

git常用命令

1、git init(当前文件夹一定要是空的)

`git init` 是一个 Git 命令，用于初始化一个新的 Git 仓库。这个命令会创建一个新的 `.git` 目录，这个目录包含了所有必要的 Git 仓库文件，用于跟踪和记录项目的历史和版本信息。

这个命令的组成部分如下：

- `git` 是版本控制系统的命令行工具。
- `init` 是一个子命令，用于初始化一个新的 Git 仓库。

使用 `git init` 命令时，你可以在现有的目录中创建一个新的 Git 仓库，或者创建一个新的目录并初始化为 Git 仓库。以下是两种常见的用法：

1. 在当前目录中初始化一个新的 Git 仓库：

```
bash

git init
```

这会在当前目录下创建一个 `.git` 目录，但不会自动将任何文件添加到版本控制中。你需要使用 `git add` 命令来添加文件，然后使用 `git commit` 命令来提交这些文件。

2. 创建一个新的目录并初始化为 Git 仓库：

```
bash

mkdir my-new-project
cd my-new-project
git init
```

这会创建一个名为 `my-new-project` 的新目录，进入该目录，然后初始化一个新的 Git 仓库。

2、git config

是一个 Git 命令，用于配置 Git 仓库或系统级的 Git 设置。这个命令允许你查看和修改配置选项，这些选项控制 Git 的各种行为和外观

下面举个例子，在我们创建一个新的git库的时候，我们第一步操作一般都会要求我们配置用户邮箱和用户名（个人感觉没啥用，但是又非摄制不可）

```
C:\Users\nana\Desktop\test1>git config --global user.email "annaxzl05@gmail.com"
C:\Users\nana\Desktop\test1>git config --global user.name "anna"
C:\Users\nana\Desktop\test1>
```

3、git add.

`git add .` 是一个 Git 命令，用于将工作目录中的所有修改（包括新文件和已修改的文件）添加到暂存区（staging area），准备进行下一次提交。

这个命令的组成部分如下：

- `git` 是版本控制系统的命令行工具。
- `add` 是一个子命令，用于将更改添加到暂存区。
- `.` 表示当前目录及其子目录中的所有文件。

当你在 Git 仓库中对文件进行修改或添加新文件后，这些更改最初只在工作目录中。要将这些更改包含在下一次提交中，你需要先使用 `git add` 命令将它们添加到暂存区。这样，当你执行 `git commit` 命令时，这些更改就会被记录在新的提交中。

例如，如果你在仓库中做了一些修改并希望将这些更改提交到仓库，你可以按照以下步骤操作：

1. `git add .`：将所有修改添加到暂存区。
2. `git commit -m "Your commit message"`：将暂存区的更改提交到仓库，并附上提交信息。

这样，你的更改就会被记录在 Git 仓库的历史中。

4、git commit -m

`git commit -m` 是一个 Git 命令，用于将更改提交到本地仓库。这个命令的组成部分如下：

- `git` 是版本控制系统的命令行工具。
- `commit` 是一个子命令，用于记录项目历史中的一次更改。
- `-m` 是一个选项，用于指定提交信息（commit message）。
- 紧跟在 `-m` 后面的内容是提交信息，它描述了这次提交所做的更改。

例如，如果你执行了以下命令：

```
bash
```

```
git commit -m "Add new feature"
```

这将创建一个新的提交，其中包含自上次提交以来你所做的所有更改，并且提交信息为 "Add new feature"。这个信息有助于记录你所做的更改，便于将来查阅历史记录。

 复制  分享

 

5、git push

这个命令的组成部分如下：

- `git` 是版本控制系统的命令行工具。
- `push` 是一个子命令，用于将本地的提交推送到远程仓库。

使用 `git push` 时，通常需要指定远程仓库的名称和分支。例如：

```
bash
```

```
git push origin master
```

这个命令会将本地的 `master` 分支推送到名为 `origin` 的远程仓库。如果你已经设置了默认的远程仓库和分支，有时候你只需要输入 `git push` 即可。

如果你是第一次推送到远程仓库，或者你的远程仓库是空的，你可能需要使用 `-u` 选项来设置上游引用：

```
bash
```

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

这会将本地的 `master` 分支推送到远程仓库，并设置 `origin/master` 为本地 `master` 分支的上游（tracking branch），这样以后你只需要输入 `git push` 就可以推送了。

如果你在推送时遇到权限问题，可能需要输入你的用户名和密码，或者使用更安全的认证方式，如 SSH 密钥或个人访问令牌（Personal Access Token. PAT）。

 **zhudiquan.github.io** 公共📌 针👁 取消观

🔗 主要 ▾

🔗 1分支

🏷 标签

🔍 转到文件 t

➕ 添加文件 ▾

🔼 法典 ▾

 **zdqlwe** 删除网络安全与对抗 directory 80afbee · yesterday 🕒 9次提交

 README.md

初始提交

yesterday

📄 自述文件 ✎

zhudiquan.github.io