

# Git的使用

---

## 1.Git使用前的配置

在使用git前，需要告诉git你是谁，在向git仓库中提交时需要用到。

1. 配置提交人的姓名

```
1 | git config --global user.name 提交人姓名
```

2. 配置提交人的邮箱

```
1 | git config --global user.email 提交人邮箱
```

3. 查看git配置信息

```
1 | git config --list
```

如果需要对配置信息进行修改时，重复上述命令。

配置只需要执行一次。

## 2.Git提交步骤

- 初始化git仓库

```
1 | git init
```

- 查看文件状态

```
1 | git status
```

- 追踪文件

```
1 | git add 文件列表
```

- 向仓库中提交代码

```
1 | git commit -m 提交信息
```

- 查看提交记录

```
1 | git log
```

## 3.Git撤销

- 用暂存区中的文件覆盖工作目录中的文件

```
1 | git checkout 文件
```

- 将文件从暂存区删除

```
1 | git rm --cached 文件
```

- 将git仓库中指定的更新记录恢复出来，并且覆盖暂存区和工作目录

```
1 | git rest --hard commitID
```

## Git分支

---

### 1.分支细分

- 主分支(master): 第一次向git仓库中提交更新记录时自动创建的一个分支
- 开发分支(develop): 作为开发分支，基于master分支创建
- 功能分支(feature): 作为开发具体功能的分支，基于开发分支创建

### 2.分支命令

- 查看分支

```
1 | git branch
```

- 创建分支

```
1 | git branch 分支名称
```

- 切换分支

```
1 | git checkout 分支名称  
2 | 必须保存当前分支的改变，才可以切换分支
```

- 合并分支

```
1 | git merge 来源分支
```

- 删除分支

```
1 | git branch -d 分支名称  
2 | 分支被合并后才允许删除 -D强制删除
```

## Git暂时保存更改

---

在git中，可以暂时提取分支上所有的改动并存储，临时转向其他工作。

使用场景：分支切换

- 存储临时改动

```
1 | git stash
```

- 恢复改动

```
1 | git stash pop
```

# Github

---

## 1.远程仓库

1. 

```
1 | git remote add 远程仓库地址别名 远程仓库地址
```

2. 

```
1 | git push -u 远程仓库地址别名 分支名称
```

  

```
2 | -u记住推送地址及分支，下次只需要输入git push即可
```

## 2.克隆仓库

克隆远端数据到本地：

```
1 | git clone 仓库地址
```