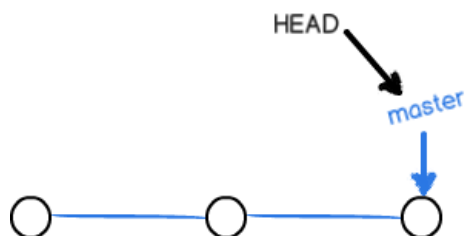


git 分支管理

<https://www.liaoxuefeng.com/wiki/0013739516305929606dd18361248578c67b8067c8c017b000/0013743862006503a1c5bf5a783434581661a3cc2084efa000>

Git都把每次提交串成一条时间线，这条时间线就是一个分支。只有一条时间线时，这个分支叫主分支，即`master`分支。
`master`指向提交，`HEAD`指向`master`，就是当前分支。



创建并切换分支

```
$ git checkout -b dev //创建dev分支，然后切换到dev分支
```

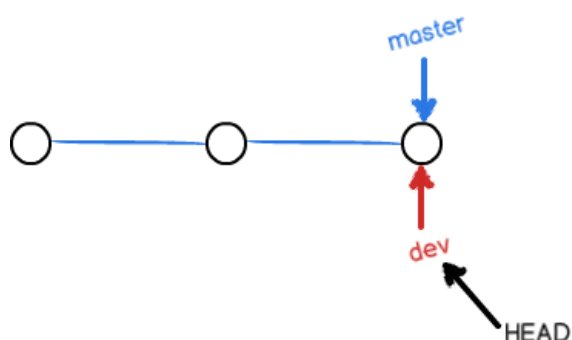
Switched to a new branch 'dev'

`git checkout`命令加上`-b`参数表示创建并切换，相当于以下两条命令：

```
$ git branch dev //创建dev分支
```

```
$ git checkout dev //切换到dev分支
```

Switched to branch 'dev'



查看分支

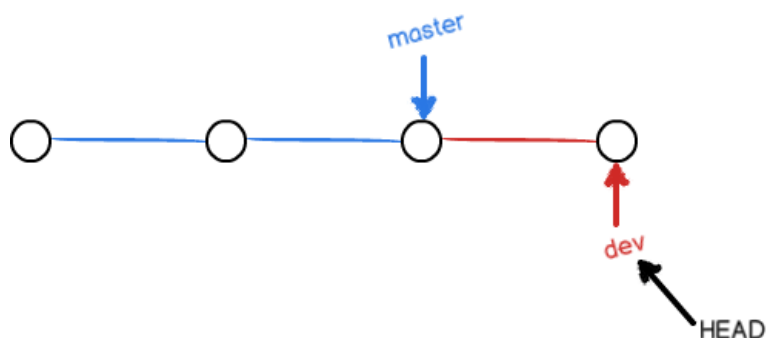
```
$ git branch //列出所有分支，当前分支前面会标一个*号。
```

```
* dev
```

```
master
```

从现在开始，对工作区的修改和提交就是针对`dev`分支了。

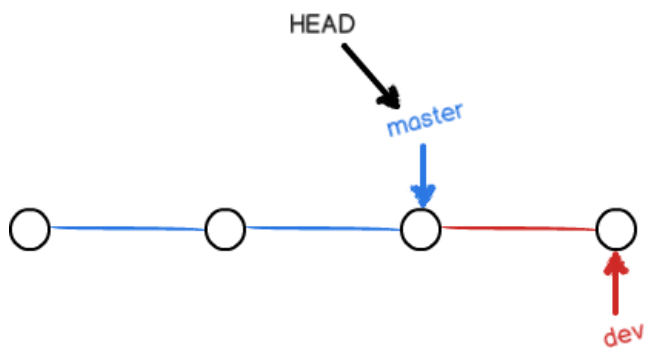
对`readme.txt`做个修改，然后提交。



切换分支

```
$ git checkout master //切换回master分支
```

Switched to branch 'master'



合并分支

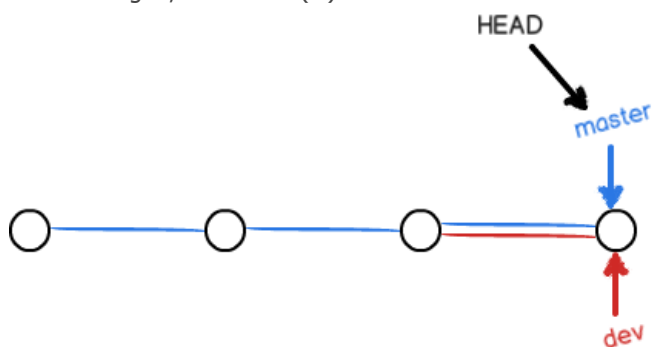
\$ git merge dev //把指定的dev分支的工作成果合并到当前分支master分支上

Updating d17efd8..fec145a

Fast-forward

readme.txt | 1 +

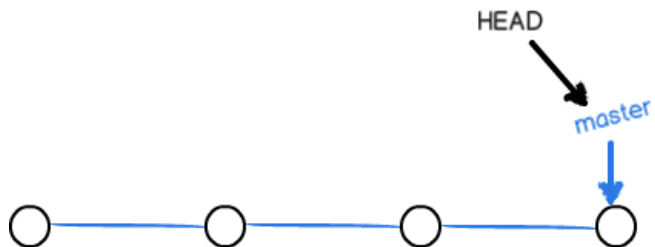
1 file changed, 1 insertion(+)



删除分支

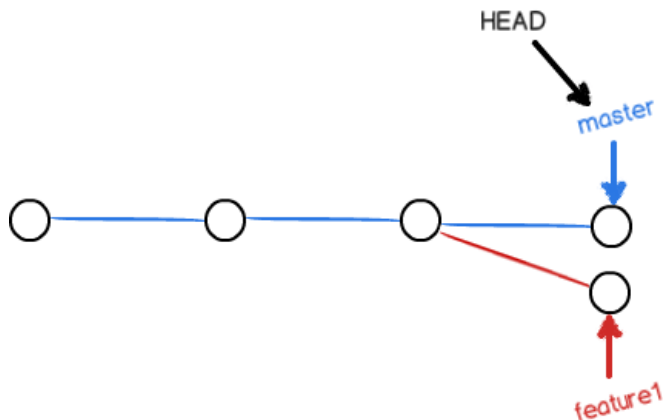
\$ git branch -d dev //删除dev分支

Deleted branch dev (was fec145a).



解决冲突

创建新的feature1分支后，在master分支和feature1分支各自都分别有新的提交



此时合并就可能会有冲突

\$ git merge feature1

Auto-merging readme.txt

CONFLICT (content): Merge conflict in readme.txt

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

`git status`也可以告诉我们冲突的文件

```
$ git status
# On branch master
# Your branch is ahead of 'origin/master' by 2 commits.
#
# Unmerged paths:
#   (use "git add/rm <file>..." as appropriate to mark resolution)
#
#       both modified:   readme.txt
#
no changes added to commit (use "git add" and/or "git commit -a")
```

直接查看readme.txt的内容

Git用<<<<<<, =====, >>>>>>标记出不同分支的内容

Git is a distributed version control system.

Git is free software distributed under the GPL.

Git has a mutable index called stage.

Git tracks changes of files.

<<<<<< HEAD

Creating a new branch is quick & simple.

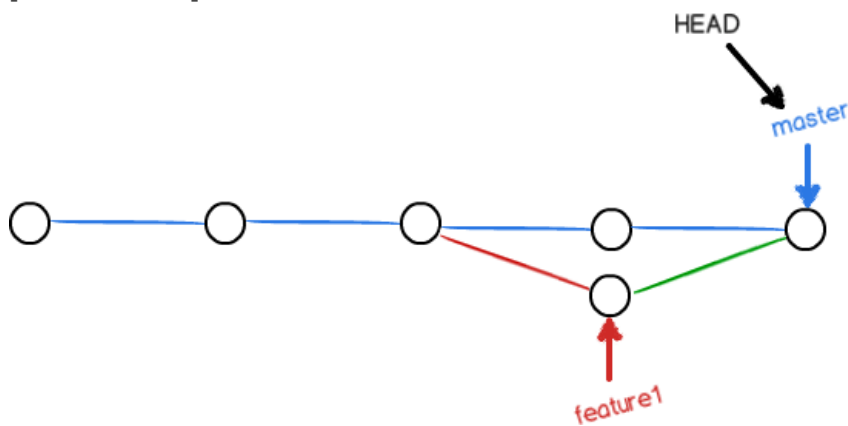
=====

Creating a new branch is quick AND simple.

>>>>>> feature1

修改后保存再提交:

```
$ git add readme.txt
$ git commit -m "conflict fixed"
[master 59bc1cb] conflict fixed
```



用带参数的`git log`也可以看到分支的合并情况:

```
$ git log --graph --pretty=oneline --abbrev-commit
* 59bc1cb conflict fixed
|\
| * 75a857c AND simple
* | 400b400 & simple
|/
* fec145a branch test
...
```

最后，删除`feature1`分支。