

# Git分支使用方法

---

## 前言

- 一、查看可用分支
- 二、创建新分支
- 三、切换到新分支
- 四、在新分支上进行工作
- 五、提交更改
- 六、切换回主分支
- 七、删除分支
- 八、合并分支
- 九、合并冲突
  - 1、冲突是什么？
  - 2、解决冲突的步骤：

## 前言

分支是指在同一个代码仓库中的不同版本线。它们可以被用来同时开展不同的开发任务、修复bug或实现新功能，而不会影响到主要的代码线。每个分支都可以独立地进行修改、提交和合并操作。

## 一、查看可用分支

在终端或命令提示符中，导航到你的Git项目所在的目录，使用 `cd` 命令切换目录；

可以使用以下命令查看当前仓库中已存在的分支：

▼

XML | 复制代码

```
1 git branch
```

该命令会列出所有已存在的分支，并在当前分支前面标记一个星号。

```
MINGW64:/d/a

A@DESKTOP-ND FME7E MINGW64 ~
$ cd D:\a

A@DESKTOP-ND FME7E MINGW64 /d/a (master)
$ git branch
* master
```

## 二、创建新分支

使用以下命令创建一个新的分支：

XML 复制代码

```
1 git branch <分支名>
```

例如，要创建一个名为branch1的新分支，可以运行：

XML 复制代码

```
1 git branch branch1
```

```
MINGW64:/d/a

A@DESKTOP-ND FME7E MINGW64 ~
$ cd D:\a

A@DESKTOP-ND FME7E MINGW64 /d/a (master)
$ git branch
* master

A@DESKTOP-ND FME7E MINGW64 /d/a (master)
$ git branch branch1
```

## 三、切换到新分支

创建分支后，使用以下命令切换到新创建的分支：

XML 复制代码

```
1 git checkout <分支名>
```

或者，可以使用以下命令来创建并切换到新分支：

```
1 git checkout -b <分支名>
```

例如，要切换到名为branch1的新分支，可以运行：

```
MINGW64:/d/a
A@DESKTOP-NDPME7E MINGW64 ~
$ cd D:\a

A@DESKTOP-NDPME7E MINGW64 /d/a (master)
$ git branch
* master

A@DESKTOP-NDPME7E MINGW64 /d/a (master)
$ git branch branch1

A@DESKTOP-NDPME7E MINGW64 /d/a (master)
$ git checkout branch1
Switched to branch 'branch1'
```

## 四、在新分支上进行工作

切换到新创建的分支后，你可以在该分支上进行修改、添加、删除文件等操作。在这个分支上的更改不会影响其他分支。

## 五、提交更改

在新分支上进行了一些修改后，使用以下命令将更改提交到该分支：

```
1 git add <文件名>
2 git commit -m "提交信息"
```

## 六、切换回主分支

完成在新分支上的工作后，你可以切换回主分支或其他任何分支。

在 Git 中，主分支通常被称为 **master** 或 **main** 分支。这是默认的主要开发分支，用于存储项目的稳定版本和主要代码线。

```
1 git checkout <主分支名>
```

具体使用哪个名称取决于你的 Git 版本和项目设置，你可以通过查看分支列表

```
MINGW64:/d/a
A@DESKTOP-NDFME7E MINGW64 /d/a (branch1)
$ git branch
* branch1
  master

A@DESKTOP-NDFME7E MINGW64 /d/a (branch1)
$ git checkout master
Switched to branch 'master'

A@DESKTOP-NDFME7E MINGW64 /d/a (master)
$ git branch
  branch1
* master
```

## 七、删除分支

1、确保你不在要删除的分支上，运行以下命令确认当前所在的分支：

▼ XML 复制代码

```
1 git branch
```

2、切换到其他分支：选择一个不需要删除的分支，例如 "master" 分支运行以下命令来切换分支：

▼ XML 复制代码

```
1 git checkout master
```

3、删除分支：现在你可以运行以下命令尝试删除：

▼ XML 复制代码

```
1 git branch -d 删除分支名
```

4、强制删除分支：使用强制删除选项 **-D**：

▼ XML 复制代码

```
1 git branch -D 删除分支名
```

## 八、合并分支

1、切换到要接受更改的目标分支；示例：将 **branch1** 分支合并到 **master** 分支中，执行以下命令：

▼XML复制代码

```
1 git checkout master
```

2、将目标分支合并到当前分支，将branch1分支合并到master分支中，， 执行以下命令：

▼XML复制代码

```
1 git merge branch1
```

3、当合并完成后，你可以使用git log命令来查看合并提交的历史记录。

▼XML复制代码

```
1 git log
```

## 九、合并冲突

### 1、冲突是什么？

冲突在Git中发生于两个不同的分支对同一文件的相同部分进行了不同的修改。当你尝试将这两个分支合并时，Git无法自动确定应该接受哪个修改，因此产生了冲突。

### 2、解决冲突的步骤：

执行git merge命令后，如果存在冲突，Git会告诉你冲突发生的文件。

打开冲突的文件，你会看到类似下面的标记：

▼Plain Text复制代码

```
1 <<<<<<< HEAD
2 // 当前分支的代码
3 =====
4 // 要合并的分支的代码
5 >>>>>>> branch-name
```

在冲突标记之间编辑代码，将冲突的部分修改为你期望的代码，或你可以完全删除冲突标记并重新编写整个部分。

保存文件后，标记为已解决的文件会自动加入暂存区，重复上述步骤，直到解决所有冲突。

当所有冲突都解决并已经暂存，执行git commit命令以完成合并提交。

