Git的一个方便之处就在于对历史版本的管理。借助分散仓库的优势，可以在不影响其他仓库的前提下对历史版本进行操作。

1. 回溯历史版本

首先让我们先回溯到上一节 feature-A 分支创建之前，创建一个名为fix-B 的特性分支。

要让仓库的 HEAD、暂存区、当前工作树回溯到指定状态，需要用到 git rest --hard命令。只要提供目标时间点的哈希值，就可以完全恢复到这个时间节点的状态。

输入：

git reset --hard d2e76a62c7dee9ed5473af119570c950c3c1b4a1

提示：

HEAD is now at d2e76a6 测试git pull命令

回到了创建分支feature-A之前的状态。

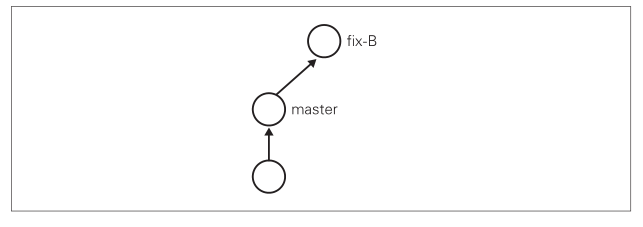
创建分支feature-B，输入：

git checkout -b feature-B

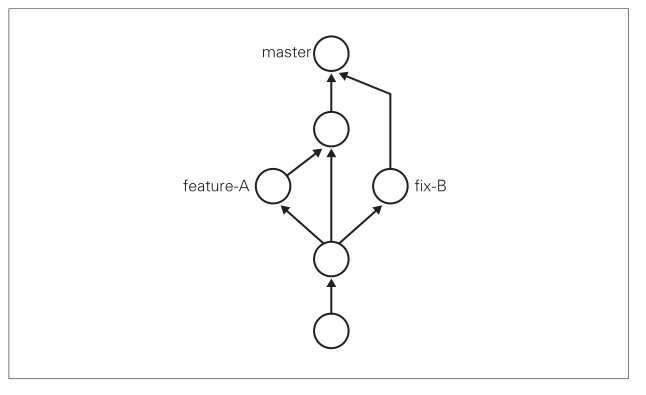
提示：

Switched to a new branch 'feature-B'

我们现在的状态如下图所示：



接下来我们的目标是：



即主干分支合并 feature-A 分支的修改后，又合并了 fix-B 的修改。

1. 推进至 feature - A 分支合并后的状态

git log命令只能查看以当前状态为终点的历史日志。所以这里要使用 git reflog命令，查看当前仓库的操作日志。在日志中找出回溯历史之前的哈希值，通过 git reset --hard命令恢复到回溯历史前的状态。

输入：

git reflog

提示：

954dc49 (HEAD -> feature-B) HEAD@{0}: commit: Fix B

d2e76a6 (origin/master, master) HEAD@{1}: checkout: moving from master to feature-B

d2e76a6 (origin/master, master) HEAD@{2}: reset: moving to d2e76a62c7dee9ed5473af119570c950c3c1b4a1

9dfab48 HEAD@{3}: merge feature-A: Merge made by the 'recursive' strategy.

d2e76a6 (origin/master, master) HEAD@{4}: checkout: moving from feature-A to master

0ec858c (feature-A) HEAD@{5}: checkout: moving from master to feature-A

d2e76a6 (origin/master, master) HEAD@{6}: checkout: moving from feature-A to master

0ec858c (feature-A) HEAD@{7}: checkout: moving from master to feature-A

d2e76a6 (origin/master, master) HEAD@{8}: checkout: moving from feature-A to master

0ec858c (feature-A) HEAD@{9}: commit: Add feature-A

d2e76a6 (origin/master, master) HEAD@{10}: checkout: moving from master to feature-A

d2e76a6 (origin/master, master) HEAD@{11}: pull origin master: Fast-forward

b47cee8 HEAD@{12}: commit: 随便添加的

5fc5bd5 HEAD@{13}: commit: add new line

32280c3 HEAD@{14}: commit: add 222222

68e8d7d HEAD@{15}: commit: third edit

757378f HEAD@{16}: commit: add hello world

:...skipping...

954dc49 (HEAD -> feature-B) HEAD@{0}: commit: Fix B

d2e76a6 (origin/master, master) HEAD@{1}: checkout: moving from master to feature-B

d2e76a6 (origin/master, master) HEAD@{2}: reset: moving to d2e76a62c7dee9ed5473af119570c950c3c1b4a1

9dfab48 HEAD@{3}: merge feature-A: Merge made by the 'recursive' strategy.

d2e76a6 (origin/master, master) HEAD@{4}: checkout: moving from feature-A to master

0ec858c (feature-A) HEAD@{5}: checkout: moving from master to feature-A

d2e76a6 (origin/master, master) HEAD@{6}: checkout: moving from feature-A to master

0ec858c (feature-A) HEAD@{7}: checkout: moving from master to feature-A

d2e76a6 (origin/master, master) HEAD@{8}: checkout: moving from feature-A to master

0ec858c (feature-A) HEAD@{9}: commit: Add feature-A

d2e76a6 (origin/master, master) HEAD@{10}: checkout: moving from master to feature-A

d2e76a6 (origin/master, master) HEAD@{11}: pull origin master: Fast-forward

b47cee8 HEAD@{12}: commit: 随便添加的

5fc5bd5 HEAD@{13}: commit: add new line

32280c3 HEAD@{14}: commit: add 222222

68e8d7d HEAD@{15}: commit: third edit

757378f HEAD@{16}: commit: add hello world

5c1c086 HEAD@{17}: commit (initial): create new file

在日志中，我们可以看到 commit、checkout、reset、merge 等 Git 命令的执行记录。只要不进行 Git 的 GC（Garbage Collection，垃圾回收），就可以通过日志随意调取近期的历史状态，就像给时间机器指定一个时间点，在过去未来中自由穿梭一般。即便开发者错误执行了 Git 操作，基本也都可以利用 git reflog命令恢复到原先的状态。

这个命令主要是获取每次操作生成的hash值，我们得根据这个值才能进行版本管理工作。

因为现在我们要恢复到合并feature-A分支的状态，所以，找到merge这条操作日志：9dfab48 HEAD@{3}: merge feature-A: Merge made by the 'recursive' strategy.

9dfab48是对应这次操作的hash值。我们可以使用这个hash值，将HEAD、暂存区和工作树恢复到这个时间点状态。

输入：

git checkout master

git reset --hard 9dfab48

提示：

HEAD is now at 9dfab48 Merge branch 'feature-A'

现在我们回到了合并feature-A分支后的状态。

1. 合并分支feature-B

现在要合并feature-B分支，输入：

git merge --no-ff feature-B

提示：

Auto-merging readme.md

CONFLICT (content): Merge conflict in readme.md

Automatic merge failed; fix conflicts and then commit the result.

没有合并成功，合并出现了冲突（Conflict）。系统在合并 README.md 文件时，feature-A 分支更改的部分与本次想要合并的 fix-B 分支更改的部分发生了冲突。

现在看一下readme.md的内容：

<<<<<<< HEAD

# Git教程

feature-A

=======

# Git 教程

- fix-B

>>>>>>> feature-B

=======以上的内容是当前HEAD的内容，以下的内容是要合并的feature-B分支的内容，现在在编辑器中修改：

# Git教程

- feature-A

- fix-B

让两个分支的内容并存。

然后我们提交解决冲突后的readme.md文件。

输入：

git add readme.md

git commit -m "Fix conflict"

提示：

[master c987ebf] Fix conflict

记录提交信息为：Fix conflict。

总结，如果合并分支，遇到了冲突，需要我们手动去仔细修改冲突的地方。然后进行提交。

1. 修改提交信息

要修改上一条提交信息，可以使用 git commit --amend命令。

我们将上一条提交信息记为了 "Fix conflict"，但它其实是 fix-B 分支的合并，解决合并时发生的冲突只是过程之一，这样标记实在不妥。于是，我们要修改这条提交信息。

输入：

git commit --amend

然后就会启动编辑器，编辑器的内容是：

Fix conflict

# Please enter the commit message for your changes. Lines starting

# with '#' will be ignored, and an empty message aborts the commit.

#

# Date: Sun Jul 7 10:29:42 2019 +0800

#

# On branch master

# Your branch is ahead of 'origin/master' by 4 commits.

# (use "git push" to publish your local commits)

#

# Changes to be committed:

# modified: readme.md

#

然后我们将提交的信息部分：Fix conflict，修改为：Merge branch 'fix-B'。保存后，关闭编辑器。

提示：

[master e6affd0] Merge branch 'fix-B'

然后我们输入：

git log --graph

输出：

\* commit e6affd036d5340a9afbac52a2b2123b995a37252 (HEAD -> master)

|\ Merge: 9dfab48 954dc49

| | Author: windofme1109 <wuzengyan1314@163.com>

| | Date: Sun Jul 7 10:29:42 2019 +0800

| |

| | Merge branch 'fix-B'

| |

| \* commit 954dc491ab591e6b5eb8079993985e808a727200 (feature-B)

| | Author: windofme1109 <wuzengyan1314@163.com>

| | Date: Sat Jul 6 16:15:32 2019 +0800

| |

| | Fix B

| |

\* | commit 9dfab4846a04195c570756b052215070a9b796c3

|\ \ Merge: d2e76a6 0ec858c

| |/ Author: windofme1109 <wuzengyan1314@163.com>

|/| Date: Sat Jul 6 12:50:27 2019 +0800

| |

| | Merge branch 'feature-A'

| |

| \* commit 0ec858c09800eb041a05981b3f18f6c78b21a268 (feature-A)

|/ Author: windofme1109 <wuzengyan1314@163.com>

| Date: Sat Jul 6 12:33:16 2019 +0800

|

| Add feature-A

|

\* commit d2e76a62c7dee9ed5473af119570c950c3c1b4a1 (origin/master)

| Author: windofme1109 <41136323+windofme1109@users.noreply.github.com>

| Date: Fri Jul 5 22:20:14 2019 +0800

|

| 测试git pull命令

|

| 远端仓库更新，使用git pull 命令同步更新本地仓库到最新状态

|

\* commit b47cee83602d07515d10f37a3ec79621534e1fb1

| Author: windofme1109 <wuzengyan1314@163.com>

| Date: Fri Jul 5 20:47:37 2019 +0800

|

| 随便添加的

|

\* commit 5fc5bd5347c42af1e53a7c4cbf9566f88849e509

| Author: windofme1109 <wuzengyan1314@163.com>

| Date: Fri Jul 5 19:45:43 2019 +0800

|

| add new line

|

| I add this line is to exercise how to use git

|

\* commit 32280c38087be1033440ef84b5ddc85589fef53f

| Author: windofme1109 <wuzengyan1314@163.com>

| Date: Fri Jul 5 12:24:31 2019 +0800

|

| add 222222

|

\* commit 68e8d7df3727f27a303a70d18661cb70ec4533f0

| Author: windofme1109 <wuzengyan1314@163.com>

| Date: Fri Jul 5 12:20:56 2019 +0800

|

| third edit

|

\* commit 757378fe76e80337679e448374796d1c75fec3fd

| Author: windofme1109 <wuzengyan1314@163.com>

| Date: Fri Jul 5 12:12:30 2019 +0800

|

| add hello world

|

\* commit 5c1c086a2fe61281cbffbc361f6ae622f8a410de

Author: windofme1109 <wuzengyan1314@163.com>

Date: Fri Jul 5 12:05:31 2019 +0800

create new file

可以看出，我们的提交信息已经被修改了。

1. 压缩历史记录

在合并特性分支之前，如果发现已提交的内容中有些许拼写错误等，不妨提交一个修改，然后将这个修改包含到前一个提交之中，压缩成一个历史记录。

创建一个新的分支：feature-C。

输入：

git checkout -b feature-C

提示：

Switched to a new branch 'feature-C'

作为 feature-C 的功能实现，我们在 README.md 文件中添加一行文字，并且故意留下拼写错误，以便之后修正。

# Git教程

- feature-A

- fix-B

- faeture-C

提交这部分内容。这个小小的变更就没必要先执行 git add命令再执行 git commit命令了，我们用 git commit -am命令来一次完成这两步操作。

输入：

git commit -am "Add feature-C"

提示：

[feature-C 6ea5bed] Add feature-C

1 file changed, 2 insertions(+)

现在我们修改readme.md中的拼写错误，然后使用git diff命令查看修改前后的变化。

输入：

git diff

提示：

**diff --git a/readme.md b/readme.md**

**index 6ab8878..b38d249 100644**

**--- a/readme.md**

**+++ b/readme.md**

@@ -23,5 +23,5 @@ eeeeeeeeeeeeeeeeeeee

- fix-B

-- faeture-C

+- feature-C

然后进行提交：

git commit -am "Fix typo"

提示：

[feature-C 2f112d8] Fix typo

1 file changed, 1 insertion(+), 1 deletion(-)

错字漏字等失误称作 typo，所以我们将提交信息记为 "Fix typo"。

实际上，我们不希望在历史记录中看到这类提交，因为健全的历史记录并不需要它们。

对于拼写错误等一下小的改动，我们没有必要在次次记录下这种提交。因此需要和别的提交合并成一次完美提交。

将 "Fix typo"修正的内容与之前一次的提交合并，在历史记录中合并为一次提交。为此，我们要用到git rebase命令。

输入：

git rebase -i HEAD~2

用上述方式执行 git rebase命令，可以选定当前分支中包含HEAD（最新提交）在内的两个最新历史记录为对象。然后会启动编辑器，编辑器的内容是：

pick 6ea5bed Add feature-C

pick 2f112d8 Fix typo

# Rebase e6affd0..2f112d8 onto e6affd0 (2 commands)

#

# Commands:

# p, pick <commit> = use commit

# r, reword <commit> = use commit, but edit the commit message

# e, edit <commit> = use commit, but stop for amending

# s, squash <commit> = use commit, but meld into previous commit

# f, fixup <commit> = like "squash", but discard this commit's log message

# x, exec <command> = run command (the rest of the line) using shell

# b, break = stop here (continue rebase later with 'git rebase --continue')

# d, drop <commit> = remove commit

# l, label <label> = label current HEAD with a name

# t, reset <label> = reset HEAD to a label

# m, merge [-C <commit> | -c <commit>] <label> [# <oneline>]

# . create a merge commit using the original merge commit's

# . message (or the oneline, if no original merge commit was

# . specified). Use -c <commit> to reword the commit message.

#

# These lines can be re-ordered; they are executed from top to bottom.

#

# If you remove a line here THAT COMMIT WILL BE LOST.

#

# However, if you remove everything, the rebase will be aborted.

#

# Note that empty commits are commented out

我们将2f112d8的Fix typo压缩到6ea5bed的Add feature-C。所以，要将

2f112d8左侧的pick命令改为fixup命令。

pick 6ea5bed Add feature-C

fixup 2f112d8 Fix typo

下面的注释中，对不同的命令都有说明，比如fixup命令，表示压缩，而且是抛弃这条提交日志信息。

完成修改后，保存，并关闭编辑器。提示：

Successfully rebased and updated refs/heads/feature-C.

表示我们已经rebase成功了。将Fix typo成功的合并到Add feature-C这次提交中。

然后我们使用git log --graph命令查看日志：

git log --graph

输出：

\* commit 5d830bdaec55236f892f879c94c288ec49206acb (HEAD -> feature-C)

| Author: windofme1109 <wuzengyan1314@163.com>

| Date: Sun Jul 7 11:50:04 2019 +0800

|

| Add feature-C

|

\* commit e6affd036d5340a9afbac52a2b2123b995a37252 (master)

|\ Merge: 9dfab48 954dc49

| | Author: windofme1109 <wuzengyan1314@163.com>

| | Date: Sun Jul 7 10:29:42 2019 +0800

| |

| | Merge branch 'fix-B'

| |

| \* commit 954dc491ab591e6b5eb8079993985e808a727200 (feature-B)

| | Author: windofme1109 <wuzengyan1314@163.com>

| | Date: Sat Jul 6 16:15:32 2019 +0800

| |

| | Fix B

| |

\* | commit 9dfab4846a04195c570756b052215070a9b796c3

|\ \ Merge: d2e76a6 0ec858c

| |/ Author: windofme1109 <wuzengyan1314@163.com>

|/| Date: Sat Jul 6 12:50:27 2019 +0800

Administrator@USER-20181123NM MINGW64 /e/GitForCoding/hello\_world (feature-C)

$ git log --graph

\* commit 5d830bdaec55236f892f879c94c288ec49206acb (HEAD -> feature-C)

| Author: windofme1109 <wuzengyan1314@163.com>

| Date: Sun Jul 7 11:50:04 2019 +0800

|

| Add feature-C

|

\* commit e6affd036d5340a9afbac52a2b2123b995a37252 (master)

|\ Merge: 9dfab48 954dc49

| | Author: windofme1109 <wuzengyan1314@163.com>

| | Date: Sun Jul 7 10:29:42 2019 +0800

| |

| | Merge branch 'fix-B'

| |

| \* commit 954dc491ab591e6b5eb8079993985e808a727200 (feature-B)

| | Author: windofme1109 <wuzengyan1314@163.com>

| | Date: Sat Jul 6 16:15:32 2019 +0800

| |

| | Fix B

| |

\* | commit 9dfab4846a04195c570756b052215070a9b796c3

|\ \ Merge: d2e76a6 0ec858c

| |/ Author: windofme1109 <wuzengyan1314@163.com>

|/| Date: Sat Jul 6 12:50:27 2019 +0800

| |

| | Merge branch 'feature-A'

| |

| \* commit 0ec858c09800eb041a05981b3f18f6c78b21a268 (feature-A)

|/ Author: windofme1109 <wuzengyan1314@163.com>

| Date: Sat Jul 6 12:33:16 2019 +0800

|

| Add feature-A

|

\* commit d2e76a62c7dee9ed5473af119570c950c3c1b4a1 (origin/master)

| Author: windofme1109 <41136323+windofme1109@users.noreply.github.com>

| Date: Fri Jul 5 22:20:14 2019 +0800

|

| 测试git pull命令

|

| 远端仓库更新，使用git pull 命令同步更新本地仓库到最新状态

|

\* commit b47cee83602d07515d10f37a3ec79621534e1fb1

| Author: windofme1109 <wuzengyan1314@163.com>

| Date: Fri Jul 5 20:47:37 2019 +0800

|

| 随便添加的

|

\* commit 5fc5bd5347c42af1e53a7c4cbf9566f88849e509

| Author: windofme1109 <wuzengyan1314@163.com>

| Date: Fri Jul 5 19:45:43 2019 +0800

|

| add new line

|

| I add this line is to exercise how to use git

|

\* commit 32280c38087be1033440ef84b5ddc85589fef53f

| Author: windofme1109 <wuzengyan1314@163.com>

| Date: Fri Jul 5 12:24:31 2019 +0800

|

| add 222222

|

\* commit 68e8d7df3727f27a303a70d18661cb70ec4533f0

| Author: windofme1109 <wuzengyan1314@163.com>

| Date: Fri Jul 5 12:20:56 2019 +0800

|

| third edit

|

\* commit 757378fe76e80337679e448374796d1c75fec3fd

| Author: windofme1109 <wuzengyan1314@163.com>

| Date: Fri Jul 5 12:12:30 2019 +0800

|

| add hello world

|

\* commit 5c1c086a2fe61281cbffbc361f6ae622f8a410de

Author: windofme1109 <wuzengyan1314@163.com>

Date: Fri Jul 5 12:05:31 2019 +0800

create new file

提交日志中已经没有了Fix typo这条记录。也就相当于 Add feature-C中从来没有出现过拼写错误。这算是一种良性的历史改写。

最后，将feature-C与master分支合并。

git merge --no-ff feature-C

提示：

Merge made by the 'recursive' strategy.

readme.md | 2 ++

1 file changed, 2 insertions(+)