建立了远端仓库，现在谁都可以查看并且修改仓库里面的内容。这就需要我们将远端仓库的内容下载到本地。

1. 获取远端仓库

我们切换到其他目录下，然后将将GitHub上的仓库clone到本地。

输入：

cd ..

cd hello\_world\_clone

git clone git@github.com:windofme1109/hello\_world.git

提示：

Cloning into 'hello\_world'...

remote: Enumerating objects: 18, done.

remote: Counting objects: 100% (18/18), done.

remote: Compressing objects: 100% (7/7), done.

Receiving objects: 100% (18/18), done.

Resolving deltas: 100% (1/1), done.

remote: Total 18 (delta 1), reused 18 (delta 1), pack-reused 0

git clone命令用来克隆GitHub上面的远端仓库到本地。此时，我们会默认处于master 分支下，同时系统会自动将 origin 设置成该远程仓库的标识符。也就是说，当前本地仓库的 master 分支与 GitHub 端远程仓库（origin）的 master 分支在内容上是完全相同的。

因此，我们就可以对这个本地仓库中的内容进行操作。

我们用 git branch -a命令查看当前分支的相关信息。添加 -a参数可以同时显示本地仓库和远程仓库的分支信息。

输入：

$ git branch -a

提示：

\* master

remotes/origin/HEAD -> origin/master

remotes/origin/feature-D

remotes/origin/master

结果中显示了 remotes/origin/feature-D，证明我们的远程仓库中已经有了 feature-D 分支。

1. 获取远程的 feature - D 分支

我们试着将 feature-D 分支获取至本地仓库。

输入：

git checkout -b feature-D origin/feature-D

提示：

$ git checkout -b feature-D origin/feature-D

Switched to a new branch 'feature-D'

Branch 'feature-D' set up to track remote branch 'feature-D' from 'origin'.

-b 参数的后面是本地仓库中新建分支的名称。为了便于理解，我们仍将其命名为 feature-D，让它与远程仓库的对应分支保持同名。新建分支名称后面是获取来源的分支名称。我们指定了 origin/feature-D，就是说以名为 origin 的仓库（这里指 GitHub 端的仓库）的 feature-D 分支为来源，在本地仓库中创建 feature-D 分支。这样就会将feature-D分支的内容同步过来。

向feature-D这个分支中的readme.md添加内容，然后提交，最后推送到远端仓库。

输入：

git diff

提示：

**diff --git a/readme.md b/readme.md**

**index 494aad6..ffff318 100644**

**--- a/readme.md**

**+++ b/readme.md**

@@ -27,3 +27,5 @@ eeeeeeeeeeeeeeeeeeee

- feature-D

+- feature-D\_Clone

+

提交：

git add readme.md

git commit -m "new feature-D"

提示：

[feature-D 4727624] new feature-D

1 file changed, 2 insertions(+)

推送到远端仓库的feature-D分支：

输入：

git push

提示：

Enumerating objects: 5, done.

Counting objects: 100% (5/5), done.

Delta compression using up to 4 threads

Compressing objects: 100% (2/2), done.

Writing objects: 100% (3/3), 294 bytes | 294.00 KiB/s, done.

Total 3 (delta 1), reused 0 (delta 0)

remote: Resolving deltas: 100% (1/1), completed with 1 local object.

To github.com:windofme1109/hello\_world.git

1dfb490..4727624 feature-D -> feature-D

此时远端仓库的feature-D分支就同步更新了。

1. 获取最新的远端仓库

如果远端仓库中的内容发生了变化，我我们可以使用git pull命令，将本地的仓库更新到最新的状态。

输入：

git pull origin master

提示：

remote: Enumerating objects: 5, done.

remote: Counting objects: 100% (5/5), done.

remote: Compressing objects: 100% (2/2), done.

remote: Total 3 (delta 1), reused 0 (delta 0), pack-reused 0

Unpacking objects: 100% (3/3), done.

From github.com:windofme1109/hello\_world

\* branch master -> FETCH\_HEAD

b47cee8..d2e76a6 master -> origin/master

Updating b47cee8..d2e76a6

Fast-forward

readme.md | 5 ++++-

1 file changed, 4 insertions(+), 1 deletion(-)

我们只有一个master分支（主分支）。我们在GitHub上的远程仓库对readme.md进行了修改，添加的内容是：#### 哈哈哈哈哈哈哈哈哈哈。

使用git pull命令，origin是远端仓库的标识名，master是分支（可以从其他分支同步更新本地仓库）。

查看本地文件的内容。

输入：

cat readme.md

输出：

hello world

11111111111111

22222222222222

where amazing happens

aaaaaaaaaaaaaaaaaaaa

eeeeeeeeeeeeeeeeeeee

#### 哈哈哈哈哈哈哈哈哈哈

可以看出，已经与远端仓库保存同步更新了。

在刚刚克隆到本地仓库的分支feature-D（hello\_world\_clone/hello\_world/）的readme.md中添加了新的内容：feature-D\_Clone。这部分内容已经推送到远端仓库了。而原本的仓库的分支feature-D（hello\_world）却没有得到更新，内容与之前的一样。所以需要从远端仓库的分支feature-D获取最新的内容。

首先切换到原来的仓库中，然后查看所处的分支：

git branch

提示：

feature-A

feature-B

feature-C

\* feature-D

master

目前处在feature-D分支下。使用 git pull 命令，将本地的 feature-D 分支更新到最新状态。

输入：

git pull

提示：

remote: Enumerating objects: 5, done.

remote: Counting objects: 100% (5/5), done.

remote: Compressing objects: 100% (1/1), done.

remote: Total 3 (delta 1), reused 3 (delta 1), pack-reused 0

Unpacking objects: 100% (3/3), done.

From github.com:windofme1109/hello\_world

1dfb490..4727624 feature-D -> origin/feature-D

Updating 1dfb490..4727624

Fast-forward

readme.md | 2 ++

1 file changed, 2 insertions(+)

查看文件内容：

cat readme.md

输出：

# Git教程

- feature-A

- fix-B

- feature-C

- feature-D

- feature-D\_Clone

此时本地仓库的feature-D分支与远端仓库的feature-D分支保持了最新的同步状态。