**Git分支 - 基本分支操作（一）**

**1、创建分支**

**（1）创建分支**

Git 是怎么创建新分支的呢？

很简单，就是要创建一个可以移动的新的指针。

比如，创建一个testing分支， 你需要使用命令：git branch testing。

示例：

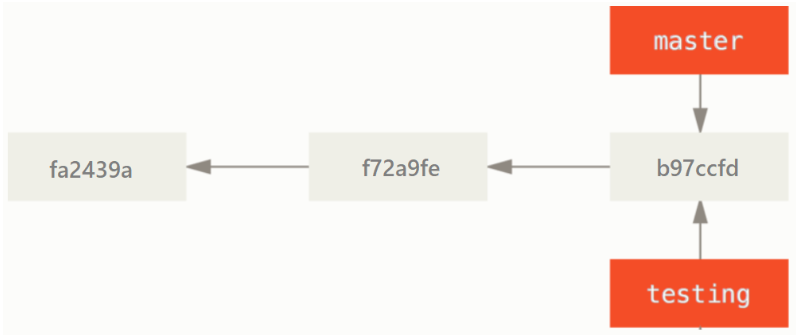
|  |
| --- |
| Bash # 1.查看本地版本库历史提交 L@DESKTOP-T2AI2SU MINGW64 /j/git-repository/learngit (master) $ git log --oneline b97ccfd (HEAD -> master) 第3次提交，新增内容：branch test v3 f72a9fe 第2次提交，新增内容：branch test v2 fa2439a 第1次提交，新增readme.txt文件  # 2.新建分支testing  L@DESKTOP-T2AI2SU MINGW64 /j/git-repository/learngit (master) $ git branch testing  # 3.再次查看本地版本库历史提交 L@DESKTOP-T2AI2SU MINGW64 /j/git-repository/learngit (master) $ git log --oneline b97ccfd (HEAD -> master, testing) 第3次提交，新增内容：branch test v3 f72a9fe 第2次提交，新增内容：branch test v2 fa2439a 第1次提交，新增readme.txt文件 |

可以看到(HEAD -> master, testing)，说明本地版本库中，已经多了一个testing分支。

**（2）图示理解**

如上操作，会在当前所在的提交对象上创建一个新的指针。

如下图：

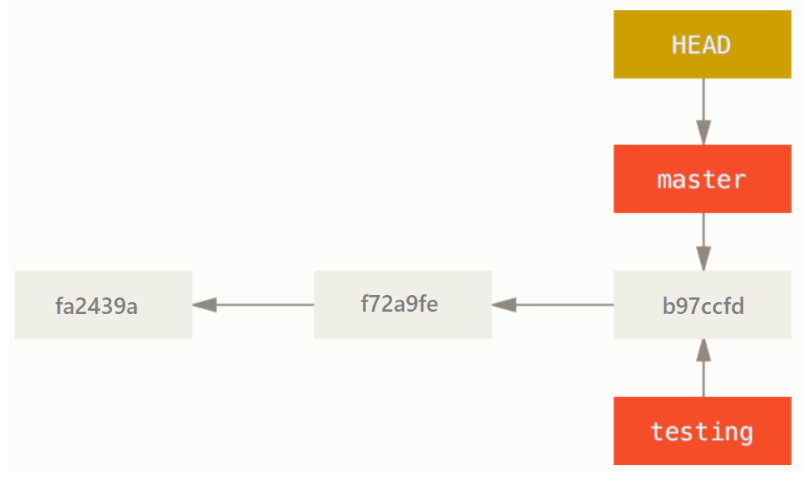


那么Git 又是怎么知道当前在哪一个分支上呢？ 也很简单，它有一个名为 HEAD 的特殊指针，HEAD指向当前所在的本地分支上。

在本例中，是仍然在 master 分支上。 因为 git branch 命令仅仅 **创建** 一个新分支，并不会自动切换到新分支中去。

因为上面日志信息：(HEAD -> master, testing)，也说明HEAD指针指向的是master分支。

如下图所示：



如图所示，当前 master 和 testing 分支均指向校验和以 b97ccfd 开头的提交对象。

**2、查看分支列表**

执行命令：git branch。不用加任何参数。

如下图：



该命令会列出当前本地版本库中存在的所有分支，且当前正在工作的分支前会显示\*。

和命令行路径后面( )中的内容相对应。

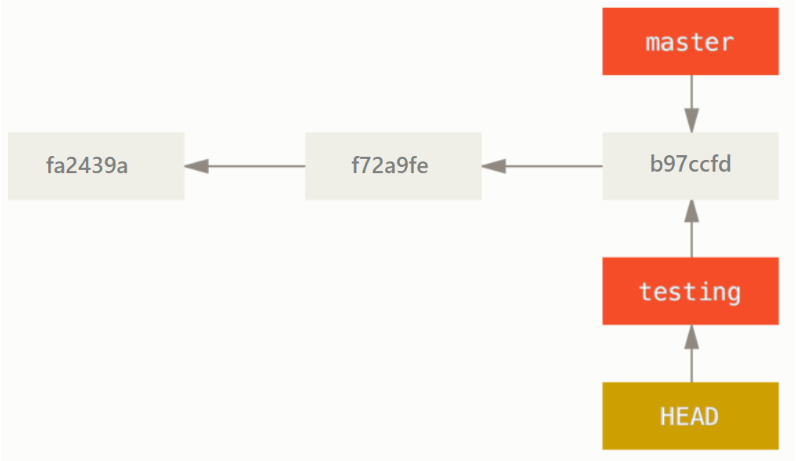
**3、分支切换**

要切换到一个已存在的分支，你需要使用 git checkout branchname 命令。

我们现在切换到新创建的 testing 分支去：

|  |
| --- |
| Bash # 1.切换到testing分支上 L@DESKTOP-T2AI2SU MINGW64 /j/git-repository/learngit (master) $ git checkout testing Switched to branch 'testing' # 提示：切换到分支“testing”  # 2.查看当前正在工作的分支。 L@DESKTOP-T2AI2SU MINGW64 /j/git-repository/learngit (testing) # 这里 $ git branch  master \* testing # 这里 |

现在 HEAD 就指向 testing 分支了。（即：HEAD 会指向当前所在的分支）



|  |
| --- |
| 提示：  在创建分支的时候，添加-b选项，可以直接切换到新创建的分支上。  例如：git checkout -b testing  该命令等价于如下两条命令：   * git branch testing：创建分支。 * git checkout testing：切换分支。 |

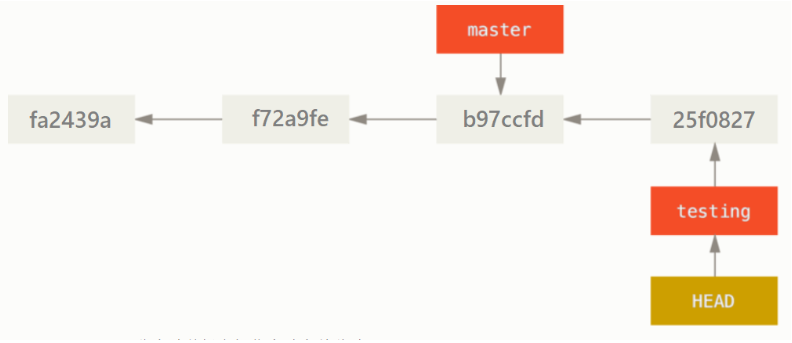
**4、查看所有分支的最后一个提交**

使用git branch -v命令可以查看，所有分支的最后一次提交。

我们在testing 分支修改readme.txt文件，再进行一次提交。

|  |
| --- |
| Bash # 1.查看当前工作的分支 L@DESKTOP-T2AI2SU MINGW64 /j/git-repository/learngit (testing) $ git branch  master \* testing  # 2.修改文件，提交新版本 L@DESKTOP-T2AI2SU MINGW64 /j/git-repository/learngit (testing) $ echo "branch test v4" >> readme.txt  L@DESKTOP-T2AI2SU MINGW64 /j/git-repository/learngit (testing) $ git commit -a -m '第4次提交，新增内容：branch test v4' warning: LF will be replaced by CRLF in readme.txt. The file will have its original line endings in your working directory [testing 25f0827] 第4次提交，新增内容：branch test v4  1 file changed, 1 insertion(+)  # 3.查看所有分支的最后一次提价 L@DESKTOP-T2AI2SU MINGW64 /j/git-repository/learngit (testing) $ git branch -v  master b97ccfd 第3次提交，新增内容：branch test v3 \* testing 25f0827 第4次提交，新增内容：branch test v4  # 4.查看版本库历史提交记录 L@DESKTOP-T2AI2SU MINGW64 /j/git-repository/learngit (testing) $ git log --graph --oneline \* 25f0827 (HEAD -> testing) 第4次提交，新增内容：branch test v4 \* b97ccfd (master) 第3次提交，新增内容：branch test v3 \* f72a9fe 第2次提交，新增内容：branch test v2 \* fa2439a 第1次提交，新增readme.txt文件 |

通过上面的日志信息，此时 HEAD 指针的状态如下图：



可以看到 HEAD 随着在testing 分支上的提交操作，自动向前移动。

**5、删除分支**

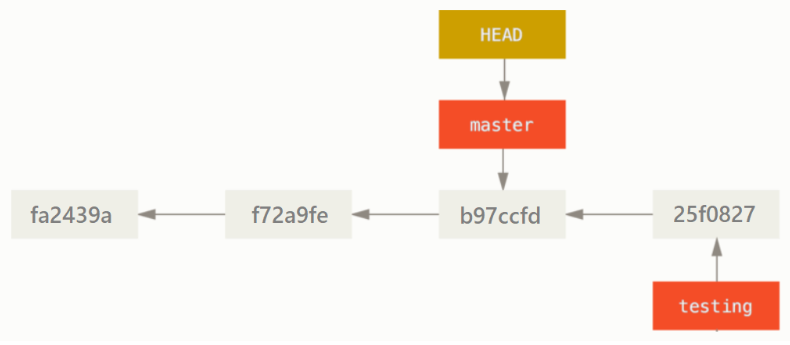
使用命令：git branch -d branchname。

但是要注意的是：**若要删除某分支，必须要保证当前工作的分支，不能是该分支。**

这个时候我们需要先切换回master分支。

|  |
| --- |
| Bash L@DESKTOP-T2AI2SU MINGW64 /j/git-repository/learngit (testing) $ git checkout master Switched to branch 'master'  L@DESKTOP-T2AI2SU MINGW64 /j/git-repository/learngit (master) $ git branch \* master  testing |

此时 HEAD 指针的状态如下图：



|  |
| --- |
| 说明：git checkout master这条命令做了两件事。   1. 是使 HEAD 指回 master 分支。 2. 是将工作目录中文件，恢复成 master 分支所指向提交的快照内容。   **在切换分支时，一定要注意你工作目录里的文件会被改变，你的工作目录会恢复到该分支最后一次提交时的样子。如果 Git 不能干净利落地完成这个任务，它将禁止切换分支。**（下面分支实战中有详细说明）  也就是说，你现在做修改的话，项目将始于 master 分支版本。 本质上来讲，这就是忽略 testing分支所做的修改，以便于向另一个方向进行开发。 |

继续我们开始删除 testing 分支。

|  |
| --- |
| Bash L@DESKTOP-T2AI2SU MINGW64 /j/git-repository/learngit (master) $ git branch -d testing error: The branch 'testing' is not fully merged. If you are sure you want to delete it, run 'git branch -D testing'. |

提示你：分支“testing”没有合并到主分支，如果确定要删除它，请运行git branch -D testing命令。。

所以我们使用-D选项删除 testing 分支。

|  |
| --- |
| Bash # 1.删除分支 L@DESKTOP-T2AI2SU MINGW64 /j/git-repository/learngit (master) $ git branch -D testing Deleted branch testing (was 25f0827).  # 2.查看本地库中的分支，就只有master分支了 L@DESKTOP-T2AI2SU MINGW64 /j/git-repository/learngit (master) $ git branch \* master  # 3.查看提交历史记录 L@DESKTOP-T2AI2SU MINGW64 /j/git-repository/learngit (master) $ git log --graph --oneline \* b97ccfd (HEAD -> master) 第3次提交，新增内容：branch test v3 \* f72a9fe 第2次提交，新增内容：branch test v2 \* fa2439a 第1次提交，新增readme.txt文件 # 可以看到对testing分支的修改，只要没有合并，对master分支是没有影响的。  # 4.查看可回退的提交历史记录 L@DESKTOP-T2AI2SU MINGW64 /j/git-repository/learngit (master) $ git reflog b97ccfd (HEAD -> master) HEAD@{0}: checkout: moving from testing to master 25f0827 HEAD@{1}: commit: 第4次提交，新增内容：branch test v4 b97ccfd (HEAD -> master) HEAD@{2}: checkout: moving from master to testing b97ccfd (HEAD -> master) HEAD@{3}: commit: 第3次提交，新增内容：branch test v3 f72a9fe HEAD@{4}: commit: 第2次提交，新增内容：branch test v2 fa2439a HEAD@{5}: commit (initial): 第1次提交，新增readme.txt文件  # 可以看到只要HEAD移动过，所有历史轨迹都会保留在Git版本库中。 # （也就是.git/objects目录中的文件只能无限的增多，存储了就不会删除掉。） |

|  |
| --- |
| 提示：合并后的分支不仅无用，它的存在还会引起不必要的麻烦，一般都会删除无用的分支。 |

作者：繁华似锦Fighting 链接：https://www.jianshu.com/p/10b3218f9d92 来源：简书 著作权归作者所有。商业转载请联系作者获得授权，非商业转载请注明出处。