

```
## merge
```

```
#   ---o---o master
#       \
#         --o
#         dev
```

```
(master)
$ git merge dev
```

```
#   ---o---o---x master
#       \       /
#         --o--
#         dev
```

```
(master)
$ git merge pu maint
```

```
#   ---o---o---x master
#       |\       /|
#       | --o-- | pu
#       \       /
#       `--o--' maint
```

```
# Linux kernel has 66 branches merged together
# in 2014
```

```
## --ff (fast-forward) merge
```

```
#   ---o master
#       \
#       --o---o
#               dev
#
```

```
(master)
$ git merge dev
```

```
#   ---o
#       \      master
#       --o---o
#               dev
#
```

```
# Where was master previously?
```

```
## --no-ff merge
```

```
#   ---o master
#       \
#       --o---o
#               dev
```

```
(master)
```

```
$ git merge dev --no-ff
```

```
#   ---o-----x master
#       \       /
#       --o---o
#               dev
```

```
# .gitconfig
```

```
[merge]
```

```
ff = false
```

```
# --no-ff flag default
```

```
$ git merge dev
```

```
# for ff merge
```

```
$ git merge dev --ff
```

```
# .gitconfig
```

```
[branch "master"]
```

```
mergeOptions = --no-ff
```

```
# same for next, dev
```

```
## merge conflicts
```

```
# There are 3 things certain in life.
```

```
(death)
```

```
$ git merge taxes
```

```
<<<<<< HEAD
```

```
death
```

```
||||||| merged common ancestors
```

```
merge conflicts
```

```
=====
```

```
taxes
```

```
>>>>>> taxes
```

```
(master)
```

```
$ git merge dev
```

```
# conflicts
```

```
$ git status -s
```

```
UU a.txt # conflict (Unmerged, Updated)
```

```
 M b.txt # modified
```

```
 A c.txt # added
```

```
 R d.txt # renamed
```

```
 D e.txt # deleted
```

```
# get current|other branch's copy
```

```
$ git difftool --ours|--theirs file.txt
```

```
# get current branch's copy
```

```
$ git checkout --ours file.txt  
--theirs
```

```
# undo merge, even a successful merge
$ git reset --merge orig_head
# use again to go back to merge commit
```

```
[alias]
```

```
    undomerge = reset --merge orig_head
$ git undomerge
```

```
# How to see resolved conflicts before push?
```

```
$ git show head
diff --cc file.txt
- my change
- other change
++resolved change
```

```
# this does not work
$ git difftool head^!
```

```
## merge pitfalls
```

```
(feature)
```

```
$ git merge dev
```

```
# conflicts
```

```
# keep all files while merging with conflicts
```

```
(feature|MERGING)
```

```
$ git status -s
```

```
UU a.txt
```

```
  M b.txt
```

```
  M c.txt
```

```
  M d.txt
```

```
  M e.txt
```