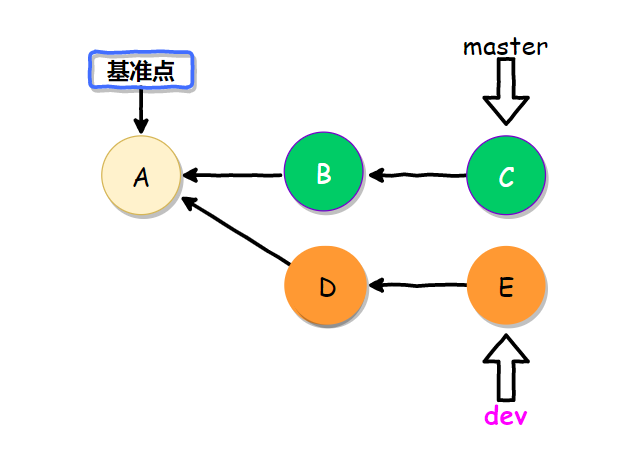
**Git操作 - rebase命令执行原理**

如图所示，master和dev分支都以提交节点A为基准点：



如果dev分支想要变换A这个基准点，那么：

**第一步**：切换到dev分支上；（想要变基哪个分支，就切换到哪个分支上执行git rebase命令）

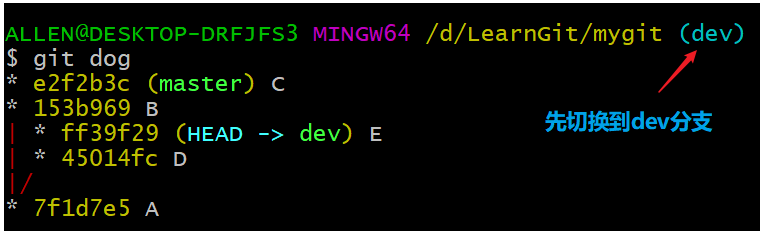
**第二步**：执行git rebase master；（master就代表变基的目标分支）

说明：上述命令中rebase参数后面指定的分支，就是变更后的基准点，如上例子中的master分支：

* 如果是分支，如master，基准点为该分支的最新提交节点，也就是C；
* 如果是一个commit\_id，基准点为该commit\_id对应的提交节点；

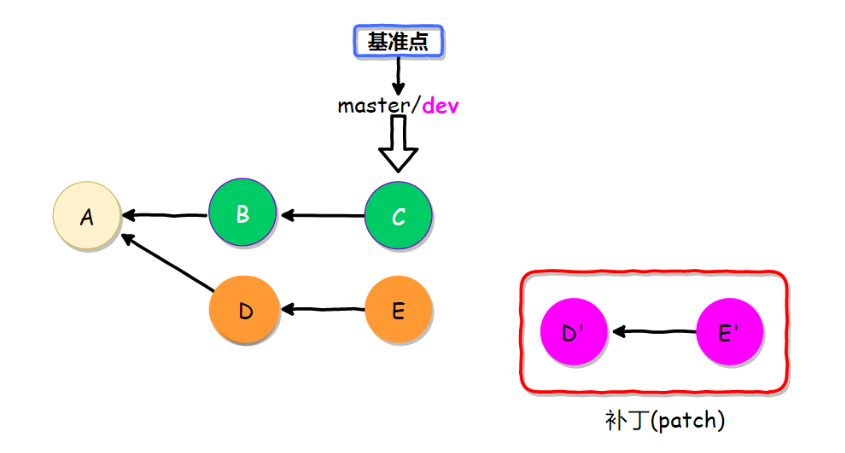
**（1）以分支为基准点示例**

沿用以上模型，查看项目的历史提交记录，如下图：



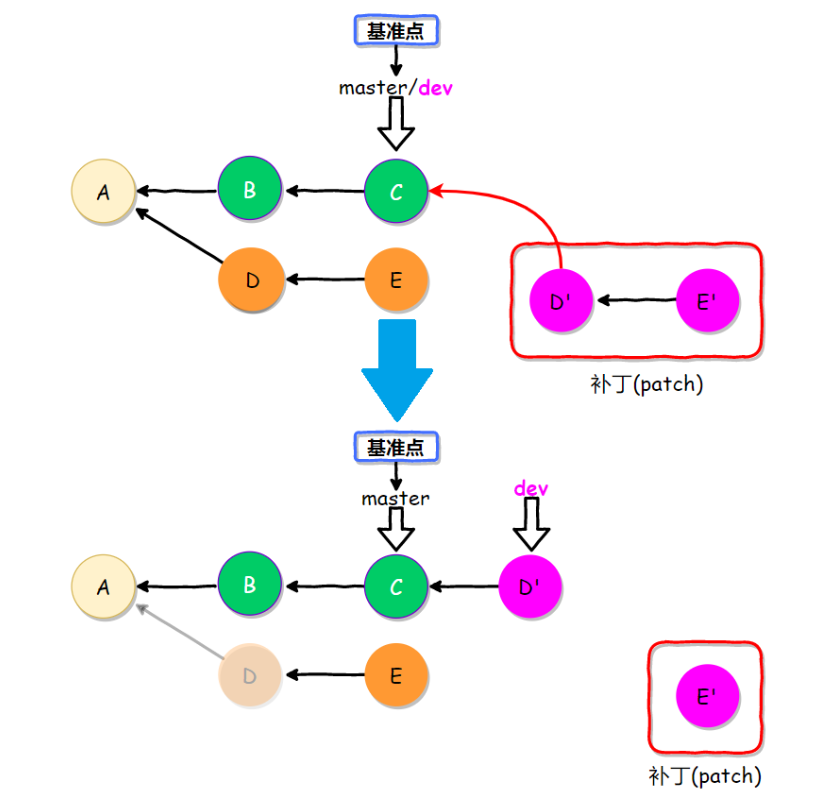
**1）首先，将dev分支上除了基准点A外的所有节点复制一份，即D'和E'，作为补丁备用，并将分支dev指向新基准点C。**

如下图：

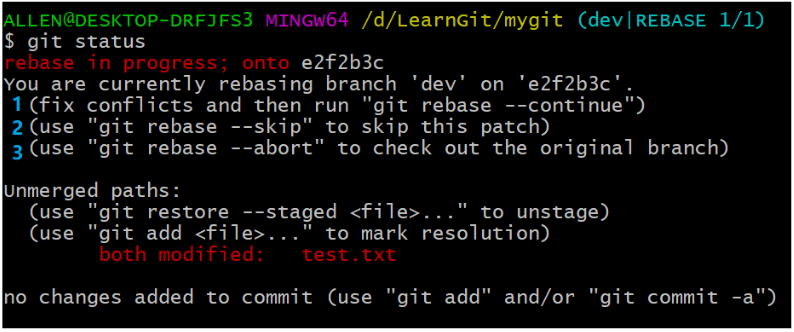


**2）然后，按原来dev上的节点顺序（D -> E）将补丁应用（Patch Applying）到新基准点C后面，并同时改变分支dev指向。**

如下图：**追加补丁D'**

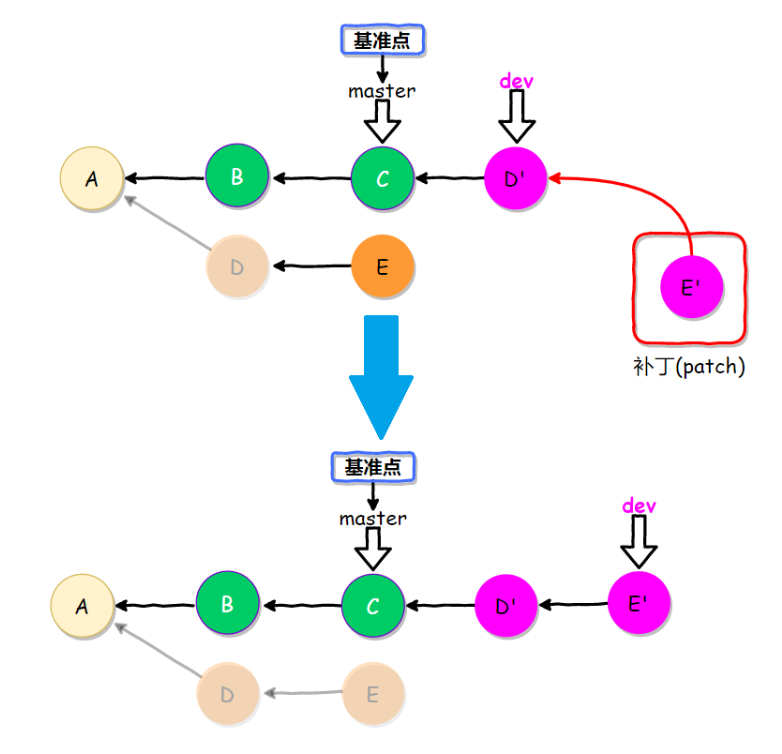


每次向新基准点应用补丁时，都会出现**三个选项**：

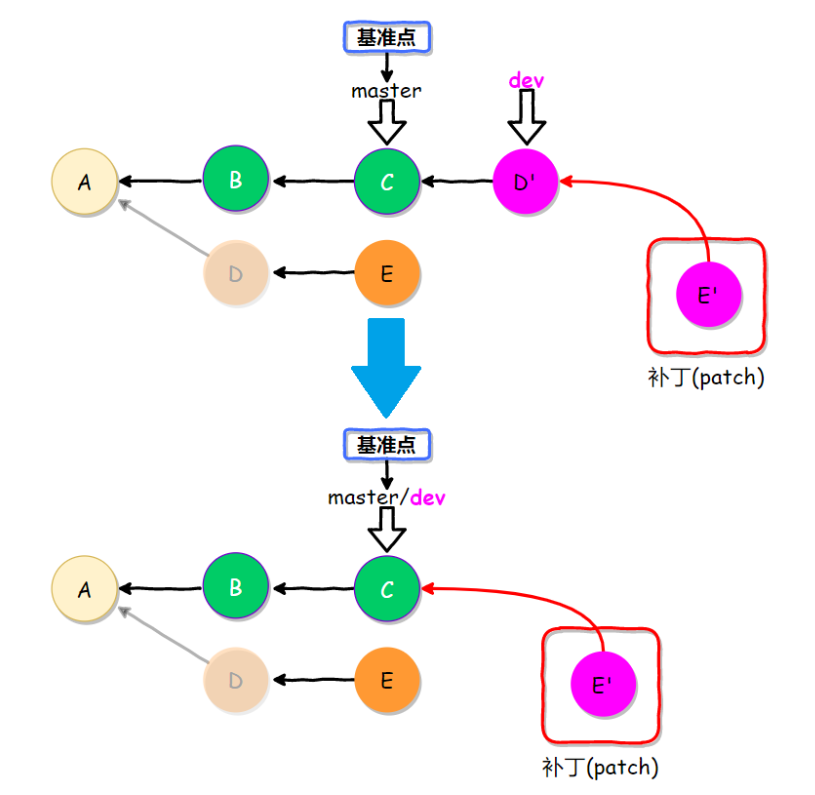


说明：

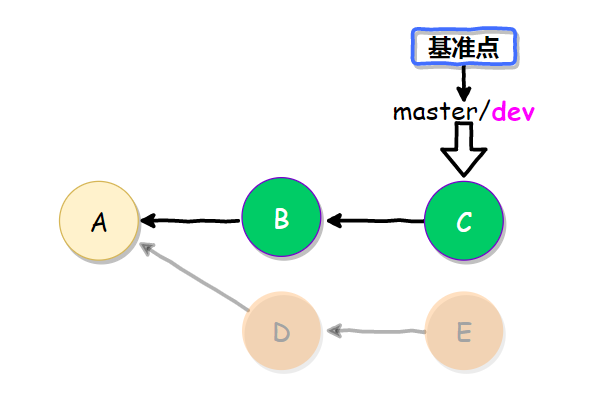
@1.git rebase --continue该选项表示：解决了合并冲突后，继续应用剩余补丁E'，如下图：



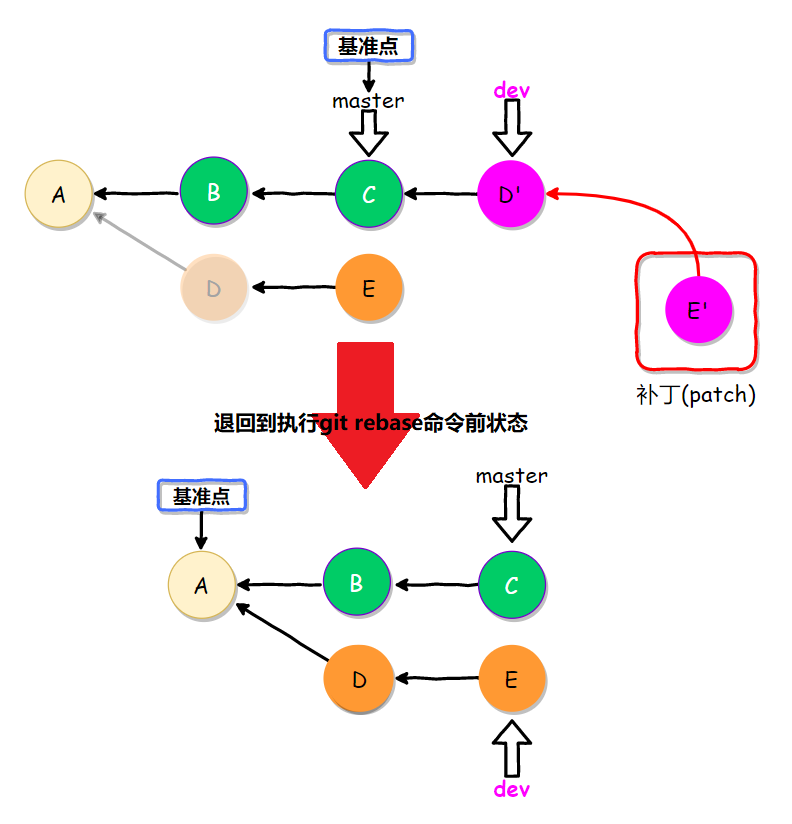
@2.git rebase --skip该选项表示：跳过当前补丁，继续应用下一个补丁：



如果一直执行该选项，直到应用完分支dev上剩余的补丁，结束git rebase命令后，两分支的状态为：

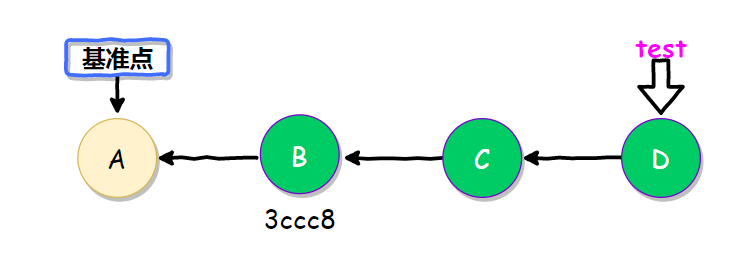


git rebase --abort该选项表示：终止rebase操作，回到执行rebase指令前的状态：



**（2）以提交为基准点示例**

如图下所示，若将提交节点B作为基准点，在当前正在工作的分支为test：

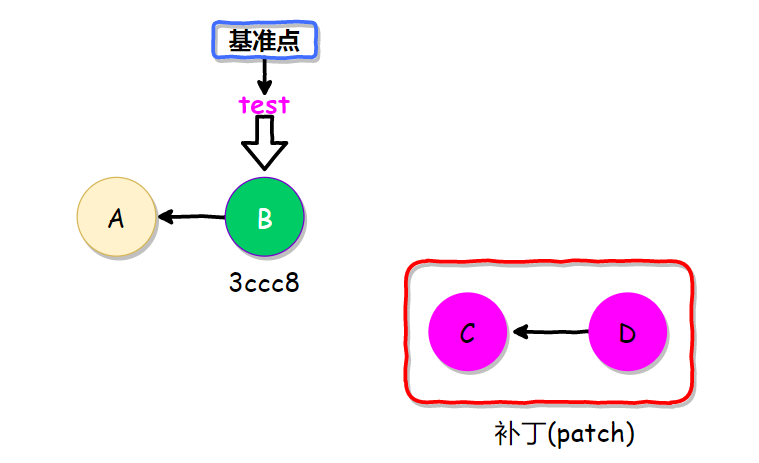


执行命令：git rebase 3ccc8。

会直接将原来的提交节点C和D，应用到新基准点B后，相当于没有发生变化，这个变基的过程为：

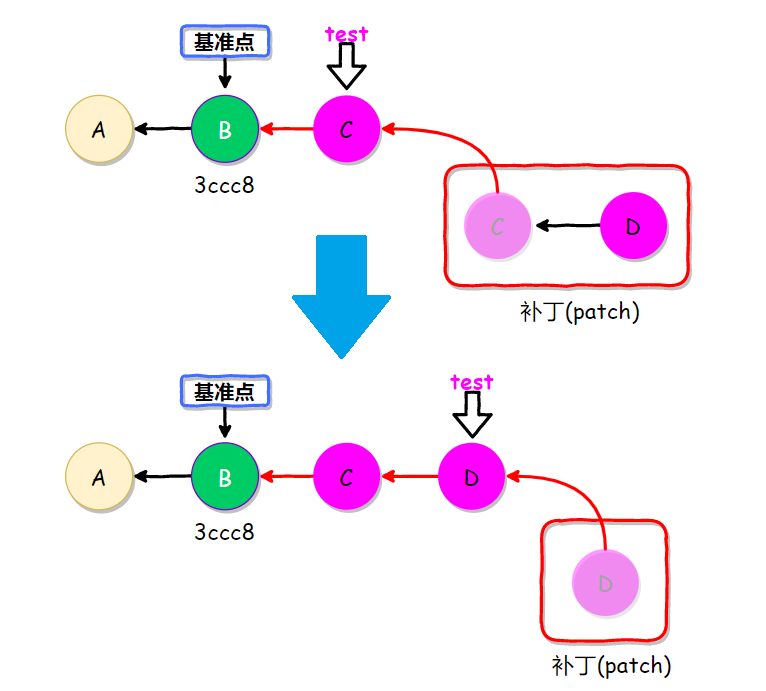
**1）首先，将基准点和test分支指向改变为节点B，并将test分支上基准点往后的提交节点作为补丁。**

如下图：

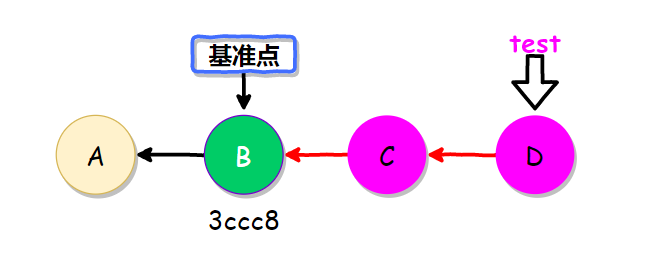


**2）然后，按顺序将补丁C和D提交节点，应用到新基准点B后面。**

如下图：



**3）最后，test分支的状态。**



所以，直接执行git rebase 3ccc8命令，历史提交记录不会有任何变化，但是C和D提交的commit-id会有变化。

**记住：基准点之后的提交会变基，不包括基准点。**

|  |
| --- |
| 参考：   * [https://blog.csdn.net/qq\_43425487/article/details/105620270](https://links.jianshu.com/go?to=https%3A%2F%2Fblog.csdn.net%2Fqq_43425487%2Farticle%2Fdetails%2F105620270) |

作者：繁华似锦Fighting 链接：https://www.jianshu.com/p/288507d9619d 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。