

创建新仓库

```
git init
```

创建新的git仓库

当你本地创建了一个工作目录，你可以进入这个目录，使用'git init'命令进行初始化,Git以后就会对该目录下的文件进行版本控制

检出仓库

```
git clone /path/to/repository
```

创建一个本地仓库的克隆版本

```
git clone username@host:/path/to/repository
```

远端服务器上的仓库，创建一个本地仓库的克隆版本

添加与提交

```
git add <filename> or git add * or git add -A
```

将改动添加到缓存区

```
git commit -m "代码提交信息"
```

提交改动，现在，你的改动已经提交到了HEAD，但是还没到你的远端仓库

推送改动

```
git push origin master
```

将这些改动提交到远端仓库，可以把 master 换成你想要推送的任何分支

```
git remote add origin(项目别名) <server>
```

如果你还没有克隆现有仓库，并欲将你的仓库连接到远程服务器，你可以使用这个命令添加,例如

```
git remote add origin https://github.com/shaoguangcheng/TDIRLP.git
```

更新与合并

```
git pull
```

更新你的本地仓库至最新改动，以在你的工作目录中 获取（fetch）并合并（merge）远端的改动

```
git merge <branch>
```

合并其他分支到你的当前分支（例如 master）

替换本地改动

```
git checkout -- <filename>
```

假如你做错事（自然，这是不可能的），你可以使用如下命令替换掉本地改动。此命令会使用 HEAD 中的最新内容替换掉你的工作目录中的文件。已添加到缓存区的改动，以及新文件，都不受影响。

分支

分支是用来将特性开发绝缘开来的。在你创建仓库的时候，master 是“默认的”。在其他分支上进行开发，完成后再将它们合并到主分支上。

```
git checkout -b feature_x
```

创建一个叫做“feature_x”的分支，并切换过去

```
git checkout master
```

切换回主分支

```
git branch -d feature_x
```

把分支删掉

```
git push origin <branch>
```

将分支推送到远端仓库

流程

1. 在github上创建一个仓库
2. git init
3. touch README.md
4. git add -a
5. git commit -m “first commit”
6. git remote add origin <https://github.com/shaoguangcheng/dd.git>
7. git push -u origin master

Contact

chengshaoguang1291@126.com

