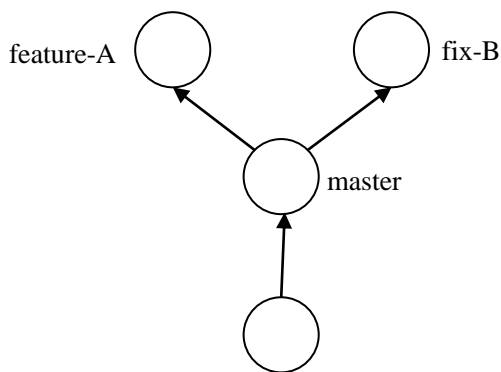


分支操作

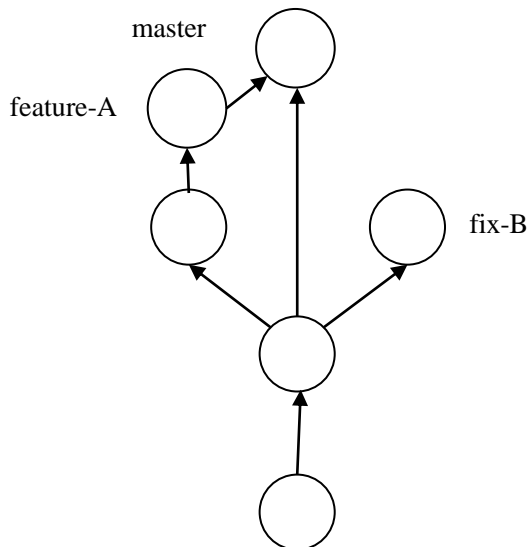
Git 最核心的操作对象是版本（commit），最核心的操作技巧是分支。

在进行多个并行作业时，我们会用到分支。在这类并行开发的过程中，往往同时存在多个最新代码状态。如图所示，从 `master` 分支创建 `feature-A` 和 `fix-B` 分支后，每个分支中都拥有自己的最新代码。`master` 分支是 Git 默认创建的分支，因此基本上所有开发都是以这个分支为中心进行的。

此外，`feature-A` 与 `fix-B` 可以称之为特性分支。所谓特性分支，是集中实现单一特性（主题），除此之外不进行任何作业的分支。在日常开发中，往往会创建数个特性分支，同时在此之外再保留一个随时可以发布软件的稳定分支。稳定分支的角色通常由 `master` 分支担当，所以 `master` 分支也称为“主干分支”或“主分支”。



不同分支中，可以同时进行完全不同的作业。等该分支的作业完成之后再与 `master` 分支合并。比如 `feature-A` 分支的作业结束后与 `master` 合并，如图所示。通过灵活运用分支，可以让多人同时高效地进行并行开发。



(1) `git branch`——显示分支一览表

用 `git branch` 命令可以将分支名列表显示出来，同时可以确认当前所在分支。

```
LZT@LZT-PC MINGW64 ~/hello (master)
$ git branch
* master
```

可以看到 `master` 分支左侧标有“*”（星号），表示这是我们当前所在的分支。也就是说，我们正在 `master` 分支下进行开发。结果中没有显示其他分支名，表示本地仓库中只存在 `master` 一个分支。

（2）`git checkout -b`——创建、切换分支

①切换到 `feature-A` 分支并进行提交

如果想以当前的 `master` 分支为基础创建新的分支，需要用到 `git checkout -b` 命令。

比如，我们使用 `git checkout -b feature-A` 创建一个名为 `feature-A` 的分支。

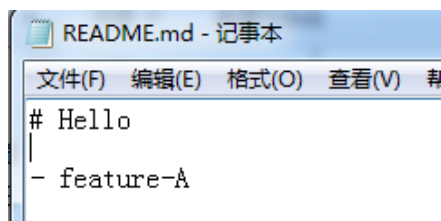
再用 `git branch` 查看分支列表，会显示我们处于 `feature-A` 分支下。

```
LZT@LZT-PC MINGW64 ~/hello (master)
$ git checkout -b feature-A
Switched to a new branch 'feature-A'

LZT@LZT-PC MINGW64 ~/hello (feature-A)
$ git branch
* feature-A
  master
```

`feature-A` 分支左侧标有“*”，表示当前分支为 `feature-A`。在这个状态下像正常开发那样修改代码、执行 `git add` 命令并进行提交的话，代码就会提交至 `feature-A` 分支。像这样不断对一个分支进行提交操作，我们称为“培育分支”。

比如我们在上节课新建的 `README.md` 文件中添加一行。



```
README.md - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)
# Hello
- feature-A
```

这里我们添加了 `- feature-A` 这样一行字母，然后进行提交。

```
LZT@LZT-PC MINGW64 ~/hello (feature-A)
$ git status
On branch feature-A
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   README.md

no changes added to commit (use "git add" and/or "git commit -a")

LZT@LZT-PC MINGW64 ~/hello (feature-A)
$ git add README.md

LZT@LZT-PC MINGW64 ~/hello (feature-A)
$ git commit -m"Add feature-A"
[feature-A 66b5c18] Add feature-A
1 file changed, 2 insertions(+), 1 deletion(-)
```

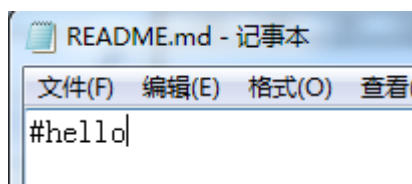
于是，这一行就添加到 feature-A 分支中了。

②切换到 master 分支

现在我们再来看一看 master 分支有没有收到影响。首先切换至 master 分支。

```
LZT@LZT-PC MINGW64 ~/hello (feature-A)
$ git checkout master
Switched to branch 'master'
```

然后查看 README.md 文件，会发现文件仍然保持原先的状态，并没有被添加文字。



feature-A 分支的更改不会影响到 master 分支，这正是在开发中创建分支的优点。只要创建多个分支，就可以在不互相影响的情况下同时进行多个功能的开发。

③切换回上一个分支

现在，我们再切换回 feature-A 分支。

```
LZT@LZT-PC MINGW64 ~/hello (master)
$ git checkout -
Switched to branch 'feature-A'
```

像上面这样用“-”（连字符）代替分支名，就可以切换至上一个分支。当然，将“-”替换为 feature-A 同样可以切换到 feature-A 分支。

(3) 合并分支

接下来，我们假设 feature-A 已经实现完毕，想要将它合并到主干分支 master 中。首先切换到 master 分支。

```
LZT@LZT-PC MINGW64 ~/hello (feature-A)
$ git checkout master
Switched to branch 'master'
```

然后合并 feature-A 分支。为了在历史记录中明确记录下本次分支合并，我们需要创建合并提交。因此，在合并时加上--no-ff 参数。

```
LZT@LZT-PC MINGW64 ~/hello (master)
$ git merge --no-ff feature-A
```

随后编辑器会启动，用于录入合并提交的信息。

```
Merge branch 'feature-A'

# Please enter a commit message to explain why this merge is necessary,
# especially if it merges an updated upstream into a topic branch.
#
# Lines starting with '#' will be ignored, and an empty message aborts
# the commit.
```

默认信息中已经包含了是从 feature-A 分支合并过来的相关内容，所以可不必做任何修改。将编辑器中显示的内容保存，关闭编辑器（方法：一直按住 ESC，再连续按大写的 z 两次就退出来了），然后就会看到下面的结果。

```
LZT@LZT-PC MINGW64 ~/hello (master)
$ git merge --no-ff feature-A
Merge made by the 'recursive' strategy.
 README.md | 3 ++-
 1 file changed, 2 insertions(+), 1 deletion(-)
```

这样一来，feature-A 分支的内容就合并到 master 分支中了。

（4）git log --graph——以图表形式查看分支

用 git log --graph 命令查看提交日志的话，能很清楚地看到 feature-A 分支创建及提交的内容被合并。

```
LZT@LZT-PC MINGW64 ~/hello (master)
$ git log --graph
*   commit d7a3995c80fdd99e60f9ff929da67d220ef3c6d0 (HEAD -> master)
  | Merge: 347b98b 66b5c18
  | Author: ZitingLou <louziting@163.com>
  | Date: Mon Nov 6 21:30:19 2017 +0800
  |
  | Merge branch 'feature-A'
  |
  *   commit 66b5c18edb28a72f65074eb91e456a684abce68f (feature-A)
  | / Author: ZitingLou <louziting@163.com>
  | / Date: Mon Nov 6 21:02:18 2017 +0800
  | /
  | / Add feature-A
  | /
  *   commit 347b98bd9ba08a13c3abfd1da56bf75d7b6ff47b
  | Author: ZitingLou <louziting@163.com>
  | Date: Mon Nov 6 20:59:51 2017 +0800
  |
  | This is a test
```