

Reference :

<http://www.liaoxuefeng.com/wiki/0013739516305929606dd18361248578c67b8067c8c017b000>

Git

`git reset --hard <commit id>`

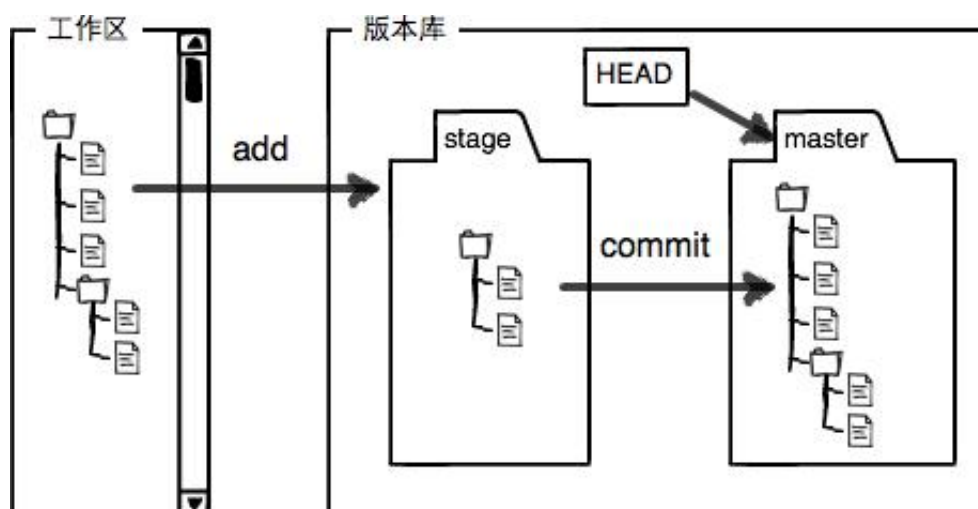
new file and modify file:

`git add <file>`

`git commit -m "comment"`

`git reflog <= check commit id history`

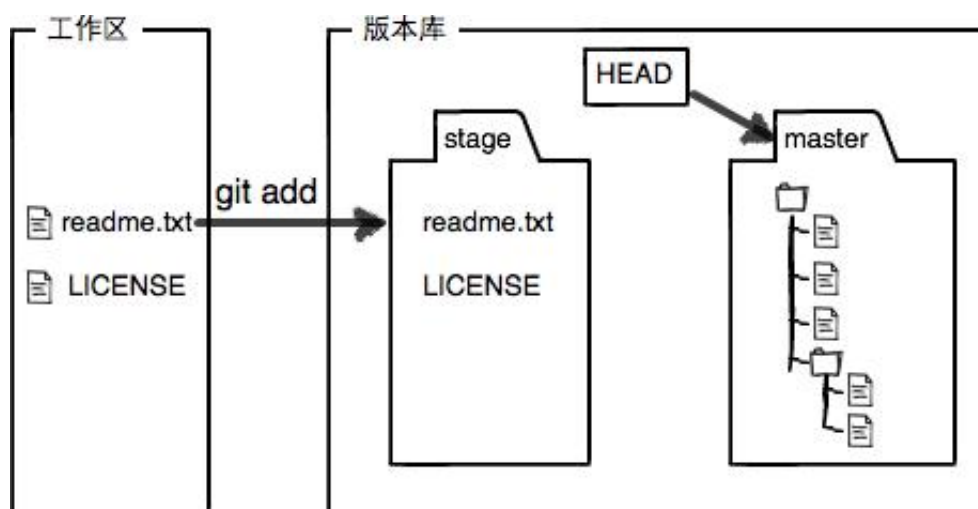
Repository



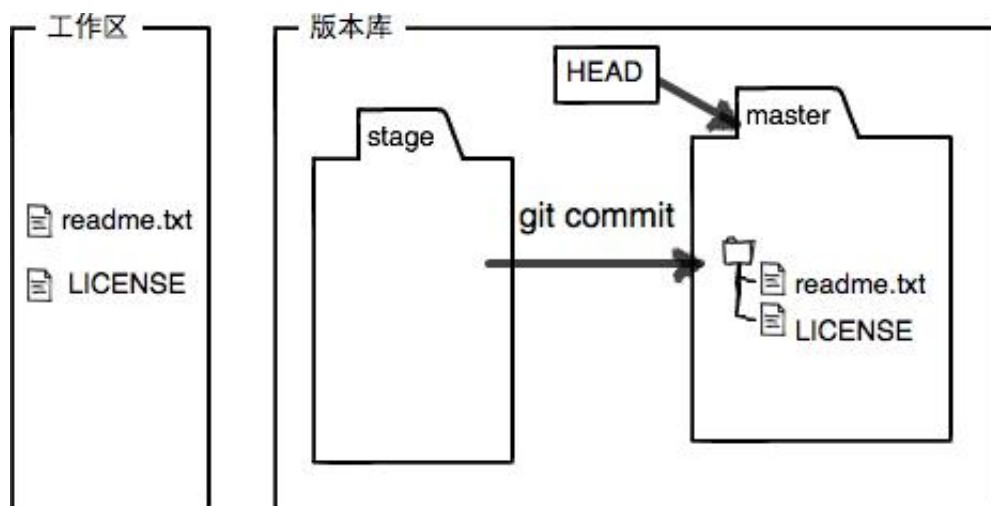
`git add` : 将文件添加到暂存区。

`git commit`: 将暂存区的内容添加到当前分支。

在创建repository时, `git` 会为我们创建一个唯一的master分支。



git commit



Git 跟踪管理的是修改而非文件

如果每次修改文件不把文件添加到暂存区, 则 commit 的时候, git 不会把修改的内容提交到 repository。

git checkout --<file>, 命令中的“--”, 很重要。没有她则变成了切换到了另外一个分支的命令。

Case 1: git reset HEAD, onstaged file, 把暂存区的修改回退到工作区

Case 2: git checkout -- file, 丢弃工作区的修改。

Case 3: 已经提交了不合适的版本到 repository 时, 用 git reset --hard <commit id>. 前提是你没有把修改提交到远程数据库。

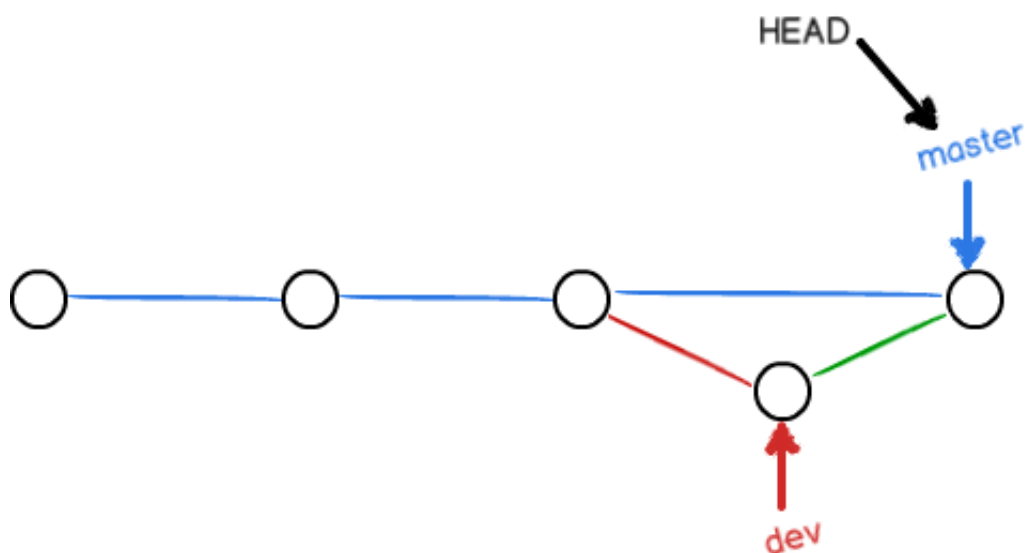
git remote add origin git@github.com:zs0723/learn git => 将本地的 repository 关联到远程

git push: 把本地库的内容推送到远程。加 -u, 会把本地 master 和远程 master 关联起来。

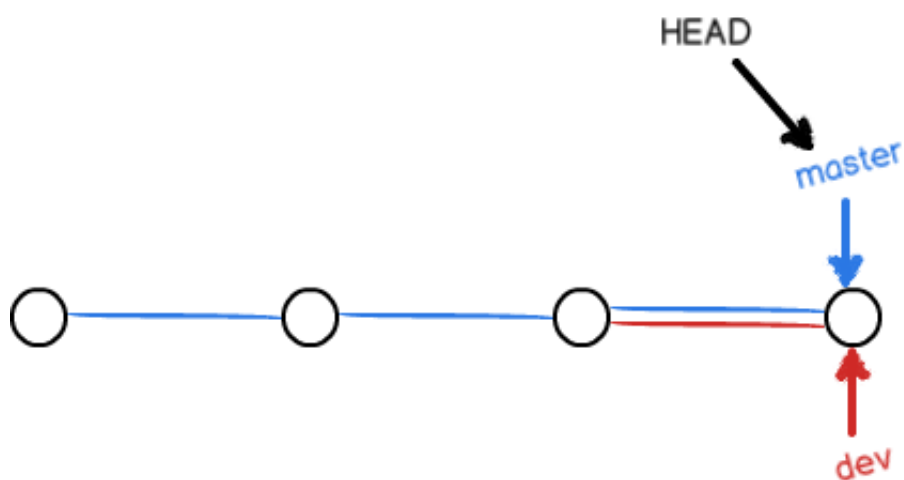
分支

git checkout -b dev: 创建 dev 分支, 并且切换到 dev

不使用 ff



使用ff



修复bug时，我们会通过创建新的bug分支进行修复。修复后，合并，然后删除。

当手头工作没有完成时，先把工作现场git stash，然后修复bug，修复后， git stash pop 回到现场。

查看远程库的信息 git remote, git remote -v

```
git remote -v
origin  git@github.com:zs0723/gitskills.git (fetch)
```

```
origin  git@github.com:zs0723/gitskills.git (push)
```

如果没有推送权限，则看不到push 地址。

git pull 失败，有两种方法：

```
git pull origin master
```

```
git branch --set-upstream-to=origin/<branch> dev
```

```
git tag -d <tagname>
```

```
git push origin :refs/tags/tagname 可以删除一个远程标签。
```

```
git push origin :refs/tags/v0.9
```

```
git book
```

```
https://git-scm.com/book/en/v2
```