

#### 1. 分支

分支与主线平行，不影响主线的运行

#### 2. 查看当前有多少分支 `git branch`

```
python@ubuntu:~/git_learn$ git branch
* master
```

以上表示当前只有主线，没有分支，指针指向master主线

#### 3. 创建分支 `git branch 分支名`

```
python@ubuntu:~/git_learn$ git branch dev
python@ubuntu:~/git_learn$ git branch
dev
* master
```

以上表示当前有主线master和分支dev，指针指向master主线

#### 4. 切换分支 `git checkout 分支名`

```
python@ubuntu:~/git_learn$ git checkout dev
切换到分支 'dev'
python@ubuntu:~/git_learn$ git branch
* dev
master
```

以上表示当前有主线master和分支dev，指针指向dev分支

#### 5. 创建+切换分支 `git checkout -b 分支名`

```
python@ubuntu:~/git_learn$ git checkout -b dev1
切换到一个新分支 'dev1'
python@ubuntu:~/git_learn$ git branch
dev
* dev1
master
```

以上表示当前有主线master和分支dev dev1，指针指向dev1分支

#### 6. 合并某分支到某分支或主线 `git merge 分支名`

```
python@ubuntu:~/git_learn$ git checkout master
切换到分支 'master'
python@ubuntu:~/git_learn$ git merge dev1
更新 53bc36d..1b1462d
Fast-forward
 cat.txt | 1 +
 1 file changed, 1 insertion(+)
```

首先将指针切换到需要合并到的主线上，再合并需要合并的分支

备注：fast-forward为快进模式，就是直接把master指向dev1的当前提交，合并速度非常快

## 7. 删除分支 `git branch -d 分支名`

```
python@ubuntu:~/git_learn$ git branch -d dev
已删除分支 dev (曾为 53bc36d) 。
python@ubuntu:~/git_learn$ git branch
* master
```

删除已经合并到主线的分支，只保留主线master

## 8. 解决冲突

(1): 创建并切换到分支，然后对文件编辑

```
python@ubuntu:~/git_learn$ git checkout dev
切换到分支 'dev'
python@ubuntu:~/git_learn$ vim cat.txt
python@ubuntu:~/git_learn$ git add cat.txt
python@ubuntu:~/git_learn$ git commit -m "分支提交1"
[dev 0a0b4fa] 分支提交1
 1 file changed, 1 insertion(+)
python@ubuntu:~/git_learn$ git log --pretty=oneline
0a0b4faf94e8cb24337a2786f20ec20cd0b2a63f 分支提交1
53bc36d3d8c8009033c99f4bcc915436c6fd9b43 删除cat1.txt
75d6c32e95ad416ace5b53d38c7eb7e5c0c70533 版本4
5dfa13d493e718adcd43bee201852857498149a7 版本3
a5f71c3a41c376bb7f163911753fee9149c7ac19 版本2
0522d5575b051135ae13e3a9875b87f330fb159f 版本一
```

(2): 切换到主线，然后对文件编辑

```

python@ubuntu:~/git_learn$ git checkout master
切换到分支 'master'
python@ubuntu:~/git_learn$ vim cat.txt
python@ubuntu:~/git_learn$ git add cat.txt
python@ubuntu:~/git_learn$ git commit -m "主线提交"
[master ebf2879] 主线提交
1 file changed, 1 insertion(+), 1 deletion(-)
python@ubuntu:~/git_learn$ git log --pretty=oneline
ebf2879ec1426a6341ce11f97cb6be3854977387 主线提交
1b1462d9cf3577ef19542d19109b470a137e56bd 分支
53bc36d3d8c8009033c99f4bcc915436c6fd9b43 删除cat1.txt
75d6c32e95ad416ace5b53d38c7eb7e5c0c70533 版本4
5dfa13d493e718adcd43bee201852857498149a7 版本3
a5f71c3a41c376bb7f163911753fee9149c7ac19 版本2
0522d5575b051135ae13e3a9875b87f330fb159f 版本一

```

(3): 合并分支到主线冲突，此时需打开文件，对文件进行编辑，然后提交此次变更

```

python@ubuntu: ~/git_learn
1 this is the frist line;
2 this is the second line;
3 this is the third line;
4 this is the forth line;
5 <<<<<< HEAD
6 a master line
7 =====
8 a branch line
9 >>>>>> dev

```

```
python@ubuntu:~/git_learn$ git merge dev
自动合并 cat.txt
冲突（内容）：合并冲突于 cat.txt
自动合并失败，修正冲突然后提交修正的结果。
python@ubuntu:~/git_learn$ vim cat.txt
python@ubuntu:~/git_learn$ git add cat.txt
python@ubuntu:~/git_learn$ git commit -m "解决冲突"
[master 6f0bfe2] 解决冲突
python@ubuntu:~/git_learn$ git log --graph --pretty=oneline
* 6f0bfe246e58b84c14b6fbd7ae6de33984e42883 解决冲突
| \
| * 0a0b4faf94e8cb24337a2786f20ec20cd0b2a63f 分支提交1
* | ebf2879ec1426a6341ce11f97cb6be3854977387 主线提交
* | 1b1462d9cf3577ef19542d19109b470a137e56bd 分支
| /
* 53bc36d3d8c8009033c99f4bcc915436c6fd9b43 删除cat1.txt
* 75d6c32e95ad416ace5b53d38c7eb7e5c0c70533 版本4
* 5dfa13d493e718adcd43bee201852857498149a7 版本3
* a5f71c3a41c376bb7f163911753fee9149c7ac19 版本2
* 0522d5575b051135ae13e3a9875b87f330fb159f 版本一
```

(4). 合并分支到主线后删除分支

```
python@ubuntu:~/git_learn$ git branch -d dev
已删除分支 dev (曾为 0a0b4fa) 。
python@ubuntu:~/git_learn$ git branch
* master
```

## 9. 分支管理策略

(1): 通常合并分支时，如果可能，git会用 fast forward 模式，但是有些快速合并并不能成但合并时没有冲突，这个时候会合并之后做一次新的提交

```

python@ubuntu:~/git_learn$ git checkout -b dev
切换到一个新分支 'dev'
python@ubuntu:~/git_learn$ vim cat3.txt
python@ubuntu:~/git_learn$ git add cat3.txt
python@ubuntu:~/git_learn$ git commit -m "创建新文件cat3.txt"
[dev cd5a290] 创建新文件cat3.txt
1 file changed, 1 insertion(+)
create mode 100644 cat3.txt
python@ubuntu:~/git_learn$ git checkout master
切换到分支 'master'
python@ubuntu:~/git_learn$ vim cat.txt
python@ubuntu:~/git_learn$ git add cat.txt
python@ubuntu:~/git_learn$ git commit -m "添加新行"
[master e7a25e7] 添加新行
1 file changed, 1 insertion(+)
python@ubuntu:~/git_learn$ git log --pretty=oneline
e7a25e73459665839f48b7c808c218f528335b8c 添加新行
6f0bfe246e58b84c14b6fbd7ae6de33984e42883 解决冲突
ebf2879ec1426a6341ce11f97cb6be3854977387 主线提交
0a0b4faf94e8cb24337a2786f20ec20cd0b2a63f 分支提交1
1b1462d9cf3577ef19542d19109b470a137e56bd 分支
53bc36d3d8c8009033c99f4bcc915436c6fd9b43 删除cat1.txt
75d6c32e95ad416ace5b53d38c7eb7e5c0c70533 版本4
5dfa13d493e718adcd43bee201852857498149a7 版本3
a5f71c3a41c376bb7f163911753fee9149c7ac19 版本2
0522d5575b051135ae13e3a9875b87f330fb159f 版本一

```

对dev分支新建一个文件，对master主线文件修改

然后git merge dev 合并分支出现下图所示画面



修改备注后 ctrl+X离开然后按Y保存

此时dev分支合并到master主线



Merge branch 'dev'



# 请输入一个提交信息以解释此合并的必要性，尤其是将一个更新后的上游分支  
# 合并到主题分支。  
#  
# 以 '#' 开头的行将被忽略，而且空提交说明将会终止提交。



 求助  
 离开



 Write Out  
 读档

 搜索  
 替换

[ 已读取6 行 ]  
 剪切文字  
 Uncut Text

 对齐  
 拼写检查

 光标位置  
 跳行


 上页  
 下页

合并dev分支

# 请输入一个提交信息以解释此合并的必要性，尤其是将一个更新后的上游分支  
# 合并到主题分支。  
#  
# 以 '#' 开头的行将被忽略，而且空提交说明将会终止提交。

储存更动过的缓冲区吗(回答 "No" 会撤销修改)?

Y 是  
N 否

 取消



```
python@ubuntu:~/git_learn$ git merge dev
Merge made by the 'recursive' strategy.
 cat3.txt | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 cat3.txt
python@ubuntu:~/git_learn$
python@ubuntu:~/git_learn$ git log --pretty=oneline
fad54b5754a335811d2ef4cca8a5a8ff5dff7179 合并dev分支
e7a25e73459665839f48b7c808c218f528335b8c 添加新行
cd5a290d564ab57a553bae799634dea98162845d 创建新文件cat3.txt
6f0bfe246e58b84c14b6fbd7ae6de33984e42883 解决冲突
ebf2879ec1426a6341ce11f97cb6be3854977387 主线提交
0a0b4faf94e8cb24337a2786f20ec20cd0b2a63f 分支提交1
1b1462d9cf3577ef19542d19109b470a137e56bd 分支
53bc36d3d8c8009033c99f4bcc915436c6fd9b43 删除cat1.txt
75d6c32e95ad416ace5b53d38c7eb7e5c0c70533 版本4
5dfa13d493e718adcd43bee201852857498149a7 版本3
a5f71c3a41c376bb7f163911753fee9149c7ac19 版本2
0522d5575b051135ae13e3a9875b87f330fb159f 版本一
```

```
python@ubuntu:~/git_learn$ git log --pretty=oneline --graph
* fad54b5754a335811d2ef4cca8a5a8ff5dff7179 合并dev分支
| \
| * cd5a290d564ab57a553bae799634dea98162845d 创建新文件cat3.txt
* | e7a25e73459665839f48b7c808c218f528335b8c 添加新行
| /
* 6f0bfe246e58b84c14b6fbd7ae6de33984e42883 解决冲突
| \
| * 0a0b4faf94e8cb24337a2786f20ec20cd0b2a63f 分支提交1
* | ebf2879ec1426a6341ce11f97cb6be3854977387 主线提交
* | 1b1462d9cf3577ef19542d19109b470a137e56bd 分支
| /
* 53bc36d3d8c8009033c99f4bcc915436c6fd9b43 删除cat1.txt
* 75d6c32e95ad416ace5b53d38c7eb7e5c0c70533 版本4
* 5dfa13d493e718adcd43bee201852857498149a7 版本3
* a5f71c3a41c376bb7f163911753fee9149c7ac19 版本2
* 0522d5575b051135ae13e3a9875b87f330fb159f 版本一
```

最后删除dev分支

```
python@ubuntu:~/git_learn$ git branch -d dev
已删除分支 dev (曾为 cd5a290) 。
python@ubuntu:~/git_learn$ git branch
* master
```

(2): 强制禁用fast forward模式, git会在merge时生成一个新的commit

首先创建并切换到分支上, 修改文件

然后切换到master主线，使用 `git merge --no-ff -m 备注 分支名` 将分支合并到主线上

```
python@ubuntu:~/git_learn$ git checkout -b dev
切换到一个新分支 'dev'
python@ubuntu:~/git_learn$ vim cat.txt
python@ubuntu:~/git_learn$ git add cat.txt
python@ubuntu:~/git_learn$ git commit -m "添加新行"
[dev 2d867f8] 添加新行
1 file changed, 1 insertion(+)
python@ubuntu:~/git_learn$ git checkout master
切换到分支 'master'
python@ubuntu:~/git_learn$ git merge --no-ff -m "禁用fast forward合并" dev
Merge made by the 'recursive' strategy.
cat.txt | 1 +
1 file changed, 1 insertion(+)
```

```
python@ubuntu:~/git_learn$ git log --graph --pretty=oneline
*    c1e00c5e8a8453a832af7300220e2cc01277c533 禁用fast forward合并
| \
| * 2d867f8ac4c59c8a42eb124ebd8c9cbcb128542b 添加新行
| /
*    fad54b5754a335811d2ef4cca8a5a8ff5dfff7179 合并dev分支
| \
| * cd5a290d564ab57a553bae799634dea98162845d 创建新文件cat3.txt
* | e7a25e73459665839f48b7c808c218f528335b8c 添加新行
| /
*    6f0bfe246e58b84c14b6fbd7ae6de33984e42883 解决冲突
| \
| * 0a0b4faf94e8cb24337a2786f20ec20cd0b2a63f 分支提交1
* | ebf2879ec1426a6341ce11f97cb6be3854977387 主线提交
* | 1b1462d9cf3577ef19542d19109b470a137e56bd 分支
| /
*    53bc36d3d8c8009033c99f4bcc915436c6fd9b43 删除cat1.txt
*    75d6c32e95ad416ace5b53d38c7eb7e5c0c70533 版本4
*    5dfa13d493e718adcd43bee201852857498149a7 版本3
*    a5f71c3a41c376bb7f163911753fee9149c7ac19 版本2
*    0522d5575b051135ae13e3a9875b87f330fb159f 版本一
```

## 10. bug分支

bug可以通过一个新的临时分支来修复，修复后，合并分支，然后将临时分支删除。

当手头工作没有完成时，先把工作现场git stash一下，然后去修复bug，修复后再git stash pop回到工作现场

(1).分支dev上目前在编辑cat3.txt文件，还没有提交，此时有一个bug需要处理，可以使用git stash将分支dev 放入暂存区



```
python@ubuntu:~/git_learn$ git branch
* master
python@ubuntu:~/git_learn$ git checkout -b dev
切换到一个新分支 'dev'
python@ubuntu:~/git_learn$ ls
cat3.txt  cat.txt
python@ubuntu:~/git_learn$ vim cat3.txt
python@ubuntu:~/git_learn$ git status
位于分支 dev
尚未暂存以备提交的变更：
  （使用 "git add <文件>..." 更新要提交的内容）
  （使用 "git checkout -- <文件>..." 丢弃工作区的改动）

        修改：      cat3.txt

修改尚未加入提交（使用 "git add" 和/或 "git commit -a"）
python@ubuntu:~/git_learn$ git stash
Saved working directory and index state WIP on dev: c1e00c5 禁用fast forward合并
HEAD 现在位于 c1e00c5 禁用fast forward合并
python@ubuntu:~/git_learn$ git status
位于分支 dev
无文件要提交，干净的工作区
```

(2). 切换回master主线，然后新建一个分支处理bug，将bug修改后的分支合并到主线，然后删除bug分支

```
python@ubuntu:~/git_learn$ git checkout master
切换到分支 'master'
python@ubuntu:~/git_learn$ git checkout -b bug001
切换到一个新分支 'bug001'
python@ubuntu:~/git_learn$ git branch
* bug001
  dev
  master
python@ubuntu:~/git_learn$ ls
cat3.txt  cat.txt
python@ubuntu:~/git_learn$ vim cat.
python@ubuntu:~/git_learn$ vim cat.
cat.      cat.txt
python@ubuntu:~/git_learn$ vim cat.txt
python@ubuntu:~/git_learn$ git add cat.txt
python@ubuntu:~/git_learn$ git commit -m "修复bug001"
[bug001 9ba45ee] 修复bug001
 1 file changed, 1 insertion(+)
python@ubuntu:~/git_learn$ git checkout master
```

```
python@ubuntu:~/git_learn$ git merge --no-ff -m "修复bug" bug001
Merge made by the 'recursive' strategy.
 cat.txt | 1 +
 1 file changed, 1 insertion(+)
python@ubuntu:~/git_learn$ git log --pretty=oneline --graph
*    4cf3f7b735eefa34fc34f67d38be66bc4d6de347 修复bug
| \
| * 9ba45eed375ba1374a490dd13fb0626d4821cb7b 修复bug001
| /
*   c1e00c5e8a8453a832af7300220e2cc01277c533 禁用fast forward合并
| \
| * 2d867f8ac4c59c8a42eb124ebd8c9cbeb128542b 添加新行
| /
*   fad54b5754a335811d2ef4cca8a5a8ff5dff7179 合并dev分支
| \
| * cd5a290d564ab57a553bae799634dea98162845d 创建新文件cat3.txt
* | e7a25e73459665839f48b7c808c218f528335b8c 添加新行
| /
*   6f0bfe246e58b84c14b6fbd7ae6de33984e42883 解决冲突
| \
| * 0a0b4faf94e8cb24337a2786f20ec20cd0b2a63f 分支提交1
* | ebf2879ec1426a6341ce11f97cb6be3854977387 主线提交
* | 1b1462d9cf3577ef19542d19109b470a137e56bd 分支
| /
*   53bc36d3d8c8009033c99f4bcc915436c6fd9b43 删除cat1.txt
*   75d6c32e95ad416ace5b53d38c7eb7e5c0c70533 版本4
*   5dfa13d493e718adcd43bee201852857498149a7 版本3
```

```
python@ubuntu:~/git_learn$ git branch -d bug001
已删除分支 bug001 (曾为 9ba45ee) 。
python@ubuntu:~/git_learn$ git branch
dev
* master
```

(3). bug修改OK后再切换回dev分支，使用git stash pop恢复暂存区里的文件

git stash list可以查看暂存区里是否有文件

```
python@ubuntu:~/git_learn$ git status
位于分支 dev
无文件要提交，干净的工作区
python@ubuntu:~/git_learn$ git stash list
stash@{0}: WIP on dev: c1e00c5 禁用fast forward合并
```

```
python@ubuntu:~/git_learn$ git stash pop
```

位于分支 dev  
 尚未暂存以备提交的变更：  
 （使用 "git add <文件>..." 更新要提交的内容）  
 （使用 "git checkout -- <文件>..." 丢弃工作区的改动）

修改： cat3.txt

修改尚未加入提交（使用 "git add" 和/或 "git commit -a"）  
 丢弃了 refs/stash@{0} (5fa092dab1a30aecdde1f479228bab40ebee3601)

```
python@ubuntu:~/git_learn$ git status
```

位于分支 dev  
 尚未暂存以备提交的变更：  
 （使用 "git add <文件>..." 更新要提交的内容）  
 （使用 "git checkout -- <文件>..." 丢弃工作区的改动）

修改： cat3.txt

修改尚未加入提交（使用 "git add" 和/或 "git commit -a"）

## 总结

