

## git 使用笔记

---

### 一、工作流程

- git Repository 仓库
  - 暂存区
  - 工作区
- 

### 二、初始化仓库

#### 基本信息配置

1. 设置用户名: `--global` 为全局配置, 不加 `--global` 为当前仓库下配置

```
git config --global user.name 'github用户名'
```

2. 设置用户邮箱:

```
git config --global user.email 'github注册邮箱'
```

3. 查看配置信息: 全局和子仓库都有时, 下边的配置会覆盖上边的

```
git config --list
```

---

### 三、linux基本命令

1. 创建文件夹

```
mkdir test
```

2. 查看目录 `-a` 查看全部文件, 包括隐藏文件  
`-s` 查看目录下文件数量

```
ls
```

3. 切换目录

```
cd
```

#### 4. 创建文件

```
touch index.html
```

#### 5. 编辑文件

```
<!-- 直接写入内容同时创建文件 -->  
echo 'something' >> index.html  
<!-- 使用vim编辑文件 -->  
vim index.html
```

i 进入编辑模式

esc 进入命令模式

: 进入底部命令行,可执行退出操作

wq 保存并退出

q 不保存, 直接退出

#### 6. 删除文件

```
<!-- 删除文件夹 递归删除 -->  
rm -rf test  
<!-- 删除文件 -->  
rm index.html
```

#### 7. 重命名或移动文件

```
<!-- index.html 重命名为demo.html -->  
mv index.html demo.html  
<!-- 移动demo.html 到 上一级目录 -->  
mv demo.html ../
```

---

## 四、具体操作

#### 1. 初始化当前目录 git init [dirname] 可以选定目录初始化仓库

```
git init  
git init [repository]
```

## 2. 将工作区内容添加至暂存区

```
git add [file name or directory name]
```

## 3. 提交至本地仓库 -m 备注 -a 直接跳过add步骤

```
git commit -m '添加备注说明'  
git commit -am '不执行add执行提交，添加备注说明'
```

## 4. 查看状态 -s 查看精简消息

```
git status
```

## 5. 克隆仓库

```
git clone [repository] [directory]
```

## 6. 取消暂存区内容，取消后就不会提交

```
git reset HEAD [file name]
```

## 7. 分支管理 checkout -b 直接创建并切换至此分支下

```
<!-- 查看分支 -->  
git branch  
<!-- 创建分支 -->  
git branch [branch name]  
<!-- 切换分支 -->  
git checkout [branch name]  
<!-- 创建并切换分支 -->  
git checkout -b [branch name]  
<!-- 删除分支 -->  
git branch -d [branch name]
```

## 8. 合并分支 切换至master合并分支 合并冲突后手动修改代码，重新提交

```
git merge [branch name]
```

## 9. 查看log

```
--author=username 查找指定用户提交日志
--oneline 按照行查看日志
--graph 图表格式查看日志
--reverse 倒叙查看
--decorate 查看标签
```

```
git log
```

## 10. 查看标签tag-a添加详细说明

```
<!-- 查看所有标签 -->
git tag
<!-- 设置标签 -->
git tag -a [tag name]
<!-- 删除标签 -->
git tag -d [tag name]
<!-- 查看版本修改内容 -->
git show [tag name]
<!-- 追加标签 git log 查询对应提交所在行 -->
git log --oneline
git tag -a [tag name] 2e2b797
```

## 11. 远程仓库

- 生成ssh key

```
ssh-keygen -t rsa -C "github注册邮箱"
```

- github主页setting中添加ssh
- 查看远程仓库

```
git remote
```

- 远程仓库详细信息

```
git remote -v
```

- 添加远程仓库

```
git remote add [shortname] [repository url]
```

- 首次提交至远程仓库使用 -u

```
git push -u origin master
```

## 12. 提交远程代码的正常步骤

- 需要ssh提前生成ssh key;
- git clone 克隆远程仓库;
- 在本地工作区, 修改, 添加等操作;
- push提交至远程仓库;