash = use commit, but meld into previous  
commit
```

```
# f, fixup = like "squash", but discard this commit's  
log message
```

```
# x, exec = run command (the rest of the line) using  
shell
```

```
# d, drop = remove commit
```

```
# scenario 1: edit, pick, drop
```

```
edit A  
pick B  
drop C  
drop D
```

```
stopped at A...  
(some changes...)  
$ git add .  
$ git rebase --continue
```

```
#               master  
#   ---o---o---o  
#           |\  
#           | A---B---C---D  
#           \  
#           A'--B' topic
```

```
# scenario 2: reorder commits
```

```
pick A  
pick C  
pick B  
pick D
```

```
#               master  
#   ---o---o---o  
#           |\  
#           | A---B---C---D  
#           \  
#           A'--C'--B'--D' topic
```

```
# scenario 3: pick, fixup, reorder, drop
```

```
pick A
```

```
fixup C
```

```
pick B
```

```
#edit D (dropping D by commenting)
```

```
# results in
```

```
AC (commits A+C)
```

```
B (commit B)
```

```
#               master
#   ---o---o---o
#           | \
#           |  A---B---C---D
#           \
#           AC'--B' topic
```

```
## push after rebase
```

```
(topic)
```

```
$ git push
```

```
$ git fetch
```

```
$ git rebase master
```

```
#                               master
#  ---o---o---o
#      \      \
#      \      \ A'---B' topic
#      \      \
#      \      \ A---B topic{-1}
#              \ origin/topic
```

```
$ git push topic
```

```
rejected (non-fast-forward)
```

```
do a pull/merge first
```

```
$ git push -f
```

```
#                               master
#  ---o---o---o
#      \      \
#      \      \ A'---B' topic
#      \      \ origin/topic
#              \
#              A---B
```

```
# golden rule of rebase
```

```
# I normally don't rebase a branch,  
# but when I do, I do master.  
#     - The Most Interesting Rebaser in the World  
#     #yolo
```

```
# don't rebase public branches
```

```
#     last      hotfix  
#   ---o---o---o  
#       \  
#       A---B master (me, others)
```

```
# I rebase master  
(master)  
$ git rebase hotfix  
$ git push -f
```

```
#             hotfix      origin/master  
#   ---o---o---o---A'---B' master (me)  
#       \  
#       A---B master (others)
```

```
# Others merge origin/master
(master)
$ git merge origin/master
$ git push
```

```
#                      hotfix
#  ---o---o---o---A'---B'
#      \                               \  origin/master
#      A---B-----x master (me, others)
```

```
# Merge is better compared to rebase
```

```
# I merge master
(master)
$ git merge hotfix
$ git push
```

```
#                      hotfix
#  ---o---o---o
#      \       \
#      A---B---x master (me, others)
#                      origin/master
```

```
# Others can fast-forward master now
```