



ZERO

---

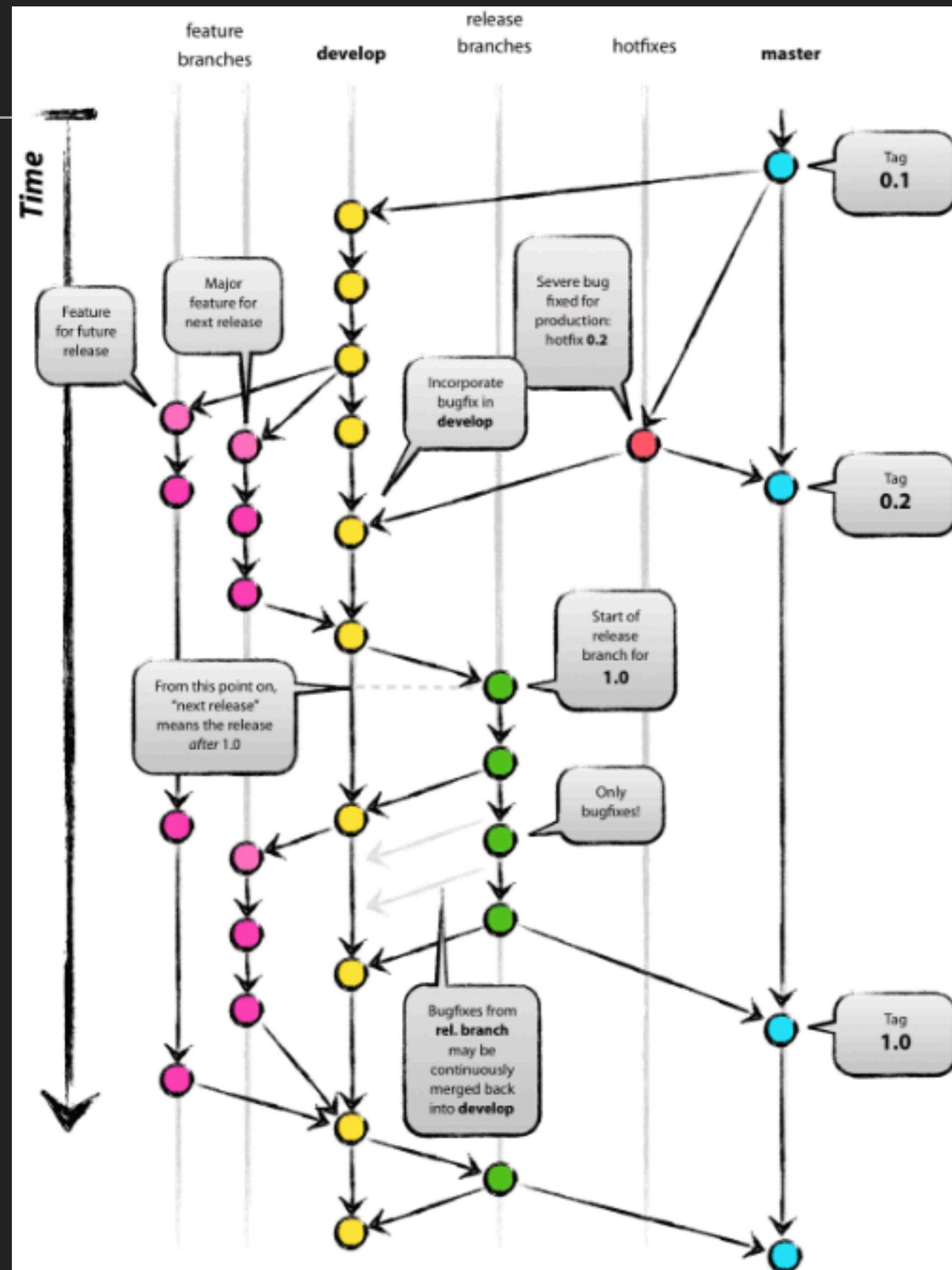
**GIT**





# BRANCH

- ▶ 本地分支
- ▶ 远程分支



## BRANCH

- ▶ `git branch --set-upstream-to=origin/<branch> localBranch`
- ▶ `git branch -D localBranch`

# ALIAS

- ▶ 少按一下是一下
- ▶ 更便于记忆

## ALIAS

- ▶ `git config --global alias.co checkout`
- ▶ `git config --global alias.br branch`
- ▶ `git config --global alias.ci commit`
- ▶ `git config --global alias.st status`
- ▶ `git config --global alias.unstage 'reset HEAD --'`
- ▶ `git config --global alias.last 'log -1 HEAD'`
- ▶ `git config --global alias.review '!sh -c \'git push origin HEAD:refs/for/$1\' -`

## 合并分支

### ▶ Rebase

- ▶ 将整个提交历史记录一并并进行移植合并
- ▶ 整个提交看起来更干净

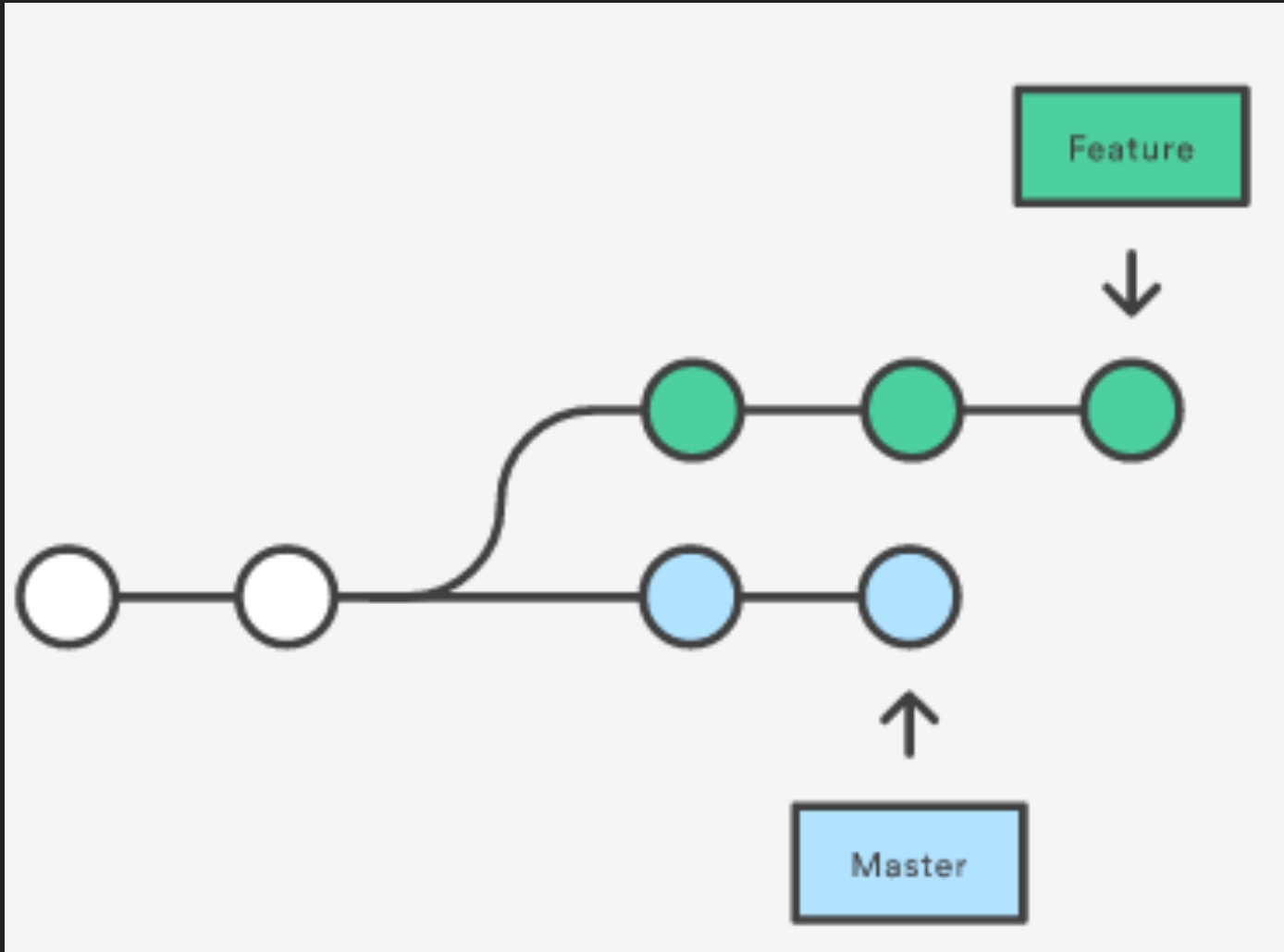
### ▶ merge

- ▶ 生成一个新的提交进行代码合并

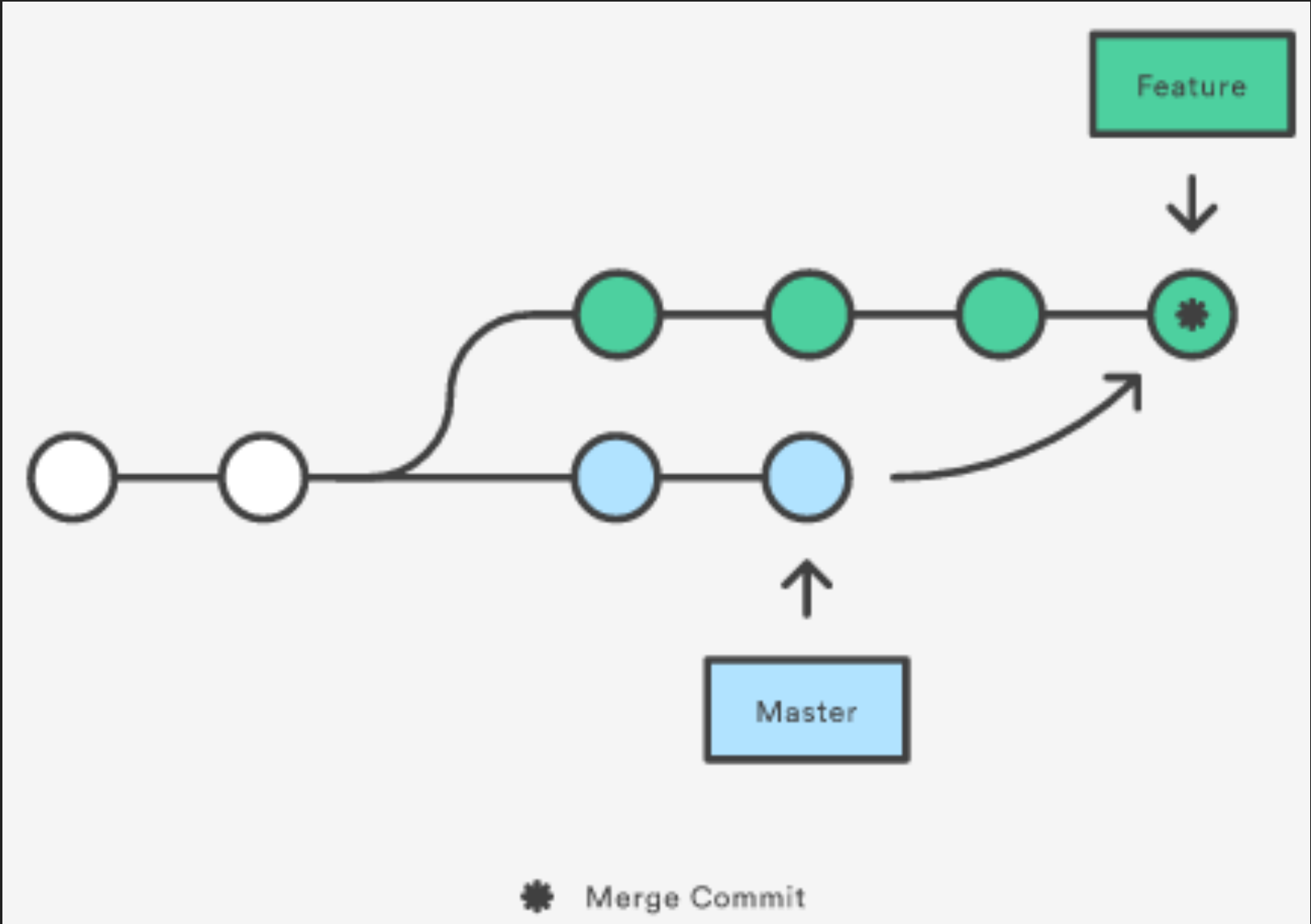
### ▶ Fast-Forwarded

- ▶ 并不会生成一个新的提交进行代码合并，而是直接将提交指针移动至最后的提交

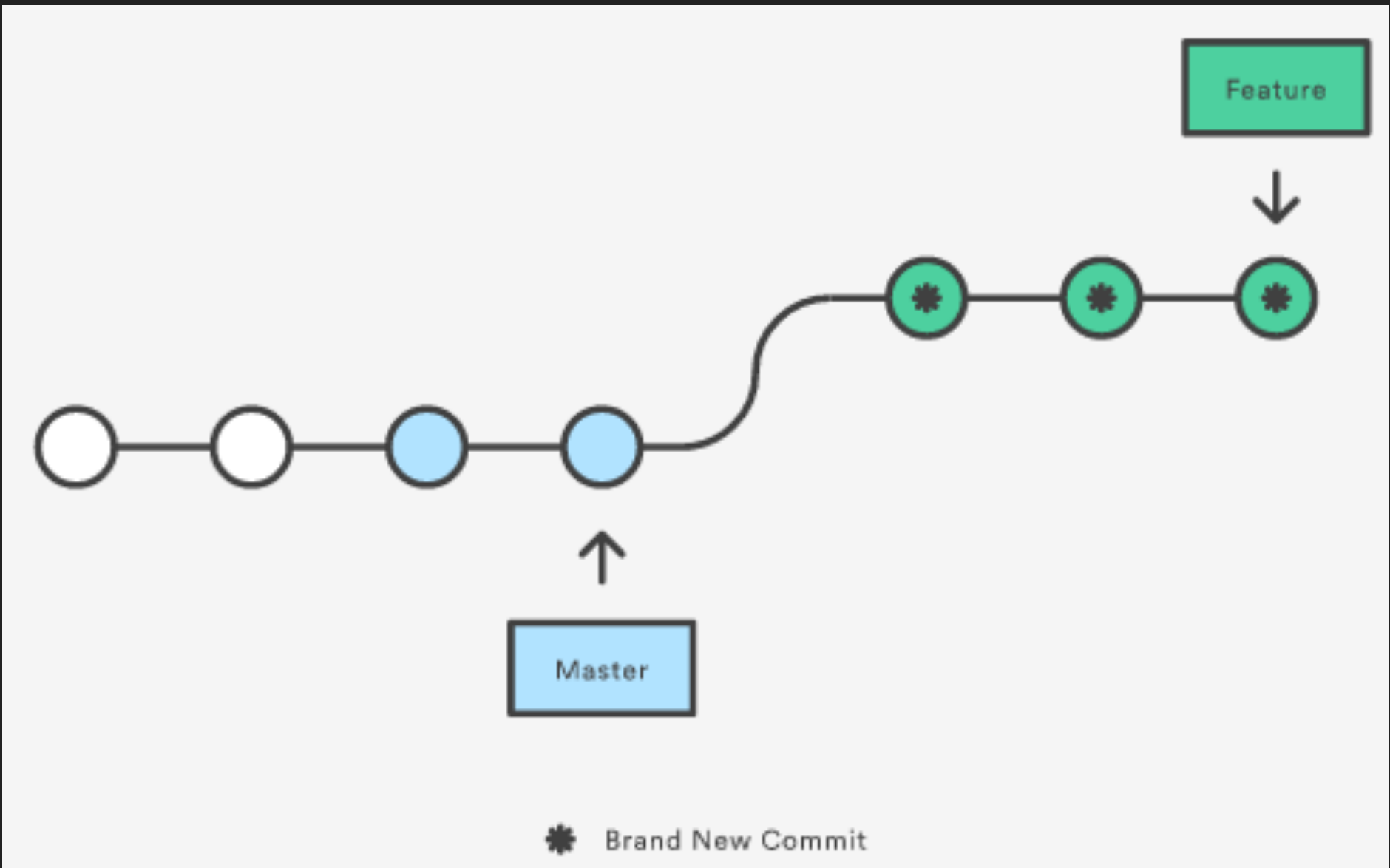
合并分支——图解



MERGE



REBASE





## 合并分支——注意事项

- ▶ 不要再公共分支进行rebase操作
- ▶ fast-forwarded 合并可以通过 `git merge --no-ff` 来添加合并提交

## REBASE还能做点什么？

- ▶ `git rebase --onto <new-parent> <old-parent>`
- ▶ `git rebase -i HEAD~3`

## 误删节点怎么办？

- ▶ reflog 命令

- ▶ 查看所有的操作内容，包括操作名称，操作的commit的sha-1 标示码
- ▶ 通过reset或者cherry-pick命令进行回滚

ZERO

---

## SHOW

- ▶ 查看操作详情
- ▶ `git show HEAD@{2}`



## CHERRY-PICK

- ▶ 用于指定某些提交合并到当前版本上
- ▶ `git cherry-pick sha-1 Code`
- ▶ `git cherry-pick master~3`

## 版本选择

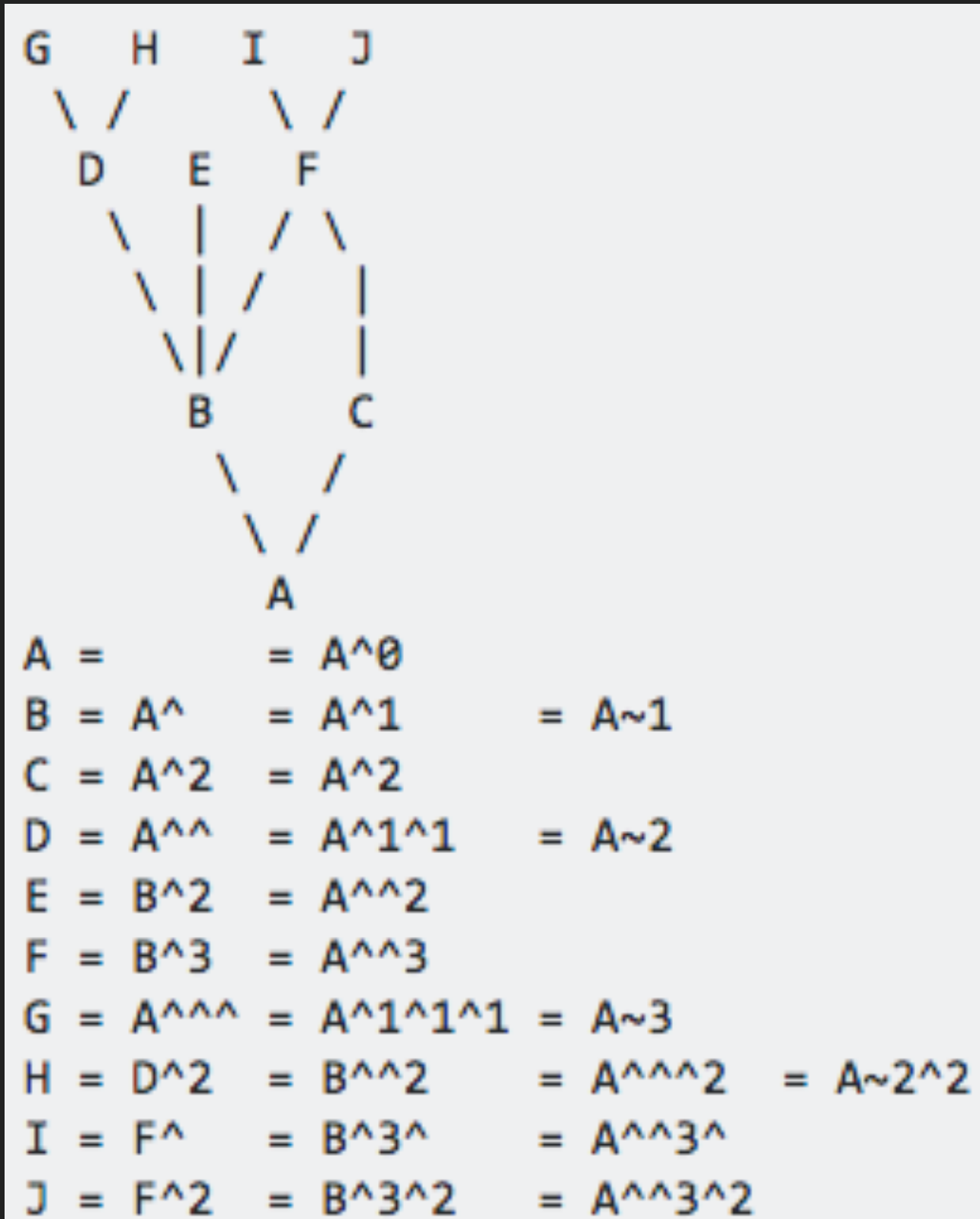
### ▶ HEAD~:

- ▶ HEAD~1 === HEAD~: 第一个父节点
- ▶ HEAD~2: 第一个父节点的父节点即祖父节点

### ▶ HEAD^

- ▶ HEAD^1 === HEAD^ === HEAD~: 第一个父节点
  - ▶ HEAD^2: 第二个父节点
- ▶ HEAD也可以使用某个sha-1来代替

# 版本选择



## 建议——COMMIT MESSAGE

- ▶ 提交信息的第一行用于概述
- ▶ 第二行为空
- ▶ 第三行起用于描述主体的修改，表述做了些什么和为什么这么做
- ▶ 如果需要结合Bug系统，可以考虑在最后加入相关的Bug号等



# GIT FLOW

我又来了

<https://github.com/nvie/gitflow>

